Sentinel-2 Agriculture

Software User Manual

<table>
<thead>
<tr>
<th>Milestone</th>
<th>Milestone 6 – End of CCN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authors</td>
<td>Cosmin UDROIU, Laurentiu NICOLA, Cosmin CARA, Mickael SAVINAUD, Nicolas BELLEMANS, Philippe MALCORPS, Sophie BONTEMPS, Pierre DEFOURY</td>
</tr>
<tr>
<td>Distribution</td>
<td>ESA - Benjamin KOETZ</td>
</tr>
</tbody>
</table>
This page is intentionally blank
Table of recorded changes

<table>
<thead>
<tr>
<th>Issue/Rev.</th>
<th>Date</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>25/10/2015</td>
<td>Document creation</td>
</tr>
<tr>
<td>1.1</td>
<td>26/02/2016</td>
<td>Changed the parameters for the processors, and added steps for installation and configuration of the missing system components (like website, database, etc.)</td>
</tr>
<tr>
<td>1.2</td>
<td>01/03/2016</td>
<td>Changed parameter list for MACCS, added MACCS configurator page</td>
</tr>
<tr>
<td>2.0</td>
<td>21/08/2016</td>
<td>Major revision of the document, with changes in the structure to better match the user point of view, and with changes in the content to reflect the system v1.3</td>
</tr>
<tr>
<td>2.1</td>
<td>10/10/2016</td>
<td>Corrections.</td>
</tr>
<tr>
<td>2.2</td>
<td>23/12/2015</td>
<td>Update to reflect the system 1.4</td>
</tr>
<tr>
<td>2.3</td>
<td>20/02/2017</td>
<td>Update to reflect the system 1.5</td>
</tr>
<tr>
<td>2.4</td>
<td>13/06/2017</td>
<td>Updates to reflect the system 1.6</td>
</tr>
<tr>
<td>2.5</td>
<td>03/10/2017</td>
<td>Updates to reflect the system 1.7 – Added local store mode for the Sentinel 2 Downloader</td>
</tr>
<tr>
<td>2.6</td>
<td>04/04/2018</td>
<td>Updates to reflect the system 1.8</td>
</tr>
<tr>
<td>2.7</td>
<td>02/07/2018</td>
<td>Updates to reflect the patch and evolutions for system 1.8 and 1.8.1</td>
</tr>
<tr>
<td>2.8</td>
<td>19/09/2018</td>
<td>Updates to reflect patch and evolutions for system 1.8.3</td>
</tr>
<tr>
<td>3.0</td>
<td>13/05/2019</td>
<td>Major revision of the document, to reflect the major change in the system related to the MAJA implementation which requires a re-installation of the system</td>
</tr>
<tr>
<td>3.1</td>
<td>02/20/2020</td>
<td>Updates for the release of version 2.0.2</td>
</tr>
<tr>
<td>3.2</td>
<td>12/11/2020</td>
<td>Updates for the release of version 2.0.3</td>
</tr>
</tbody>
</table>
Table of contents

Sentinel-2 Agriculture Software User Manual 1

1. Introduction 11
   1.1 Purpose and scope 11
   1.2 Structure of the document 11
   1.3 References 11
      1.3.1 Applicable documents 11
      1.3.2 Reference documents 12
      1.3.3 Acronyms and abbreviations 13

2. System Overview 16
   2.1 General overview 16
   2.2 Processors 18
      2.2.1 Cloud-free temporal syntheses (L3A) processor 19
      2.2.2 Biophysical vegetation (L3B) processor 21
      2.2.3 Cropland mask (L4A) processor 25
      2.2.4 Crop type map (L4B) processor 27
   2.3 Visualization tool 30
   2.4 S2-Toolbox integration 31
   2.5 System implementation 31
   2.6 The Sen2-Agri Orchestrator 32
   2.7 The Sen2-Agri Services 33

3. System deployment 34
   3.1 Installation package content 34
   3.2 Prerequisites for automatic usage 34
   3.3 Installation and configuration for automatic usage 36
      3.3.1 Installation procedure 37
      3.3.2 Update procedure 39
      3.3.3 Additional configurations 41
      3.3.4 First interaction with the Sen2-Agri system 44
   3.4 Prerequisites for manual usage 45
   3.5 Installation and configuration for manual usage 45
      3.5.1 Installing Sen2-Agri core components 45
      3.5.2 Configuring and using SNAP 47

4. How to Use the Sen2-Agri System 53
4.1 General
4.2 First steps for the automated usage of the system
   4.2.1 Configure a new site
   4.2.2 Editing the sites to define season, insitu data, strata data and processors to execute
   4.2.3 Optionally adapt the main processor parameters
   4.2.4 Activate or change the site parameters
   4.2.5 Delete a site
   4.2.6 Optionally adapt the data sources configuration
   4.2.7 Activating download of Sen2Cor L2A products
   4.2.8 Monitor the downloading
   4.2.9 Monitor the processing
   4.2.10 Where to find output products
   4.2.11 Provide in situ data during monitoring period
4.3 How to add scheduled jobs
4.4 Perform manual operations
   4.4.1 Atmospheric Correction (L2A processor)
   4.4.2 Creating a composite (L3A) product
   4.4.3 Creating a LAI vegetation status (L3B) product
   4.4.4 Creating a NDVI vegetation status (L3B) product
   4.4.5 Creating a crop mask (L4A) product
   4.4.6 Creating a crop type (L4B) product
4.5 In-situ data information
4.6 Single-tile vs multi-tile processing

5. Visualization tool
   5.1 Visualization tool deployment
      5.1.1 Prerequisites for automatic usage
      5.1.2 Installation and configuration
   5.2 How to use the visualization tool?
      5.2.1 GeoServer Interface
      5.2.2 Integration into QGIS

6. Maintenance of the system
   6.1 Uninstall procedure
   6.2 Update procedure
   6.3 How to submit an issue
   6.4 FAQ
Appendix A - Additional tools available in the system

Appendix A.1 Sen2-Agri Archiver parameters
Appendix A.2 Sen2-Agri HTTP Listener parameters
Appendix A.3 Sen2-Agri Executor parameters
Appendix A.4 Sen2-Agri Monitor Agent parameters
Appendix A.5 Sen2-Agri Persistence Manager parameters
Appendix A.6 Sen2-Agri Services download parameters
Appendix A.7 Using Sentinel2 Download Services With Local L1C Store

Appendix B - Sen2-Agri database tables for manual configuration

Appendix C - System performance example

Appendix D - Building from source

Appendix E – How to add new users

Appendix F – Advanced system configuration items

Appendix G – ESA CCI LC legend
List of figures

Figure 2-1. Suite of the four validated S2 derived agricultural products generated by the Sen2-Agri system .......................................................... 16
Figure 2-2. Logical data flow of the Sen2-Agri system ................................. 17
Figure 2-3. Logical data flow of the L3A processor ........................................ 20
Figure 2-4. Sequence for an automatic production of a monthly cloud-free composite .......... 20
Figure 2-5. 50 days composite centered on April 1st, 2016 based on S2 and L8 images over Occitanie Region, France .................................................. 21
Figure 2-6. Logical data flow of the L3B processor for generating LAI products ........... 23
Figure 2-7. Logical data flow of the L3B processor for generating NDVI products .......... 23
Figure 2-8. Sequence for an automatic production of a LAI product .................... 24
Figure 2-9. Mono-date LAI product over South-Africa (Kwazulu-Natal) based on a S2 orbit from 19-01-2017 ......................................................... 24
Figure 2-10. Logical data flow of the L4A processor ........................................ 26
Figure 2-11. Sequence for an automatic production of a series of monthly crop masks ...... 26
Figure 2-12. Ukraine 10-meter national cropland mask for the 2016 growing season generated using 7 months of S2 and L8 data ......................... 27
Figure 2-13. Logical data flow of the L4B processor and its articulation with the L4A processor .............................................................................. 28
Figure 2-14. Sequence for an automatic production of crop type maps over the season ...... 29
Figure 2-15. Crop type map over the Western Cape region, South Africa, based on S2 and L8 imagery acquired between May and December 2016 ........................................ 29
Figure 2-16. Overview of the Sen2-Agri visualization tool architecture .................... 30
Figure 2-17. Software System Architecture .................................................... 32
Figure 3-1 Structure of the installation package (including the SRTMs and MAJA) ........ 38
Figure 3-2. Demmacs launcher updating ....................................................... 40
Figure 3-3. In the tab user of the web interface, “Add new user” ............................ 44
Figure 3-4. In the case of a first login: “Set a password” in the login page ................ 44
Figure 3-5. Main page of the Sen2-Agri web interface ........................................ 45
Figure 4-1. Dialog box for site creation through the Sen2-Agri system interface .......... 54
Figure 4-2. Site page of the Sen2-Agri interface, to edit site or seasons .................... 55
Figure 4-3. Dialog box for sites edition (including season dates) through the Sen2-Agri system interface ................................................................ 56
Figure 4-4. Dialog box when a season is defined and the possibility to activate processors is visible .............................................................................. 57
Figure 4-5. The "data sources" tab of the web interface - Configure the data sources parameters ........................................................................... 61
Figure 4-6. The “monitoring” tab of the web interface - A view to the downloading .......... 62
<table>
<thead>
<tr>
<th>Ref</th>
<th>Sen2-Agri_SUM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Issue</td>
<td>Page</td>
</tr>
<tr>
<td>3.2</td>
<td>8</td>
</tr>
</tbody>
</table>

Figure 4-7. Sen2-Agri Execution Dashboard – System Overview ......................................................... 63
Figure 4-8 The jobs history information .................................................................................................... 64
Figure 4-9 – The job execution steps details ................................................................................................. 65
Figure 4-10. Sen2-Agri Execution Dashboard – Product Availability ............................................................... 66
Figure 4-11 - Adding in situ data .................................................................................................................... 67
Figure 4-12 – Processor parameters ................................................................................................................ 68
Figure 4-13 – Adding a new scheduled job ....................................................................................................... 68
Figure 4-14 – Adding “Once” scheduled type .................................................................................................. 69
Figure 4-15– Adding “Cycle” scheduled type .................................................................................................. 69
Figure 4-16 – Adding “Repeat” scheduled type ................................................................................................ 69
Figure 4-17 – Delete scheduled job.................................................................................................................. 70
Figure 4-18. Launching MACCS/MAJA processor in SNAP – Execution parameters ................................. 72
Figure 4-19. L3A Composite Processor in SNAP – Execution parameters ....................................................... 75
Figure 4-20. Manual execution of L3A processor from Web Site .................................................................... 76
Figure 4-21. Manual execution of L3A processor from Web Site (advanced parameters) .............................. 77
Figure 4-22. L3B LAI Vegetation Status Processor in SNAP – Execution parameters ................................. 79
Figure 4-23. Manual execution of LAI processor from Web Site ...................................................................... 80
Figure 4-24. Manual execution of LAI processor from Web Site (advanced parameters) .............................. 81
Figure 4-25. L3B NDVI Vegetation Phenology Processor in SNAP – Execution parameters .......................... 83
Figure 4-26. Manual execution of Phenological NDVI processor from Web Site ............................................ 84
Figure 4-27. L4A Crop Mask Processor in SNAP – Execution parameters ...................................................... 89
Figure 4-28. Manual execution of L4A processor from Web Site (with InSitu data) ......................................... 90
Figure 4-29. Manual execution of L4A processor from Web Site (with InSitu data) advanced parameters ....... 91
Figure 4-30. Manual execution of L4A processor from Web Site (without InSitu data) ................................. 92
Figure 4-31. Manual execution of L4A processor from Web Site (without InSitu data) advanced parameters ................................................................................................................................................. 93
Figure 4-32. L4B Crop Type Processor in SNAP – Execution parameters ....................................................... 96
Figure 4-33. Manual execution of CropType processor from Web Site ............................................................ 98
Figure 4-34. Manual execution of L4B processor from Web Site advanced parameters .................................. 99
Figure 5-1. Layer page of the geoserver with search bar in red ................................................................. 105
Figure 5-2. Layer preview page of the geoserver with search bar in red and the OpenLayer button in green ................................................................................................................................................. 105
Figure 5-3. Openlayer view: in red the zoom buttons .................................................................................... 106
Figure 5-4. OpenLayer view with value and position .................................................................................... 107
Figure 5-5. Main QGIS interface with browser panel and WMS line in red ................................................. 108
Figure 5-6. Creation WMS widget ................................................................................................................ 108

© UCL-Geomatics 2018

This document is the property of the Sen2-Agri partnership, no part of it shall be reproduced or transmitted without the express prior written authorisation of UCL-Geomatics (Belgium).
Figure 5-7. Visualization of a L3B product through the WMS stream ........................................ 109
Figure 5-8. Access to the metasearch plugin ................................................................................. 109
Figure 5-9. Metasearch configuration ............................................................................................... 110
Figure 5-10. MetaSearch interface .................................................................................................. 110
Figure 0-1. Sen2-Agri Configurator – parameter validation............................................................ 132
Figure 0-2. Sen2-Agri Configurator – parameter validation – error message ......................... 132
Figure 0-3. Sen2-Agri Configurator – general tab – user with admin role................................. 133
Figure 0-4. Sen2-Agri Configurator – customize parameter value .............................................. 133
Figure 0-5. Sen2-Agri Configurator – general tab – user without admin role ........................ 134
Figure 0-6. Sen2-Agri Configurator – Archiver tab .................................................................... 134
Figure 0-7. Sen2-Agri Configurator – Dashboard tab ................................................................. 135
Figure 0-8. Sen2-Agri Configurator – Executor tab .................................................................. 135
Figure 0-9. Sen2-Agri Configurator – Monitoring Agent tab ....................................................... 135
Figure 0-10. Sen2-Agri Configurator – Downloader tab ............................................................... 136
List of tables

Table 1-1. Applicable documents ........................................................................................................11
Table 1-2. Reference documents ........................................................................................................12
Table 1-3. List of acronyms and abbreviations .......................................................................................13
Table 2-1. Cloud-free composite specifications .......................................................................................19
Table 2-2. Vegetation status indicators specifications ..............................................................................22
Table 2-3. Dynamic crop masks specifications .......................................................................................25
Table 2-4. Crop type map specifications ..................................................................................................27
Table 4-1. L2A processor (demmacs) launching parameters .................................................................70
Table 4-2. L3A processor’s parameters ..................................................................................................72
Table 4-3. L3B LAI processor’s parameters ............................................................................................78
Table 4-4. L3B NDVI processor’s parameters .........................................................................................82
Table 4-5. L4A processor’s parameters ..................................................................................................84
Table 4-6. L4B Processor’s parameters ..................................................................................................94
Table 4-7. Format of the attribute table of in-situ dataset for cropland products .....................................100
Table 0-1. Advanced parameters table .................................................................................................137
1. Introduction

1.1 Purpose and scope

This document is the Software User Manual (SUM) of the Sentinel-2 for Agriculture (Sen2-Agri) project funded by the European Space Agency (ESA). The aim of this document is to detail the operational system implementation, build, installation and maintenance procedure.

The SUM is the output of Work Package (WP) 4200, named “System Documentation”, which is part of the Task 4 of the project.

1.2 Structure of the document

After this introduction, this document contains 4 main sections and the appendices:

- Section 2 provides a general overview of the system: its main objectives and properties, its processors, its architecture and integration with the Sentinel-2 Toolbox (S2-TB);
- Section 3 describes the deployment procedures of the system, both for the automatic and manual modes;
- Section 4 explains how to use the system, detailing the first steps to allow an automatic use but also the customization of the jobs for the manual operation;
- Section 5 defines the maintenance and troubleshooting procedures;
- 7 appendices are provided at the end of the document.

1.3 References

1.3.1 Applicable documents

Table 1-1. Applicable documents

<table>
<thead>
<tr>
<th>ID</th>
<th>Title</th>
<th>Reference</th>
<th>Issue/Rev.</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD.1</td>
<td>Sentinel-2 for Agriculture Statement of Work</td>
<td>EOEP-DUEP-EOPS-SW-13-0004</td>
<td>1.0</td>
<td>26/03/2013</td>
</tr>
<tr>
<td>AD.2</td>
<td>Space Engineering Software standards</td>
<td>ECSS-E-ST-40C</td>
<td></td>
<td>6/03/2009</td>
</tr>
<tr>
<td>AD.3</td>
<td>Sentinel-2 for Agriculture Technical Proposal</td>
<td></td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>AD.4</td>
<td>Sentinel-2 for Agriculture User Requirements Document</td>
<td>Sen2-Agri_URD_1.2</td>
<td>1.2</td>
<td>25/07/2014</td>
</tr>
<tr>
<td>AD.5</td>
<td>Sentinel-2 for Agriculture Product Specification Document</td>
<td>Sen2-Agri_PSD_1.3</td>
<td>1.3</td>
<td>08/06/2017</td>
</tr>
<tr>
<td>AD.6</td>
<td>Sentinel-2 for Agriculture Technical Specification</td>
<td>Sen2-Agri_TS_1.2</td>
<td>1.2</td>
<td>20/02/2015</td>
</tr>
<tr>
<td>AD.7</td>
<td>Sentinel-2 for Agriculture Design Definition File</td>
<td>Sen2-Agri_DDF_1.2</td>
<td>1.2</td>
<td>17/02/2016</td>
</tr>
<tr>
<td>ID</td>
<td>Title</td>
<td>Reference</td>
<td>Issue/Rev.</td>
<td>Date</td>
</tr>
<tr>
<td>------</td>
<td>----------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>AD.8</td>
<td>User, installation and operating manual MACCS chains</td>
<td>CS/ESG/IGA/MACCS/MU</td>
<td>08</td>
<td>07/07/2015</td>
</tr>
<tr>
<td>AD.9</td>
<td>Sentinel-2 for Agriculture Design Definition File - Algorithm Theoretical Basis Document for L3 monthly composite product</td>
<td>Sen2-Agi_DDF_1.3_ATBDComposite</td>
<td>1.3</td>
<td>08/06/2017</td>
</tr>
<tr>
<td>AD.10</td>
<td>Sentinel-2 for Agriculture Design Definition File - Algorithm Theoretical Basis Document for L3 biophysical product</td>
<td>Sen2-Agi_DDF_1.3_ATBDBegetationStatus</td>
<td>1.4</td>
<td>08/06/2017</td>
</tr>
<tr>
<td>AD.11</td>
<td>Sentinel-2 for Agriculture Design Definition File - Algorithm Theoretical Basis Document for L4 dynamic crop mask product</td>
<td>Sen2-Agi_DDF_1.3_ATBDCropMask</td>
<td>1.4</td>
<td>08/06/2017</td>
</tr>
<tr>
<td>AD.12</td>
<td>Sentinel-2 for Agriculture Design Definition File - Algorithm Theoretical Basis Document for L4 crop type product</td>
<td>Sen2-Agi_DDF_1.3_ATBDCropType</td>
<td>1.3</td>
<td>08/06/2017</td>
</tr>
<tr>
<td>AD.13</td>
<td>User, installation and operating manual MAJA</td>
<td><a href="https://github.com/CNES/Start-MAJA/blob/master/Readme.md">https://github.com/CNES/Start-MAJA/blob/master/Readme.md</a></td>
<td></td>
<td>08/05/2019</td>
</tr>
</tbody>
</table>

### 1.3.2 Reference documents

<table>
<thead>
<tr>
<th>ID</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RD.1</td>
<td>Sentinel-2 PDGS Product Specification Document</td>
</tr>
<tr>
<td>RD.3</td>
<td>User, installation and operating manual MACCS chain</td>
</tr>
</tbody>
</table>

© UCL-Geomatics 2019

This document is the property of the Sen2-Agri partnership, no part of it shall be reproduced or transmitted without the express prior written authorisation of UCL-Geomatics (Belgium).
1.3.3 Acronyms and abbreviations

