

Customer	: ESRIN	Document Ref	: S2PAD-VEGA-SUM-0001
Contract No	: 21450/08/I-EC	Issue Date	: 15 May 2015
WP No	: 1.3	Issue	: 2.0

**Title** : **Sentinel-2 MSI – Level-2A Prototype Processor Installation and User Manual**

**Abstract** : Sen2Cor is a prototype processor for Sentinel-2 Level 2A product formatting and processing. The processor performs the tasks of atmospheric-, terrain and cirrus correction and a scene classification of Level 1C input data. Level 2A outputs are: Bottom-Of-Atmosphere (BOA), optionally terrain- and cirrus corrected reflectance images, Aerosol Optical Thickness-, Water Vapour-, Scene Classification maps and Quality Indicators, including cloud and snow probabilities. The Level 2A Product Formatting performed by the processor follows the specification of the Level 1C User Product.

**Author** : \_\_\_\_\_  
Uwe Müller-Wilm

**Approval** : \_\_\_\_\_  
Christian Laroque

**Accepted** : \_\_\_\_\_  
Christine Dingeldey  
Quality Assurance Manager

**Distribution** :

**Hard Copy File:**  
**Filename:** S2PAD-VEGA-SUM-0001-2.0.docx



**Copyright © 2015 Telespazio VEGA Deutschland GmbH**

*All rights reserved.*

*No part of this work may be disclosed to any third party translated reproduced copied or disseminated in any form or by any means except as defined in the contract or with the written permission of VEGA Deutschland GmbH & Co. KG.*

**Telespazio VEGA Deutschland GmbH**  
Europaplatz 5, 64293 Darmstadt, Germany  
Tel: +49 (0)6151 8257-0 Fax: +49 (0)6151 8257-799  
[www.telespazio-vega.de](http://www.telespazio-vega.de)

***This Page Is Intentionally Blank***

## TABLE OF CONTENTS

<b>1. INTRODUCTION.....</b>	<b>10</b>
1.1 Purpose and Scope.....	10
1.2 Document Overview.....	11
1.3 Documentation and Definitions.....	11
1.3.1 Normative Reference Documents.....	12
1.3.2 Informative Reference Documents.....	12
<b>2. FUNCTIONALITY AND OPERATION.....</b>	<b>13</b>
2.1 Scene Classification (L2A_SceneClass).....	14
2.2 Atmospheric Correction (L2A_AtmCorr).....	16
2.2.1 Look Up Table Generation.....	17
2.2.2 Aerosol Optical Thickness.....	17
2.2.3 Water Vapour Retrieval.....	18
2.2.4 Cirrus Correction.....	18
2.2.5 Surface Reflectance Retrieval.....	19
2.2.6 Terrain Correction.....	21
2.3 Level-2A Processor Architecture.....	22
2.4 Product Formatting.....	24
2.5 Runtime Configuration.....	25
2.6 Processing Scheme and Logical Flow (L2A_Process).....	25
2.6.1 Pre-processing (L2A_Tables).....	26
2.6.2 The 60 m Product Processing.....	26
2.6.3 The 20 m Product Processing.....	27
2.6.4 The 10 m Product Processing.....	27
2.6.5 Post-Processing.....	27
2.7 Logging (Logger).....	28
<b>3. CONFIGURATION AND INSTALLATION.....</b>	<b>29</b>
3.1 Installation.....	29
3.1.1 Setting up the Runtime Environment.....	30
Sen2Cor.....	31
3.1.2 Installation.....	31
3.1.3 Configuration Files.....	33
3.1.4 Uninstallation.....	33
3.1.5 Versioning.....	33
3.2 Operation.....	34
3.2.1 Command line interpreter.....	34
3.2.2 Interfacing and integration into the Sentinel-2 Toolbox.....	35
3.3 The Software Development Environment (SDE).....	36
3.3.1 Requirements and Third Party Software.....	36
3.3.1.1 Eclipse, PyDev and Rinzo.....	37
3.3.1.2 Anaconda.....	37
3.3.1.3 Cython.....	37
3.3.1.4 Distutils.....	38
3.3.1.5 GDAL.....	38
3.3.1.6 Glymur.....	38
3.3.1.7 PyTables.....	38
3.3.2 Installation.....	38
3.3.3 Configuration.....	39
3.3.3.1 Configure Python.....	39
Processor.....	39
3.3.3.2 Installation.....	39
3.3.3.3 Environment Settings.....	39
3.3.4 Operation.....	40
3.3.4.1 Running the processor within Eclipse.....	40

3.3.5	Generating a Source Distribution .....	41
3.3.6	Generating a binary module for the Atmospheric Correction.....	42
<b>4.</b>	<b>REFERENCES .....</b>	<b>43</b>
<b>5.</b>	<b>APPENDIX .....</b>	<b>44</b>
5.1	Licenses.....	44
	List OF FiguresFigure 2-1 – Sentinel-2 Spectral Bands and Resolutions .....	13
	Figure 2-2 – Processing Flow, Overview .....	14
	Figure 2-3 – Scene Classification .....	15
	Figure 2-4 – Scene Classification, Processing Flow.....	16
	Figure 2-5 – Atmospheric Correction, Processing Flow .....	16
	Figure 2-6 – AOT Retrieval using Band 12.....	18
	Figure 2-7 – WV Retrieval using Bands 8a and 9.....	18
	Figure 2-8 – Cirrus Correction, Bands 2-4 with Band 10 .....	19
	Figure 2-9 – Left: Level 1C Input, Bands 2-4, RGB composite image; middle: Scene Classification, right: Level 2A Output, Bands 2-4, RGB composite image .....	20
	Figure 2-10 – BOA reflectance spectra for soils, vegetation and water, separated by Scene Classification; Black: TOA Values.....	20
	Figure 2-11 – Terrain correction; Left: Level 1C Input, Bands 2-4, RGB composite image, middle: DEM, reshaped to according scene, right: Level 2A Output, Bands 2-4, RGB composite image.	21
	Figure 2-12 – High Level processor architecture .....	22
	Figure 2-13 – Filing structure of Level-1C product on tile level .....	25
	Figure 2-14 – Processing chain for the three different resolutions .....	26
	Figure 2-15 – Level 2A product, physical format Configuration (L2A_Config) .....	28
	Figure 3-1 – Configuration of Python interpreter within Eclipse.....	39
	Figure 3-2 – Environment settings within Eclipse .....	40
	Figure 3-3 – Setting the processors Main Module .....	41
	Figure 3-3 – SCD command line arguments.....	41

LIST OF TABLES

Table 2-1 – Classification Map ..... 15

Table 2-2 – Parameter space for atmospheric correction ..... 17

Table 2-3: Listing of software modules..... 23

Table 3-1: Third Party products for the SDE ..... 37

## AMENDMENT POLICY

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

### AMENDMENT RECORD SHEET

ISSUE	DATE	DCI No	REASON
1.0	2.July 2012	-	Created
1.1	15 September 2012	-	Issue after S2PAD Phase 2 CDR. Updated according to ESA comments and discussion on CDR 02/08/2012
1.2	1.June 2013	-	Adaptation of installation procedures according to unification of environments and pre release of processor for QR
1.3	31. March 2014	-	New Section 2.2, updated Installation procedures for Windows, moved information for data IO into new created document [L2A-IODD]
1.4	27. June 2014	DCR-01, 02	Restructuring of sections 3.1, 3.2 to align the installation after upgrades of Anaconda and GDAL.
2.0	15.May 2015	-	Integration into Sentinel-2 Toolbox, Version 2.0  Complete Improvement of Installation Procedure: Integration of CONDA Packages for GDAL and GLYMUR Upgrade to PSD V12 Upgrade of JPEG-2000 Readers to OpenJPEG 2.1.0 instead of Jasper Fixes of SPRs according to Release note for Version 2.0

## DOCUMENT CHANGE RECORD

			DCR No	001
			Date	01. July 2012
			Originator	Uwe Müller-Wilm
			Approved by	Christian Laroque
1. Document Title:			Sentinel-2 Level-2A Prototype Processor Installation and User Manual	
2. Document Reference Number:			S2PAD-VEGA-SUM_SD-17	
3. Document issue / revision number:			1.0 DRAFT	
4. Page	5. Paragraph	6. Reason for change		
All	All	Created new		

			DCR No	002
			Date	15 Sept 2012
			Originator	Uwe Müller-Wilm
			Approved by	Christian Laroque
1. Document Title:			Sentinel-2 Level-2A Prototype Processor Installation and User Manual	
2. Document Reference Number:			S2PAD-VEGA-SUM_SD-17	
3. Document issue / revision number:			1.1	
4. Page	5. Paragraph	6. Reason for change		
	REF	SUM_01		
	REF	SUM_02		
	2.1	SUM_03		
	2.2	SUM_04		
	table 9	SUM_05		
	Operation	SUM_06		

			DCR No	003
			Date	01 June 2013
			Originator	Uwe Müller-Wilm
			Approved by	Christian Laroque

1. Document Title:		Sentinel-2 Level-2A Prototype Processor Installation and User Manual
2. Document Reference Number:		S2PAD-VEGA-SUM_SD-17
3. Document issue / revision number:		1.2
4. Page	5. Paragraph	6. Reason for change
	1.3.1 – 1.3.2	Consolidation of references with glossary
	2.5	Alignment with processor development
	3.1	Alignment with processor development
	3.2	Alignment with processor development
	3.3	Alignment with processor development
	3.4	Alignment with processor development

		DCR No	004
		Date	31. March 2014
		Originator	Uwe Müller-Wilm
		Approved by	Christian Laroque
1. Document Title:		Sentinel-2 Level-2A Prototype Processor Installation and User Manual	
2. Document Reference Number:		S2PAD-VEGA-SUM	
3. Document issue / revision number:		1.3	
4. Page	5. Paragraph	6. Reason for change	
	2.2.1 – 2.2.6	New Sections on processors purpose and use	
	3.1.2.2.3	moved information for data IO into new created document [L2A-IODD]	
	3.1	added Installation procedures for Windows,	
	3.2	added configuration procedure for Windows.	
	DCR No	004	
	Date	31. March 2014	
	Originator	Uwe Müller-Wilm	
	Approved by	Christian Laroque	
1. Document Title:		Sentinel-2 Level-2A Prototype Processor Installation and User Manual	
2. Document Reference Number:		S2PAD-VEGA-SUM	
3. Document issue / revision number:		1.3	
4. Page	5. Paragraph	6. Reason for change	



	2.2.1 – 2.2.6	New Sections on processors purpose and use
	3.1.2.2.3	moved information for data IO into new created document [L2A-IODD]
	3.1	added Installation procedures for Windows,
	3.2	added configuration procedure for Windows.

		DCR No	005
		Date	15. May 2015
		Originator	Uwe Müller-Wilm
		Approved by	Christian Laroque
1. Document Title:		Sentinel-2 Level-2A Prototype Processor Installation and User Manual	
2. Document Reference Number:		S2PAD-VEGA-SUM	
3. Document issue / revision number:		2.0	
4. Page	5. Paragraph	6. Reason for change	
	2.2.5	New examples for Surface Reflectance Retrieval	
	2.2.6	New examples for Terrain Correction	
	3.1	Restructuring of sections 3.1 to align the installation after upgrades of Anaconda, GDAL and OpenJPEG.	
	3.2	Restructuring of sections 3.2 to align the configuration after upgrades of Anaconda, GDAL and OpenJPEG.	
	3.3	Update of SDE according to new installation procedures	

# 1. INTRODUCTION

## 1.1 Purpose and Scope

This document is the Software *Installation and User Manual (SUM)* for the Sentinel-2 Level-2A Prototype Processor, which is labelled **Sen2Cor** for **Sentinel 2** (atmospheric) **Correction**.

The prototype implementation for the Level 2A processing of Sentinel-2 imagery over land is a combination of state-of-the-art techniques for performing Atmospheric Corrections (AC, including Cirrus clouds and terrain correction), tailored to the Sentinel-2 environment and a Scene Classification (SC) module.

Sen2Cor performs a pre-processing of Level-1C (L1C) Top of Atmosphere (TOA) image data, and applies a scene classification an atmospheric correction and a subsequent conversion into an ortho-image Level-2A Bottom-Of-Atmosphere (BOA) reflectance product. Outputs are an Aerosol Optical Thickness (AOT) map, a Water Vapour (WV) map and a Scene Classification map together with Quality Indicators data. Details of the products and its contents is provided in [L2A-PDD] of section 1.3.