Table 1-3. List of acronyms and abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD</td>
<td>Applicable Document</td>
</tr>
<tr>
<td>CESBIO</td>
<td>Centre d’Études Spatiales de la Biosphère</td>
</tr>
<tr>
<td>CNES</td>
<td>Centre National d’Études Spatiales</td>
</tr>
<tr>
<td>CS</td>
<td>Communication &amp; Systèmes</td>
</tr>
<tr>
<td>CS-R</td>
<td>CS Romania</td>
</tr>
<tr>
<td>CSW</td>
<td>Catalog Services for the Web</td>
</tr>
<tr>
<td>DDF</td>
<td>Design Definition File</td>
</tr>
<tr>
<td>DEM</td>
<td>Digital Elevation Model</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>ESA</td>
<td>European Space Agency</td>
</tr>
<tr>
<td>GIS</td>
<td>Geographic Information System</td>
</tr>
<tr>
<td>GNU</td>
<td>GNU’s Not Unix</td>
</tr>
<tr>
<td>GPL</td>
<td>General Public License</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>L1C/L1T</td>
<td>Level 1C/1T</td>
</tr>
<tr>
<td>L2A</td>
<td>Level 2A</td>
</tr>
<tr>
<td>L8</td>
<td>Landsat 8</td>
</tr>
<tr>
<td>LAI</td>
<td>Leaf Area Index</td>
</tr>
<tr>
<td>MACCS</td>
<td>Multisensor Atmospheric Correction and Cloud Screening</td>
</tr>
<tr>
<td>MAJA</td>
<td>MACCS-ATCOR Joined Algorithms</td>
</tr>
<tr>
<td>NDVI</td>
<td>Normalized Difference Vegetation Index</td>
</tr>
<tr>
<td>OGC</td>
<td>Open Geospatial Consortium</td>
</tr>
<tr>
<td>OTB</td>
<td>Orfeo ToolBox</td>
</tr>
<tr>
<td>RD</td>
<td>Reference Document</td>
</tr>
<tr>
<td>PSD</td>
<td>Product Specification Document</td>
</tr>
<tr>
<td>RPM</td>
<td>Redhat Package Manager</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root Mean Square Error</td>
</tr>
<tr>
<td>S2</td>
<td>Sentinel-2</td>
</tr>
<tr>
<td>S2-TB</td>
<td>Sentinel-2 ToolBox</td>
</tr>
<tr>
<td>SC</td>
<td>Software Component</td>
</tr>
<tr>
<td>SciHub</td>
<td>Scientific Data Hub</td>
</tr>
<tr>
<td>Sen2-Agri</td>
<td>Sentinel-2 for Agriculture</td>
</tr>
<tr>
<td>SNAP</td>
<td>Sentinel Application Platform</td>
</tr>
<tr>
<td>SoW</td>
<td>Statement of Work</td>
</tr>
<tr>
<td>STA</td>
<td>Standalone Tool Adapter</td>
</tr>
<tr>
<td>SUM</td>
<td>Software User Manual</td>
</tr>
<tr>
<td>UCL</td>
<td>Université Catholique de Louvain</td>
</tr>
<tr>
<td>USGS</td>
<td>United States Geological Survey</td>
</tr>
<tr>
<td>UTM</td>
<td>Universal Transverse Mercator</td>
</tr>
<tr>
<td>WGS84</td>
<td>World Geodetic System 1984</td>
</tr>
<tr>
<td>WMS</td>
<td>Web Mapping Service</td>
</tr>
<tr>
<td>WP</td>
<td>Work Package</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
</tbody>
</table>
2. System Overview

2.1 General overview

The Sen2-Agri operational system is a standalone processing chain which generates a set of products for agriculture monitoring from Sentinel-2 (S2) L1C and Landsat 8 (L8) L1T time series. These different products consist of (Figure 2-1):

- composites of cloud-free surface reflectance values;
- dynamic cropland masks, delivered on a monthly basis during agricultural seasons, starting from the middle of the season;
- cultivated crop type maps and area extent for main crop groups, delivered twice along agricultural seasons (at the middle and at the end of the season);
- biophysical vegetation status indicators, Normalized Difference Vegetation Index (NDVI) and Leaf Area Index (LAI), describing the vegetative development of crops on a 7 to 10 day basis.

The Sen2-Agri system is open-source, allowing any user to generate at their own premises and in an operational way products tailored to their needs. It is composed of a set of independent processing modules orchestrated by a data driven approach. These modules are composed by a set of tools which can be re-used into other systems. The logical data flow and the main interfaces of the Sen2-Agri operational system is provided in Figure 2-2.
Figure 2-2. Logical data flow of the Sen2-Agri system

The main and mandatory inputs of the operational system are:

- A time series of S2 L1C products automatically downloaded from ESA facility;
- Mandatory user parameters: Area of Interest defined as a shapefile, definition of the monitoring period, a set of parameters associated with the algorithms (see section 4.1);
- In-situ data for supervised classification and product validation (including associated metadata or nomenclature).

Additional inputs can be provided by the user if available and/or if found relevant. This is the case for the L8 L1T products but also of other products and parameters that will be detailed in section 4.1.

A Graphical User Interface (GUI) is available to define and handle all the requested parameters. This interface also controls the start of the system, indicates the main directories or files used by the system and allows users visualizing and downloading output products.

The Sen2-Agri system was designed to be modular and interactive. It includes two main components:

- A set of Sen2-Agri Software Components (SC): each Sen2-Agri SC is an independent executable that represents an algorithm or a set of algorithms;
- A Sen2-Agri Orchestrator: the Orchestrator is the main component used to manage the above Sen2-Agri SC on the system. Its role is to monitor the occurrence of new files in the processing area file system, determine the processing chain to be launched, handle user requests and monitor the execution of the current processing chains.

There are two ways of using the system:

- The **automated mode**, which would be used in operational scenarios with as little as possible operator intervention. It is based on the Orchestrator and it ensures that the system automatically downloads data, processes them until the end of the season and delivers in time the output products. In this functioning mode, the processors can also be executed on user request, by using the public website functionality, which is part of the platform. The website can be reached using a web browser and accessing...
http://[machine_name_or_ip]/main.php, where [machine_name_or_ip] is the server name or pc.

- The **manual mode**, which offers the possibility of a manual processing, independently for any processor, without installing and configuring the whole system. In this case, only the processors (which are OrfeoToolbox (OTB) applications), and optionally S2-TB / Sentinel Application Platform (SNAP) application (for visual execution) are necessary.

The modular design and standardization of the interfaces make the Sen2-Agri architecture targeted at various user profiles:

- **High-level analysts**, who want to integrate the products into their estimations related to agriculture and food production and are interested in a direct delivery of the products generated by the Sen2-Agri system;
- **Operator level users**, who aim at running the system and generating the Sen2-Agri products in their own facilities to control and validate the results, then disseminate them to their own high-level users;
- **Research users**, who would rather access and tune the components available, to generate new or adjusted products and possibly operate them as “classical” added-value producers.

The system can either be built from sources, available on a git repository, or installed from precompiled binaries packaged into Redhat Package Manager (RPM) binaries. “Appendix D - Building from source” presents the steps to be followed in order to build the RPM binaries. Section 3 describes the installation and configuration of the obtained RPM binaries (either from sources or from a distribution package). If any of the components binaries are available or installed, the corresponding step will be skipped.

### 2.2 Processors

The processors are a collection of one or several modules dedicated to of the Sen2-Agri products. These processors are split into several modules to allow different parallelization strategies. They are (Figure 2-2):

- **L2A processor**, based on one of the following applications that perform atmospheric corrections and produce bottom-of-atmosphere reflectance values with snow, water, cloud and cloud shadow masks:
  - the Multi-sensor Atmospheric Correction and Cloud Screening (MACCS) chain starting from version 1.6 but still supported also in version 2.0 for compatibility reasons;
  - the MACCS- ATCOR Joint Algorithm (MAJA) which is supported starting from the version 2.0;
- **L3A processor** that delivers monthly cloud-free temporal syntheses;
- **L3B processor** that delivers biophysical indicators on vegetation status;
- **L4A processor** that delivers binary cropland masks along the season;
- **L4B processor** that delivers crop type maps twice during the season.
2.2.1 Cloud-free temporal syntheses (L3A) processor

The cloud-free temporal syntheses processor provides a cloud-free composite of surface reflectance values in the 10 S2 bands designed for land observation and keeping their native spatial resolution (10 or 20 meters). The product is delivered on a monthly basis. Complete specifications are provided in Table 2-1.

Table 2-1. Cloud-free composite specifications

<table>
<thead>
<tr>
<th>Properties</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available bands at 10m spatial resolution</td>
<td>Blue (B2), green (B3), red (B4), near infrared (B8)</td>
</tr>
<tr>
<td>Available bands at 20m spatial resolution</td>
<td>Vegetation red-edge (B5, B6, B7, B8a), shortwave infrared (B11, B12)</td>
</tr>
<tr>
<td>Temporal resolution</td>
<td>1 month with 30 to 50 days moving window</td>
</tr>
<tr>
<td>Geometric accuracy</td>
<td>Same than L1C input data</td>
</tr>
<tr>
<td>Format</td>
<td>GeoTIFF raster images</td>
</tr>
<tr>
<td>Projection</td>
<td>UTM / WGS84</td>
</tr>
<tr>
<td>Metadata</td>
<td>XML file</td>
</tr>
</tbody>
</table>

The product is delivered with several masks that will help appraising its quality:

- the number of valid observations over the period used to generate the composite;
- the status of the pixel over the period: no-data (pixel never observed during the compositing period), cloud (pixel always cloudy during the compositing period), snow (pixel always covered by snow for the available cloud-free observations), water (pixel always covered by water for the available cloud-free observations), land (pixel free from cloud / cloud shadow, snow and water at least once during the compositing period);
- the weighted average of dates used in the synthesis (which informs about the date within the compositing period from which the synthesis is representative).

The main and mandatory input of this L3A processor is a S2 time series, optionally completed by an additional L8 time series, turned into L2A products through the L2A processor.

The L3A processor is based on the weighted average composite approach [RD.4, RD.5, RD.6], includes the correction of directional effects to take into account changes in observation angles and therefore in reflectance values among the different images that are stitched to create the product. It takes as input each L2A tile product available at time \( t \) but also the temporary product generated at the previous date \( t-1 \). This \( t-1 \) product is updated iteratively with the new data until the compositing period is completed. The process is illustrated in Figure 2-3 and Figure 2-4; an example of composite is shown in Figure 2-5, with a product based on S2 and L8 over the Occitanie Region, France.

More information on the algorithms underlying this processor can be found in [AD.9].
Figure 2-3. Logical data flow of the L3A processor

Figure 2-4. Sequence for an automatic production of a monthly cloud-free composite
2.2.2 Biophysical vegetation (L3B) processor

The biophysical vegetation processor provides three types of products informing about the evolution of the green vegetation:

- The **NDVI**, the most popular indicator operationally used for vegetation monitoring, provided to ensure continuity with existing long-term time series and thus, allowing anomalies detection;

- The **LAI**, an intrinsic canopy primary variable that should not depend on observation conditions, which determines the size of leaf interface for exchange of energy and mass between the canopy and the atmosphere;

- **Phenology indices**, also referred to as **NDVI metrics**, which inform about specific key parameters of the growing season (starting date, season length and date of maximum growth rate).

With regard to the LAI product, the system proposes three processing options (not mutually exclusive). First, LAI time series can be produced in near-real time based on a mono-date approach. For the second and third options, the LAI time series are built on a multi-temporal approach, i.e. each LAI value is estimated using more than one date as input. In the second option, the LAI value is also produced in near-real time based on “n” last acquisitions. The third option consists in a re-processing at the end of the season by fitting a phenological model, which should allow having a higher quality time series.
Complete specifications of all products generated by the biophysical vegetation processor are provided in Table 2-2.

### Table 2-2. Vegetation status indicators specifications

<table>
<thead>
<tr>
<th>Properties</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available vegetation indicators</td>
<td>NDVI, LAI, NDVI metrics</td>
</tr>
<tr>
<td>Spatial extent</td>
<td>Global (not only crop area)</td>
</tr>
<tr>
<td>Spatial resolution</td>
<td>10 m</td>
</tr>
<tr>
<td>Temporal resolution</td>
<td>NDVI and LAI: 10 days (7 days with Sentinel-2A and -2B)</td>
</tr>
<tr>
<td></td>
<td>NDVI metrics: at the middle of the season (start date) and at the end of the season (length and date of maximum growth)</td>
</tr>
<tr>
<td>Geometric accuracy</td>
<td>Same than L1C input data</td>
</tr>
<tr>
<td>Format</td>
<td>GeoTIFF raster images</td>
</tr>
<tr>
<td>Projection</td>
<td>UTM / WGS84</td>
</tr>
<tr>
<td>Metadata</td>
<td>XML file</td>
</tr>
</tbody>
</table>

The products are delivered with several masks that will help appraising its quality:

- **For the NDVI maps:**
  - The number of valid observations (i.e. cloud-free and land) over the period used to generate the NDVI;
  - The number of observations associated with the other statuses (“water”, “snow”, “cloud”, “cloud shadow”, “no data’) over the same period;

- **For the mono-date LAI maps:**
  - The number of valid observations (i.e. cloud-free and land) over the period used to generate the LAI;
  - The number of observations associated with the other statuses (“water”, “snow”, “cloud”, “cloud shadow”, “no data’) over the same period;
  - The uncertainty, which will represent Root Mean Square Error (RMSE) as a function of the estimated value.

- **For the multi-date LAI based on “n” last acquisitions:** the number of valid observations (i.e. cloud-free and land) over the search window; knowing that the multi-date LAI will not be computed if this number is lower than 2);

- **For the LAI re-processed at the end of the season and for the NDVI metrics:**
  - The number of valid observations (i.e. cloud-free and land) over the season, knowing that the re-processed LAI and the NDVI metrics will not be computed if this number is lower than 4);
  - The number of observations associated with the other statuses (“water”, “snow”, “cloud”, “cloud shadow”, “no data’) over the same period.

The main and mandatory input of this L3A processor is a S2 time series, optionally completed by an additional L8 time series, turned into L2A products through the L2A processor.

The NDVI is computed using a standard formulation applied to the S2 red (B4) and vegetation red-edge (B8a) bands. The LAI retrieval is performed from the bands 3, 4, 5, 6, 7, 8, 9, 12, 13
using machine learning to build a non-linear regression model. The regression model is estimated using reflectance values simulated using the ProSail model [RD.7, RD.8]. The NDVI metrics are extracted from NDVI time profiles through a double logistic function fitted to each pixel.

The process is illustrated in Figure 2-6, Figure 2-7 and Figure 2-8; an example of mono-date LAI product over South Africa is shown in Figure 2-9.

More information on the algorithms underlying this processor can be found in [AD.10].

Figure 2-6. Logical data flow of the L3B processor for generating LAI products

Figure 2-7. Logical data flow of the L3B processor for generating NDVI products
Figure 2-8. Sequence for an automatic production of a LAI product

Figure 2-9. Mono-date LAI product over South-Africa (Kwazulu-Natal) based on a S2 orbit from 19-01-2017
2.2.3 Cropland mask (L4A) processor

The cropland mask consists in a binary map separating annual cropland areas and other areas at 10 meters spatial resolution. This mask is produced along the agricultural season, on a monthly basis to serve for instance as a mask to monitor crop growing conditions. Its accuracy is expected to increase along the season as long as new images are integrated. Complete specifications are provided in Table 2-3.

Table 2-3. Dynamic crop masks specifications

<table>
<thead>
<tr>
<th>Properties</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial resolution</td>
<td>10 meters</td>
</tr>
<tr>
<td>Temporal resolution</td>
<td>First product delivered after an initialization period (up to 6 months), then updated monthly</td>
</tr>
<tr>
<td>Legend</td>
<td>Binary (crop - no crop)</td>
</tr>
<tr>
<td>Geometric accuracy</td>
<td>Same than L1C input data</td>
</tr>
<tr>
<td>Thematic accuracy</td>
<td>Progressively increasing along the season F1-score of 50% at the middle of the season and of 80% at the end</td>
</tr>
<tr>
<td>Format</td>
<td>GeoTIFF raster images</td>
</tr>
<tr>
<td>Projection</td>
<td>UTM / WGS84</td>
</tr>
<tr>
<td>Metadata</td>
<td>XML file</td>
</tr>
</tbody>
</table>

The product is delivered with several masks that will inform about the number of dates (i.e. L2A products) which are associated with the different classes resulting from the cloud detection (land, water, snow, cloud, cloud shadow and no data) during the period used to generate the mask. The metadata file contains the thematic accuracy figures (overall accuracy and contingency matrix).

The main and mandatory input of this L4A processor is a S2 time series, optionally completed by an additional L8 time series, turned into L2A products through the L2A processor.

Two alternative approaches are available for the development: the first one requires actual in-situ data distinguishing between crop and non-crop classes, whereas the second one builds its own sample collection from a “reference crop map”. A by-default reference crop map exists in the system but the users can provide its own one. In the first approach, in-situ data is mandatory to allow running the processor. In both cases, the product is generated using a Random Forest classifier, complemented on demand by an a posteriori filtering to smooth out the resulting mask [RD.9, RD.10].

The processor can be run at single- or multi-tile level. In the single-tile mode, the processed tile needs to be covered by crop and non-crop samples (coming either from in-situ data or a reference map). Conversely, the multi-tile mode allows running the algorithm over a larger area without the obligation of having samples over all tiles included in the area.

The process is illustrated in Figure 2-10 and Figure 2-11 and an example of crop mask is shown in Figure 2-12, with a 10-meter national cropland mask based on S2 and L8 data over Ukraine for the 2016 growing season.

More information on the algorithms underlying this processor can be found in [AD.11].
Figure 2-10. Logical data flow of the L4A processor

Figure 2-11. Sequence for an automatic production of a series of monthly crop masks
2.2.4 Crop type map (L4B) processor

The cropland type product is a map of the main crop types or groups at 10 meters spatial resolution. Crop types are classified only over the crop area identified in the crop mask. The map is generated twice over the season, with a first delivery at the middle of the season and the second one at the end. The accuracy expected to increase along the season as long as new images are integrated and crops are growing. Complete specifications are provided in Table 2-4.

Table 2-4. Crop type map specifications

<table>
<thead>
<tr>
<th>Properties</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial resolution</td>
<td>10 meters</td>
</tr>
<tr>
<td>Temporal resolution</td>
<td>Twice over the growing season: first product delivered after an initialization period (up to 6 months); second delivery at the end</td>
</tr>
<tr>
<td>Legend</td>
<td>Same as in-situ data provided by users</td>
</tr>
<tr>
<td>Geometric accuracy</td>
<td>Same than L1C input data</td>
</tr>
<tr>
<td>Thematic accuracy</td>
<td>Progressively increasing along the season Overall accuracy of 50% at the middle of the season and of F1-Score higher than 65% for the main classes</td>
</tr>
<tr>
<td>Format</td>
<td>GeoTIFF raster images</td>
</tr>
</tbody>
</table>
The product is delivered with several masks that will inform about the number of dates (i.e. L2A products) which are associated with the different classes resulting from the cloud detection (land, water, snow, cloud, cloud shadow and no data) during the period used to generate the mask. The metadata file contains the thematic accuracy figures (overall accuracy and contingency matrix).

The main and mandatory input of this L4A processor is a S2 time series, optionally completed by an additional L8 time series, turned into L2A products through the L2A processor. In-situ data for each expected crop type is also needed, as well as the crop masks generated by the system for the same dates.

The processing chain is based on a supervised Random Forest classification built on in-situ data to be collected during the growing season [RD.11]. In-situ information will be split for algorithm calibration and products validation. As for the Cropland mask (L4A) processor, the processing chain can be run at single- or multi-tile level.

The process is illustrated in Figure 2-13 and Figure 2-14 and an example is shown in Figure 2-15, with a crop type map over the Western Cape region (South Africa), based on S2 and L8 imagery acquired between May and December 2016.

More information on the algorithms underlying this processor can be found in [AD.12].
Figure 2-14. Sequence for an automatic production of crop type maps over the season

Figure 2-15. Crop type map over the Western Cape region, South Africa, based on S2 and L8 imagery acquired between May and December 2016
2.3 Visualization tool

A visualization tool is embedded in the Sen2-Agri system, since the version 2.0. It is a new Sen2-Agri service, which aims at meeting specific users’ requirements in terms of products visualization and analysis. The visualization tool is based on the GeoServer\(^1\) project.

GeoServer is an open source tool (under GNU General Public License (GPL)) which provides the possibility to share and publish geospatial data based on open standards. It is possible to display geospatial data into a map through a web interface. Moreover, it allows streaming data to other applications like Google Maps/Earth or to Geographic Information System (GIS) applications as QGIS or ESRI ArcGIS. These functionalities are based on the Open Geospatial Consortium (OGC) Web Mapping Service\(^2\) (WMS) and Catalog Services for the Web\(^3\) (CSW) standards. For example, the QGIS software supports natively the visualization of WMS data\(^4\) and also supports the search and download in a CSW catalog\(^5\).

Figure 2-16 provides an overview of the architecture and the components of the tool.

The tool is embedded into a Docker container to be easily deployed and to be independent from the other components of the Sen2-Agri system. It can be deployed on the Sen2-Agri platform or in any other machine having access to the output directory of the system.

---

\(^1\) [http://geoserver.org/](http://geoserver.org/)

\(^2\) [https://www.opengeospatial.org/standards/wms](https://www.opengeospatial.org/standards/wms)

\(^3\) [https://www.opengeospatial.org/standards/cat](https://www.opengeospatial.org/standards/cat)

\(^4\) [https://docs.qgis.org/3.4/en/docs/user_manual/working_with_ogc/ogc_client_support.html](https://docs.qgis.org/3.4/en/docs/user_manual/working_with_ogc/ogc_client_support.html)

2.4 S2-Toolbox integration

As explained in section 2.1, the system can be operated in manual mode, i.e. each processor can be run manually without installing and configuring the whole system. This can be done from a terminal but also through the S2-TB.

Integrating Sen2-Agri processors in the S2-TB does not require any special consideration since each processor is an independent python script based on an OTB application. This integration is controlled by the S2-TB via its Standalone Tool Adapter (STA) module. The integration is thus done on a per-processor basis. The main requirement for the STA module is a single execution point entry (i.e. a single executable). This is in line with the processors being independent OTB applications.

However, certain aspects have to be considered when integrating a Sen2-Agri processor:

- If there are specific pre-requisites for a processor, they should be addressed in the pre-execution template of the STA module definition. For example, certain processors require the source product to be split by tiles. The STA adapter accepts the full product as source. Therefore, a pre-processing step of splitting the source product into tiles has to be created. This will be done in STA by means of a pre-processing STA template;

- The command line parameters of the processor have to be described in the command line template of the STA module definition. The exhaustive list of command line parameters, of their types and of their constraints are available in this manual (see section 4.4);

- Any post-processing actions (e.g. cleaning up the temporary files) should be addressed in the post-execution template of the STA module definition. For example, for the processors that work on a tile-basis, a post-processing step of aggregating the tiles into a single product will be necessary. This will also be done in STA by means of a post-processing STA template.

NOTE: This integration is interesting to test parameterizations or run the processor over specific cases but it is more to be used for one-shot processing than for running the processors over large areas and/or seasons.

2.5 System implementation

The Sen2-Agri system is composed of the following layers:

- User interface layer
- Service layer
- Persistence layer

The architecture of the system as well as the relationship between the components identified is given in Figure 2-17.
2.6 The Sen2-Agri Orchestrator

Automated and manual (created from the Sen2-Agri website) processing jobs are executed via the Sen2-Agri Orchestrator. This is a system component that translates a job request into actual command invocations. The Sen2-Agri Orchestrator handles:

- parametrization of jobs: system-default and job-specific parameters are merged by the Orchestrator, allowing for any parameter to be customized by the user;
- job decomposition: the Orchestrator splits jobs into multiple steps, some of which can be executed in parallel;
- job invocation: the steps of a job are sent to SLURM¹, allowing the system to take advantage of a processing cluster, if configured as such;
- job monitoring: the Orchestrator is notified of failed steps and cancels the remaining steps that depend on those;
- automated job creation: when processing jobs are scheduled via the website, the Orchestrator decides whether a job can be executed and chooses the input data.

As such, if running in the automated mode is desired, the Orchestrator is an essential part of the Sen2-Agri system.

¹ SLURM is an open source, fault-tolerant, and highly scalable cluster management and job scheduling system for large and small Linux clusters (https://Slurm.schedmd.com/)
2.7 The Sen2-Agri Services

Sen2Agri Services is an evolution of the former download component. It is decoupled from the Sen2Agri system and it offers the following functionalities:

- Pluggable architecture for querying and fetching products from remote repositories: out-of-the box, the module comes with plugins for Copernicus SciHub (for Sentinel-2 L1C products), for USGS (for Landsat-8), for AWS (for Sentinel-2 and Landsat-8) and for PEPS (for Sentinel-2; due to the workflow implemented by PEPS – a product may not be immediately downloaded - this plugin should not be used).

- More configurable support for L1 products querying and retrieval:
  - Exposure of query parameters so that each data source can be customized;
  - Separation of query and download sources; for example, SciHub can be configured for querying the list of available products for a site while AWS can be configured as download site for the products returned by the SciHub query;
  - Independent parallelism thresholds for data sources; for example, SciHub can be configured to allow maximum 2 connections (which is its actual limit), while AWS can be configured to allow 4+ connections.

- A RESTful API for accessing/controlling different functionalities of the Sen2Agri System from an external application. The operations that can be performed using the services are:
  - Querying the list of sites, delete a specific site or get the seasons of a site;
  - Enable or disable downloading for a certain site and/or satellite and extract the status of downloading for a site and/or satellite;
  - Stop or start the downloading of a certain site and/or satellite;
  - Get the data sources configuration;
  - Receive download progress notifications from the sen2agri system.
3. System deployment

3.1 Installation package content

The installation package of the Sen2-Agri system consists in the following main folders:

- **install_script** - contains the installation scripts that are used to create the distribution and to install the system and the tool needed for the integration of the Sen2-Agri processors in SNAP;
- **rpm_binaries** – the RPM files for all other system components (SLURM, orchestrator, downloader, processors);
- **sen2agri-services** - the folder containing the Sen2Agri services to be installed.

Additionally, the following archives are provided beside the system installation package:

- **gipp_maja.zip** - the gipp files used by the L2A processor (MAJA) during the processing of the L1C products. These gipp files correspond to the MAJA 3.1 and 3.2.2 version. Please note that if another more recent MAJA version will be installed, these gipp files should be updated accordingly;
- **srtm.zip and swbd.zip** - these archives contain the Digital Elevation Model (DEM) files needed by the L2A processor (MAJA).

Although the additional archives are part of the installation package, they are provided separately due to their size and because they will normally not change from a version to another. Before starting the installation of the system, these two files should be copied inside the installation package, in the “Sen2AgriDistribution”, as described in the paragraph 3.3.1.

**IMPORTANT NOTE:** The L2A processor, named MAJA, is not provided with the system installer and shall be downloaded separately from CNES site. If MAJA is installed before running the Sen2Agri installer, the Sen2Agri install script will look for the executable and eventually prompt for its path. Nevertheless, for an easier installation, MAJA version 3.2.2 installer can be downloaded and copied in a subfolder called “maja” in the root of the installer (on the same level with **install_script** and **rpm_binaries** folders) and the system will automatically install it from here. This is the recommended method as the Sen2Agri installation script will take care of some steps like setting some rights to the MAJA installation path (by default MAJA is installing in /opt/maja but its installer does not give read rights for this folder to other users than root and because of this the sen2agri-service user fails to access this folder). Please see next section for downloading MAJA 3.2.2.

3.2 Prerequisites for automatic usage

Before starting the installation of the Sen2-Agri system for an automatic use, a set of preconditions is expected to be met in terms of system hardware, operating system or packages installed.

The minimum hardware requirements are:

- disk space for system installation: 80 GB;
- disk space for the resulted products (/mnt/archive - see section 3.3.1): 10 TB are enough for a national case (i.e. a study area ≈ 500,000 km²), even if 15 TB is recommended;
• disk space for the internal directory where the shapefiles of the created sites will be uploaded in order to be used by the system (/mnt/upload - see section 3.3.1): 1 GB is needed;
• RAM: 64 GB for a national case and 48 GB for a local case;
• number of CPUs: 1 dual-core for a local case (i.e. one S2 swath) and 2 quad-core for a national case.

CentOS 7 64-bit is required by the Sen2-Agri system. The graphic interface (X Server) should be also installed if it is intended to use the Sen2Agri Configurator (see Appendix F.1 Processing System Configuration Utility).

The Linux user who will perform the installation should have the root rights (i.e. be part of the sudoers list), the installation process will perform a lot of yum installation.

Also, in case of proxy server, the “/etc/yum.conf” file should be properly configured. Complete instructions to perform this configuration of the proxy server in yum.conf see are given in https://www.centos.org/docs/5/html/yum-sn-yum-proxy-server.html. For example, some lines with the ones below should be added in yum.conf file:

```bash
# The proxy server - proxy server:port number
proxy=http://mycache.mydomain.com:3128

# The account details for yum connections
proxy_username=yum-user
proxy_password=qwerty
```

The Sen2Agri installation package does not contains anymore the installation package for MAJA, which has to be downloaded separately from the CNES website, performing the following steps:

• For MAJA:
  o Go to https://logiciels.cnes.fr/en/node/58?type=desc
  o Click on the “Download” tab;
  o Click on the “Download” button of the version 3.2.2 TM version;
  o Read the license carefully;
  o Fill the form with your information;
  o Don’t forget to accept the license;
  o The downloading is starting.

The MAJA application is needed for the generation of L2A products from S2 L1C and L8 L1T products and it has to be installed prior to the Sen2-Agri system installation. Instructions for installation are provided by CNES inside of the MAJA installation package.

In any case, the Sen2-Agri installer will check if MAJA is already installed and will prompt the user to provide the installation path if it is not found in the default one (which normally is /opt/maja). For any information about MAJA, its own Software User Manual can be checked [AD.8; AD.13].

IMPORTANT NOTE: If MAJA is installed before the system installation and not by the Sen2Agri installer, by default, MAJA is installed in the /opt/maja directory but with read/write
For downloading S2 and L8 acquisitions from the distribution server, the system will need 2 accounts to be provided after the installation:

- An account and a password for the ESA Sentinels Scientific Data Hub (SciHub), allowing downloading S2 products. This account can be obtained accessing [https://scihub.copernicus.eu/dhus/#/home](https://scihub.copernicus.eu/dhus/#/home) and then accessing the “Sign up” section;

- An account and a password for the United States Geological Survey (USGS) portal, allowing downloading L8 products. The account can be created accessing the link [https://ers.cr.usgs.gov/login/](https://ers.cr.usgs.gov/login/) and following the “Create new account” option. Nevertheless, in order to activate the downloads for your account the following additional operations need to be performed:
  
  - Login to your account on USGS
  - Go to [https://ers.cr.usgs.gov/profile/access](https://ers.cr.usgs.gov/profile/access)
  - Push the “Request Access” button and fill the information in the form that will be displayed:
    
    ▪ “Datasets Required” – Landsat Collection 1 Level-1
    ▪ “Data Products Required” - Landsat 8 OLI/TIRS C1 Level -1
    ▪ “Scenes required” – In your case you can fill 2 monthly or 24 yearly (depends on your site dimension). You can also put more if you consider you will have additional sites to monitor
    ▪ “Scripting Capabilities”: python, java
    ▪ For the other fields, you will have to fill the requested information

  After that, you should receive after several hours an email from USGS confirming the activation. Once this is done, the L8 should start also being downloaded.

Sub-section 3.3.3.1 describes how to configure these accounts in the system.

Also, although this file is needed later after the system is completely installed, an .zip archive containing the shapefile delineating your site of interest is required. This archive should include a “.shp”, a “.prj”, a “.dbf” and a “.shx” file. More information about the way this archive is used is given in the sub-section 4.2.1).

### 3.3 Installation and configuration for automatic usage

Depending on your current usage of the Sen2-Agri system (already using or new user), a specific installation procedure or update procedure must be followed:

- **it is my first installation of the system**: follow the instructions given in the section 3.3.1 (installation procedure) to install the new version 2.0;

- **my current installed version of the system is before version 1.7**: you need to uninstall your current version and install again the new version 2.0 following the instructions given in the section 3.3.1 (installation procedure);

- **my current installed version of the system is version 1.7 or later**: you can update your system to the new version 2.0 following the instructions given in the section 3.3.2
(update procedure). Please note that upgrading the existing version to the version 2.0 will not perform any upgrade from MACCS to MAJA since this operation should be done manually by the user.

**IMPORTANT NOTE:**

It is not advisable to switch from MACCS to MAJA while a season production is still in progress as unpredictable behavior might occur. The switching from MACCS to MAJA should be done only if all the sites were removed from the system.

### 3.3.1 Installation procedure

The following procedure allows installing the system for automatic usage. It is worth noting that the installation for automatic usage implies that all functionalities of the system will be installed and made available, including those for running the manual mode.

Before of the installation, some default directories have to be created in the system either physically or mounted. The default directories are:

- `/mnt/archive` – this is the default working directory for the system;
- `/mnt/upload` – the folder where the files from the user are uploaded (for example shapefiles with in situ data).

These directories can be changed after installation using the “`sen2agri-configurator`” application (see Appendix F.1 Processing System Configuration Utility).

The user `sen2agri-service` (which will be added by the installer, see below) should have full access rights (read/write/execution) over these default directories. In order to do this, launch the following command from a terminal, as a user which is in sudoers list:

```
sudo mkdir /mnt/archive
sudo mkdir /mnt/upload
sudo chmod -R a+rwx /mnt/archive /mnt/upload
```

The directories “`/mnt/archive`” and “`/mnt/upload`” can also be mount points to another external storage.

The users on a system and the root user are not recommended to be used as there cannot be known a priori. During the installation, the Sen2-Agri installer will create the user ‘`sen2agri-service`’. This “`sen2agri-service`” user will be the user under which all the Sen2-Agri services are executed. The “`sen2agri-service`” user will be created by the installer; he should have write access to these mount points. Since the “`sen2agri-service`” user might not be already created before the installation, these external directories should therefore have initially access rights for all users.

As explained in section 3.1, the MAJA gipp files, SRTM and SWBD files are provided separately, because of their size and also because they will normally not change from one version to another and don’t need to be re-downloaded at each update. The following options exist to have these dataset usable by the Sen2Agri system:
• Copy the srtm.zip and swbd.zip in the root folder of the installation package (in the Sen2AgriDistribution directory). In this case, the installer will automatically unzip them and copy them into the right location (/mnt/archive);
• Or, the user unzip the files from the two archives in folders /mnt/archive/srtm and /mnt/archive/swbd respectively. In this case, the installer will detect their presence in the right location and will use them directly.
• Unzip the MAJA gipp archive in the Sen2AgriDistribution directory (on the same level as install_script, rpm_binaries etc. directories);

An example of a structure of the package, including the SRTMs and MAJA is presented in Figure 3.1.

![Figure 3.1](image.png)

Figure 3.1 Structure of the installation package (including the SRTMs and MAJA)

To install and configure the Sen2-Agri system, with all its dependencies, it is enough to run the following script that can be found in the distribution package:

```bash
# open a terminal -- go into /install_script folder:

cd /path/to/install_script

# Run the install script

sudo ./sen2agriPlatformInstallAndConfig.sh
```

This script will automatically install the system (SLURM, orchestrator, processors, downloader, website, the database and all other dependencies). The installation is completely automatic, requiring minimum interaction from the user.
A PostgreSQL\(^1\) database is used to keep track of the system information. The database is installed and configured by the installation script. The reference tables are already filled in with all the necessary details when the database is installed. For a list of reference tables and access for advanced users, the reader is referred to Appendix B - Sen2-Agri database tables for manual configuration.

**IMPORTANT NOTE:** The installation scripts should have execution rights. If by some operations, these rights are lost, the following command could be optionally executed before starting the installation:

```
chmod -R a+x /path/to/Sen2AgriDistribution/
```

### 3.3.2 Update procedure

If you already installed the Sen2-Agri system (version 1.7 or later), you can install the version 2.0 through an update procedure, described here below.

The version 2.0 includes different updates that are described in the “changelog” tab of our website. The version 2.0 also includes the Ground Image Processing Parameters (GIPP) used by the MAJA atmospheric correction processor aligned with those used by the CNES production centre.

**IMPORTANT NOTE:**

The system cannot be turned from using MACCS to use MAJA and this GIPP files cannot be updated if you are currently using the system. Indeed, to ensure the compatibility of the L2A time series, the same parameters need to be used for the whole season and the update should not be done before all the downloads and L2A processing for all existing sites defined in the system. All the sites should have all L2A processed till the end of the seasons and after the installation of MAJA, new seasons should not be added to the current sites.

This section identifies varying scenarios depending on your current use of the Sen2-Agri system and for each of them, describes the update procedure.

1. **I have installed but not yet started to run my Sen2-Agri system**

In this case, it is advisable before upgrading to version 2.0 to switch also to MAJA. The steps for downloading and installing MAJA are described in the section 3.2.

After installing MAJA, the Sen2-Agri configuration should be updated in order to specify MAJA as the L2A processor. In order to do this, the “sen2agri-config” application could be used to update the corresponding entry in the database:

---

\(^1\) PostgreSQL is a powerful, open source object-relational database system – see [https://en.wikipedia.org/wiki/PostgreSQL](https://en.wikipedia.org/wiki/PostgreSQL)
Optionally, the entry can be changed also using directly a command from the terminal:

```
sudo -u postgres psql sen2agri -c "update config set value = '<PATH_TO_MAJA_EXECUTABLE>' where key = 'demmacs.maccs-launcher';"
```

You can download the new installation package and simply upgrade your system by running these two update scripts that will make the general Sen2-Agri update and replace the GIPP files (you will be asked by the script the L2A processor for which MAJA should be installed):

```
cd ~/Downloads/Sen2AgriDistribution/install_script
sudo ./update.sh
sudo ./update_gipp.sh
```

2. **I am running my Sen2-Agri system over one or multiple sites**
   a. **All processing over these sites are completed**
      ➔ Sites are *disabled* on GUI

      *All the processing over the sites you previously created are completed and you intend to launch new sites in the future?*

      You can follow the steps from point the point 1. with the condition to not activate or add new seasons to the existing sites otherwise unpredictable behaviour can be encountered if mixing L2A products produced by MACCS with the ones produced by MAJA.

   b. **Processing over (some of) these sites are still ongoing**
      ➔ Sites are *enabled* on GUI
The processing over the sites you previously created is not completed?

Still, you can already make the general update of your system but you will have to update the GIPP files later on, when the current processing will be completed otherwise errors could occur during processing.

You can download the new installation package and follow the here-below instructions:

- **Upgrade your system by running the generic script that will update the sen2agri-services application**

  ```bash
  cd ~/Downloads/Sen2AgriDistribution/install_script
  sudo ./update.sh
  ```

- **Update the GIPP files once the download and L2A processing’s are completed**

  To ensure the compatibility of the L2A time series, you should not update the gipp before all the download and L2A processing’s over these sites are completed.

  Once this is the case and before launching a new site, you can run the `update_gipp` script, after installing MAJA as described at point 1 above:

  ```bash
  cd ~/Downloads/Sen2AgriDistribution/install_script
  sudo ./update_gipp.sh
  ```

- **Recover missing downloads**

  As sometimes, even if a retry mechanism is implemented, a product cannot be downloaded (from various reasons starting with the unavailability of SciHub or some errors caused by the missing disk space). In these cases, the download of the product is completely aborted after expiring the configured retries.

  In order to recover the missing S2 / L8 downloads marked as aborted, you will have to run a specific procedure to update a system database called `downloader_history`, which is linking each product to a series of properties. As explained here, each S2/L8 product is related to a `status_id` informing about the status of its processing (from the download to the L2A generation). The missing products are flagged with a `status_id` = ‘4’, i.e. ‘download aborted’.

  To force the sen2agri-services application to retry the download of these products, you have to change the status of these products to 3 (i.e. ‘failed but will retry’) and the number of retries to ‘0’. You can do so by running to two here-below commands:

  ```bash
  sudo -u postgres psql sen2agri -c "update downloader_history set no_of_retries = '0' where status_id = '4'"
  sudo -u postgres psql sen2agri -c "update downloader_history set status_id = '3' where status_id = '4'"
  ```

### 3.3.3 Additional configurations

Some additional steps are needed to complete the installation of the system. Some of them are mandatory, while other ones are optional.
3.3.3.1 Configure user accounts in the Sen2Agri-Services for the S2 and L8 download [mandatory]

This step is mandatory.

There is one file which has to be modified after the installation in order to provide the user name and password for both SciHub and USGS servers.

The files can be edited using the vim\(^1\) editor (or other editor preferred by the user) as it is usually available on any Linux operating system using the following command:

```
# sudo vim /usr/share/sen2agri/sen2agri-services/config/services.properties
```

**IMPORTANT NOTE:**

After modifying the file sen2agri-services.properties, the sen2agri-services should be restarted using the command:

```
# sudo systemctl restart sen2agri-services
```

3.3.3.2 Configure your proxy for the Sen2Agri-Services downloaders [optional]

This step is optional.

If a proxy server is used for the internet connection, the file mentioned in section 3.3.3.1 have to be manually edited by adding as root user the following configuration keys in the /usr/share/sen2agri/sen2agri-services/config/sen2agri-services.properties file:

- `proxy.host=<host of the proxy server>`
- `proxy.port=<port of the proxy server>`
- `proxy.user=<user for the proxy>`
- `proxy.password=<password of the proxy user>`
- `proxy.type=<Type of the proxy>` that can be one of the following values:
  - `DIRECT` - Represents a direct connection, or the absence of a proxy;
  - `HTTP` - Represents proxy for high level protocols such as HTTP or FTP;
  - `SOCKS` - Represents a SOCKS (V4 or V5) proxy.