Level-2A (L2A) products are re-sampled as L1C products with a constant GSD (Ground Sampling Distance) of 10m, 20m and 60 m according to the native resolution of the different spectral bands. If applicable, Level-2A products are provided for each MSI channel at coarser resolution (i.e. 20 m and 60 m) as well.

A large database of look-up tables (LUTs) has been compiled using an atmospheric radiative transfer model based on libRadtran<sup>1</sup>. The LUTs are generated for a wide variety of atmospheric conditions, solar geometries, and ground elevations and are calculated with a high spectral resolution of 0.6 nm. This database has been subsequently resampled with the Sentinel-2 spectral responses, in order to obtain the sensor-specific functions needed for the atmospheric correction.

This Installation and User Manual mainly addresses the following aspects of the software:

1. Introduction into the main functionality of the system
2. Description of the system functionality, operations, architecture and modules
3. Overview of the Sen2Cor system and its operational environments:
  - Description of the Software Development Environment (SDE)
  - Description of the Source Distribution (SCD)
  - Description of the Runtime Environment (RTE)
4. Installation, configuration and operation of SDE, SCD and RTE
5. Creation of distribution media for all environments

The configuration parameter and their descriptions are now part of the IODD [L2A-IODD]

---

<sup>1</sup> <https://www.libradtran.org>

#### Restrictions:

1. The Look Up tables are only compiled for the rural aerosol and the mid-latitude summer (MS) atmosphere with its corresponding ozone column (331 DU for sea level. Other Look up Tables can be generated on request but are not part of the current baseline.
2. The current version of the processor will not give validated results over water or on coastal regions. The design of the current implementation does not prevent such an extension, but this is not part of the current baseline.

## 1.2 Document Overview

The configuration and user manual consists of the following chapters and sections:

Chapter, Section	Description
1	this chapter
2	Introduces the Sen2Cor system. What is the general purpose of the application, how is it structured, what are the processing schemes. It lists the general system architecture modules and functionality and gives a brief overview on its functionality and operation.
3	Introduces the two different environments (runtime and development) of the system in general and the source code distribution.
3.1	How to install and configure the processor software and the according runtime environment.
3.2	How to run the processor software in the according runtime environment.
3.3	Describes the setup configuration and operation of the software development environment, set up on top of Eclipse and PyDev. Contains also a detailed overview on all used third party tools.

With these sections the configuration and user manual enables developers to upgrade and maintain the software and users of the software to operate the system within their specific hardware environment.

What this document will not provide is the scientific background of the application. This is part of the corresponding ATBD [L2A-ATBD], see below. Also the configuration of the processor has been moved into [L2A-IODD]. In order to avoid redundancies and inconsistencies between the different project documents, this content will thus not be repeated here. If it is necessary for the understanding of the operation, this SUM will refer to the according chapters of [L2A-IODD], [L2A-PDD] and [L2A-DPM].

## 1.3 Documentation and Definitions

The reference list of all project related documents with their full version numbers and issue dates is given in:

[L2A-GLODEF] S2PAD Project Glossary, S2PAD-VEGA-GLO-0001, version 3.3, 31.03.2014

### **1.3.1 Normative Reference Documents**

[L2A-IODD]	Sentinel-2 MSI – Input Output Data Definition
[L2A-PFS]	Sentinel-2 MSI – Product Format Specification
[L2A-PDD]	Sentinel-2 MSI – Level-2A Product Definition Document
[L2A-ATBD]	Sentinel-2 MSI – Level 2A Products, Algorithm Theoretical Basis
[L2A-PERF]	Sentinel 2 MSI – Performance Tests on 64 and 32 bit Python

### **1.3.2 Informative Reference Documents**

Document

[L2A-DPM]	Sentinel-2 MSI – Level 2A Detailed Processing Model
-----------	---

## 2. FUNCTIONALITY AND OPERATION

Sen2Cor is a prototype processor for Sentinel-2 Level 2A product formatting and processing. The processor performs the tasks of atmospheric-, terrain and cirrus correction and a SC of Level 1C input data. Level 2A outputs are: Bottom-Of-Atmosphere (BOA), optionally terrain- and cirrus corrected reflectance images, AOT-, WV-, SC maps and Quality Indicators, including cloud and snow probabilities. The Level 2A Product Formatting performed by the processor follows the specification of the Level 1C User Product. Details are given in [L2A-PFS].

The Sentinel-2 Multi-Spectral Instrument (MSI) consists of 13 spectral bands with three different resolutions (10m, 20m and 60m) as shown in Figure 2-1. The instrument covers a 290 km swath. The Level-1C image product, which serves as the input for the Level-2A processing consists of a series of n tiles, each with a 100 km square. Each tile consists of thirteen compressed JPEG-2000 images, each image representing one single band. The thirteen bands have three different resolutions (10m, 20m and 60m) which lead to different image dimensions of the Level-1C input product. These details are given in [L2A-PDD] and [L2A-IODD].

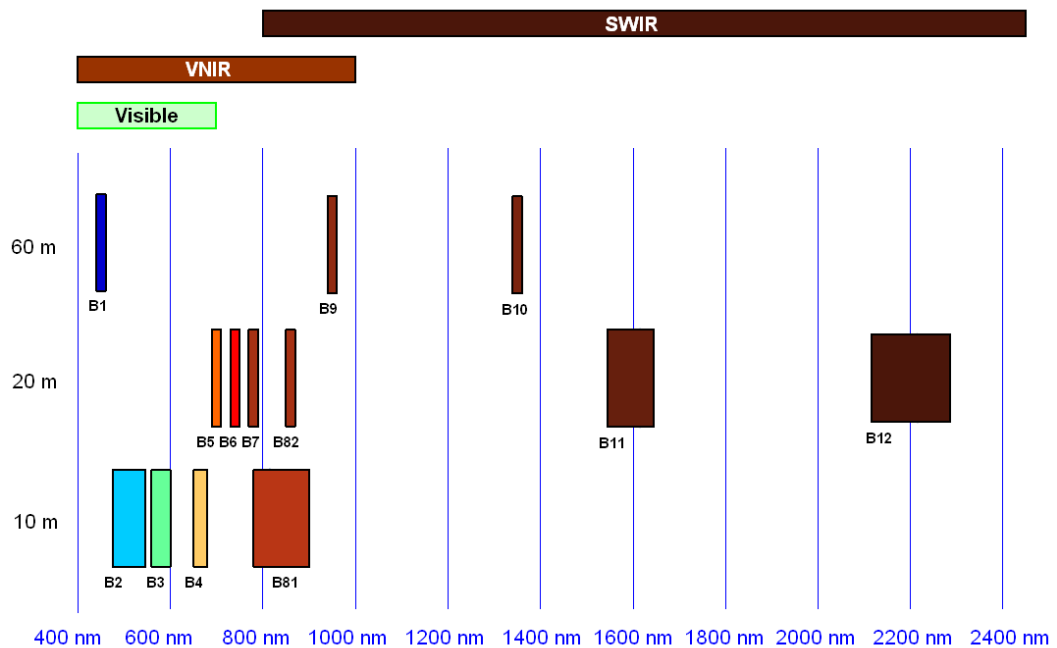


Figure 2-1 – Sentinel-2 Spectral Bands and Resolutions

Figure 2-2 below shows the main processing workflow. After reading and processing the input parameter and data the main processing module triggers the creation of an internal temporary database, which is then used by the SC and the AC module to retrieve and to store the data and intermediate products. The processing can act in a loop, dependent on the number of different product resolutions to be generated.

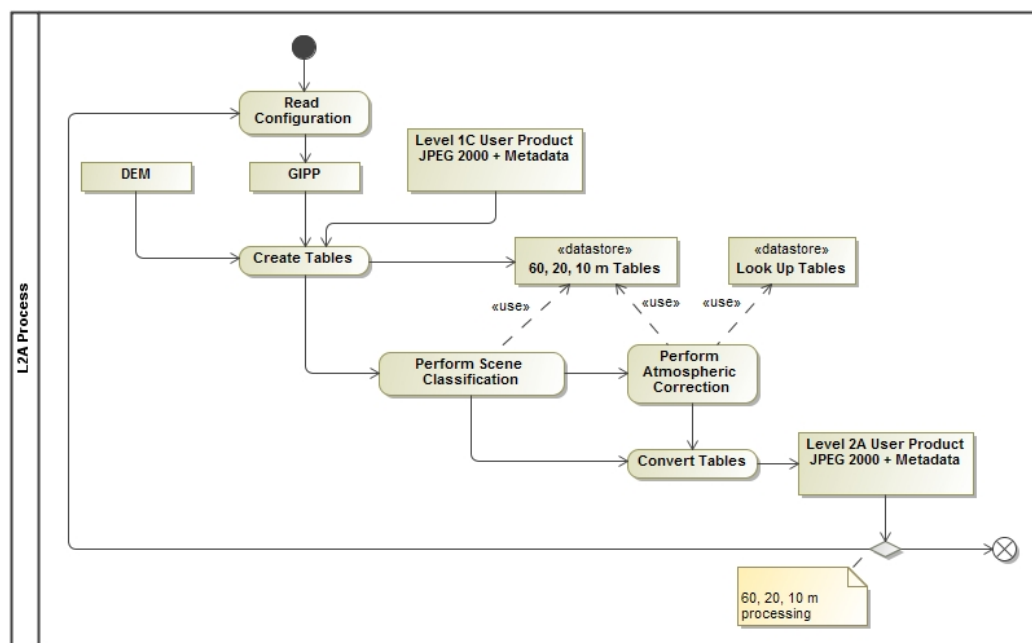
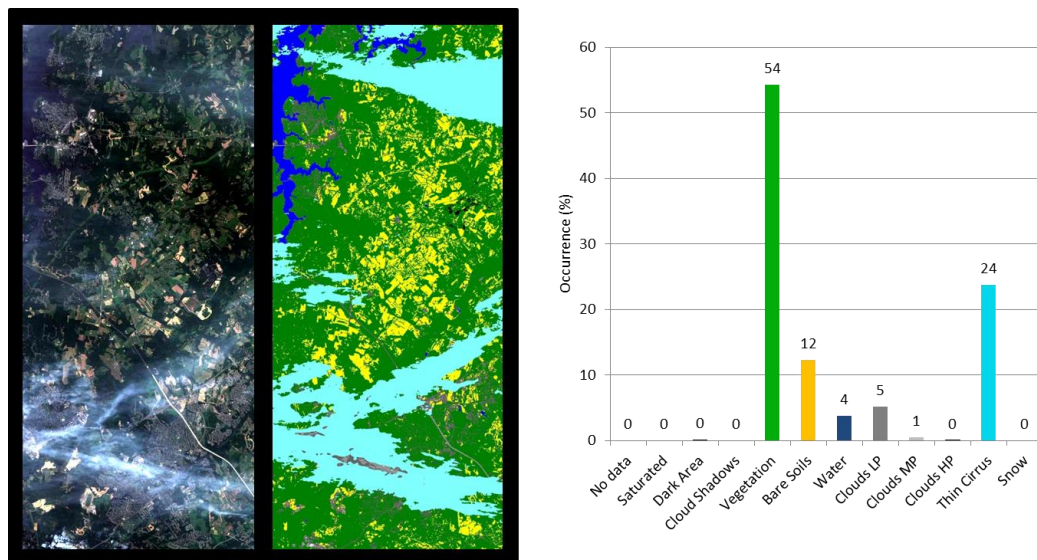


Figure 2-2 – Processing Flow, Overview

## 2.1 Scene Classification (L2A\_SceneClass)

The SC algorithm allows to detect clouds, snow and cloud shadows and to generate a classification map, which consists of 4 different classes for clouds (including cirrus), together with six different classifications for shadows, cloud shadows, vegetation, soils / deserts, water and snow. The algorithm is based on a series of threshold tests that use as input top-of-atmosphere reflectance from the Sentinel-2 spectral bands. In addition, thresholds are applied on band ratios and indexes like the Normalized Difference Vegetation - and Snow Index (NDVI, NDSI [3]). For each of these thresholds tests, a level of confidence is associated. At the end of the processing chain a probabilistic cloud mask quality map and a snow mask quality map is produced. The algorithm uses the reflective properties of scene features to establish the presence or absence of clouds in a scene. Cloud screening is applied to the data in order to retrieve accurate atmospheric and surface parameters, either as input for the further processing steps below or for being valuable input for processing steps of higher levels. Figure 2-2 below shows the results of a SC (right side) based on modified AVIRIS test data (left side). Twelve different classifications are provided.