**IMPORTANT NOTE:**

After modifying the file sen2agri-services.properties, the sen2agri-services should be restarted using the command:

```
# sudo systemctl restart sen2agri-services
```

These values will be used for the first execution of the system, after a fresh installation, when nothing is configured in the system. The next configurations of the system need to be performed only via the IHM of the system that has a dedicated page for this, accessible from the “data sources” item of the main menu (see 4.2.6).

---

\(^1\) Vim is a highly configurable text editor built to make creating and changing any kind of text very efficient. It is included as “vi” with most UNIX systems and with Apple OS X (http://www.vim.org/download.php)
3.3.3.3 Configure your proxy for the web browser [optional]

This step is optional.

In order to allow the web browser having the map world interface functional, it can be needed to add your HTTP/HTTPS proxy information in the Sen2-Agri terminal or directly into the Sen2-Agri web browser.

From the terminal, you can use the following command lines with the right information:

```
export http_proxy=http://$PROXY_USER:$PROXY_PASSWORD@$PROXY_HOSTNAME:$PROXY_PORT
export https_proxy=https://$PROXY_USER:$PROXY_PASSWORD@$PROXY_HOSTNAME:$PROXY_PORT
```

From the Sen2-Agri web browser, you should set the proxy information in our settings. For example for Firefox, you should set in the Advanced section / Network / Settings the correct information.

3.3.3.4 Sen2-Agri website users management [optional]

This step is optional.

By default, the website interface is provided with the following user having admin rights:

Username: sen2agri
Password: sen2agri

The website uses a user hierarchy to change the user parameters. There are two types of users: admin users and non-admin users. One might take advantage of this “non-admin users”, when several sites are run in parallel on the same Sen2-Agri system. In this case, a “non-admin user” can be assigned to each site. In this case, in the website interface, the non-admin user can see only information related to the assigned site.

To add users in the database, a special tab in the web interface is available since the version 1.8.3 of the system. Using this tab, you can check the current information corresponding to a user and add a new one (Figure 3-3). You can also define the role of a specific user and, in the case of a regular user (not admin), limit the access only to certain sites.
After a user is added, the corresponding password need to be set during the first login, by selecting “Set a password” from the login page (Figure 3-4).

Additionally, users can also be added in the database using the procedure described in section “Appendix E – How to add new users”.

3.3.4 First interaction with the Sen2-Agri system

After the installation is finished and all the additional configurations are performed, the web interface of the system can be accessed from a web browser if the user is connected to the same machine as the Sen2-Agri system. The following address needs to be inserted: http://localhost/login.php or http://127.0.0.1/login.php

The following interface (Figure 3-5) will be displayed.
Figure 3-5. Main page of the Sen2-Agrí web interface

The interface can be also accessed in the same manner from another PC that has access to the port 80 of the Sen2-Agrí system, by providing in the browser the IP address or the computer name of the Sen2-Agrí system. For example, the local instance of the Sen2Agri system website can be accessed via http://127.0.0.1/login.php or http://localhost/login.php.

After the user provided a correct user and password (by default, “sen2agri” and “sen2agri”), the user is automatically redirected to the “Sites” page where he will be invited to configure his site to monitor (see section 4.2.1).

3.4 Prerequisites for manual usage

One way to operate the system in the manual mode is to run the processor through the S2-TB, i.e. the processors will be executed and integrated in SNAP. To do so, Java Runtime Environment (minimum version 8.0) and SNAP software (minimum version 5.0.5) must be installed.

In order to install Java, the following command can be executed:

```
$ sudo yum -y install java
```

The SNAP application can be downloaded from: http://step.esa.int/main/download/. Clear guidelines on how to download and install SNAP can also be found http://step.esa.int/main/toolboxes/snap/. When downloading, make sure to choose the right installer for your operating system.

!! IMPORTANT NOTE: you need at least the SNAP 5.0.5 version to use the adapters !!

This version is available at: http://step.esa.int/downloads/5.0/installers/esa-snap_sentinel_unix_5_0.sh.

3.5 Installation and configuration for manual usage

3.5.1 Installing Sen2-Agrí core components

The Sen2-Agrí core components are those that allow manual execution of the processors, either by the command line, or by integration with SNAP.

The core components installation includes:

- the OTB applications used by the processors;
• the Sen2-Agri processors provided as python scripts.

The core components installation does not include:

• the database;
• the Sen2-Agri Orchestrator and SLURM;
• the web interface of the Sen2-Agri system.

If only the core components are needed, the following script (available in the distribution package) must be executed:

```bash
## open a terminal -- go into /install_script folder:
cd /path/to/install_script
## Run the install script
sudo ./sen2agriPlatformInstallAndConfigCore.sh
```

In order to quickly check if OTB and the Sen2-Agri processors are successfully installed, a command like the following one could be executed:

```bash
otbcli BandsExtractor
```

If the processors are successfully installed, the expected output is the following one:

```
ERROR: Waiting for at least one parameter...
```

This is the BandsExtractor application, version 5.0.0

Extract only the needed bands from a temporal series of images.

Parameters:

- progress <boolean> Report progress
MISSING -il <string list> The xml files (mandatory)

- out <string> [pixel] The concatenated images [pixel=uint8/uint16/int16/uint32/int32/float/double] (default value is float) (optional, off by default)

 Conversely, the installation has not been successful if the output is similar to the following one.

```
ERROR: Could not find application " BandsExtractor"
ERROR: Module search path: /bin/../lib/otb/applications:
ERROR: Available modules :
BVImageInversion
BVInputVariableGeneration
```

© UCL-Geomatics 2019

This document is the property of the Sen2-Agri partnership, no part of it shall be reproduced or transmitted without the express prior written authorisation of UCL-Geomatics (Belgium).
In that case, the installation process should be performed again, by checking if all the installation steps were correct and the user has the right to perform the installation.

Using only these components, no custom configuration is needed. At this point, the processors can be executed by using the command line functionality. In order to execute one of the processors, one of the following scripts needs to be executed:

- composite_processing.py – used to create free cloud composite L3A products;
- lai_retrieve_processing.py – to create LAI L3B products;
- pheno_processing.py – to create phenological NDVI L3B products;
- CropMaskFused.py – to create crop mask single-tile or multi tile (stratification) L4A products;
- CropTypeFused.py – to create crop type single-tile or multi tile (stratification) L4B products.

Complete examples on how to manually execute these scripts are given in section 4.4.

**REMINDER:** The manual mode is already available if the system was installed completely, for automatic mode.

### 3.5.2 Configuring and using SNAP

In order to execute the Sen2-Agri processors through SNAP, the SNAP software shall be installed in advanced (see section 3.4).

The SNAP adapters (.nbm files) for the Sen2-Agri processors are located in the installation package in the directory `/install_script/converters`. They must be installed and configured in SNAP, by following the steps:

1. Copy the .nbm files on disk, all in the same directory. The necessary .nbm files are:
   - CompositeProcessor.nbm
   - CropMaskProcessor.nbm
   - CropTypeProcessor.nbm
   - VegetationPhenoProcessor.nbm
   - VegetationStatusProcessor.nbm
   - MACCSSProcessor.nbm
2. Start SNAP Desktop
3. Go to menu **Tools -> Plugins**
4. In the **Plugins** window, select the tab **Downloaded**, and click the button **Add Plugins**…

5. In the **Add Plugins** dialog, navigate to the directory where the `.nbm` files were copied. Select all files by using the CTRL key, and click the button **Open**

6. After the files are added, in the **Plugins** dialog, check the boxes associated with the plugins (corresponding to the adapters) and click the button **Install**
7. In the Plugin Installer dialog, where the plugins are enumerated, click Next, check the box ‘I accept the terms…’, and click the button Update.

8. Accept the validation warning by clicking the button Continue.

9. Check the radio button Restart Now and click the button Finish.
10. Wait for SNAP to restart

11. Go to menu Tools -> Manage External Tools. The adapters should be visible in the table (along with other adapters that were added before)

12. For each adapter, follow the steps:
   - Select the adapter in the table
   - Click the button Edit
- In the **System Variables** tab, observe the defined variables. Change their values (if necessary) as follows:
  
  o **WORKING_DIR_PATH**: the path where the temporary files can be created and where the final product can be saved. This folder must have a subfolder for each processor (called **Composite, CropMask, CropType, Vegetation, VegetationLAI**) and each subfolder must have a subfolder called **out**, where the final product will be saved.

  The folders where the temporary files and the final products will be saved can be configured separately in the adapter, but as being a preliminary version of the delivery, it is preferable to leave it as it is.

  o **SCRIPT_PATH**: `[sen2agri-sources]/scripts` (where `[sen2agri-sources]` is the location where the git repository was cloned)

- Click the button **OK**

**NOTE:**

- The variables have values independent of user path, so the values should not need to be modified
- It is possible that the variables values are auto-updated after the first adapter was updated. Please check all the variables from all the adapters!

As already mentioned, detailed explanations on how to manually run a processor from SNAP are given in dedicated paragraphs in section 4.4.
<table>
<thead>
<tr>
<th>Ref</th>
<th>Sen2-Agri_SUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issue</td>
<td>Page</td>
</tr>
<tr>
<td>3.2</td>
<td>52</td>
</tr>
</tbody>
</table>
4. How to Use the Sen2-Agri System

4.1 General

To put it simple, the Sen2-Agri system transforms S2 L1C and L8 L1T products into higher level products by means of its L2A (atmospheric corrections), L3A (cloud-free composite), L3B (vegetation status), L4A (crop mask) and L4B (crop type) processors.

The normal functioning state of the system is the **automated mode**, i.e. the automatic download and processing of the S2 L1C and L8 L1T products and production of higher level products.

The automatic execution of the system is based on the following items:

- At least one site needs to be defined inside the system. Additional sites can be created and edited from the “Sites” page of the Sen2-Agri website;
- For each site, the extent and the season to monitor need to be defined. At least one season (winter or summer) should be defined, specifying the start and end dates of the season. These parameters are associated to by-default values, which can be modified by the user. The season dates are defined by a month, a day and a year;
- Based on these two information (extent and season definition), the system automatically launches the downloader and the processors to generate the higher-level products.

As for the download, some precisions need to be done:

- The system starts to download products 3 months before the start of the season date. For example, considering a start of the season on 2016-05-01, products download will thus start on 2016-02-01. These 3 first months of data are needed to initialize the L2A processor (see section 4.4.1) but will not be used to produce the higher-level products;
- Similarly, the system also tries to download products during 2 months after the end season (so considering an end of season of 2016-09-01, it means until 2016-11-01), to avoid losing any products inside the season which may be added after the end of season day;

Each processor is associated with main and advanced parameters and each parameter has by-default values. Each processor is automatically run using these by-default values except if the user changes them. More information regarding the processors parameters can be found in section 4.4. This by-default parameterization results from consistent studies carried out during the project over globally distributed sites [RD.9, RD.10, RD.11, RD.12]. It is therefore recommended to be cautious when modifying them.

As an alternative to the **automated mode**, each processor can also be run in the **manual mode**. In this case, processors can be invoked in various ways:

- from a terminal window, by executing a specific Python script;
- from S2-TB and SNAP software;
- from the Sen2-Agri web GUI, for L3A, L3B, L4A and L4B processors. In this case, immediate execution is not performed. Instead, a scheduled job is created and the processor will be executed when the system decides it has enough resources to do it. This is done with the purpose of not interfering with the system production jobs.
The following sections present how running the system in the automated and manual modes, and also gives information on the way to deal with in-situ data that are required for some processors.

4.2 First steps for the automated usage of the system

4.2.1 Configure a new site

By-default, the Sen2-Agri platform does not contain any geographical site. This is to the user to create at least one site. This site is then used by the downloader processor (see Appendix F.7 Sen2-Agri Downloader) to download S2 L1C and L8 L1T products. The first step for the automated usage is therefore to create and configure a new site.

In order to create a new site, the Sen2-Agri system interface should be used. Select the “Sites” tab and press the “Create new site” button. At that moment, the dialog box for configuring a new site is displayed (Figure 4-1).

![Figure 4-1. Dialog box for site creation through the Sen2-Agri system interface](image)

In the “Add New Site” dialog box, the following information should be provided:

- Site name: this should be a unique site name defined in the system;
- Upload a shapefile with the site extent. The uploaded file will have to be in a “.zip” archive that contains all the “.shp”, “.dbf”, “.prj” and “.shx” files. The site can be defined by more than a polygon, i.e. the shapefile can contain a multi-part polygon. Make sure that your polygon (or each polygon in the case of a multi-part polygon) is not made of more than 150 points;

**IMPORTANT NOTE:** Starting from the version 1.6 the season to monitor is not defined anymore when creating the site. In other terms, the site will contain no season after its creation by default. Season dates will be added or removed later on, when editing the site (see section 4.2.2).
After providing all information in the dialog box, the user should choose “Save New Site” and the site is created.

An error will be given if the information provided is not correct (for example in the case of invalid shapefile archive).

Once at least one season is added and the site is “Enabled”, the download and processing will be automatically started. If the user needs to check and/or change some of the processors parameters in the system, it is highly indicated to disable the site and re-enable it when all the parameters are set.

4.2.2 Editing the sites to define season, insitu data, strata data and processors to execute

As specified in the previous paragraph, by default, a new created site is not associated with season dates, which are to be defined in a second step. To this end, on the Sen2-Agri system interface, go to the “Sites” tab and press the “Edit” button corresponding to your site (Figure 4-2).

Figure 4-2. Site page of the Sen2-Agri interface, to edit site or seasons

The dialog shown in Figure 4-3 will then be opened.
In order to add a season, press the + icon in the “List of Seasons” section. It will allow you specifying the following parameters: “Season Name”, “Season start”, “Season mid” and “Season End”. It will also give you the possibility to decide if the season should be enabled or not.

At this moment, the user has the possibility to specify which processor will be run in the automated mode for the site and season just defined. This is done simply by ticking the boxes associated with the processors you want to activate, as illustrated in Figure 4-4. Note that the L2A processor is always run automatically.

Also, the user can select what sensor to be enabled for downloads: by default, L8 and S2 sensors are “Enabled” but only the L8 sensor can be disabled. The dialog presents also the tiles for each satellite, the user having the possibility to remove from these, if not all tiles intersecting the site are interesting for him. The editors for the tiles become enabled once the site is enabled.
By selecting the processors to be active for the season, the corresponding scheduled jobs will be added to the season for automatic processing. Please refer to section 4.3 for additional information about the scheduled jobs. The processors that can be active are:

- **L2A** – this option is selected and enabled by default (cannot be modified) as this processor is the one that provide input products for all other processors;

- **L3A** – a “Repeat” type schedule will be created that will execute every 1st day of the month in order to create the L3A corresponding to the beginning of the previous month.

- **L3B** – a “Cycle” type schedule will be created that will execute every 1 day in order to build one L3B products for each L2A products that occurred during the last interval.

- **L4A** – a “Cycle” type schedule will be created to run each month, the first run being at the defined season middle date.

- **L4B** – a “Once” type cycle scheduled that is executed at the end of the season.

**IMPORTANT NOTE**: This possibility to specify the active processors is only available when the season is defined. It cannot be edited later on.

Once the season is defined, it can be saved using the icon. The changes can be cancelled using the icon .

© UCL-Geomatics 2019

This document is the property of the Sen2-Agri partnership, no part of it shall be reproduced or transmitted without the express prior written authorisation of UCL-Geomatics (Belgium).
At this moment, after the creation of the season the user has the possibility to enable the site. In this section the user can also upload the insitu data and strata data (in the form of shapefiles) by selecting the button “Upload” that will open the following dialog with “Insitu data” or “Strada data” options (Figure 4-4-1).

After the successful creation of a season, a set of scheduled jobs corresponding to the selected processors are automatically created for that season. These jobs are available in the “Dashboard” section of the Sen2-Agri system interface and will start to execute once the site is Enabled.

### 4.2.3 Optionally adapt the main processor parameters

As introduced in sections 2.1 and 2.2, the system is made of 4 main processors, allowing producing temporal syntheses, vegetation status indicators, crop masks and crop type maps. Each of these processors is parameterized by-default when the system is installed. The automated mode will be run using this by-default parameterization except if the user wants to change them. The change of the by-default parameters can be performed by using the “sen2agri-config” application or by manually changing the parameters values in the database (see Appendix B - Sen2-Agri database tables for manual configuration and Appendix F – Advanced system configuration items). It should be noted that this by-default parameterization were found to be the best generic values after a consistent benchmarking carried out during the project over globally distributed sites [RD.9, RD.10, RD.11, RD.12].

The parameters of each processor are presented in detail in the section 4.4 about manual operations.
4.2.4 Activate or change the site parameters

If the site was not activated during the season definition, it can be “Enabled” at any time by navigating in the “Sites” tab and accessing to the Site Edition dialog by pressing the “Edit” button (Figure 4-5).

![Edit Site dialog](image)

This dialog box allows activating the site, but also changing site parameters like the season, insitu data or strata data.

Seasons can be edited using the button or deleted using . The list of activated processors for a certain season can be viewed by moving the mouse over the text “hover to reveal” but not modified.

Insitu data and strata data can be uploaded using the button “Browse” from each section. The upload of the file is actually performed when the “Save Site” button is pressed.

The user has the possibility to enable or disable the L8 downloads by unchecking the “L8” button from the “Enable sensor” area.

Please note that there exist two levels of enabling options:

- At season level: the user can enable or disable different seasons of a site;
- At site level: the site, with all seasons (if more than one season has been defined) can be enabled or disabled at once.

© UCL-Geomatics 2019
This document is the property of the Sen2-Agris partnership, no part of it shall be reproduced or transmitted without the express prior written authorisation of UCL-Geomatics (Belgium).
4.2.5 Delete a site

The user has the possibility to delete a site at any time by navigating in the “Sites” tab and accessing to the Site Edition dialog by pressing the “Edit”. From this dialog box the user should press the button “Delete Site” (Figure 4-5-1)

![Edit Site dialog box](image)

By default, this dialog box allows the user to delete the site with all associated products. But, the user can keep some products of this site by unchecking the boxes associated with the products. Then the user will select the button “Confirm Delete Site” (Figure 4-5-2)
4.2.6 Optionally adapt the data sources configuration

Since version 1.8.3, it is possible to adapt the data sources configuration directly from the web interface (Figure 4-5). For example, you can define which downloading platform to use for the “query” and “download” tasks of the downloading, the user and password corresponding to your downloading platform, define the destination folder for the downloaded products, and many more.

Figure 4-5. The "data sources" tab of the web interface - Configure the data sources parameters
4.2.7 Activating download of Sen2Cor L2A products

By default, the system is downloading L1C products from ESA SciHub. These L1C are pre-processed into L2A products using MAJA. Nevertheless, the system can be configured in order to download directly the L2A products created with Sen2Cor from ESA SciHub.

In order to activate globally the L2A download, the key 'downloader.use.esa.l2a' can be set to true into the config table by executing the following command:

```
sudo -u postgres psql sen2agri -c "UPDATE config SET value = 'true' WHERE key = 'downloader.use.esa.l2a';"
```

The configuration can be performed per site by altering/inserting the above key into the config table for each site:

```
sudo -u postgres psql sen2agri -c "INSERT INTO config(key, site_id, value) VALUES ('downloader.use.esa.l2a', <site_id>, 'true');"
```

Where `<site_id>` is the id of the site that needs to be configured.

4.2.8 Monitor the downloading

The Monitoring tab of the web interface (available since version 1.8.3) gives various information concerning the downloading of the data. In the “Download statistics” bar, you can find:

- The number of images that were downloaded (and %);
- The number of images that are downloading currently (and %);
- The number of images from which the downloading failed (and %);
- The number of images from which the downloading failed but will retried (and %).

It gives you also the estimated number of products still to download.

![Monitoring tab of the web interface](image)

Figure 4-6. The “monitoring” tab of the web interface - A view to the downloading

4.2.9 Monitor the processing

The Execution Dashboard enables the user to monitor the Sen2-Agri system. It consists of a web page that dynamically loads data from the Sen2-Agri HTTP listener service.
The System Overview tab of this Dashboard (Figure 4-7) presents the jobs that are currently being executed by the system as well as the performance of the server(s).