**Figure 2-3 – Scene Classification**

The generated classification map is specified as follows:

**Table 2-1 – Classification Map**

Label	Classification
0	NO_DATA
1	SATURATED_OR_DEFECTIVE
2	DARK_AREA_PIXELS
3	CLOUD_SHADOWS
4	VEGETATION
5	BARE_SOILS
6	WATER
7	CLOUD_LOW_PROBABILITY
8	CLOUD_MEDIUM_PROBABILITY
9	CLOUD_HIGH_PROBABILITY
10	THIN_CIRRUS
11	SNOW

Associated quality indicators on snow and cloud probability are generated from the results. These Quality indicators calculate the probability (0-100%) that the earth surface is obstructed by clouds or optically thick aerosol (ice or snow).

The SC processing consists of six different steps:

1. Snow detection;
2. Cloud detection;
3. Cloud shadow detection;
4. Cirrus detection;
5. Classification map generation.

The processing is shown in the flow diagram below.

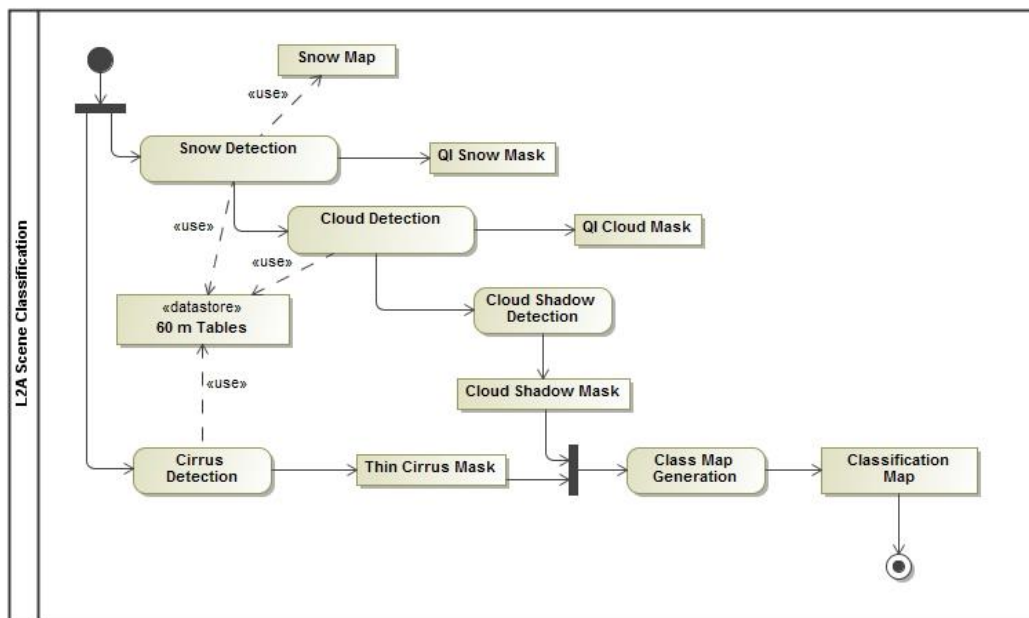


Figure 2-4 – Scene Classification, Processing Flow

Details of this algorithm, especially on the different threshold conditions are given in chapter 3 of [L2A-ATBD] and chapter 2.5 of [L2A-DPM]

## 2.2 Atmospheric Correction (L2A\_AtmosCorr)

The AC processing consists of a set of four different subtasks, (AOT, WV and terrain retrieval (optional with terrain and cirrus correction, having three different user products as output: AOT and WV tables on pixel level and the BOA corrected reflectance images for all bands measured. Figure 2-5 below shows the processing flow for the atmospheric correction module.

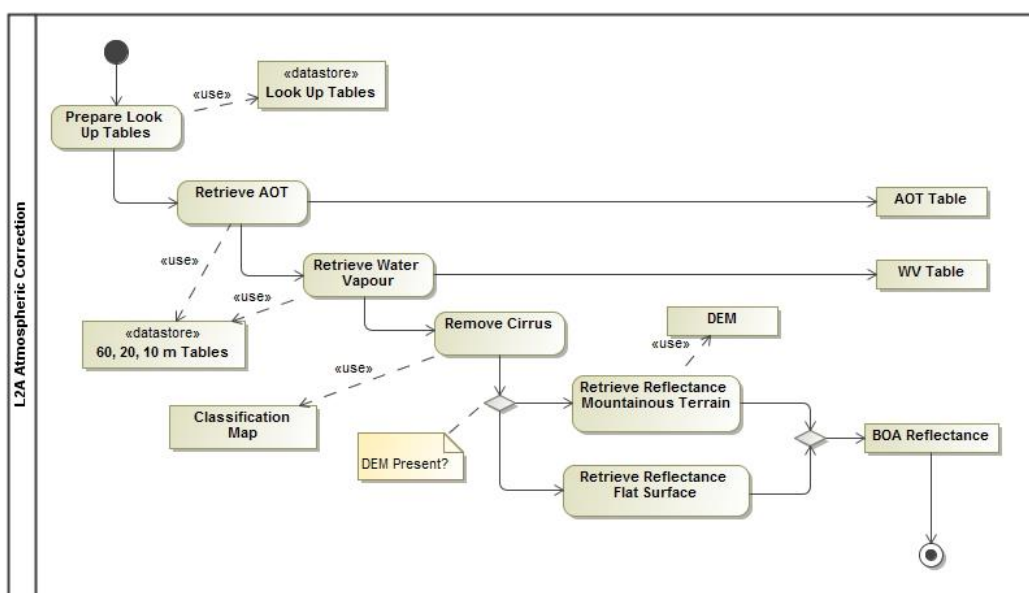


Figure 2-5 – Atmospheric Correction, Processing Flow



## 2.2.1 Look Up Table Generation

The L2A\_Atmos module is dependent on calculating a database of radiative transfer function for different sensor and solar geometries, ground elevations, and atmospheric parameters. The following list presents a 6-dimensional parameter space and the grid spacing for each parameter. The atmospheric correction processor reads the Look Up Tables (LUTs) pertaining to this parameter space and interpolates, if required. The LUTs are generated via libRadtran, a library for the calculation of solar and thermal radiation in the Earth's atmosphere.

**Table 2-2 – Parameter space for atmospheric correction**

Parameter	Range	Increment / grid points
Solar zenith angle	0 -70°	10°
Sensor view angle	0 -10°	10°
Relative azimuth angle	0 -180°	30° (180° = backscatter)
Ground elevation	0 -2.5 km	0.5 km
Visibility	5 -120 km	5, 7, 10, 15, 23, 40, 80, 120 km
Water vapour, summer	0.4 -5.5 cm	0.4, 1.0, 2.0, 2.9, 4.0, 5.0 cm
Water vapour, winter	0.2 -1.5 cm	0.2, 0.4, 0.8, 1.1 cm

Baseline processing of the current processor version is the rural/continental aerosol type. Other Look-Up tables can also be used according to the scene geographic location and climatology, but are not present at this first release.

## 2.2.2 Aerosol Optical Thickness

AOT retrieval provides a measure for the visual transparency of the atmosphere. It is derived using the DDV (Dense Dark Vegetation) algorithm [5], using the short wave infrared (SWIR) band 12 and correlates its reflectance with bands 4 (red) and 2 (blue). The algorithm requires that the scene contains reference areas of known reflectance behaviour, preferably Dark Dense Vegetation (DDV) and/or dark soil and water bodies.

The algorithm starts with a user-defined visibility (default: 20 km) as input. If the scene contains no dark vegetation or soil pixels, the surface reflectance threshold of band 12 will be successively iterated in order to include medium brightness reference pixels in the sample. If the scene contains no reference and no water pixels the scene is processed with the start visibility instead. The algorithm delivers an AOT map as shown in Figure 2-6 below.

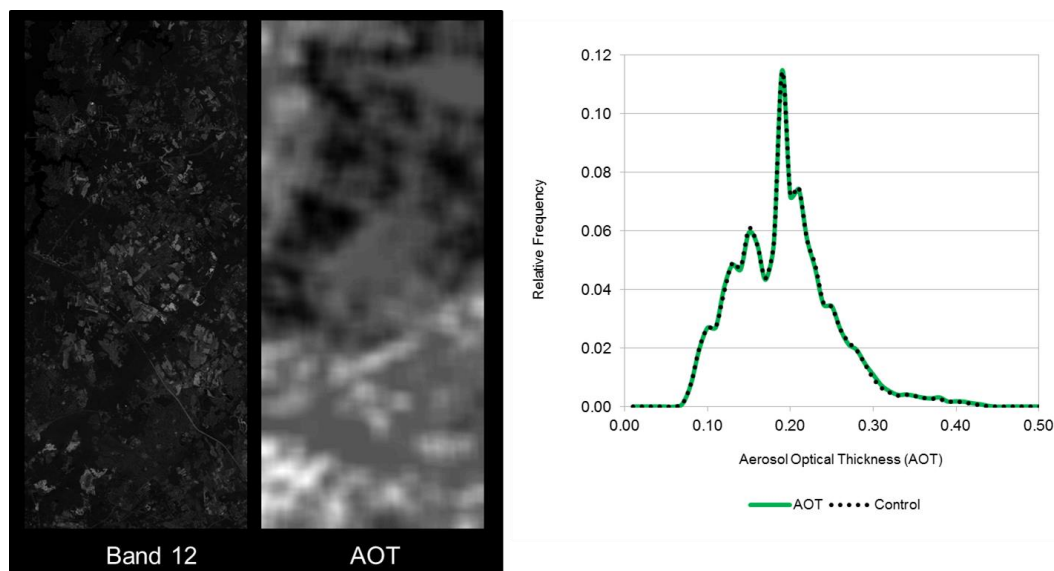


Figure 2-6 – AOT Retrieval using Band 12

### 2.2.3 Water Vapour Retrieval

WV retrieval over land is performed with the Atmospheric Pre-corrected Differential Absorption algorithm (APDA, [6]) which is applied to the two Sentinel-2 bands B8a, and B9 (Fig. 4). Band 8a is the reference channel in an atmospheric window region. Band B9 is the measurement channel in the absorption region. The absorption depth is evaluated by calculating the radiance for an atmosphere with no WV, assuming that the surface reflectance for the measurement channel is the same as for the reference channel. The absorption depth is then a measure of the WV column content.

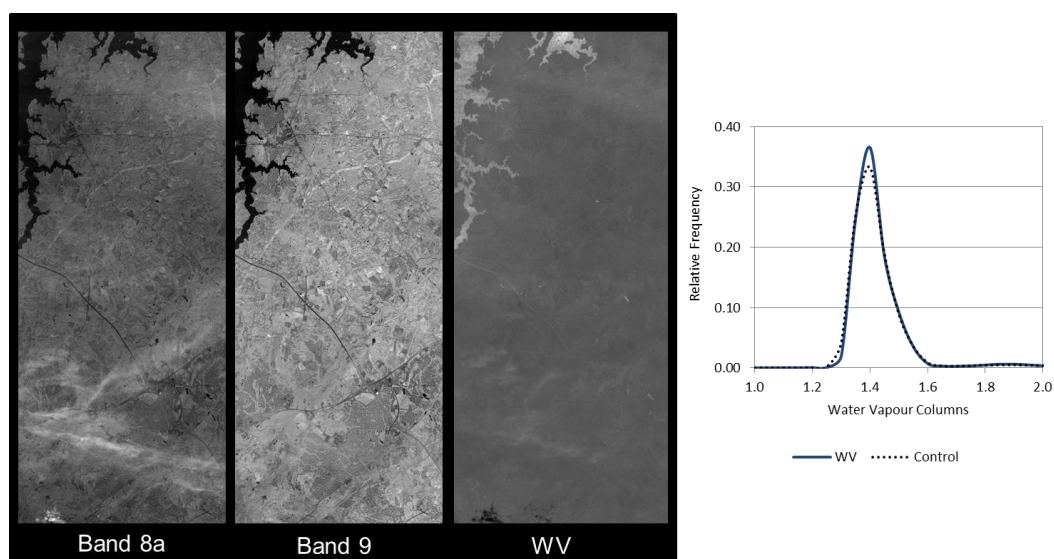


Figure 2-7 – WV Retrieval using Bands 8a and 9

### 2.2.4 Cirrus Correction

The Cirrus Correction algorithm uses the sentinel (cirrus) channel 10. Thin cirrus clouds affect the visible, near- and shortwave infrared spectral regions. They are partially

transparent and thus difficult to detect with broad-band multispectral sensors, especially over spatially inhomogeneous land areas.