![System Overview Dashboard](image)

Figure 4-7. Sen2-Agri Execution Dashboard – System Overview

For each server (top of the system overview), the following resources are presented:

- CPU (current value and a graph showing the evolution over the last 15 minutes);
- RAM (current value and a graph showing the evolution over the last 15 minutes);
- Swap (current value and a graph showing the evolution over the last 15 minutes);
- Disk – the used disk space from the available disk space;
- Load (current value and a graph showing the evolution over the last 15 minutes) – the load on the server is provided by the operating system in the form of 3 numbers that represent the 1/5/15 minutes averages of the run-queue length: the sum of the number of processes currently running plus the number of processes waiting (queued) to be run.

For every job, the following information is presented:

- Job Id;
- Processor;
- Site;
- Triggered By - the possible values are:
o “Scheduler request” – when the job was created by the automated scheduling mechanism;
o “User request” when the job was created with a custom job request;

- Triggered On;
- Status (of the job) - the possible values are:
  o Running;
  o Canceled;
  o Paused;
- Tasks Completed/Running;
- For the current task:
  o Module;
  o Tiles Completed/Running;
- Actions (depending on the status):
  o As possible actions for a job, we have “Pause”, “Resume”, “Cancel” and “View Config”;
  o “View config” – will show the parameters used when the job was submitted.

Each job has associated two buttons:

- Pause or Resume:
  o The button “Pause” is displayed when the current job is running and allows the user to interrupt the job execution
  o The button “Resume” is displayed when the current job is interrupted and allows the user to resume the job execution
- Cancel : This button allows the user to cancel the job execution

Detailed informations about each job (both the currently executing ones and also the finished jobs) can be found also in the “Monitoring” tab, section “Jobs history”. These information refer to:

- the job id as an unique identifier;
- the time when the job finished;
- the name of the processor that launched the job;
- the site for which was launched;
- the status of the job - still running, finished OK or finished with error;
- the job start type – if it was a custom job or was a job triggered by a schedule job.
- a link to a page presenting details about the steps that were executed by the processing during that job.

Figure 4-8 The jobs history information
Accessing the “output” link for a job, the user will be redirected to a page showing the steps executed, the information about each step consisting in:

- the name of the step.
- The command that was invoked for that step;
- The output generated during execution;
- The errors that were generated during execution;
- The exit code of the executable that was invoked during the execution of the step.

The exit code of the step is displayed only when the execution of the step finished (successfully or with error) in the other cases only an icon is displayed for describing the current status of the step:

- ⏳ - the step is pending for being executed (submitted but not yet executed);
- ⌛ - the step is currently executing;
- ⏪ - the job was paused by the user;
- ✗ - the step gave an error;
- ❌ - the job was cancelled;
- ⚠ - the step is waiting for some inputs;
- ❓ - the status of the step is unknown.

Figure 4-9 – The job execution steps details

4.2.10 Where to find output products

The “Product” tab of the Execution Dashboard (Figure 4-10) shows a tree view with the products that are available. On clicking the site node, all the products available for this site become visible. If the icon of the product is clicked, the overview is visible on the map. Conversely, if the download symbol in the right of the product is clicked, the product is downloaded locally.
The paths where the products can be found are the following ones:

- **L1C / L1T** products found in `/mnt/archive/dwn_def/` (see configuration parameters “Write directory for L8” and “Write directory for S2” from Appendix F.7 Sen2-Agri Downloader)
- **L2A** products found in `/mnt/archive/maccs_def` (see configuration parameter “demmacs.output-path” from Appendix F.8 Sen2-Agri Demmacs_launcher)
- **L3A** products stored in a folder having the short name of the site*, which can be found in “/mnt/archive”. For example, “/mnt/archive/south_africa/l3a”.
- **L3B** products stored in a folder having the short name of the site*, which can be found in “/mnt/archive”. For example, “/mnt/archive/south_africa/l3c” or “/mnt/archive/south_africa/l3d”.
- **L4A** products stored in a folder having the short name of the site*, which can be found in “/mnt/archive”. For example, “/mnt/archive/south_africa/l4a”.
- **L4B** products stored in a folder having the short name of the site*, which can be found in “/mnt/archive”. For example, “/mnt/archive/south_africa/l4b”.

### 4.2.11 Provide in situ data during monitoring period

For the L4A and L4B processors, in-situ data can be provided. This is not mandatory to run the L4A processor, but it is for the L4B. When the L4A processor runs without in-situ data, it makes use of a reference land cover map. The system has a by-default map (the 2015 ESA-CCI landcover map), which can be changed by the user. More information about the in-situ data

* This could be changed via the key “archiver.archive_path” from the sen2agri-configuration
shapefile format and the reference land cover map required by the system is given in section 4.5. This section also explains how to upload the data in the system.

In order to upload a new in-situ data shapefile, the user should select the “Edit” site option from the “Sites” page of the Sen2Agri GUI and then upload the in-situ or strata data in the sections found on the “Upload Files” page.

By selecting the button “Insitu data” or “Strata data” (Figure 4-4-1)

![Figure 4-11 - Adding in situ data](image)

If there exist already an in-situ data previously uploaded for this site, the “Existing file” file edit will display the current shapefile name.

By clicking “Browse” button, a zip file containing in-situ data can be selected.

Please note that the in-situ data is not actually uploaded until the site is not saved.

The same mechanism applies for the “Strata data”

Once the user selected a file, the “Insitu data” and/or “Strata data” text will be marked with a red star that will indicate that a file was selected for that section (useful if the user is collapsing the section).

### 4.3 How to add scheduled jobs

Scheduled jobs tab of the provide the possibility to define executions that are performed at certain moments of time (one time executions or periodic executions) and that use the products available at that moment of time, according also to the configuration defined in the database.

The scheduled tasks can be added for all the processors excepting the L2A processor that is a built-in execution module. The scheduled tasks can be added from the “Dashboard” page of the Sen2Agri website.

The scheduled tasks will use the default parameters configured in the database for that processor. The default parameters used can be seen also in each tab of the processor from the “Dashboard” page:
The adding of a scheduled task is similar for all processors.

In order to add a scheduled task the following steps should be followed:

- Press the “Add Job” button from the tab corresponding to the processor.

A new entry will be added in the list of scheduled tasks:

- In the “Job Name” field add an identifier for the added task (ex. “myjob”).
- In the “Site” field should be selected one of the existing sites.
- From the “Schedule” field, the following options are available:
  - “Once” – this option specifies to the scheduled task to run only once at the specified date. Selecting this option, a new field “Date” will be added to the entry where the user can select the date when the scheduled task should run:
Figure 4-14 – Adding “Once” scheduled type

- “Cycle” – this option allows to schedule a task at a certain date and then to reschedule the task after a number of specified days. For example, a configuration with the start date of 1st of March and “Repeat after” set to 4 days will execute the scheduler on 1st of March then on 5th of March, then on 10th of March and so on.

Figure 4-15 – Adding “Cycle” scheduled type

- “Repeat” – this option allows to schedule a task at a certain date and then to reschedule the task at a certain day of the month. For example, a configuration with the start date of 1st of March and “On every” set to 4 will execute first on 1st of March, then on 4th of March, then on 4th of April, then on 4th of May and so on.

Figure 4-16 – Adding “Repeat” scheduled type

In order to delete an added scheduled job, the “Delete” button can be used.
4.4 Perform manual operations

The manual operations of the Sen2-Agri system can be done by:

- Executing the processors from the command line using a specific Python scripts for each processor. This execution is available both in automated and manual mode installation of the system. Manually invoking the processors from the command-line does not use the SLURM scheduler. These manual executions from command line are intended for special scenarios like SNAP integration in which SLURM might not be available. When running in manual mode, some of the potential parallelism present in a job is not exploited. If needed, multiple manual jobs can be started at the same time;

- Executing the processors from SNAP, using the Standalone Tools Adapters. These tools will actually invoke the same Python scripts like in the case of the command line invocations;

- Executing the processors from the web interface. This execution is similar as with the command line invocation except that the execution will be performed using the orchestration mechanism that allows executions of parallel tasks and optionally on different machines and priorities (according the SLURM configuration). Though, this option is available only if the system was installed for an automated usage.

**NOTE:** When running the system components in the manual mode, make sure that the output location is writable by the current user;

The following sections explain how to manually run the system for a set of operations.

### 4.4.1 Atmospheric Correction (L2A processor)

An atmospheric correction on a L1C and L1T products is performed using the L2A processor (also called demmacs) which is used in the automatic mode by the demmaccs_launcher processor. For each tile to process, the MACCS/MAJA executable needs the output of DEM processor. The demmaccs processor automatically runs both DEM and MACCS processors for each tile. It has the parameters listed in Table 4-1.

**Table 4-1. L2A processor (demmaccs) launching parameters**

<table>
<thead>
<tr>
<th>input</th>
<th>input L1C directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>-w</td>
<td>--working-dir</td>
</tr>
<tr>
<td></td>
<td>working directory</td>
</tr>
<tr>
<td>--srtm</td>
<td>SRTM dataset path</td>
</tr>
<tr>
<td>--swbd</td>
<td>SWBD dataset path</td>
</tr>
<tr>
<td>--gipp-dir</td>
<td>directory where gipp files are to be found</td>
</tr>
<tr>
<td>--maccs-launcher</td>
<td>the path where MACCS/MAJA binary file is installed</td>
</tr>
<tr>
<td>--processes-number-dem</td>
<td>maximum number of parallel processes which run DEM processor. The DEM</td>
</tr>
</tbody>
</table>
processor handles one tile only, thus multiple DEM processors may be run in the same time to minimize the processing time. NOTICE: the value for this key should be handled with caution and should not exceed 60% from the server number of logical CPUs (default 20)

--processes-number-maccs

maximum number of parallel processes which run MACCS/MAJA processor. The MACCS/MAJA processor handles one tile only, thus multiple MACCS/MAJA processors may be run in the same time to minimize the processing time. NOTICE: the value for this key should be handled with caution and should not exceed 60% from the server number of logical CPUs (default 8)

--maccs-address

MACCS/MAJA has to be run from a remote host. This should be the ip address of the pc where MACCS/MAJA is to be found.

--prev-l2a-tiles

the MACCS/MAJA processor may be run in 2 modes: L2INIT and L2NOMINAL. The former will be used if there aren’t any previously processed tiles by MACCS/MAJA. The latter will be used if there are already previously processed tiles and can be used by MACCS/MAJA to enhance the final result. The tiles should be inserted as a list separated by spaces

--prev-l2a-products-paths

the path for each previously processed tile found in the –prev-l2a-tiles list. For each present tile there should be a corresponding path. The paths are separated by spaces

output

output location

4.4.1.1 Executing L2A processor from a terminal window

The L2A processor can be launched from terminal by using the script demmaccs.py:

```bash
./demmaccs.py --working-dir /mnt/archive/temp_maccs/workingDir/ --srtm /mnt/archive/srtm/ --swbd /mnt/archive/swbd/ --processes-number-dem 20 --glipp-dir /mnt/archive/glipp/ --maccs-launcher /opt/maja/3.2.2/bin/maja /mnt/archive/Sen2Agri_DataSets/2015/Ukraine/EO/L8/LC81810252015240LGN00/ /mnt/archive/temp_maja/
```

4.4.1.2 Executing L2A processor from SNAP

The L2A Processor can be launched from SNAP, after setting the parameters, by following the steps (Figure 4-18):

- Open SNAP
- Go to menu Tools -> External Tools
• Select MACCSPprocessor
• In the execution dialog, go to tab Processing Parameters*. These parameters should be updated according to the L2A processor that is used (MACCS or MAJA) for GIPP and executable location.
• Change the parameters as desired and click the button Run
• A progress bar will appear, and when it finishes, the produced product can be found in the corresponding location

![MACCS Processor]

Figure 4-18. Launching MACCS/MAJA processor in SNAP – Execution parameters

4.4.2 Creating a composite (L3A) product

An overview of the processor is given in the section 2.2.1 and a more detailed description of the algorithms can be found in [AD.9]. This product will be formatted according to the Product Specification Document (PSD) [AD.5].

4.4.2.1 Processors parameters

When a new L2A product is available or when a user requests a composite product, the L3A processor can be invoked with the parameters given in Table 4-2.

| Table 4-2. L3A processor’s parameters |

* Definition of all parameters is given in Appendix F.9 All advanced parameters table
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>input</td>
<td>the list of products descriptors (xml files)</td>
<td>file list</td>
</tr>
<tr>
<td>res</td>
<td>resolution of the output image: if the value is 0, the resolution of the input image will be kept; if the value is 10 or 20, the image will be resampled at 10 or 20.</td>
<td>integer</td>
</tr>
<tr>
<td>outdir</td>
<td>the output directory</td>
<td>string</td>
</tr>
<tr>
<td>syntdate</td>
<td>the date of the synthesis, which must be defined as the central date of the synthesis time window. The format of the date should be YYYYMMDD.</td>
<td>integer</td>
</tr>
<tr>
<td>synthalf</td>
<td>the half synthesis period (days)</td>
<td>integer</td>
</tr>
<tr>
<td>applocation</td>
<td>processors build folder</td>
<td>string</td>
</tr>
<tr>
<td>bandsmap</td>
<td>the file that contains the mapping of the bands. It has a structure like: primary_sensor_name,secondary_sensor_name primary_band_index,target_resolution,secondary_band_index ... For example, for Sentinel 2 and Landsat 8, having the Sentinel2 as a primary mission, we will have the following structure. Please note the -1 for the Sentinel2 bands for which there is no Landsat 8 band. SENTINEL-2A,LANDSAT_8 2,10,2 3,10,3 4,10,4 5,20,-1 6,20,-1 7,20,-1 8,10,5 9,20,5 12,20,6 13,20,7</td>
<td>string</td>
</tr>
<tr>
<td>scatteringcoef</td>
<td>Scattering coefficient file. This file is requested in S2 case ONLY</td>
<td>file</td>
</tr>
<tr>
<td>tileid</td>
<td>Tile id (optional)</td>
<td>string</td>
</tr>
<tr>
<td>siteid</td>
<td>The site id, represented as 2 digits (optional)</td>
<td>string</td>
</tr>
<tr>
<td>lut</td>
<td>Path to the colour mapping table, for example /usr/share/sen2agri/composite.map (optional)</td>
<td>string</td>
</tr>
</tbody>
</table>

#### 4.4.2.2 Manual execution from terminal

The L3A Processor can be launched from terminal, as for instance:

```bash
./composite_processing.py --syntdate 20160101 --synthalf 25 --input "/mnt/archive/maccs/south_africa/l2a/1C81710792015346LGN00_L2A/L8_TEST_L8C_L2VALD_171079_20151221.HDR"

"/mnt/archive/maccs/south_africa/l2a/S2A_OPER_PRD_MSIL2A_PDMC_20160108T173225_R135_V20160108T082023_20160108T082023.SAFE/S2A_OPER_SSC_L2VALD_35JMK____20160108.HDR"
```

© UCL-Geomatics 2019

This document is the property of the Sen2-Agri partnership, no part of it shall be reproduced or transmitted without the express prior written authorisation of UCL-Geomatics (Belgium).
"/mnt/archive/maccs/south_africa/l2a/S2A_OPER_PRD_MSIL2A_PDMC_20160114T023112_R035_V20160111T082928_20160111T082928.SAFE/S2A_OPER_SSC_L2VALD_35JMK____20160111.HDR" \ 
--res 10 --outdir "/mnt/archive/temp/20160101/35JMK" --bandsmap
/usr/share/sen2agri/bands_mapping_s2_L8.txt --scatteringcoef
/usr/share/sen2agri/scattering_coeffs_10m.txt

4.4.2.3 Manual execution from SNAP

The L3A Processor can be launched from SNAP, after setting the parameters, by following the steps (Figure 4-19):

- Open **SNAP**
- Go to menu **Tools -> External Tools**
- Select **CompositeProcessor**
- In the execution dialog, go to tab **Processing Parameters**
- Change the parameters as desired and click the button **Run**
- A progress bar will appear, and when it finishes, the produced product can be found in the corresponding location

\* Definition of all parameters is given in Appendix F.9 All advanced parameters table
4.4.2.4 Manual execution from Web Interface

The L3A Processor can be also executed manually from the web interface by accessing the “Custom Jobs” tab in the Web Interface and by selecting the “L3A processor” tab.

This editor allows the user running the L3A processor (i) by using only a subset of the available L2A input products and/or (ii) by running it with other parameters than the default ones defined in the automated processing.

When the editor is opened (Figure 4-20), the user has to select the site of interest and then, he has to define the following parameters:

- The set of L2A input products to be used: when the user selects his site of interest, the “Available input files” list is populated with the existing L8 and S2 L2A products available in the system. The user can select a subset of products and enable / disable one of the two sensors S2 and L8;
- The synthesis date (“syndate” in Table 4-2);
- The half synthesis interval (“synthalf” in Table 4-2): the by-default value is 25 days. It can be modified;
- Spatial resolution (“res” in Table 4-2).
Figure 4-20. Manual execution of L3A processor from Web Site

The advanced parameters of the processor can be accessed ticking the "Show advanced parameters" box. The advanced parameters will be displayed as illustrated in Figure 4-21. Their definition can be found in Appendix F.9 All advanced parameters table.
4.4.3 Creating a LAI vegetation status (L3B) product

As outlined in section 2.2.2, the generation of LAI vegetation status is included in the L3B processor. A more detailed description of the algorithms can be found in [AD.10]. This product will be formatted according to the Product Specification Document (PSD) [AD.5].

4.4.3.1 Processors parameters

When a new L2A product is available or when a user requests a LAI product, the L3B LAI vegetation status processor can be invoked with the parameters given in Table 4-3.
Table 4-3. L3B LAI processor’s parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>application</td>
<td>the path where the sen2agri is built</td>
<td>string</td>
</tr>
<tr>
<td>input</td>
<td>the list of products descriptors (xml files)</td>
<td>file list</td>
</tr>
<tr>
<td>res</td>
<td>resolution of the output image:</td>
<td>integer</td>
</tr>
<tr>
<td></td>
<td>- if the value is 0, the resolution of the input image will be kept;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- if the value is 10 or 20, the image will be resampled at 10 or 20.</td>
<td></td>
</tr>
<tr>
<td>outdir</td>
<td>the output directory</td>
<td>file</td>
</tr>
<tr>
<td>inlaimonodir</td>
<td>Directory where input mono-date LAI products are located (only for reprocessing) (optional)</td>
<td>file</td>
</tr>
<tr>
<td>rsrfile</td>
<td>The RSR file (/path/filename)</td>
<td>file</td>
</tr>
<tr>
<td>rsrconfig</td>
<td>The RSR configuration file specific to each mission (/path/filename)</td>
<td>file</td>
</tr>
<tr>
<td>tileid</td>
<td>Tile Id (optional)</td>
<td>string</td>
</tr>
<tr>
<td>modelsfolder</td>
<td>The folder where the models are located. If not specified, is considered the outdir (optional)</td>
<td>string</td>
</tr>
<tr>
<td>generatemodel</td>
<td>Generate the model (YES/NO) (optional) Option used for re-generating the models (needed if there are new products for which it was not created yet a L3B product)</td>
<td>string</td>
</tr>
<tr>
<td>generatemonodate</td>
<td>Generate the mono-date LAI (YES/NO) (optional)</td>
<td>string</td>
</tr>
<tr>
<td>genreprocessedlai</td>
<td>Generate the reprocessed N-Days LAI (YES/NO) (optional)</td>
<td>string</td>
</tr>
<tr>
<td>genfittedlai</td>
<td>Generate the Fitted LAI (YES/NO) (optional)</td>
<td>string</td>
</tr>
<tr>
<td>siteid</td>
<td>The site id, represented as 2 digits (optional)</td>
<td>string</td>
</tr>
</tbody>
</table>

4.4.3.2 Manual execution from terminal

The L3B Processor, with S2 tiles, can be launched from terminal, as for instance:

```
lai_retrieve_processing.py --input
/mnt/archive/maccs_def/mali/l2a/S2A_OPER_PRD_MSIL2A_PDMC_20160718T093045_R008_V2016
0717104833_20160717104833.SAFE/S2A_OPER_SSC_L2VA__20160717.HDR --res 10
--outdir
/mnt/archive/temp/test_lai --rsrconfig
/usr/share/sen2agri/rsr_cfg.txt --modelsfolder
/mnt/archive/temp/test_lai --generatemodel YES --generatemonodate YES
--genreprocessedlai NO --genfittedlai NO
```

Note: The “input” parameter must be the .HDR file of a Sentinel-2 or LANDSAT 8 L2A products.