WV, in contrast, dominates in the lower troposphere of 0-5 km. A narrow spectral band in a spectral region of very strong WV absorption (Band 10) will thus absorb the ground reflected signal, but will receive the scattered cirrus signal.

Cirrus reflectance of band 10 can therefore be correlated with other bands in the VNIR and SWIR region and the cirrus contribution can thus be removed from the radiance signal to obtain a cirrus-corrected scene. This is shown in Fig. 5 below as a qualitative result.

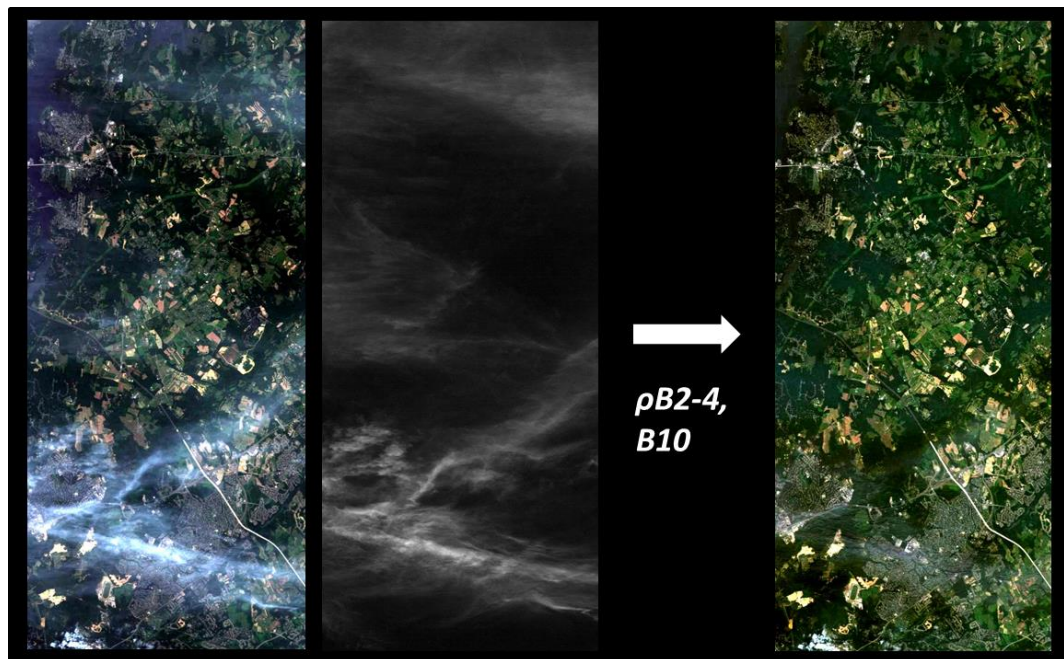


Figure 2-8 – Cirrus Correction, Bands 2-4 with Band 10

### 2.2.5 Surface Reflectance Retrieval

Surface Reflectance retrieval is performed for each sequential Band B1 – B12. Figure 2-9 below shows the Level 1C input data, their scene classification and the corresponding Level 2A output. Figure 2-10 shows the corresponding analysis of the reflectance spectra before (left) and after atmospheric correction (right), using the Scene Classifier for the pixel separation of the three areas for Bare Soils, Vegetation and Water.

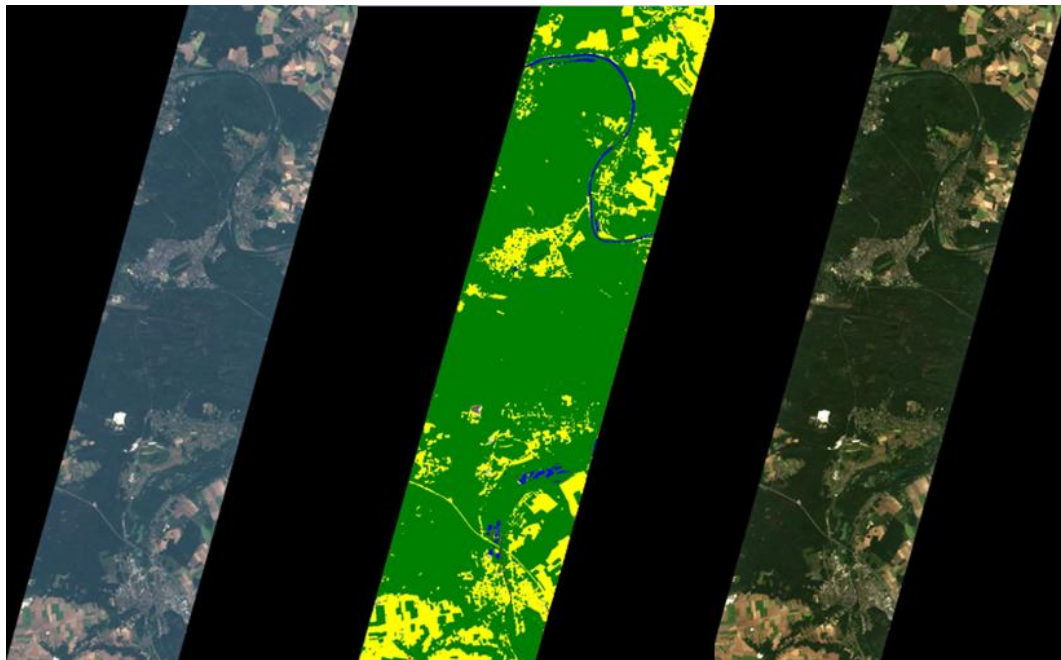


Figure 2-9 – Left: Level 1C Input, Bands 2-4, RGB composite image; middle: Scene Classification, right: Level 2A Output, Bands 2-4, RGB composite image

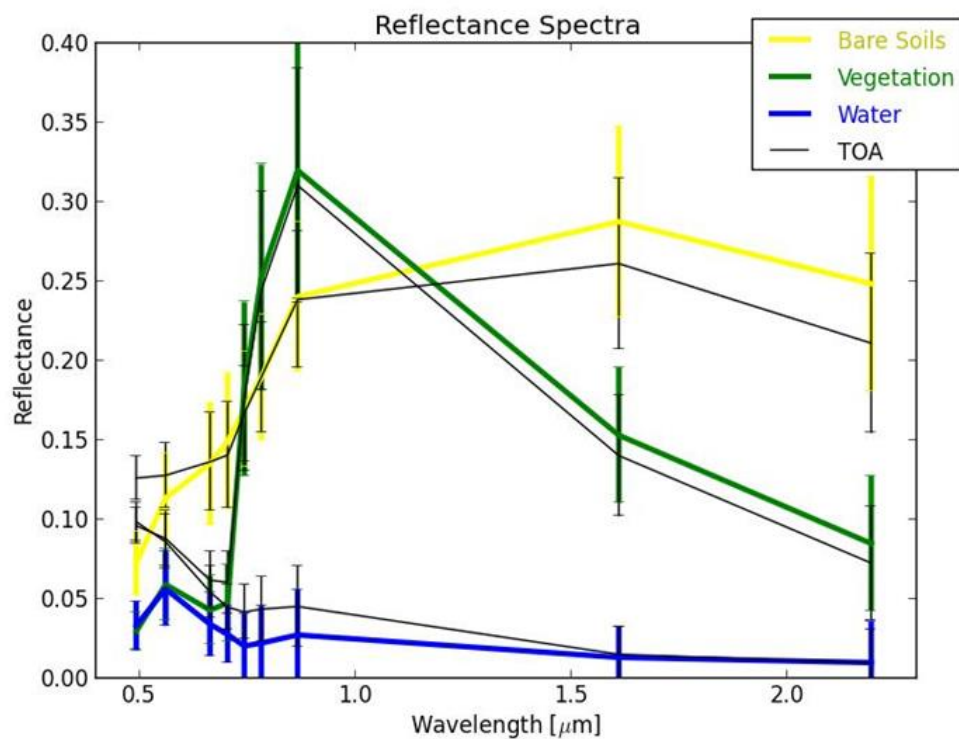


Figure 2-10 – BOA reflectance spectra for soils, vegetation and water, separated by Scene Classification; Black: TOA Values

## 2.2.6 Terrain Correction

The algorithm for rugged terrain requires the existence of an appropriate Digital Elevation Map (DEM) and a set of derived products like slope, aspect and terrain shadow maps. The retrieval of the water vapour map also includes this terrain elevation.

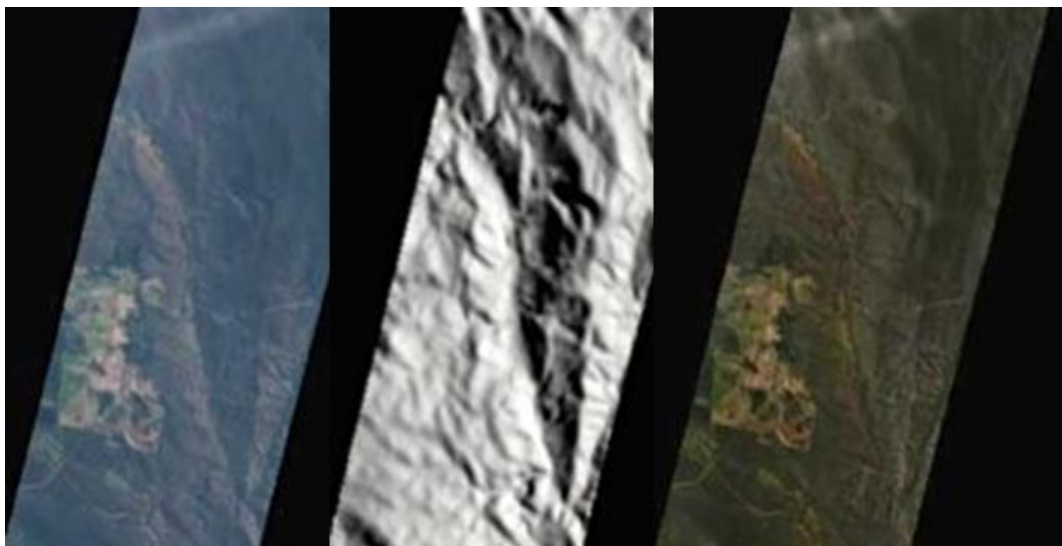
SEN2COR can optionally read SRTM data from a database. The retrieval of these products works online, during processing or from a private archive. As an example the 90m Digital Elevation Database from CGIAR-CSI (<http://www.cgiar-csi.org/>) is used for the generation of the DEM shown in Figure 2-11 below.

SEN2COR reads the geo-coordinates and angular information from the Level-1C metadata. The area of interest is created and fitted to the input images according to the information retrieved from the metadata and the conversion is performed using the GDAL libraries from OSGeo.

The retrieval of the spectral reflectance cube consists of the following steps:

1. iterations for terrain reflectance;
2. empirical BRDF correction;
3. adjacency correction;
4. spherical albedo correction.

The algorithm retrieves one or more DEMs from an external database, reformats resamples and assembles the DEMs based on the geo locations obtained either from the Level-1C metadata or from the JPEG-2000 images. Finally, it creates the according slope, aspect and a hill shadow map using the GDAL tools.



**Figure 2-11 – Terrain correction; Left: Level 1C Input, Bands 2-4, RGB composite image, middle: DEM, reshaped to according scene, right: Level 2A Output, Bands 2-4, RGB composite image.**

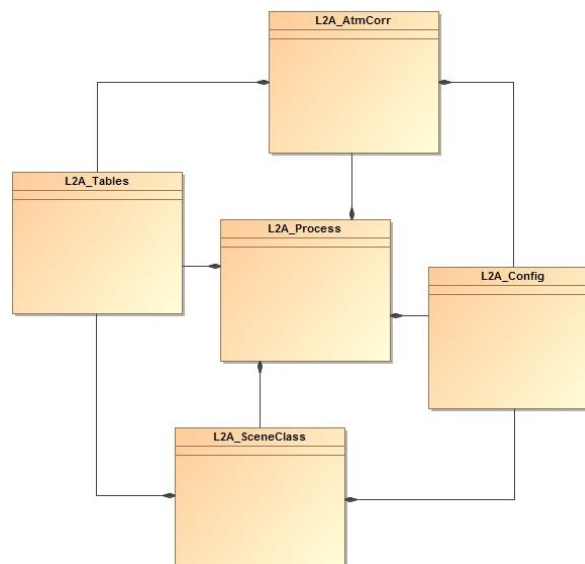


## 2.3 Level-2A Processor Architecture

The processor currently operates on tiling level. An extension of the processing on data strip level is proposed in Appendix A of [L2A-ATBD] and will be implemented in a next release. It follows a high level description of the processor architecture:

The Sen2Cor application is designed by the following five essential modules (classes):

1. **L2A\_Process:** the general operator module, which coordinates the interaction between the other modules.
2. **L2A\_SceneClass:** performs the coarse classification of the input images into their different contents like clouds, snow, water, soil etc. and provides statistical analysis.
3. **L2A\_AtmosCorr:** transforms the input from top of atmosphere (TOA) to bottom of atmosphere (BOA) representation and performs the atmospheric correction of the input.
4. **L2A\_Config:** a helper class providing the configuration parameters to the other modules.
5. **L2A\_Tables:** a helper class, managing the conversion of the JPEG-2000 based input data to an internal format (and vice versa) and providing a high performance access to the data for the other modules (see section 2.6.1).



**Figure 2-12 – High Level processor architecture**

The purpose, functioning and interaction between these five modules have been shown in the context for the Figs. 2.2, 2.4 and 2.5. Additionally to these five essential modules, a set of helper routines, libraries and schema files are part of the software. Table 2-3 below shows a comprehensive listing of all modules.

Table 2-3: Listing of software modules

Module	Description
Folder <cfg>:	
L2A_GIPP.xml	GIPP configuration file
Folder <src>:	
L2A_AtmCorr.py	Main modules as described above
L2A_Config.py	
L2A_Process.py	
L2A_SceneClass.py	
L2A_Tables.py	
Libraries and Utilities:	
L2A_Borg.py	Base class for main modules (see above)
L2A_Library.py	Helper routines as used by the atmospheric correction module
L2A_XmlParser.py	Convenience functions for easy access of xml data structures
Folder <lib>	
./10/h99000_wv29_rura.atm	libRadtran Look up Tables for 10 m resolution, see L2A_IODD for details.
./10/h99000_wv04_rura.atm	
./10/h99000_wv10_rura.atm	
./10/h99000_wv20_rura.atm	
./10/h99000_wv40_rura.atm	
./10/h99000_wv50_rura.atm	
./20_60/h99000_wv29_rura.atm	libRadtran Look up Tables for 20 and 60 m resolution, see L2A_IODD for details.
./20_60/h99000_wv04_rura.atm	
./20_60/h99000_wv10_rura.atm	
./20_60/h99000_wv20_rura.atm	
./20_60/h99000_wv40_rura.atm	

Module	Description
./20_60/h99000_wv50_rura.atm	
./20_60/GlobalSnowMap.tiff	
<b>Folder &lt;templates&gt;</b>	
S2_PDI_Level-1C_Datastrip_Metadata.xsd	XSD scheme files for product format description, see L2A_PFS for details
S2_PDI_Level-1C_Tile_Metadata.xsd	
S2_PDI_Level-2A_DataStrip_Metadata.xsd	
S2_PDI_Level-2A_Tile_Metadata.xsd	
S2_User_Product_Level-1C_Metadata.xsd	
S2_User_Product_Level-2A_Metadata.xsd	

## 2.4 Product Formatting

The Level 2A Product format is closely related to the Level 1C Top-of-Atmosphere (TOA) reflectance product which serves as an input to the processor. It consists of 13 JPEG-2000 images, associated to the 13 Sentinel-2 spectral bands at three different spatial resolutions with a ground sampling distance of 10, 20, and 60m.

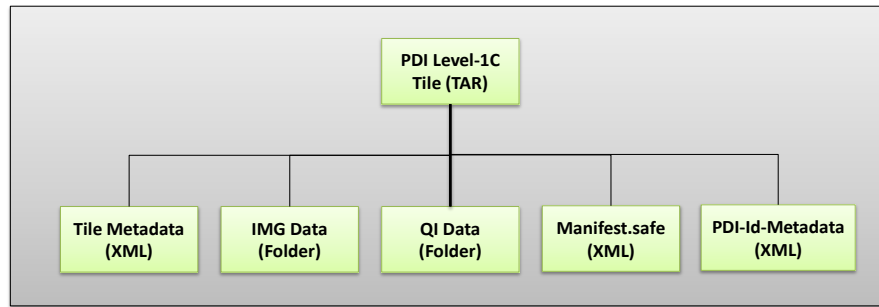
The generated Level 2A BOA reflectance output images are resampled and generated with an equal spatial resolution for all bands, based on three user selectable resolutions of 10, 20 and / or 60m.

The tile level contains different components, based on the user selected resolution:

- a 10 m resolution product contains spectral bands 2, 3, 4 and 8 and an AOT map resampled from 20m
- a 20 m product contains band 2 – 7, the bands 8a, 11 and 12 and an AOT and WV map
- a 60m product contains all components of the 20m product and additionally the 60m bands 1 and 9

The Cirrus band 10 will be omitted in the Level 2A output, as it does not contain surface information. Fig. 2 shows the Level 2A user product on tile level.





**Figure 2-13 – Filing structure of Level-1C product on tile level**

The complete specifications for all inputs are provided in [L2A-PFS] and [L2A-IODD]. Level-2A Inputs are derived from the Level-1C data schemes as described in [S2-PFS], section 0.

## 2.5 Runtime Configuration

Ground Image Processing Parameter (GIPP) are configured in an XML file named L2A\_GIPP.xml, located in the <cfg> subdirectory of the users workspace (see example in the appendix), where they can be configured. For each processed Level 2A tile the GIPP xml file will be renamed to S2A\_USER\_GIP\_L2A\_TL\_<TILE\_ID> (see section A.2.3 of [L2A\_PFS]) and subsequently copied into the AUX\_DATA subfolder of the corresponding granule. The GIPP are listed in their current context. These parameters, together with all input, output and other auxiliary data are comprehensive listed in the corresponding [L2A-IODD] document and thus not repeated here.

## 2.6 Processing Scheme and Logical Flow (L2A\_Process)

The L2A\_Process module keeps references of the four processor main modules and coordinates the interaction among the modules. It also determines the sequential steps in the processing of the different products according to the user requests on product generation.

The image processing is performed in up to three different processing cycles, dependent on the user's requirements for the image resolution to be processed <res == 10, 20 or 60m>. For the highest resolution (10m), the processing of the two lower processing cycles (20m and 60m) in advance is always a necessary intermediate step, as the algorithm needs the output of the lower resolutions as necessary input data (see section 2.4 of [L2A-ATBD]). The 20m product requires the pre-processing of the 60m product only. The 60m product can be obtained in one single processing cycle. The processing always checks, whether a lower resolution product is already available <20proc, 60proc>, triggers the generation of the lower resolution product(s) only, if they are not already apparent and finally performs the generation of its own product. If a 20m product can be found, it is implicit that a 60m product has also been generated in ahead. The processing chain is summarized in Figure 3

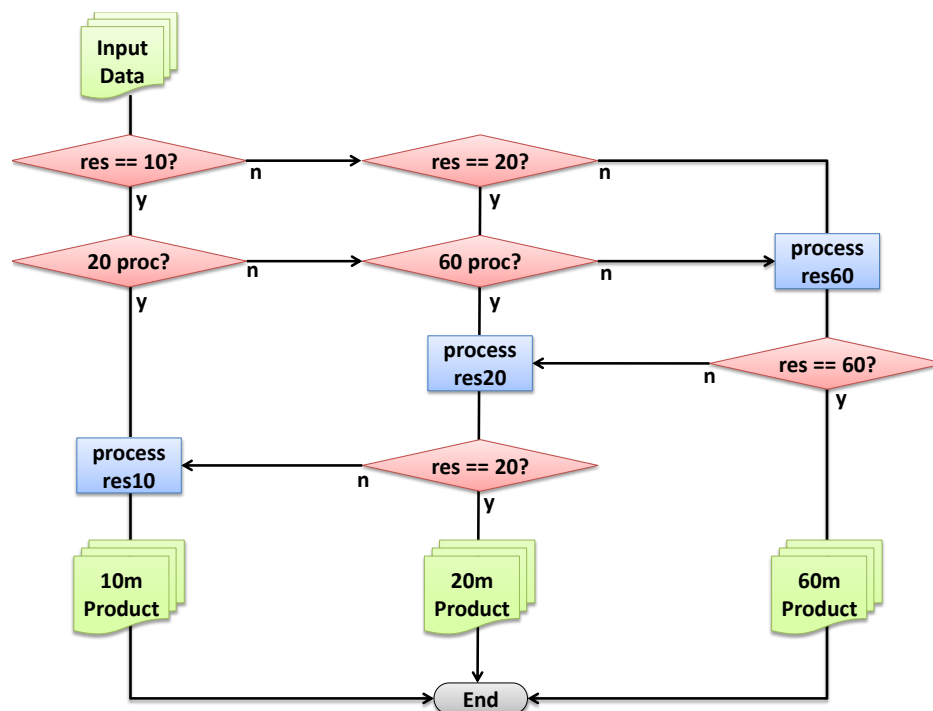


Figure 2-14 – Processing chain for the three different resolutions

### 2.6.1 Pre-processing (L2A\_Tables)

The Level-1C input data is expected to be present in a folder structure at product level, as specified in section 3.1 of [L2A-PDD]. The location of the Level-1C input archive can be specified via the command line argument.

Due to the three different resolutions of the Level-1C input images, conversion routines will serve for an up- and down-sampling of parts of the images into the appropriate resolution as well as the transfer from JPEG-2000 into a different internal format (see below). This has been implemented using the OpenJPEG library, which will be installed during the Sen2Cor setup.

Due to the relative huge size of the image data (especially for the high resolution 10 bands B02 – B04 and B08) data in- and output has been shown as rather time and memory consuming.

An intermediate data conversion thus was selected which converts the JPEG-2000 images and the necessary metadata (like the DEM) into an internal format based on HDF5. HDF5 is a data model, library, and file format for storing and managing data. It is especially designed for flexible and efficient I/O and for high volume and complex data. HDF5 is portable and extensible, allowing applications to evolve in their use and platforms to be supported.

The interfacing with the internal data format is implemented via PyTables. PyTables is a package for managing hierarchical datasets, designed to efficiently cope with extremely large amounts of data. It is built on top of the HDF5 library, the Python language and the NumPy package. PyTables and HDF5 are described in section 3.3.1.7

### 2.6.2 The 60 m Product Processing

The 60m product processing takes the three 60m bands B01, B09 and B10 and a 60m DEM as inputs and converts them directly into the internal HD5 format. The three 10m

bands B02-B04 will first be down-sampled to 60m. The same is true for the 20m bands B05 – B8A and B11, B12. The 10m band B08 is not used for the 60m processing.

Beneath the twelve optical channels (the cirrus channel B10 is excluded as it does not represent surface information) the 60m product processing generates a SC map according to the classification below, quality indicators for presence of snow and clouds (in percentage), an Atmospheric Optical Thickness (AOT), a WV map, a Visible Index file and a preview image, covering the three visible bands 2-4 within an JPEG compressed image, having a 320m resolution.

### **2.6.3 The 20 m Product Processing**

The 20m product processing takes the 20m bands B05 – B8A and B11, B12 and a 20m DEM as inputs and converts them directly into the internal HD5 format. The three 60m bands B01, B09 and B10 will first be up-sampled to 20m, the three 10m bands B02-B04 will be down-sampled to 20m. The 10m band B08 is not used for the 20m processing.

The 20m product processing covers nine optical channels a 20m AOT and a WV and a Visibility Index file corresponding to the AOT. The three resampled 60m bands B01, B09 and B10 are omitted in order to avoid spectral artefacts due to mixed signatures and resampling.

### **2.6.4 The 10 m Product Processing**

Inputs for the 10m product processing are the four 10m bands B02-B04 and B08, a 10m DEM and an AOT, up-sampled from 20m. As the WV influence is very small, only the scene-average WV needs to be used for the surface reflectance retrieval.

The 10m product processing covers only four optical channels and a 10m AOT and WV map and the Visible Index map. The other channels, not used for the calculation will be omitted.

### **2.6.5 Post-Processing**

The OpenJPEG Library is used for generating the final L2A-Product, transferring the internal HDF5 based tables back into the JPEG-2000 format. It keeps for all generated products the final (resampled) resolution.