4.4.3.3 Manual execution from SNAP

The L3B LAI Processor can be launched from SNAP, after setting the parameters, by following the steps (Figure 4-22):
- Open SNAP
- Go to menu Tools -> External Tools
- Select VegetationStatusProcessor
- In the execution dialog, go to tab Processing Parameters*
- Change the parameters as desired and click the button Run
- A progress bar will appear, and when it finishes, the produced product can be found in the corresponding location

![Figure 4-22. L3B LAI Vegetation Status Processor in SNAP – Execution parameters](image)

4.4.3.4 Manual execution from Web Interface

The L3B LAI Processor can be also executed manually from the web interface by accessing the “Custom Jobs” tab in the Web Interface and by selecting the “L3B LAI processor” tab.

This editor allows the user running a LAI product (i) by using only a subset of the available L2A input products and/or (ii) by running it with other parameters than the default ones defined in the automated processing.

When the editor is opened (Figure 4-23), the user has to select the site of interest and then, he has to define the following parameters:

* Definition of all parameters is given in Appendix F.9 All advanced parameters table
• The set of L2A input products to be used: when the user selects his site of interest, the “Available input files” list is populated with the existing L8 and S2 L2A products available in the system. The user can select a subset of products and enable / disable one of the two sensors S2 and L8;

• Spatial resolution (“res” in Table 4-3);

• Select what type of product should be generated:
  o L3B – mono-date LAI;
  o L3C – multi N-date LAI;
  o L3D – fitted LAI at the end of the season.

• Define the backward window, which is the N-Day window to be used for LAI reprocessing.

![Figure 4-23. Manual execution of LAI processor from Web Site](image)

The advanced parameters of the processor can be accessed ticking the “Show advanced parameters” box. The advanced parameters will be displayed as illustrated in Figure 4-24. Their definition can be found in Appendix F.9 All advanced parameters table.
4.4.3.5 Activate a new version of the algorithm

A new version of the algorithm used to generate LAI products has been developed, based on the BV-Net method developed by INRA and already implemented in the Sentinel-2 Toolbox. This new version is not yet available by-default but can be activated using the following procedure:

1. Enter in the Sen2-Agri database, via the command:

   ```
   sudo -u postgres psql sen2agri
   ```

2. In the table “config”, activate the new calculation method, via the command:

   ```
   update config set value='1' where key='processor.l3b.lai.use_inra_version'
   ```

4.4.4 Creating a NDVI vegetation status (L3B) product

Another output of the L3B processor is the NDVI phenological metrics, as explained in the section 2.2.2. A more detailed description of the algorithms can be found in [AD.10]. This product will be formatted according to the Product Specification Document (PSD) [AD.5].
NOTE: This processor can only be launched from the Custom Jobs; it is not part of the automated mode. It has been validate using SPOT Take 5 products, but not yet using S2 data. It is recommended to cautiously assess the quality of the products before using them.

4.4.4.1 Processors parameters

When a new L2A product is available or when a user requests an NDVI product, the L3B NDVI phenological metrics processor can be invoked with the parameters given in Table 4-4.

Table 4-4. L3B NDVI processor’s parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>application</td>
<td>the path where the sen2agri is built</td>
<td>string</td>
</tr>
<tr>
<td>input</td>
<td>the list of products descriptors (xml files)</td>
<td>file list</td>
</tr>
<tr>
<td>outdir</td>
<td>the output directory</td>
<td>file</td>
</tr>
<tr>
<td>tileid</td>
<td>Tile id (optional)</td>
<td>string</td>
</tr>
<tr>
<td>res</td>
<td>Resample to this resolution. Use the same resolution as input files if you don't want any resample.</td>
<td>integer</td>
</tr>
<tr>
<td>siteid</td>
<td>The site id, represented as 2 digits (optional)</td>
<td>string</td>
</tr>
</tbody>
</table>

4.4.4.2 Manual execution from terminal

The L3B NDVI Processor can be executed from a terminal, as for instance:

```
./pheno_processing.py --input \\
"/mnt/archive/test/Borno_South/S2A_OPER_PRD_MSIL2A_PDMC_20160607T134539_R136_V20160606T094258_20160606T094258.SAFE/S2A_OPER_SSC_L2VALD_32PRS____20160606.HDR" \\
"/mnt/archive/test/Borno_South/S2A_OPER_PRD_MSIL2A_PDMC_20160817T185332_R136_V20160815T093042_20160815T094108.SAFE/S2A_OPER_SSC_L2VALD_32PRS____20160815.HDR" \\
"/mnt/archive/test/Borno_South/S2A_OPER_PRD_MSIL2A_PDMC_20160917T211632_R136_V20160915T184521_20160915T184521.SAFE/S2A_OPER_SSC_L2VALD_32PRS____20160915.HDR" \\
"/mnt/archive/test/Borno_South/S2A_OPER_PRD_MSIL2A_PDMC_20161017T155352_R136_V20161016T154013_20161016T154013.SAFE/S2A_OPER_SSC_L2VALD_32PRS____20161016.HDR" \\
--resolution 10 --outdir /mnt/archive/test/L3B/Nigeria_Borno/
```

4.4.4.3 Manual execution from SNAP

The L3B NDVI Processor can be launched from SNAP, after setting the parameters, by following the steps (Figure 4-25):

- Open SNAP
- Go to menu Tools -> External Tools
- Select VegetationPhenoProcessor
• In the execution dialog, go to tab Processing Parameters*
• Change the parameters as desired and click the button Run
• A progress bar will appear, and when it finishes, the produced product can be found in the corresponding location

![Figure 4-25. L3B NDVI Vegetation Phenology Processor in SNAP – Execution parameters](image)

**4.4.4.4 Manual execution from Web Interface**

The L3B NDVI metrics Processor can be also executed manually from the web interface by accessing the “Custom Jobs” tab in the Web Interface and by selecting the “L3E PHENO processor” tab.

This editor allows the user running the Phenological NDVI processor (i) by using only a subset of the available L2A input products and/or (ii) by running it with other parameters than the default ones defined in the automated processing.

When the editor is opened (Figure 4-26), the user has to select the site of interest and then, he has to define the following parameters:

• The set of L2A input products to be used: when the user selects his site of interest, the “Available input files” list is populated with the existing L8 and S2 L2A products available in the system. The user can select a subset of products and enable / disable one of the two sensors S2 and L8;
• Spatial resolution (“res” in Table 4-4).

* Definition of all parameters is given in Appendix F.9 All advanced parameters table
4.4.5 Creating a crop mask (L4A) product

An overview of the processor is given in the section 2.2.3 and a more detailed description of the algorithms can be found in [AD.11]. This product will be formatted according to the Product Specification Document (PSD) [AD.5].

### 4.4.5.1 Processors parameters

When a new L2A product is available or when a user requests a crop mask product, the L4A crop mask processor can be invoked with the parameters given in (Table 4-5).

**Table 4-5. L4A processor’s parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>main-mission-segmentation</td>
<td>Only use main mission products for the segmentation</td>
<td>String (SPOT/SENTINEL/LANDSAT)</td>
</tr>
<tr>
<td>refp</td>
<td>the reference polygons <em>(only for with in-situ data mode)</em></td>
<td>file</td>
</tr>
<tr>
<td>ratio</td>
<td>the ratio between the validation and training polygons</td>
<td>decimal</td>
</tr>
<tr>
<td>input</td>
<td>the list of products descriptors (XML files)</td>
<td>file list</td>
</tr>
<tr>
<td>trm</td>
<td>The temporal resampling mode</td>
<td>String (resample/gapfill – by default: gapfill)</td>
</tr>
<tr>
<td>nbtrsample</td>
<td>the number of samples included in the training set <em>(only for without in-situ data mode)</em></td>
<td>Integer</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>rseed</td>
<td>the random seed used for training</td>
<td>integer</td>
</tr>
<tr>
<td>window</td>
<td>The window, expressed in number of records, used for the temporal features extraction <em>(only for with-in situ data mode)</em></td>
<td>Integer (by default: 6)</td>
</tr>
<tr>
<td>lmbd</td>
<td>the lambda parameter used in data smoothing <em>(only for without in-situ data mode)</em></td>
<td>Integer (by default: 2)</td>
</tr>
<tr>
<td>nbcomp</td>
<td>the number of components used by dimensionality reduction</td>
<td>Integer (by default: 6)</td>
</tr>
<tr>
<td>spatialr</td>
<td>the spatial radius of the neighbourhood used for segmentation</td>
<td>Integer (by default: 10)</td>
</tr>
<tr>
<td>refr</td>
<td>the reference raster when in situ data is not available <em>(only for without in-situ data mode)</em></td>
<td>String</td>
</tr>
<tr>
<td>eroderad</td>
<td>the radius used for erosion <em>(only for without in-situ data mode)</em></td>
<td>Integer (by default: 1)</td>
</tr>
<tr>
<td>alpha</td>
<td>the parameter alpha used by the mahalanobis function <em>(only for without in-situ data mode)</em></td>
<td>Decimal (by default: 0.01)</td>
</tr>
<tr>
<td>rfnbtrees</td>
<td>the number of trees used for training</td>
<td>Integer (by default: 100)</td>
</tr>
<tr>
<td>rfmax</td>
<td>maximum depth of the trees used for Random Forest classifier</td>
<td>Integer (by default: 25)</td>
</tr>
<tr>
<td>rfmin</td>
<td>minimum number of samples in each node used by the classifier</td>
<td>Integer (by default: 5)</td>
</tr>
<tr>
<td>minarea</td>
<td>The minimum number of pixel in an area where, for an equal number of crop and nocrop samples, the crop decision is taken</td>
<td>Integer (by default: 20)</td>
</tr>
<tr>
<td>pixsize</td>
<td>the size, in meters, of a pixel</td>
<td>Integer (by default: 10)</td>
</tr>
<tr>
<td>targetfolder</td>
<td>The folder where the target product is built</td>
<td>File</td>
</tr>
<tr>
<td>keepfiles</td>
<td>Keep all intermediate files (optional)</td>
<td>String (by default: false)</td>
</tr>
<tr>
<td>siteid</td>
<td>The site id, represented as 2 digits (optional)</td>
<td>String</td>
</tr>
<tr>
<td>lut</td>
<td>Color LUT for previews <em>(see /usr/share/sen2agri/crop-mask.lut)</em></td>
<td>String</td>
</tr>
<tr>
<td>outprops</td>
<td>Output properties file</td>
<td>String (optional)</td>
</tr>
<tr>
<td>strata</td>
<td>Shapefiles with polygons for the strata</td>
<td>Shape file (.shp)</td>
</tr>
<tr>
<td>stratum-filter</td>
<td>The list of strata to use in training and classification</td>
<td></td>
</tr>
<tr>
<td>tile-filter</td>
<td>The list of tiles to apply the classification to</td>
<td>List of tile ids</td>
</tr>
<tr>
<td>classifier</td>
<td>The classifier <em>(rf or svm)</em> used for training <em>(default rf)</em></td>
<td>String</td>
</tr>
<tr>
<td>red-edge</td>
<td>Include Sentinel-2 vegetation red edge bands</td>
<td>Boolean (by default: True)</td>
</tr>
<tr>
<td>sp</td>
<td>Per-sensor sampling rates</td>
<td>Integer (by default: Sentinel = 10; SPOT = 5; Landsat = 16)</td>
</tr>
</tbody>
</table>
The L4A Crop Mask processor generates 2 outputs: a raw crop mask and a filtered crop mask. At this point of time, the tests that we have done don’t allow us guaranteeing the quality of the post-filtering. The processor has been run over more than 10 sites and the quality has been variable. In most of the cases, the filtering allowed improving the accuracy of the raw crop mask. Yet, in some cases, it was not the case. As a result, it is recommended to check the quality of the filtered crop mask before using it. This check has to be done not only by looking at the accuracy metrics but also by making a visual assessment of the product.

4.4.5.2 Manual execution from terminal

The L4A Crop Mask Processor can be executed from a terminal, as for instance (using S2 tiles and using the `CropMaskFused.py` script which supports the stratification):

```
```
The following example presents the usage when multiple satellites are used.

The intersecting L8 tile comes right after the HDR files of a given S2 tile. Note also in this example that the L8 tiles are duplicated for each S2 tile that they intersect, and are counted also for that tile.

```bash
./CropMaskFused.py
  -main-mission-segmentation
  -nbtrsample 4000
  -pixsize 10
  -refr /mnt/archive/reference_data/ESACCI-LC-L4-LCCS-Map-300m-P5Y-2010-v1.6.1.tif
  -outdir /mnt/data/archive/nigeria_borno/l4a/ 
  -input 
  /mnt/archive/test/Borno_South/S2A_OPER_PRD_MSIL2A_PDMC_20160607T134539_R136_V201606
  06T094258_20160606T094258.SAFE/S2A_OPER_SSC_L2VALD_32PQS____20160606.HDR 
  /mnt/archive/test/Borno_South/S2A_OPER_PRD_MSIL2A_PDMC_20160607T185332_R136_V201608
  15T093042_20160817T185332_R136_V201608
  06T094258_20160606T094258.SAFE/S2A_OPER_SSC_L2VALD_32PQS____20160815.HDR 
  /mnt/archive/test/Borno_South/S2A_OPER_PRD_MSIL2A_PDMC_20160607T044353_R136_V201606
  06T094258_20160606T094258.SAFE/S2A_OPER_SSC_L2VALD_32PQT___20160606.HDR 
  /mnt/archive/test/Borno_South/S2A_OPER_PRD_MSIL2A_PDMC_20160817T214327_R136_V201608
  15T093042_20160817T214327_R136_V201608
  06T094258_20160606T094258.SAFE/S2A_OPER_SSC_L2VALD_32PQT___20160815.HDR 
  /mnt/archive/test/Borno_South/S2A_OPER_PRD_MSIL2A_PDMC_20160607T134539_R136_V201606
  06T094258_20160606T094258.SAFE/S2A_OPER_SSC_L2VALD_32PQS____20160606.HDR 
  /mnt/archive/test/Borno_South/S2A_OPER_PRD_MSIL2A_PDMC_20160817T185332_R136_V201608
  15T093042_20160817T185332_R136_V201608
  06T094258_20160606T094258.SAFE/S2A_OPER_SSC_L2VALD_32PQS____20160815.HDR 
  /mnt/archive/test/Borno_South/S2A_OPER_PRD_MSIL2A_PDMC_20160607T044353_R136_V201606
  06T094258_20160606T094258.SAFE/S2A_OPER_SSC_L2VALD_32PRT___20160606.HDR 
```

4.4.5.3 Manual execution from SNAP

The L4A Crop Mask Processor can be launched from SNAP, after setting the parameters, by following the steps (Figure 4-27):

- Open SNAP
- Go to menu Tools -> External Tools
- Select CropMaskProcessor
- In the execution dialog, go to tab Processing Parameters
- Change the parameters as desired and click the button Run
- A progress bar will appear, and when it finishes, the produced product can be found in the corresponding location.

* Definition of all parameters is given in Appendix F.9 All advanced parameters table
Figure 4-27. L4A Crop Mask Processor in SNAP – Execution parameters

4.4.5.4 Manual execution from Web Interface

The L4A CropMask Processor can be also executed manually from the web interface by accessing the “Custom Jobs” tab in the Web Interface and by selecting the “L4A IN-SITU processor” or “L4A w/o IN-SITU processor” tab.

This editor allows the user running the L4A processor (i) by using only a subset of the available L2A input products and/or (ii) by running it with other parameters than the default ones defined in the automated processing.

When the “L4A IN-SITU processor” editor is opened (Figure 4-28), the user has to select the site of interest and then, he has to define the following parameters:

- The set of L2A input products to be used: when the user selects his site of interest, the “Available input files” list is populated with the existing L8 and S2 L2A products available in the system. The user can select a subset of products and enable / disable one of the two sensors S2 and L8;
- Upload a custom shapefile containing in-situ data. The shapefile should be a .zip archive containing all shapefile components;
- Resolution (“pixsize” in Table 4-5);
- Ratio (“ratio” in Table 4-5).
Figure 4-28. Manual execution of L4A processor from Web Site (with InSitu data)

The advanced parameters of the processor can be accessed ticking the “Show advanced parameters” box. The advanced parameters will be displayed as illustrated in Figure 4-29. Their definition can be found in Appendix F.9 All advanced parameters table.
Random seed:

The random seed used for training.

Window records:

The number of records used for the temporal features extraction.

Number of components:

The number of components used by dimensionality reduction.

Range radius:

The range radius (expressed in radian unit).

Erosion radius:

The radius used for erosion.

Alpha:

The parameter alpha used by the minimum distance function.

Segmentation parameters:

Spatial radius:

The spatial radius of the neighborhood used for segmentation.

Minimum size of a region:

Minimum size of a region (in pixel unit) in segmentation.

Classifier parameters:

Training trees:

The number of trees used for training.

Max depth:

Maximum depth for the trees used for Random Forest classifier.

Minimum number of samples:

Minimum number of samples in each node used by the classifier.

The minimum number of pixels:

The minimum number of pixels in an area where an equal number of crop and non-crop samples the crop decision is taken.

Submit

Figure 4-29. Manual execution of L4A processor from Web Site (with InSitu data) advanced parameters

When the “L4A w/o IN-SITU processor” editor is opened (Figure 4-30), the user has to select the site of interest and then, he has to define the following parameters:

- The set of L2A input products to be used: when the user selects his site of interest, the “Available input files” list is populated with the existing L8 and S2 L2A products available in the system. The user can select a subset of products and enable / disable one of the two sensors S2 and L8;
• Upload a custom reference file for the site containing in-situ data. The file should be a tif file. If the reference raster is not provided, the CCI Land Map reference map will be used by default (see section 4.5);
• Resolution (“pixsize” in Table 4-5);
• Ratio (“ratio” in Table 4-5).

![Figure 4-30. Manual execution of L4A processor from Web Site (without InSitu data)](image)

The advanced parameters of the processor can be accessed ticking the “Show advanced parameters” box. The advanced parameters will be displayed as illustrated in Figure 4-31. Their definition can be found in Appendix F.9 All advanced parameters table.
Training set samples:

The number of samples included in the training set.

Random seed:

The random seed used for training.

Window records:

The number of records used for the temporal features extraction.

Lambda:

The kernel parameter used in data smoothing.

Number of components:

The number of components used by the dimensionality reduction.

Range radius:

The range radius (expressed in radiometric unit).

Erosion radius:

The radius used for erosion.

Alpha:

The parameter alpha used by the multilaminar function.

Segmentation parameters:

Spatial radius:

The spatial radius of the neighborhood used for segmentation.

Minimum size of a region:

The minimum size of a region in pixels used in segmentation.

Classifier parameters:

Training trees:

The number of trees used for training.

Max depth:

The maximum depth for the trees used by random forest classifier.

Minimum number of samples:

The minimum number of samples in each node used by the classifier.

The minimum number of pixels:

The minimum number of pixels in an area where an equal number of crop and non-crop samples the classification is taken.

Figure 4-31. Manual execution of L4A processor from Web Site (without InSitu data) advanced parameters
4.4.6 Creating a crop type (L4B) product

An overview of the processor is given in the section 2.2.4 and a more detailed description of the algorithms can be found in [AD.12]. This product is formatted according to the Product Specification Document (PSD) [AD.5].

4.4.6.1 Processors parameters

When a new L2A product is available or a when user requests a crop type product, the L4B Crop Type Processor can be invoked with the parameters given in Table 4-6. As shown in this table, the L4A Crop Mask product must be obtained with the same parameters, and provided as input to the L4B Crop Type processor.