Post-processing also packs the folder back to a Level-2A archive which is shown in its unpacked version in Figure 2-15. This archive will be placed in parallel to the Level-1C input archive. The location of the Level-2A output archive can be configured in the L2A\_GIPP.xml configuration file, which can be found in the path where the SEN2COR\_HOME environment variable points to.

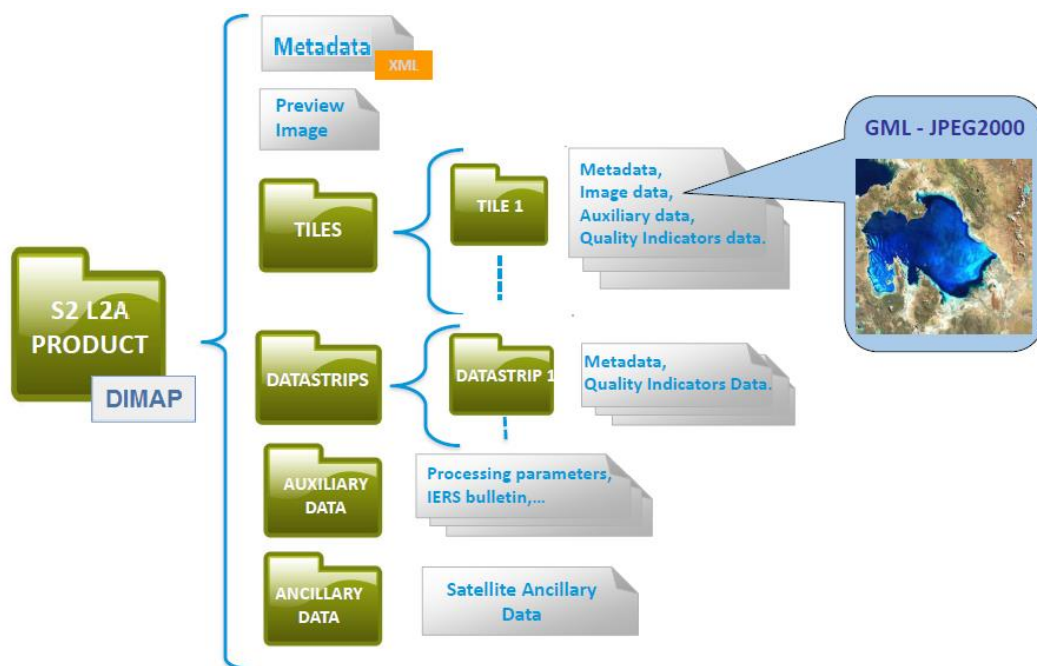


Figure 2-15 – Level 2A product, physical format Configuration (L2A\_Config)

## 2.7 Logging (Logger)

The module L2A\_Config keeps an instance of a logger object. Each of the processing modules writes its own diagnosis and status messages into a common log-file, located in the QI\_DATA folder of each generated L2A tile.

The log level can be configured in the GIPP (L2A\_GIPP.xml) as:

DEBUG, INFO, WARNING, ERROR, CRITICAL, NOTSET

The log-level is hierarchical: if e.g. set to **DEBUG**, all higher messages such as **INFO**, **WARNING**, **ERROR** or **CRITICAL** will be displayed as well. Setting the log-level to **ERROR** displays only **ERROR** and **CRITICAL** messages.

Beneath the message itself, the logger displays the system time stamp when the message was generated, the log-level, the module which has generated the message and the function and code line of the module which has generated the message.

### 3. CONFIGURATION AND INSTALLATION

The Sen2Cor software is a command-line oriented application, written in the programming language python. It consists of a set of modules, as listed in section 2.3. Due to license restrictions the module for the atmospheric correction is a binary library, containing those parts of the algorithm not public available. Table 3-1 lists the third party products for the runtime environment (RTE) for the three Operating systems Linux, Mac OS X and Windows (64 bit is a requirement).

**Table 3-1 – Third Party products for the RTE (64 bit)**

Operating System			
Product	Name	Version	Source
Linux (alternatives)	CentOS	>= 6.0	<a href="http://www.centos.org/">http://www.centos.org/</a> <a href="http://www.de.redhat.com/products/rhel/desktop/">http://www.de.redhat.com/products/rhel/desktop/</a> <a href="http://www.ubuntu.com/download/server">http://www.ubuntu.com/download/server</a>
	RHEL	>= 6.0	
	Ubuntu	>= 12.04.2 LTS 64 bit	
Mac OS X	Yosemite	>= 10.10.3	<a href="http://www.apple.com/osx/">http://www.apple.com/osx/</a>
Windows	Windows 7	Enterprise, 64 bit, SP1	
Operational Tools and Libraries			
Product	Name	Version	Source
Python Distribution Package (including Python 2.7.9, NumPy + SciPy, PyTables, Cython, Distutils and more ...	Anaconda	>= 2.2.0, 64 bit	<a href="http://continuum.io/downloads">http://continuum.io/downloads</a>
Conda GDAL package	GDAL	>= 1.11.2	<a href="https://conda.binstar.org/osgeo">https://conda.binstar.org/osgeo</a>
Conda Glymur oackage	Glymur	>= 0.8.0	<a href="https://binstar.org/sunpy/glymur">https://binstar.org/sunpy/glymur</a>
Image Processing and JPEG-2000 Support	OpenJPEG	== 2.1.0	<a href="http://www.openjpeg.org">http://www.openjpeg.org</a>

#### 3.1 Installation

In order to avoid dependency problems between the three different platforms supported and its according python packages, the Sen2Cor application works under the umbrella of the cross platform Python distribution Anaconda. Further information can be obtained here:

<https://store.continuum.io/cshop/anaconda/>

Current version of Anaconda at the writing of this SUM is: 2.2.0.

Current Python version used as a runtime environment at the writing of this SUM is 2.7.9

The sen2cor application is independent from the anaconda base installation, as all dependencies will be updated during the installation procedure. However, it is recommended always to start with the recent Anaconda version.

**Two important Remarks:**

- **It is strongly advised** to follow the recommendations choosing a **local Anaconda installation** on your machine, in order not to run into conflicts with the default python interpreter at your platform.
- **Avoid any trials to install the Sen2Cor application under a different python distribution outside of Anaconda.** There are numerous dependencies to distinct versions of GDAL, pyTables and the OpenJPEG library and to its configurations, which will very probable not be fulfilled by your default python installation.

The installation of the whole system is then performed in two steps:

### 3.1.1 Setting up the Runtime Environment

Download the recent version of the Anaconda python distribution for your operating system from:

<http://continuum.io/downloads>

and install it according to the default recommendations of the anaconda installer, which can be found here:

<http://docs.continuum.io/anaconda/install.html>

If you are working under Linux you should add the anaconda binary directory to your PATH environment variable at the end of the installation process. For the other installations (Windows and Mac) you will be asked if you want to allow this modification automatically. Please read the Anaconda installation procedure for your platform carefully before performing the installation.

If you already work with a different python interpreter outside of this new setup, you should add the PATH information for anaconda/bin to your environment only temporary. This is done as follows:

```
export PATH=<your installation directory>/anaconda/bin:$PATH
```

If you have no other python installed, you can add this line permanently, either to your .bashrc or .profile script in your home directory.

make sure that the the anaconda/bin directory is finally added to your path and check the proper installation by typing **conda -V**. If the command answers with the display of its version number, the installation was successful (the version number can be different). Example:

```
$: conda -V
```

```
conda 3.12.0
```

If this command can be found, you are ready for the second step, the installation of the processor. Continue with the next section.

The installation of the runtime environment is a one pass process. For subsequent installations of Sen2Cor you do not have to repeat the anaconda installation. Simply follow the steps in the next section.

### 3.1.2 Sen2Cor Installation

#### For Windows:

Download the zipped archive of the recent processor version from:

<http://s2tbx.telespazio-vega.de/sen2cor/>

and extract it with an unzip utility.

Step into the folder *sen2cor-<x.y.z>*, type

*python setup.py install*

and follow the instructions. The setup will install the Sen2Cor application and all its dependencies under the Anaconda python distribution.

At the end of the installation you can select your home directory for the Sen2Cor configuration data. This is by default:

*C:\Users\<your user account>\documents\sen2cor*

The setup script generates the following three environment variables:

**SEN2COR\_HOME**: this is the directory where the user configuration data are stored (see above). This can be changed later by you in setting the environment variable to a different location.

**SEN2COR\_BIN**: this is a pointer to the installation of the Sen2Cor package. This is located in the "site-packages" folder of Anaconda. **Do not change this.**

**GDAL\_DATA**: this is a pointer to the directory where the GDAL coordinate system info is stored. This is located in the

*site-packages\osgeo\data\gdal*

folder of Anaconda. **Do not change this.**

Open a new command line window, to be secure that your new environment settings are updated. From this new command line window, perform the following test:

Call the processor via "L2A\_Process --help". This will give you a list of possible further options.

If no errors are displayed, your installation was successful.

A basic L1C example test data set for further testing of the processor can be downloaded at:

<http://s2tbx.telespazio-vega.de/testdata/V12/L1C/>

**Workaround for problems with long pathnames:** Windows had a path length limitation of 260 characters in the past. Unfortunately, Windows Explorer on Windows 7 x64 is still subject to this path limit. As a consequence, the very long pathnames of a typical Level-1C or Level-2A user product can sometimes not be handled correctly. To overcome this issue the following workaround can be applied:

Map the directory of your input data to a network drive like:

**`C:\Users\<yourName>\AppData\Local\testdata`**

to a network drive, like e.g.: Z:

The following article gives an introduction how to map a folder to a network drive:

<http://windows.microsoft.com/en-us/windows/create-shortcut-map-network-drive#1TC=windows-7>

#### **For Linux and Mac:**

Download the zipped tar archive of the recent processor version from:

<http://s2tbx.telespazio-vega.de/sen2cor/>

and extract it with:

**`tar -xvzf sen2cor-<x.y.z>.tar.gz`**

Step into the folder **`sen2cor-x.y.z`**, type **`python setup.py install`** and follow the instructions. The setup will install the Sen2Cor application and all its dependencies under the Anaconda python distribution.

At the end of the installation you can select your home directory for the Sen2Cor configuration data. By default this is the directory where your **`$HOME`** environment variable points to.

The setup script generates a script called **`L2A_Bashrc`** and places it into your home directory. It contains the following three environment variables:

**`SEN2COR_HOME`**: this is the directory where the user configuration data are stored (see above). This can be changed later by you in setting the environment variable to a different location.

**`SEN2COR_BIN`**: this is a pointer to the installation of the Sen2Cor package. This is located in the “site-packages” folder of Anaconda. Do not change this.

**`GDAL_DATA`**: this is a pointer to the directory where the GDAL coordinate system info is stored. This is located in the **`site-packages/osgeo/data/gdal`** folder of Anaconda. Do not change this.

These settings are necessary for the execution of the processor. There are two possibilities how you can finish the setup:

1. You can call this script automatically via your **`.bashrc`** or **`.profile`** script (OS dependent). For this purpose, add the line:



`source <location of your script>/L2A_Bashrc`

to your script.

2. You can call this script also manually via `source L2A_Bashrc` every time before starting the processor or include it into a script of your choice.

Finally, to check the installation after sourcing the `L2A_Bashrc`, call the processor via

`L2A_Process --help.`

This will give you a list of possible further options how to operate.

A basic L1C example test data set for further testing of the processor is provided for download at:

<http://s2tbx.telespazio-vega.de/testdata/V12/L1C/>

### 3.1.3 Configuration Files

Located under `$SEN2COR_HOME/cfg` the GIPP file of the application (which is an xml formatted file) can be configured by the user for individual purposes. If a different configuration shall be used, the `$SEN2COR_HOME` directory can be reconfigured to a different directory:

`export SEN2COR_HOME = <directory of your choice>`

This allows the operation with multiple configuration settings.

### 3.1.4 Uninstallation

If you have only one version of Sen2Cor installed, uninstallation is relatively easy: simply type:

`pip uninstall sen2cor`

from your command line, and the application will be removed.

If you have multiple versions installed (see next section), it might be necessary to repeat this command several times, until the last version of the Sen2Cor application is finally removed.

There are no side dependencies outside of the Anaconda folder, except the settings of the environment variables, which you might possibly need to remove by hand. If you are sure that you want to have a complete uninstallation of every processor you have installed in the past (**and it's complete runtime environment**) you can simply remove the Anaconda folder.