Table 4-6. L4B Processor’s parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>refp</td>
<td>the reference polygons (only for with in-situ data mode)</td>
<td>file</td>
</tr>
<tr>
<td>ratio</td>
<td>the ratio between the validation and training polygons</td>
<td>decimal</td>
</tr>
<tr>
<td>input</td>
<td>the list of products descriptors (XML files)</td>
<td>file list</td>
</tr>
<tr>
<td>trm</td>
<td>The temporal resampling mode</td>
<td>String (resample/gapfill – by default: gapfill)</td>
</tr>
<tr>
<td>classifier</td>
<td>the classifier used for training (either rf (random forest) or svm (support vector machine))</td>
<td>string</td>
</tr>
<tr>
<td>rseed</td>
<td>the random seed used for training</td>
<td>integer</td>
</tr>
<tr>
<td>mask</td>
<td>the crop mask</td>
<td>file</td>
</tr>
<tr>
<td>pixsize</td>
<td>the size, in meters, of a pixel</td>
<td>integer (by default: 10)</td>
</tr>
<tr>
<td>outdir</td>
<td>the output directory</td>
<td>string</td>
</tr>
<tr>
<td>buildfolder</td>
<td>Build folder</td>
<td>File</td>
</tr>
<tr>
<td>targetfolder</td>
<td>The folder where the target product is built</td>
<td>File</td>
</tr>
<tr>
<td>rfnbtrees</td>
<td>The number of trees used for training</td>
<td>Integer (by-default: 100)</td>
</tr>
<tr>
<td>rfmax</td>
<td>maximum depth of the trees used for RF classifier</td>
<td>Integer (by-default: 25)</td>
</tr>
<tr>
<td>rfmin</td>
<td>minimum number of samples in each node used by the classifier</td>
<td>integer (by-default: 5)</td>
</tr>
<tr>
<td>keepfiles</td>
<td>Keep all intermediate files</td>
<td>string (by-default: false)</td>
</tr>
<tr>
<td>siteid</td>
<td>The site id, represented as 2 digits (optional)</td>
<td>string</td>
</tr>
<tr>
<td>lut</td>
<td>Color LUT for previews (see /usr/share/sen2agri/crop-mask.lut)</td>
<td>String</td>
</tr>
<tr>
<td>outprops</td>
<td>Output properties file</td>
<td>String (optional)</td>
</tr>
<tr>
<td>strata</td>
<td>Shapefiles with polygons for the strata</td>
<td>Shape file (.shp)</td>
</tr>
<tr>
<td>stratum-filter</td>
<td>The list of strata to use in training and classification</td>
<td></td>
</tr>
</tbody>
</table>
### Parameter Description Type

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>tile-filter</td>
<td>The list of tiles to apply the classification to</td>
<td>List of tile ids</td>
</tr>
<tr>
<td>include-raw-mask</td>
<td>Include the unmasked crop type map even when a crop mask was used</td>
<td>Boolean (by-default: false)</td>
</tr>
<tr>
<td>maskprod</td>
<td>A crop mask product for the same tiles</td>
<td>File</td>
</tr>
<tr>
<td>red-edge</td>
<td>Include Sentinel-2 vegetation red edge bands</td>
<td>Boolean (by-default: True)</td>
</tr>
<tr>
<td>sp</td>
<td>Per-sensor sampling rates</td>
<td>String list (by default: Sentinel = 10; SPOT = 5; Landsat = 16)</td>
</tr>
<tr>
<td>mode</td>
<td>The execution mode</td>
<td>String ['prepare-site', 'train', 'classify', 'postprocess-tiles', 'compute-quality-flags', 'validate']</td>
</tr>
<tr>
<td>max-parallelism</td>
<td>Number of tiles to process in parallel</td>
<td>Integer</td>
</tr>
<tr>
<td>tile-threads-hint</td>
<td>Number of threads to use for a single tile, except for the segmentation step</td>
<td>Integer (by-default: 2)</td>
</tr>
</tbody>
</table>

#### 4.4.6.2 Manual execution from terminal

The L4B Crop Type Processor can be executed from a terminal, as for instance:

```
./CropTypeFused.py
   -refp /mnt/archive/insitu/ukraine/UA_ALLC_LC_SM_201606_ALL.shp
   -rseed 0
   -outdir /mnt/archive/14b/
   -strata /mnt/archive/strata/ukraine/UKr_Strata_All.shp
   -input
   /mnt/archive/maccs_def/ukraine/l2a/S2A_OPER_PRD_MSIL2A_PDMC_20160407T093031_20160407T093031.SAFE/S2A_OPER_SSC_L2VALD_34UE_20160407.HDR
   /mnt/archive/maccs_def/ukraine/l2a/S2A_OPER_PRD_MSIL2A_PDMC_20160417T093031_20160417T093031.SAFE/S2A_OPER_SSC_L2VALD_34UB_20160417.HDR
```

© UCL-Geomatics 2019

This document is the property of the Sen2-Agri partnership, no part of it shall be reproduced or transmitted without the express prior written authorisation of UCL-Geomatics (Belgium).
4.4.6.3 Manual execution from SNAP

The L4B Crop Type Processor can be launched from SNAP, after setting the parameters, by following the steps (Figure 4-32):

- Open SNAP
- Go to menu Tools -> External Tools
- Select CropTypeProcessor
- In the execution dialog, go to tab Processing Parameters*
- Change the parameters as desired and click the button Run
- A progress bar will appear, and when it finishes, the produced product can be found in the corresponding location

* Definition of all parameters is given in Appendix F.9 All advanced parameters table
4.4.6.4 Manual execution from Web Interface

The L4B CropType Processor can be also executed manually from the web interface by accessing the “Custom Jobs” tab in the Web Interface and by selecting the “L4B processor” tab.

This editor allows the user running the L4B processor (i) by using only a subset of the available L2A input products and/or (ii) by running it with other parameters than the default ones defined in the automated processing.

When the editor is opened (Figure 4-33), the user has to select the site of interest and then, he has to define the following parameters:

- The set of L2A input products to be used: when the user selects his site of interest, the “Available input files” list is populated with the existing L8 and S2 L2A products available in the system. The user can select a subset of products and enable / disable one of the two sensors S2 and L8;
- Select a crop mask from the list of “Crop masks”
- Upload a custom shapefile containing in-situ data. The shapefile should be a zip archive containing all shapefile components;
- Resolution (“pixsize” in Table 4-6);
- Ratio (“ratio” in Table 4-6).
Figure 4-33. Manual execution of CropType processor from Web Site

The advanced parameters of the processor can be accessed ticking the “Show advanced parameters” box. The advanced parameters will be displayed as illustrated in Figure 4-34. Their definition can be found in Appendix F.9 All advanced parameters table.
4.5 In-situ data information

L4A Crop Mask and L4B Crop Type processors make use of in-situ data. In order to generate the crop mask, the processors makes use of data that distinguish between crop and non-crop classes while the L4B processor needs to have information about crop type.

The in-situ data should be gathered in a single shapefile, projected in the WGS 84/UTM grid system, containing both cropland and no-cropland information and in a corresponding XML file containing the metadata. Both files have the same name:

`CountryID_SiteID_DataContentID_CollectingMethodID_YearEndofSeason`
For instance, the shapefile named **MG_ANTS_LC_FO_201612.shp** corresponds to:

- **MG** for Madagascar (country ID)
- **ANTS** for Antsirabe (site ID)
- **LC** for Land Cover (data content ID)
- **FO** for Field Observations (collecting method ID)
  - Other options are: PI for Photo Interpretation, II for Institutional Information and SM for Several Methods (note that if several methods were used, these must be explained in the metadata file);
- **201612** (Year and Month of the data collection)

The format of the attribute table is presented in Table 4-7.

**Table 4-7. Format of the attribute table of in-situ dataset for cropland products**

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
<th>Var. type</th>
<th>Var. length</th>
<th>Var. value/comments</th>
<th>Null accepted</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Feature ID</td>
<td>Numeric</td>
<td>6</td>
<td>000000-999999</td>
<td>N</td>
</tr>
<tr>
<td>CROP</td>
<td>Binary class</td>
<td>Numeric</td>
<td>1</td>
<td>0 if non-cropland 1 if cropland</td>
<td>N</td>
</tr>
<tr>
<td>LC</td>
<td>Land cover</td>
<td>Text</td>
<td>70</td>
<td>Name of the crop type or the non-crop type class</td>
<td>N</td>
</tr>
<tr>
<td>CODE</td>
<td>Code of the class</td>
<td>Numeric (Long integer)</td>
<td>8</td>
<td>Cfr JECAM Guidelines [RD.14RD.14]</td>
<td>N</td>
</tr>
<tr>
<td>IRRIGATION</td>
<td>Information about irrigation of the field</td>
<td>Numeric</td>
<td>1</td>
<td>Only if known: 0 = not concerned or information not available 1 = irrigated 2 = not irrigated</td>
<td>Y</td>
</tr>
</tbody>
</table>

The metadata file of the in-situ dataset is an XML file containing the following information:

- **General**:
  - Information contained in the file
  - Start of data collection period
  - End of data collection period
  - Name of the site
  - Country of the site
  - Type of geometry
  - Number of features contained in the file
  - Spatial reference information
  - Date the shapefile was last modified
While in-situ data are mandatory to generate a crop type map, the L4A processor has the possibility to run in a “without in-situ mode”. In this case, a reference land cover map is used instead of field data. By-default, the Sen2-Agri uses the ESA CCI Land Cover map*, considering as “crop” the CCI classes 11 (herbaceous rainfed cropland) and 20 (irrigated cropland). The user has the possibility to provide his own map. In this version of the system, the provided map must be in the same format than the ESA CCI LC one, i.e. use labels of 11 or 20 to indicate cropland. In a future version, the reference map will be a binary image. For information, the ESA CCI LC legend is given in the Appendix G – ESA CCI LC legend.

The default folder where the Sen2Agri system expects reference data (“shp” or “tif”, the “shp” having the highest priority) is the folder “/mnt/archive/insitu{site}”. This is the by-default configuration entry; the system will then replace the {site} placeholder with the site's short name. The short name is obtained from the site's name given by the user, by taking the lower case version and replacing spaces with underscores. For example, if the site name is “South Africa Cape Town”, the short name will be south_africa_cape_town and the shapefile path will be /mnt/archive/insitu/south_africa_cape_town. This folder can be changed at any time in the system configuration, using the Processing System Configuration utility and providing a new value in the key “CropMask folder where in-situ data are checked” for the Crop Mask processor. Yet, it is advisable to keep the “{site}” if he is working with several sites.

The in-situ data can be uploaded for each site from the web interface of the application, by editing the site in the “Sites” tab of the GUI (see 4.2.2 and 4.2.11).

Optionally, for advanced users, the in-situ files can be manually copied directly in the specified directory.

4.6 Single-tile vs multi-tile processing

As briefly mentioned in sections 2.2.3 and 2.2.4, the Cropland mask (L4A) and Crop type map (L4B) processors have the possibility to be run in single- or multi-tile modes. This option is to be configured in the processors parameters as explained in sections 4.4.5.1 and 4.4.6.1.

The single-tile mode of the processors assumes that each tile of the site to monitor is covered by samples (crop and non-crop classes for the L4A and crop types for the L4B). The classification algorithm, which relies on a random forest classifier, is run by tile: a random forest model is estimated for each tile and each tile is classified independently from the neighboring tiles.

Conversely, the multi-tile mode of the processors allows running the algorithm over multiple tiles using a single random forest model for all these tiles. This mode requires a stratification which divides the region to monitor in different regions. Each region is processed independently from the other but all tiles within the region are classified with the same random forest model. The processor will consider all samples within the region to train a “global” classifier and build a “global” model. This “global” model will then be applied to each tile. This multi-tile mode thus ensures a higher spatial consistency within the final product. It also enables to process tiles which are not covered by samples.

This stratification is to be defined by the user. It is recommended to define regions as homogeneous as possible. Different aspects can be taken into consideration, such as the agricultural practices, the meteorological and agro-ecological conditions, etc. Regions should make sense for the classification as they will be classified using the same model.

This stratification is to be provided by the user to the system. The default folder where the system expects reference data (“shp” or “tif”) is the folder “/mnt/archive/insitu/{site}/strata”. This is the same folder than for the in-situ data information (see section 4.5), with “strata” appended. This is the by-default configuration entry; the system will then replace the {site} placeholder with the site's short name. The short name is obtained from the site's name given by the user, by taking the lower case version and replacing spaces with underscores. For example, if the site name is "South Africa Cape Town", the short name will be south_africa_cape_town and the shapefile path will be /mnt/archive/insitu/south_africa_cape_town. The user can change the key in the configuration but it is advisable to keep the “{site}” if he is working with several sites (see Appendix F – Advanced system configuration items). This is the same directory that for the in-situ data. The stratification needs to be uploaded as an uncompressed shapefile, i.e. not archived.

The strata data can be uploaded for each site from the web interface of the application, by editing the site in the “Sites” tab of the GUI (see 4.2.2 and 4.2.11).

Optionally, for advanced users, the strata files can be manually copied directly in the specified directory.
5. Visualization tool

5.1 Visualization tool deployment

5.1.1 Prerequisites for automatic usage

The minimum hardware requirements depend on the Sen2-Agri products that have to be displayed. The bigger the products are, the highest the performances have to be:

- **Minimum hardware configuration:**
  - 4 CPUs;
  - 16 GB RAM;
  - 50Gb free disk space;

- **Recommended hardware configuration:**
  - 8 CPUs;
  - 32Gb RAM;
  - 50Gb free disk space;

The software configuration is as follows:

- Docker-CE compatible OS : CentOS 7, Ubuntu 16.04 or 18.04 for example;
- Docker-CE;
- Own a user with administrator rights.

5.1.2 Installation and configuration

Once the docker-ce is installed:

- Import the Sen2Agri2Geoserver image:

```bash
sudo docker import <archive_path>/sen2agri2geoserver.tar sen2agri2geoserver:latest
```

where `<archive_path>` is the path to the 'sen2agri2geoserver.tar' archive

- Starts a Sen2Agri2Geoserver instance:

```bash
sudo docker run --name "sen2agri2geoserver" -tid -p 8080:8080 -v <base_folder>:/s2agri_products sen2agri2geoserver /Sen2Agri2Geoserver/script/start.sh
```

where:

- `docker run` starts a new instance of the "sen2agri2geoserver:latest" image
- `--name "sen2agri2geoserver"` names the container
- `-tid` starts in detached interactive terminal
- `-p 8080:8080` maps the localhost 8080 port to the 8080 port of the container
- `-v <base_folder>:/s2agri_products` mounts the `<base_folder>` folder as the "/s2agri_products" folder in the container. `<base_folder>` is a folder containing the "sites/processor/s2agri_products" sub-folders.
- `sen2agri2geoserver` name of the image to use
You can check the installation with the following command:

```
sudo docker ps -a
```

## should return a "Up..." status:

<table>
<thead>
<tr>
<th>IMAGE</th>
<th>COMMAND</th>
<th>CREATED</th>
<th>STATUS</th>
<th>NAMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>sen2agri2geoserver</td>
<td>&quot;/Sen2Agri2Geoserver...&quot;</td>
<td>About a minute ago</td>
<td>Up About a minute</td>
<td>0.0.0.0:8080-8080/tcp sen2agri2geoserverh</td>
</tr>
</tbody>
</table>

After a few minutes (the geoserver tomcat server takes up to 3 minutes to start), open your favourite web browser on the docker host (i.e. the machine hosting the docker daemon) with the following credentials:

- User: admin
- Password: geoserver

```
firefox http://localhost:8080/geoserver
```

A large part of the configuration can be done through the web interface. To install the additional style file provided in the archive, you can run the following commands:

```
## Connects to the 'sen2agri2geoserver' instance:
sudo docker exec -ti sen2agri2geoserver bash
STYLE_FILES=`ls /Sen2Agri2Geoserver/styles/raster_*.zip`
for STYLE_FILE in ${STYLE_FILES};
do
curl -v -u admin:geoserver -XPOST -H "Content-type: application/zip" --data-binary @${STYLE_FILE} http://localhost:8080/geoserver/rest/styles
done
```

You can customize these styles directly in the web interface.

If you want to authorize the download of the data from QGIS for example, you need to enable the DirectDownload option from the CSW panel in the geoserver interface. You can also customize the information about the Service Metadata in the geoserver interface.

### 5.2 How to use the visualization tool?

#### 5.2.1 GeoServer Interface

The documentation of the GeoServer is available on [https://docs.geoserver.org/](https://docs.geoserver.org/).

This section presents the way the Sen2-Agri user can visualize his product.
After a few minutes, the Sen2-Agri visualization tool will begin to collect and ingest existing Sen2-Agri products from the Sen2-Agri system instance. These products will be available in the Layers page (Figure 5-1 - accessible after login).

The products can be filtered using the search tool bar. Be careful to not remove or add layer manually if you are not familiar with GeoServer.

In order to visualize the products in a web browser:

- go to the Layer Preview page (accessible without login);
- click on the OpenLayers button to visualize data in a new tab;
Figure 5-3. Openlayer view: in red the zoom buttons

- in this new tab, you can navigate and zoom into the layer and retrieve information about position and scale.
5.2.2 Integration into QGIS

The tool can be integrated into QGIS either by adding a set of layers from a WMS server or by directing connecting the GeoServer catalogue. The integration from a WMS server is for users mainly interested in products visualization. The procedure is described here below:

- on the browser panel, click on the WMS line to add a new connection;
in the widget, insert the name of the connection (Sen2-Agri for example) and add the URL (http://<IP-address>:<PortNumber>/geoserver/s2agri/wms?). IP address is the IP address of the geoserver docker and PortNumber is the port number of the container (by default 8080).

click on “Ok” to get the list of layers available in the Browser Panel. If you double-click on one layer, you will add it into the main view;
it is also possible to filter the layer with the search toolbar of the browser panel.

In order to perform advanced research and download data, the geoserver catalog can also be connected to QGIS as follows:

- in the QGIS interface, open the Web -> MetaSearch -> MetaSearch plugin;

- add a new service by clicking to the New button and set a Name (Sen2-Agri for example) and the URL (http://<IP-address>:{PortNumber}/geoserver/s2agri/csw). IP address is the IP address of the geoserver docker and PortNumber is the port number of the container (by default 8080).
• if everything is correct, you should have the “Service Metadata” available in the panel below the New button.

• go the Search Tab and perform a Search with a Global Coverage to have all the layers or use a custom box or use the current map extent of QGIS.
• when you want to have more information about a result you can click on it, a new window is open and you can click on the access link to download the data.

More information about QGIS usage is available into the QGIS User Guide [RD.15].
6. Maintenance of the system

6.1 Uninstall procedure

The following steps should be run as root user in order to uninstall the Sen2-Agri system from the computer.

- Stop and disable the running applications:

```bash
# systemctl stop sen2agri-orchestrator sen2agri-http-listener sen2agri-services
# systemctl stop sen2agri-monitor-agent
# systemctl disable sen2agri-orchestrator sen2agri-http-listener sen2agri-monitor-agent
```

- Uninstall the application packages:

```bash
# yum remove sen2agri-processors sen2agri-website sen2agri-downloading-demmacs
# rm -rF /usr/share/sen2agri/sen2agri-services
```

- Stop and disable SLURM and MUNGE:

```bash
# systemctl stop Slurm Slurmctld Slurmd munge
# systemctl disable Slurm Slurmctld Slurmd munge
```

- Uninstall SLURM and MUNGE:

```bash
# yum remove Slurm Slurm-munge Slurm-Slurmd-direct Slurm-sql Slurm-torque Slurm-sjstat Slurm-plugins Slurm-Slurmd Slurm-devel Slurm-sjobexit Slurm-perlapi Slurm-pam_Slurm munge-devel munge-libs munge
```

- Stop and disable MariaDB:

```bash
# systemctl stop mariadb
# systemctl disable mariadb
```

- Uninstall MariaDB:

```bash
# yum remove mariadb-server mariadb-devel mariadb
```

- Remove the PostgreSQL database:

```bash
# su -l postgres -c 'psql -c "drop database sen2agri;"'
```

- Stop and disable PostgreSQL:

```bash
# systemctl stop postgresql-9.4
# systemctl disable postgresql-9.4
```
• Uninstall PostgreSQL:

```
# yum remove postgresql94-server postgresql94-contrib postgresql94
```

• Uninstall OTB and GDAL 2.0:

```
# yum remove otb gdal-local
```

• Stop and disable Apache:

```
# systemctl stop http
# systemctl disable http
```

• Uninstall Apache and PHP:

```
# yum remove httpd php
```

• Uninstall dependencies:

```
# yum remove boost tinyxml tinyxml-devel qt qt5-qtbase qt-x11 fftw gdal geos libgeotiff libsvm muParser opencv openjpeg2 openjpeg2-tools proj proj-epsg swig qt5-qtbase-postgresql gsl cifs-utils
```

• Uninstall MACCS/MAJA:

Follow the guidelines available in the MACCS SUM [RD.3] and/or in the MAJA installation instructions [AD.13].

**NOTE:** Other applications on the system might depend on some of the packages above. Before continuing, please make sure to double-check the list of packages to be removed.

### 6.2 Update procedure

Upgrading versions of the Sen2-Agrí system is possible. The following upgrades are possible:

- From version 1.4 to 1.5;
- From version 1.5 to version 1.6;
- From version 1.6 to version 1.7;
- From version 1.7 to version 1.8;
- From version 1.8 to version 2.0;
- From version 2.0 to version 2.0.1.
- From version 2.0.1 to version 2.0.2.
- From version 2.0.2 to version 2.0.3.