### 3.1.5 Versioning

By default, the installation of a new Sen2Cor version will not overwrite previous versions. If you explicitly only want to work with the most recent version, perform an uninstallation of the old processor first, as it is described in the previous section. Else you can add the new version to the already installed versions. You can then start a dedicated version by typing the command:

`L2A_Process-x.y.z`

where **x.y.z** denotes your version number. The command:

**L2A\_Process**

without a dedicated version number will always start the version of the processor which was installed at last. You can check it's version by typing:

**L2A\_Process --help**

## 3.2 Operation

The processor can be operated in two different ways:

- either as a purely command line driven application;
- or from the Sentinel-2 Toolbox

### 3.2.1 Command line interpreter

Calling the script L2A\_Process with the option '-h' via command line displays the following menu:

```
bash-4.1$ L2A_Process -h
usage: L2A_Process [-h] [--resolution {10,20,60}] [--sc_only] [--ac_only]
                  directory

Sentinel-2 Level-2A Prototype Processor (Sen2Cor), 1.0.1, created:
2014.06.16

positional arguments:
  directory              Directory where the Level-1C input files are located

optional arguments:
  -h, --help            show this help message and exit
  --resolution {10,20,60}
                        Target resolution, must be 10, 20 or 60 [m]
  --sc_only             Perform only the scene classification at 60m
                        resolution
```

**<directory>** can be any directory where a L1C User product is stored. The target directory can have any name, providing that a "L1C\_" is part of the directory name. The output directory for the L2A data can be configured in the L2A\_GIPP.xml configuration file. By default, the processor will create an output directory of the same name in the source directory, but replacing "L1C\_" with "L2A\_".

**--resolution** is the target resolution for the product to be processed. See section 2.6.1 for details. If resolution is omitted, the product will be processed with the lowest resolution, which is 60 m.

A typical run of the processor via command line in debug mode looks as follows:

```
L2A_Process /home/sen2cor/testdata/Level-1C_User_Product --resolution 60
```

```
Sentinel-2 Level-2A Prototype Processor (Sen2Cor), 1.0.1, created: 2013.06.01 started ...
Procedure: L2A_Process: start of Pre Processing, elapsed time[s]: 0.052
Procedure: L2A_Tables: start import, elapsed time[s]: 0.014
Procedure: L2A_Tables: stop import, elapsed time[s]: 24.095
Procedure: L2A_Process: start of Scene Classification, elapsed time[s]: 0.001
Procedure: L2A_SC init , elapsed time[s]: 0.097
Procedure: L2A_CSND_1_1, elapsed time[s]: 0.064
Procedure: L2A_CSND_1_2, elapsed time[s]: 0.088
Procedure: L2A_CSND_2_0, elapsed time[s]: 0.000
Procedure: L2A_CSND_2_1, elapsed time[s]: 0.064
Procedure: L2A_CSND_2_2, elapsed time[s]: 0.040
Procedure: L2A_CSND_2_3, elapsed time[s]: 0.036
Procedure: L2A_CSND_2_4, elapsed time[s]: 0.050
Procedure: L2A_CSND_2_5, elapsed time[s]: 0.045
Procedure: L2A_CSND_3 , elapsed time[s]: 0.087
Procedure: L2A_CSND_4 , elapsed time[s]: 0.080
Procedure: L2A_CSND_5_1, elapsed time[s]: 0.141
Procedure: L2A_CSND_5_2, elapsed time[s]: 0.126
Procedure: L2A_CSND_6 , elapsed time[s]: 0.074
Procedure: L2A_CSND_7 , elapsed time[s]: 0.129
Procedure: postprocess , elapsed time[s]: 0.093
Procedure L2A_SceneClass, overall time[s]: 1.117
Procedure: L2A_Process: start of Atmospheric Correction, elapsed time[s]: 0.001
Procedure: L2A_AtmCorr: starting data processing, elapsed time[s]: 0.004
Procedure: L2A_AtmCorr: end of calculation terrain maps, elapsed time[s]: 0.149
Procedure: L2A_AtmCorr: start of AOT retrieval @550nm, elapsed time[s]: 0.001
Procedure: L2A_AtmCorr: end of altitude grid calculation, elapsed time[s]: 1.240
Procedure: L2A_AtmCorr: end of internal classification, elapsed time[s]: 0.613
Procedure: L2A_AtmCorr: end of interpolation LUTs, elapsed time[s]: 1.125
Procedure: L2A_AtmCorr: end retrieving reference pixels for dark areas, elapsed time[s]:
1.384
Procedure: L2A_AtmCorr: end of check for negative reflectance pixels, elapsed time[s]:
3.117
Procedure: L2A_AtmCorr: end of visibility index calculation, elapsed time[s]: 1.673
Procedure: L2A_AtmCorr: start of wv retrieval, elapsed time[s]: 0.001
Procedure: L2A_AtmCorr: end of wv retrieval preparation, elapsed time[s]: 0.895
Procedure: L2A_AtmCorr: end of wv retrieval, elapsed time[s]: 19.914
Procedure: L2A_AtmCorr: start of rho retrieval, elapsed time[s]: 0.001
Procedure: L2A_AtmCorr: end of rho retrieval preparation, elapsed time[s]: 8.363
Procedure: L2A_AtmCorr: end of rho retrieval step 1, elapsed time[s]: 15.760
Procedure: L2A_AtmCorr: end of rho retrieval step 2, elapsed time[s]: 3.754
Procedure: L2A_Process: start of Post Processing, elapsed time[s]: 0.015
Procedure: L2A_Tables: start export, elapsed time[s]: 0.000
Procedure: L2A_Tables: stop export, elapsed time[s]: 15.408
Application terminated successfully
```

### **3.2.2 Interfacing and integration into the Sentinel-2 Toolbox**

The integration of the Sen2Cor in the Sentinel 2 Toolbox is provided via the Standalone Tools Adapter and is done normally automatically, if you install the toolbox. Read the Toolbox documentation for further information.

### **3.3 The Software Development Environment (SDE)**

**The content of this chapter can be ignored by the normal Sen2Cor users.**

This chapter describes all necessary steps to install, configure and operate the necessary tools in order to extend or maintain the prototype processor software. It is only kept as reference for the developer teams.

#### **3.3.1 Requirements and Third Party Software**

The Sen2Cor development distribution is a collection of essential open source tools, which provides in common the development environment for the processor. The following table specifies a comprehensive overview on all required items, including the runtime environment. All tools are Open Source except the IDL development environment, the original ATCOR code and the python module for the atmospheric correction (L2A\_AtmosCorr.py) derived by ATCOR.

IDL is only needed in that case, that a 1:1 testing is required with the initial ATCOR code, as this is written in IDL. The original ATCOR IDL code can be requested at ESA only for maintainers of the Sen2Cor project.

**Table 3-2: Third Party products for the SDE**

Operating System			
Product	Name	Version	Source
Linux (alternatives)	CentOS	>= 6.0	<a href="http://www.centos.org/">http://www.centos.org/</a>
	RHEL	6.0	<a href="http://www.de.redhat.com/products/rhel/desktop/">http://www.de.redhat.com/products/rhel/desktop/</a>
	Ubuntu	12.04.2 LTS	<a href="http://www.ubuntu.com/download/server">http://www.ubuntu.com/download/server</a>
Software Development Environment			
Product	Name	Version	Source
IDE	Eclipse Classic	>= 3.7.1	<a href="http://www.eclipse.org/downloads/packages/">http://www.eclipse.org/downloads/packages/</a>
Python IDE for Eclipse	PyDev	>= 4.0.0	<a href="http://pydev.org/">http://pydev.org/</a>
Operational Tools and Libraries (see run_time_environment)			
Product	Name	Version	Source
Python Distribution Package (including Python 7.2, NumPy + SciPy, PyTables, Cython, Distutils and more ...)	Anaconda	>= 2.2.0	<a href="https://store.continuum.io">https://store.continuum.io</a>

### 3.3.1.1 Eclipse, PyDev

Eclipse Classic is used and distributed as the standard development environment for the processor. Together with PyDev, it provides a complete development environment, including editor, syntax checker, debugger and runtime environment for python. For installation, it is referred to the installation procedures of the different tools as can be found in the references given in Table 3-24 above.

### 3.3.1.2 Anaconda

The Sen2Cor application uses a huge amount of python libraries, where “numpy” and “scipy” are only the most prominent ones. One requirement of this project was, to keep the installation overhead as low as possible. Thus, it was decided to use a free python distribution. We decided for Anaconda after several tests with different Python distributions. Anaconda is a completely free enterprise-ready Python distribution for large-scale data processing, predictive analytics, and scientific computing. Anaconda comes with its own installers, which facilitates the setup of the development environment considerably. Anaconda is able to support other operating systems (Mac OS and Windows) and thus avoiding the common problem of version mismatches between the different operating systems.

### 3.3.1.3 Cython

Cython is a python compatible language, which enables to convert native python code into C, so that the code can be compiled into a shared library and can be executed without exposing the readable python code. This technique is used here in order to protect the licensed ATCOR algorithm in the runtime distribution from being read and analysed. Cython is an integrated part of the Anaconda distribution described above.

### 3.3.1.4 Distutils

The Distutils are the Python standard for producing distribution packages. It is used here for generating the runtime installation package. Distutils is an integrated part of the Anaconda distribution described above.

### 3.3.1.5 GDAL

GDAL is a translator library for raster geospatial data formats that is released under an [X/MIT](#) style [Open Source](#) license by the [Open Source Geospatial Foundation](#). As a library, it presents a [single abstract data model](#) to the calling application for all supported formats. It also comes with a variety of useful [commandline utilities](#) for data translation and processing, which is used here (among others) for preparing the Digital Elevation Map.

### 3.3.1.6 Glymur

Glymur is an interface to the OpenJPEG library which allows one to read and write JPEG 2000 files from Python. Glymur supports both reading and writing of JPEG 2000 images, but writing JPEG 2000 images is currently limited to images that can fit in memory. Glymur can read images using OpenJPEG library versions as far back as 1.3, but it is strongly recommended to use version 2.1.0, which is the most recently released version of OpenJPEG at this time.

### 3.3.1.7 PyTables

PyTables is a package for managing hierarchical datasets and designed to efficiently and easily cope with large amounts of data.

PyTables is built on top of the [HDF5 library](#), using the [Python](#) language and the [NumPy](#) package. It features an object-oriented interface that, combined with C extensions for the performance-critical parts of the code (generated using [Cython](#)), makes it a fast, yet extremely easy to use tool for interactively browse, process and search very large amounts of data. One important feature of PyTables is that it optimizes memory and disk resources so that data takes much less space (especially if on-flight compression is used) than other solutions such as relational or object oriented databases. PyTables is an integrated part of the Anaconda distribution described above.

## 3.3.2 Installation

Development of Sen2Cor was performed under (1) Eclipse using (2) Anaconda as the Python distribution, (3) the additional package for Python development, called PyDev and (4) the Eclipse XML and XSD extension packages. It is recommended to install these third party tools from the original software providers, exactly in the range as listed below and following the corresponding installation steps as given in the links:

- Anaconda: <http://docs.continuum.io/anaconda/install.html>
- Eclipse: <https://eclipse.org/users/>
- PyDev: <http://marketplace.eclipse.org/content/pydev-python-ide-eclipse>  
[http://pydev.org/manual\\_101\\_root.html](http://pydev.org/manual_101_root.html)

As the setup of these tools changes frequently, we can only refer to the original installation and configuration instructions above.

### 3.3.3 Configuration

#### 3.3.3.1 Configure Python

In order to select Anaconda as the Python Interpreter of your choice, within Eclipse, first click on [/Project/Properties/PyDev – Interpreter](#), second, click on [Configure an Interpreter not listed](#).

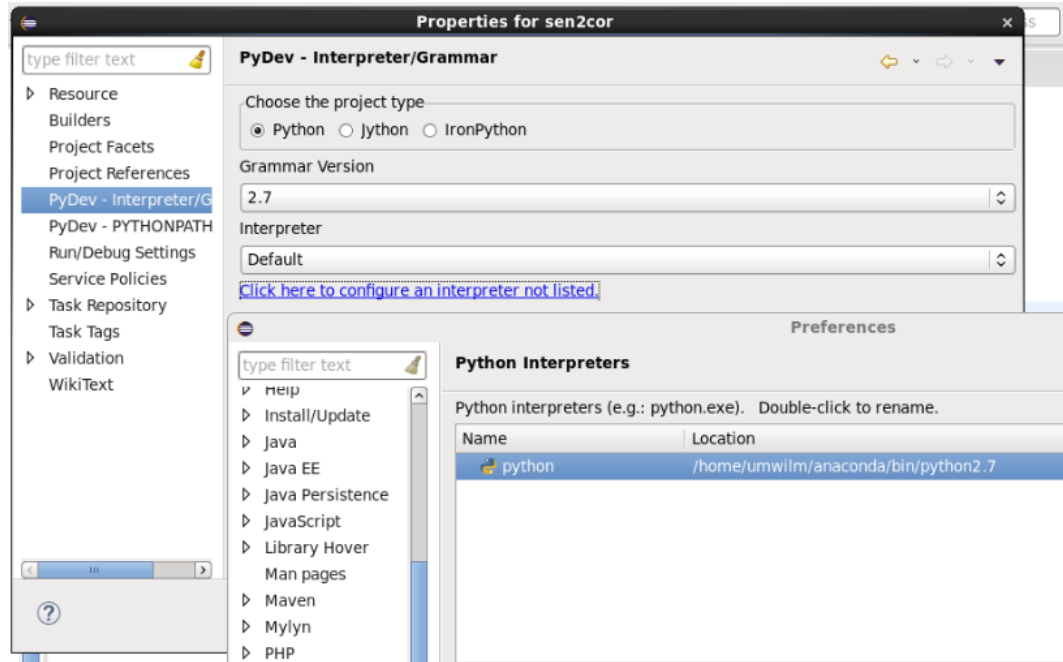


Figure 3-1 – Configuration of Python interpreter within Eclipse

Then click on [New](#) and select the Python Interpreter in the folder where Anaconda was installed. An example installation is shown above.

#### 3.3.3.2 Processor Installation

Download the archive of the recent processor version from:

<http://s2tbx.telespazio-vega.de/sen2cor/>

Unpack the processor as is described in section 3.1.2 at a location of your choice.

Within Eclipse, create a new project pointing to the root folder of your processor (e.g. sen2cor-2.0.3).

Perform a refresh. The directory tree will then show your project.

The main() operation of the Sen2Cor is located in the L2A\_Process module within the <sen2cor> subfolder.

#### 3.3.3.3 Environment Settings

Three environment variables are necessary to be set up in order to obtain a full configured system, as described in section 3.1.2:

**SEN2COR\_HOME**: this is the directory where the user configuration data are stored (see above).

**SEN2COR\_BIN**: this is a pointer to the installation of the Sen2Cor package. This is located in the same folder as the **L2A\_Process** module.

**GDAL\_DATA**: this is a pointer to the directory where the GDAL coordinate system info is stored. For Windows this is located in the **share/gdal**, for Linux and MacOSX in the **site-packages/osgeo/data/gdal** folder of Anaconda.

These three environment variables can be configured inside of the Eclipse workspace via **/Run/Run Configurations/Environment**, for the **L2A\_Process** module as shown below for Figure 3-2.

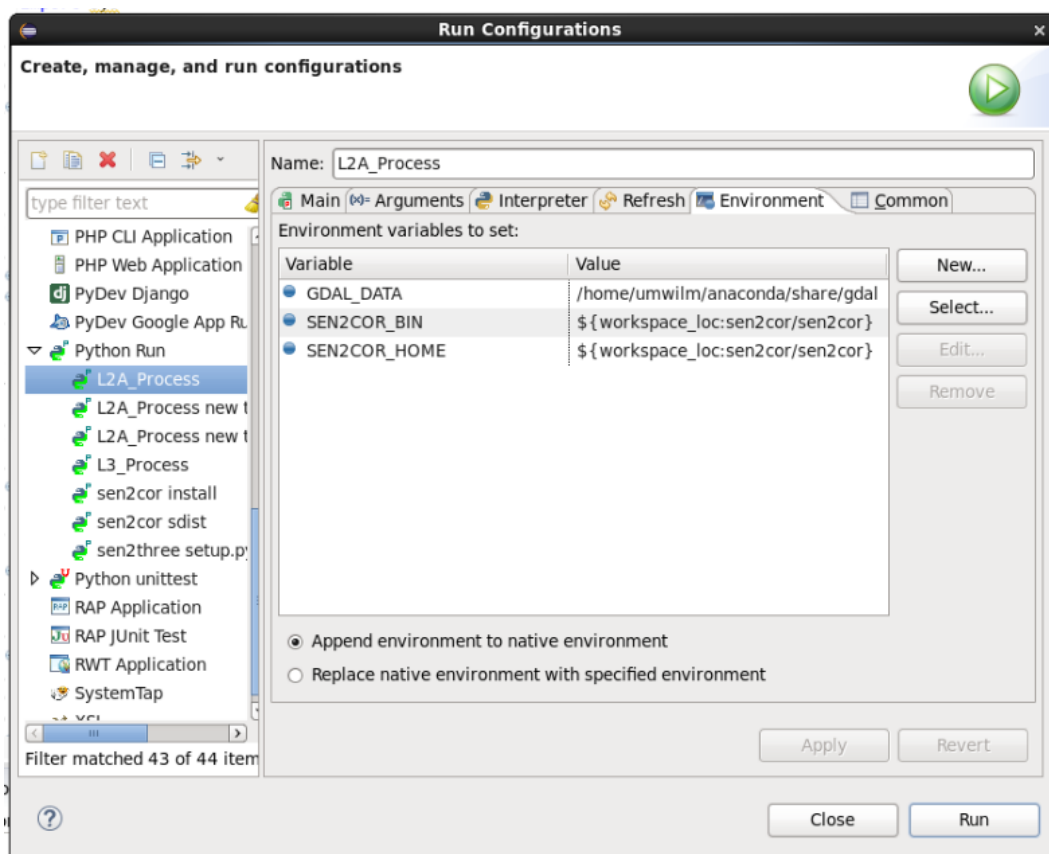


Figure 3-2 – Environment settings within Eclipse

### 3.3.4 Operation

#### 3.3.4.1 Running the processor within Eclipse

Open the **/Run/Run Configurations/Main** dialog and select **L2A\_Process** as the main module for the execution of the processor, as shown in Figure 3-3 below.



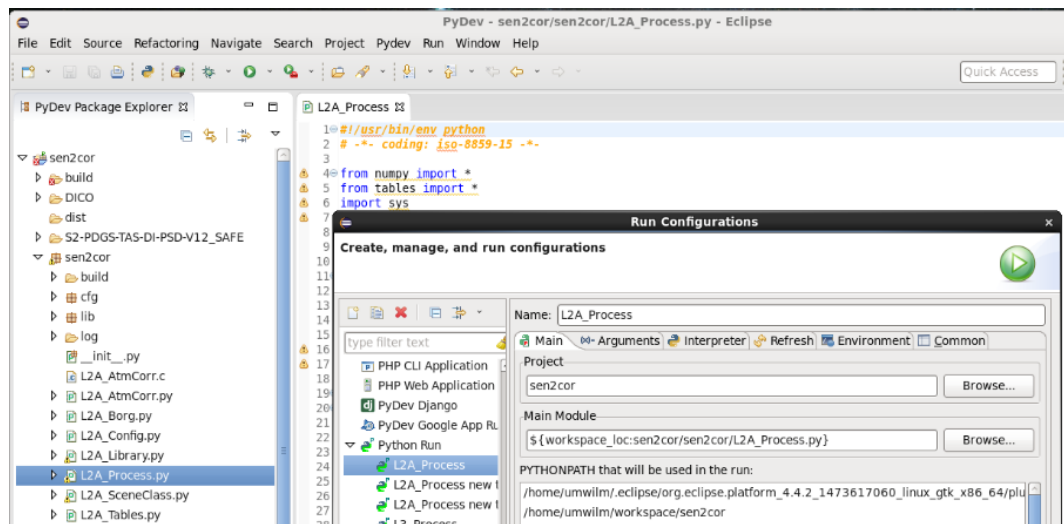


Figure 3-3 – Setting the processors Main Module

For testing purposes within Eclipse the whole application can be operated in two different ways:

1. Runtime Mode: **/Run/Run**
2. Debug Mode: **/Run/Debug**

In order to simulate a scenario for testing purposes a command line can be configured within Eclipse via:

**/Run/Run Configurations/Arguments**

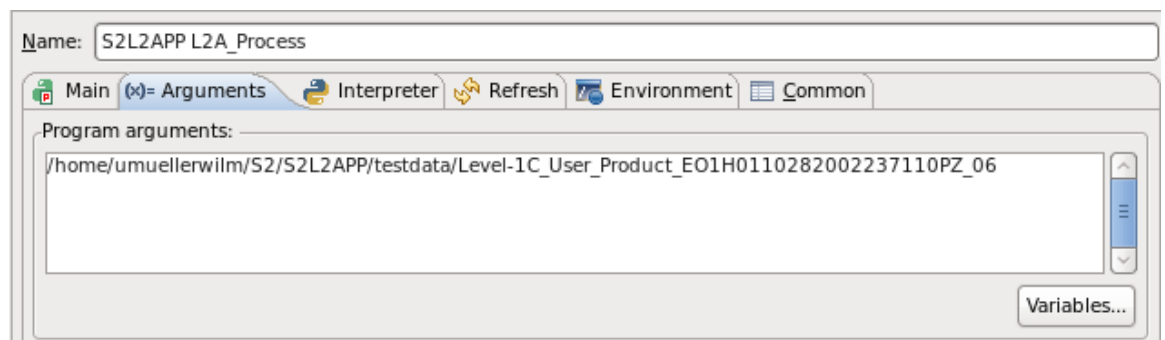


Figure 3-4 – SCD command line arguments

All other operational issues are equivalent with the scenario described for section 3.2.1. After this configuration is finished, the processor can be executed and debugged within Eclipse.

### 3.3.5 Generating a Source Distribution

From the Eclipse workspace or via command line shell execute:

**python setup.py sdist**

This will generate an archive of the form:

**sen2cor-<VERSION>.tar.gz**

in the `<dist>` folder below the sen2cor root directory, which then can be distributed.

### 3.3.6 Generating a binary module for the Atmospheric Correction

The source code of the original ATCOR software is protected by a license agreement from DLR. As a consequence, the Python source code of the processor core algorithm (L2A\_AtmosCorr.py) is only contained in a sen2Cor developer's distribution, as it is not allowed to be distributed to the end user.

To deliver a binary runtime module from this protected code, Cython is used as an intermediate development step. Cython is able to generate C source code from python code. The generated C code can then be compiled platform specific on each target platform and can finally be redistributed in a non-human readable shared object or dynamic link library. Following target platforms are currently tested and will be supported:

- Linux x86, 64 bit (CENTOS 6.5, alternatively RHEL 6.5)
- MacOSX, x86, 64 (Yosemite)
- Windows, 64 bit (Tested on Windows 7 Enterprise)

The compilation itself is done via the command `python setup.py build_ext` started from the module folder (this is the location where the L2A\_AtmosCorr.py module is located and must be executed on the target platform for which the distribution package shall be created. **Only developer packages will contain the L2A\_AtmosCorr.py module.**

## **4. REFERENCES**

1. Richter, R., Wang, X., Bachmann, M. and Schlaepfer, D. (2011). Correction of cirrus effects in Sentinel-2 type of imagery. *International Journal of Remote Sensing*, **32**, 2931-2941.
2. Louis, J., A. Charantonis, A. and Berthelot, B (2010). Cloud Detection for Sentinel-2. *Proceedings of ESA Living Planet Symposium*.
3. Salomonson V.V., I. Appel I. (2004). Estimating fractional snow cover from MODIS using the normalized difference snow index. *Remote Sensing of Environment* **89**, 351–360.
4. Vane, G., Green, R. O., Chrien, T. G., Enmark, H. T., Hansen, E. G., and Porter, W. M. (1993). The airborne visible / infrared imaging spectrometer (AVIRIS). *Remote Sens. Environ.* **44**, 127-143.
5. Kaufman, Y., et al. (1997). The MODIS 2.1  $\mu\text{m}$  channel – correlation with visible reflectance for use in remote sensing of aerosol. *IEEE TGRS*, **35**, 1286 – 1298.
6. Schläpfer, D. et al. (1998). Atmospheric precorrected differential absorption technique to retrieve columnar water vapour. *Remote Sens. Environ.*, **65**, 353-366.

## 5. APPENDIX

### 5.1 Licenses

Anaconda:

<http://docs.continuum.io/anaconda/eula.html>

#### GDAL/OGR Licensing

=====

GDAL/OGR General

GDAL/OGR is licensed under an MIT/X style license with the following terms:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included

in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.