All upgrades can be performed using the same script `Sen2AgriDistribution/install_script/update.sh`:

```
# cd <SEN2AGRI_ROOT>/Sen2AgriDistribution/install_script/
```
Between versions 1.4 and 1.5, a new version of MACCS was also included. This means that a transition from version 1.4 to 1.5 or to 1.6 will need also to perform the upgrade of MACCS. Also, on the CNES site it should be checked if a newer version of MACCS/MAJA is available.

Starting with version 2.0 the MAJA L2A processor is supported by the system and is recommended to be used instead of MACCS. The upgrade to MAJA assumes both (as described in section 3.3.2):
- MAJA package installation
- The GIPP copying

**IMPORTANT NOTICE:**

MACCS and MAJA formats are not compatible so products produced by one processor cannot be used by the other processor in the NOMINAL mode. This is why, the change to MAJA should not be made during production of a site but only once the L2A processing finished. Also, no season should be added after MAJA installation to a site created with MACCS.

This operation is not performed automatically by the system and should be performed manually by the user according to the instructions described in the MACCS/MAJA documentation.

Nevertheless, after a reinstall of MACCS/MAJA, the following operations should be performed in the Sen2Agri system in order to use the new MACCS/MAJA version. For example, in the case of MAJA 3.2.2:

```
$ rm -rf /mnt/archive/gipp/*
$ cp <SEN2AGRI_ROOT>/Sen2AgriDistribution/gipp/* /mnt/archive/gipp
$ sudo -u postgres psql sen2agri
sen2agri=# update config set value = '/opt/maja/3.2.2/bin/maja' where key = 'demmaccs.maccs-launcher';
```

During version 1.8, the old sentinel-downloader and landsat-downloader scripts were removed and their functionality was taken by the new introduced “sen2agri-services” service.

### 6.3 How to submit an issue

To submit an issue, you need an account on the Sentinel-2 for Agriculture portal. The issue tracker is accessible to logged-in users. Also, additional information can be found in the Sen2Agri forum accessible at forum.esa-sen2agri.org.

### 6.4 FAQ

A complete list of Frequently Asked questions can be found on the Sentinel-2 for Agriculture Portal, in the “Resources” section, accessible using the following link: http://www.esa-sen2agri.org/resources/faq/.
Appendix A - Additional tools available in the system

Appendix A.1 Sen2-Agri Archiver parameters

The Sen2-Agri Archiver is in charge with archiving processed products.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Parameter Value</th>
<th>Description</th>
<th>Configuration Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive path</td>
<td>/mnt/archive/{site}/{processor}/{product_type}/</td>
<td>The path on the disk where the product should be moved</td>
<td>Sen2Agri Configurator</td>
</tr>
<tr>
<td>L2A Product Max Age</td>
<td>1</td>
<td>The max age, in days, of a L2A product before being archived</td>
<td>Sen2Agri Configurator</td>
</tr>
<tr>
<td>L3A Product Max Age</td>
<td>1</td>
<td>The max age, in days, of a L3A product before being archived</td>
<td>Sen2Agri Configurator</td>
</tr>
<tr>
<td>L3B Product Max Age</td>
<td>1</td>
<td>The max age, in days, of a L3B product before being archived</td>
<td>Sen2Agri Configurator</td>
</tr>
<tr>
<td>L4A Product Max Age</td>
<td>1</td>
<td>The max age, in days, of a L4A product before being archived</td>
<td>Sen2Agri Configurator</td>
</tr>
<tr>
<td>L4B Product Max Age</td>
<td>1</td>
<td>The max age, in days, of a L4B product before being archived</td>
<td>Sen2Agri Configurator</td>
</tr>
</tbody>
</table>
Appendix A.2 Sen2-Agri HTTP Listener parameters

The Sen2-Agri HTTP Listener is a component of the Sen2-Agri system that allows interaction between the website of the system with the underlying components (database, orchestrator etc.)

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
<th>Configuration Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dashboard Listen</td>
<td>The port use to communicate with the execution dashboard</td>
<td>Sen2Agri Configurator</td>
</tr>
<tr>
<td>Port</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Root Path</td>
<td>The path where the files for the dashboard can be found</td>
<td>Sen2Agri Configurator</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix A.3 Sen2-Agri Executor parameters

The Sen2-Agri Executor is in charge with the execution of the processors applications, using SLURM for executing these applications.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
<th>Configuration Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executor IP Address</td>
<td>The address where the executor is found</td>
<td>Sen2Agri Configurator</td>
</tr>
<tr>
<td>Executor Port</td>
<td>The port where the executor is found</td>
<td>Sen2Agri Configurator</td>
</tr>
</tbody>
</table>
Appendix A.4 Sen2-Agri Monitor Agent parameters

The Sen2-Agri Monitor Agent is in charge with the monitoring the resources of the system and the status of the execution of the various applications from the processors.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Parameter Value</th>
<th>Description</th>
<th>Configuration Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk Path To Monitor For Space</td>
<td>/mnt/scratch</td>
<td>The path on the disk to be monitored for space</td>
<td>Sen2Agri Configurator</td>
</tr>
<tr>
<td>Measurement Interval (s)</td>
<td>3600</td>
<td>The interval at which the disk is monitored</td>
<td>Sen2Agri Configurator</td>
</tr>
<tr>
<td>Monitor/ServiceUrl</td>
<td>http://[IpAdress]:8080/statistics</td>
<td>the HTTP Listener endpoint address</td>
<td>Config file /etc/sen2agri/sen2agri-monitor-agent.conf</td>
</tr>
</tbody>
</table>
Appendix A.5 Sen2-Agri Persistence Manager parameters

The Sen2-Agri Persistence Manager offers an API for other components to access the database (configuration but also the execution tables).

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Parameter Value</th>
<th>Description</th>
<th>Configuration Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database/HostName</td>
<td>sen2agri-dev</td>
<td>The name of the server where the database is hosted</td>
<td>Config /etc/sen2agri/sen2agri-persistence.conf file</td>
</tr>
<tr>
<td>Database/DatabaseName</td>
<td>sen2agri</td>
<td>The name of the Sen2Agri database</td>
<td>Config /etc/sen2agri/sen2agri-persistence.conf file</td>
</tr>
<tr>
<td>Database/UserName</td>
<td>admin</td>
<td>The username used to connect to the database</td>
<td>Config /etc/sen2agri/sen2agri-persistence.conf file</td>
</tr>
<tr>
<td>Database/Password</td>
<td></td>
<td>The password used to connect to the database</td>
<td>Config /etc/sen2agri/sen2agri-persistence.conf file</td>
</tr>
</tbody>
</table>
## Appendix A.6 Sen2-Agri Services download parameters

The Sen2-Agri Services Download module is a component that is managing the downloading of L1C products (Sentinel-2 and Landsat8) from SciHub and USGS respectively.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Parameter Value</th>
<th>Description</th>
<th>Configuration Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write Directory for S2</td>
<td>/mnt/dwn_def/s2</td>
<td>The directory where the downloaded sentinel products will be saved</td>
<td>Sen2Agri Configurator</td>
</tr>
<tr>
<td>Maximum cloud coverage</td>
<td>80</td>
<td>The name of the Sen2-Agri database</td>
<td>Sen2Agri Configurator</td>
</tr>
<tr>
<td>Write Directory for L8</td>
<td>/mnt/dwn_def/l8</td>
<td>The directory where the downloaded landsat products will be saved</td>
<td>Sen2Agri Configurator</td>
</tr>
<tr>
<td>site</td>
<td>Ex:</td>
<td>The sites defined in the database, from which the downloaders are searching online products; the polygon of each site is projected in WGS84.</td>
<td>Sen2-Agri database, site table; this table is prefilled and the manual editing in the database is not advised.</td>
</tr>
<tr>
<td>shape_tiles_s2</td>
<td>Shapes defined for the acquisition plan</td>
<td>The shapes defined for the acquisition plan will be intersected with the polygon of each site, and the resulting shapes will be interrogated by the downloader</td>
<td>Sen2-Agri database, shape_tiles_s2 table; this table is prefilled and the manual editing in the database is not advised.</td>
</tr>
<tr>
<td>shape_tiles_l8</td>
<td>Shapes defined for the acquisition plan of Landsat 8</td>
<td>The shapes defined for the acquisition plan will be intersected with the polygon of each site, and the resulting shapes will be interrogated by the landsat downloader</td>
<td>Sen2-Agri database, shape_tiles_l8 table; this table is prefilled and the manual editing in the database is not advised.</td>
</tr>
</tbody>
</table>

© UCL-Geomatics 2019

This document is the property of the Sen2-Agri partnership, no part of it shall be reproduced or transmitted without the express prior written authorisation of UCL-Geomatics (Belgium).
The sen2agri-services service is able to query for the list of products available from one source and to download the products from another source. For example, the list of available products for a season can be queried from SciHub but the download of the products to be performed from Amazon or from a local store. The “datasources” table contains the data sources used for querying and for downloading L1C products. The structure of the table is the following:

<table>
<thead>
<tr>
<th>Column name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>satellite_id</td>
<td>The satellite ID for which the datasource is configured</td>
</tr>
<tr>
<td>name</td>
<td>The name of the datasource</td>
</tr>
<tr>
<td>scope</td>
<td>The scope of the datasource and may have the following values:</td>
</tr>
<tr>
<td></td>
<td>1 = query, 2 = download, 3 = query &amp; download</td>
</tr>
<tr>
<td>fetch_mode</td>
<td>The product extraction mode:</td>
</tr>
<tr>
<td></td>
<td>1 - Product will be downloaded from the remote site and the corresponding</td>
</tr>
<tr>
<td></td>
<td>local product, if exists, it will be overwritten</td>
</tr>
<tr>
<td></td>
<td>2 - Product will be downloaded from the remote site and, if a corresponding</td>
</tr>
<tr>
<td></td>
<td>local product exists, the download will be resumed from the current</td>
</tr>
<tr>
<td></td>
<td>length of the local product</td>
</tr>
<tr>
<td></td>
<td>3 - The product will be copied from a local (or shared) folder into the</td>
</tr>
<tr>
<td></td>
<td>output folder. No remote download will be performed.</td>
</tr>
<tr>
<td></td>
<td>4 - Only a symlink to the product file system location, into the output</td>
</tr>
<tr>
<td></td>
<td>folder, will be created. No remote download will be performed.</td>
</tr>
<tr>
<td></td>
<td>5 - The path of the product from the local repository is directly added</td>
</tr>
<tr>
<td></td>
<td>into the database (just a check is performed). No remote download will be</td>
</tr>
<tr>
<td></td>
<td>performed.</td>
</tr>
<tr>
<td>username</td>
<td>The user used for the remote connexion</td>
</tr>
<tr>
<td>password</td>
<td>The user used for the remote connexion</td>
</tr>
<tr>
<td>download_path</td>
<td>The path were the products will be downloaded (should not be changed)</td>
</tr>
<tr>
<td>specific_params</td>
<td>Specific parameters for the datasource (should not be changed)</td>
</tr>
<tr>
<td>maximum_connexions</td>
<td>Maximum connections to be used at a moment to the datasource</td>
</tr>
<tr>
<td>local_root</td>
<td>The local root repository in case the products are already downloaded</td>
</tr>
<tr>
<td></td>
<td>in a local store.</td>
</tr>
<tr>
<td>enabled</td>
<td>Specifies if the datasource is enabled or not</td>
</tr>
</tbody>
</table>

Some of the parameters can also be edited in the `/usr/share/sen2agri/sen2agri-services/config/sen2agri-services.properties.

In the case of using a local repository that does not have the structure in the format `.../2018/01/08`, the following keys can be added/updated in the configuration file:
In this way, repositories having the folder structure like “.../2018/1/8” will be considered.
Other keys that can be configured via the services.properties are given below.

**Parameters that should not be modified unless administration is performed:**

```java
server.port = 8081
```

- This is the port on which the server listens for calls from the interface. Should not be changed unless port is already taken by another application. In this case, the port should be also updated in the file /var/www/html/ConfigParams.php

```java
spring.datasource.url
```
- This key gives the connexion string to the database

```java
spring.datasource.username=admin
spring.datasource.password=sen2agri
```
- Credentials for the sen2agri database

**Parameters that can be optionally changed:**

```java
database.config.polling=15
```
- Interval for checking for changes in the database and for execution jobs (ex. queries for datasources to check for new products or retry products interval)

```java
network.connexions.timeout=30
```
- If some datasources are responding very slow, this parameter can be set to a higher value. Pay attention that this applies to all datasources and might impact the performances of the system. To be increased if it is absolutely necessary.

```java
SciHubDataSource.Sentinel2.auto.uncompress=true
```
- Setting this key to true will force unzipping the Sentinel2 L1C or L2A products right after the download. If using MAJA and L1C download mode, it can be set to false. If downloading Sen2Cor L2A products from SciHub, it must be set to true.

**NOTE:** Changing one of the keys in the “services.properties” file will require restarting the application using:

```
sudo systemctl restart sen2agri-services
```

Also, after changing datasources parameters in “services.properties” and restarting the application, should be checked the correct update of the “datasource” table using:

```
sudo -u postgres psql sen2agri “select * from datasource;”
```

If the parameters do not match, manual update of the datasource table might be necessary and restart the application.

© UCL-Geomatics 2019
This document is the property of the Sen2-Agri partnership, no part of it shall be reproduced or transmitted without the express prior written authorisation of UCL-Geomatics (Belgium).
NOTE: In the sen2agri-services, compared with the old downloaders, the philosophy of downloading changed as searches are not performed anymore each time from the beginning of the season(s) but instead it searches in an incremental manner, from the last downloaded/aborted product.

With the sen2agri-services, in order to try to retry the aborted products they should be set to status 3 (FAILED) and in the same time reset the no_of_retries column to 0 (there is a retry job in the application that will perform all the retries automatically).

Nevertheless, in order to force a query from the beginning of the season (in the case of missed products), a small plugin was added to the application in order to be able to perform this operation.

This option should be activated only in exceptional situations or if the user really wants this and is aware about the disadvantages (much more requests to datasources, more processing time consuming on machine, possible higher delays in product availability and incompatibility with the near-realtime mode). This is why, this option is disabled by default.

To use this option, add in the “config” table the key “scheduled.lookup.all_products.enabled = true”.

Once you notice that the products are up-to-date, you should remove (or set to false) the key from the database as it is not compatible with the near-realtime download mode.

To change the logging level of the sen2agri-services, the file /usr/share/sen2agri/sen2agri-services/config/application.properties can be edited for changing especially the following keys:

```
logging.level.ro.cs.tao=TRACE
logging.level.org.esa.sen2agri=TRACE
```

Where the possible values are for log levels are: ERROR, WARN, INFO, DEBUG, or TRACE.
Appendix A.7 Using Sentinel2 Download Services With Local L1C Store

In the case where are available Sentinel 2 L1C products that are already downloaded (like the IPT store), the Sentinel 2 Services can be configured in order to use directly this store instead of downloading the L1C products from SciHub or from Amazon. In this case, after the successful installation of the Sen2Agri system, the “datasource” table can be edited in order to specify another store, other from SciHub or Amazon.

In order to activate the local store mode, the line corresponding to the download datasource for the S2 satellite (having the column satellite_id = 1 and the scope=2) should be updated in the following manner:

- Set the “local_root” column to the root directory where the L1C products are stored (see "<PATH_TO_LOCAL_EO_DATA>" below);
- Set the “fetch_mode” column value to 4 (symlink mode) or 3 (copy products mode)

The "<PATH_TO_LOCAL_EO_DATA>" should be the root directory where the L1C products are stored in a hierarchical folder structure YYYY/MM/DD/<Products> like in the following example, where the first level is the year, the second level of folders is the month of the year, the third level of folders is the day of the month:

```
2015
2016
2017
01
02
03
01
02
03
04
S2A_MSIL1C_<xxxx1>.SAFE
S2A_MSIL1C_<xxxx1>.SAFE
```

These parameters can also be edited in the /usr/share/sen2agri/sen2agri-services/config/sen2agri-services.properties by setting, for example, for IPT repository and the option of creating symlinks:

```
AWSDataSource.Sentinel2.local_archive_path=/eodata/Sentinel-2/MSI/L1C
AWSDataSource.Sentinel2.fetch_mode=4
```

After changing configuration file or the database, the changes normally should be refreshed in maximum 1 minute otherwise the Sen2Agri Services service should be restarted using:

```
sudo systemctl restart sen2agri-services
```
Appendix B - Sen2-Agri database tables for manual configuration

The table below presents the main configuration tables that might present interest for the advanced users.

<table>
<thead>
<tr>
<th>Table name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site</td>
<td>The table contains all sites that are used in the system</td>
</tr>
<tr>
<td>Processor</td>
<td>The table contains all processors that are used in the system</td>
</tr>
<tr>
<td>Config_category</td>
<td>The table contains the configuration for the type of applications that can be configured. Examples: Archiver, Executor, L4A Crop Mask Processor, etc.</td>
</tr>
</tbody>
</table>
| Config_metadata   | The table contains the parameters that can be configured for each type of application contained in Config_category table. The following columns can be modified:  
                    - the friendly name for the parameter  
                    - the type of the parameter  
                    - is_advanced – specifies if the parameter can be configured by an user with admin role  
                    - config_category_id – the id from Config_category table |
| Config            | The table contains default values for the parameters of each configured parameters:  
                    - key – the name of the parameter  
                    - site_id – the id of the site, if the parameter is configured for a specific site  
                    - value – the default value of the parameter  
                    - last_updated – the time when the parameter was last updated                                                                |

Apart from these tables, there are also the execution tables that are used by the system but these tables not recommended to be edited by the users.

In order to access the database, the following command can be issued in order to access to the PostgreSQL console:

```
$ sudo su - postgres -c "psql -d sen2agri"
sen2agri=#
```

The prompt “sen2agri=#” can be used to issue SQL commands like the following that displays the content of the table “config”:

```
sen2agri=# select * from config;
```

Please note that all commands in this console should end by a semicolon.

Another option, if the Linux was installed with an X server, is to use the much easier to use application, pgadmin3. In order to install and run this application, the following commands can be executed:
$ sudo yum install --y pgadmin3
$ pgadmin3 &

In this application, a new connexion to the database server can be created using the parameters from the picture below (database password is “sen2agri”):
## Appendix C - System performance example

System performances on the Sen2-Agri Production Platform used as calculation basis (2 x Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz - 40 threads, 128 gb ram)

<table>
<thead>
<tr>
<th>Products</th>
<th>Tiles</th>
<th>Out Tiles</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download from Amazon</td>
<td>1</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>MACCS</td>
<td>1</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>Composite</td>
<td>17</td>
<td>201</td>
<td>61</td>
</tr>
<tr>
<td>LAI Mono date with model generation</td>
<td>17</td>
<td>201</td>
<td>201</td>
</tr>
<tr>
<td>LAI Mono date with model generation</td>
<td>4</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>LAI Mono date without model generation</td>
<td>4</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>LAI Reprocessing 3 Dates</td>
<td>4</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>LAI Fitted</td>
<td>4</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>CropMask</td>
<td>6</td>
<td>4 (S2)</td>
<td>2 (L8)</td>
</tr>
<tr>
<td>CropType</td>
<td>6</td>
<td>4 (S2)</td>
<td>2 (L8)</td>
</tr>
</tbody>
</table>

**Measured times on the Sen2-Agri production platform**

NOTE: During production, it was not identified a significant difference between the execution with strata or without strata.

<table>
<thead>
<tr>
<th>Products</th>
<th>Tiles /acq</th>
<th>Duration per acq (min)</th>
<th>Nb Acq</th>
<th>Duration for all months (min)</th>
<th>Duration for products in one month (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download from Amazon</td>
<td>80</td>
<td>104</td>
<td>3</td>
<td>312</td>
<td>5.2</td>
</tr>
<tr>
<td>MACCS</td>
<td>80</td>
<td>320</td>
<td>3</td>
<td>960</td>
<td>16</td>
</tr>
<tr>
<td>Composite</td>
<td>80</td>
<td>180</td>
<td>3</td>
<td>540</td>
<td>9</td>
</tr>
<tr>
<td>LAI Mono date with model generation</td>
<td>80</td>
<td>240</td>
<td>3</td>
<td>720</td>
<td>12</td>
</tr>
<tr>
<td>LAI Mono date without model generation</td>
<td>80</td>
<td>100</td>
<td>3</td>
<td>300</td>
<td>5</td>
</tr>
<tr>
<td>LAI Reprocessing 3 Dates</td>
<td>80</td>
<td>120</td>
<td>3</td>
<td>360</td>
<td>6</td>
</tr>
<tr>
<td>LAI Fitted *</td>
<td>80</td>
<td>160</td>
<td>3</td>
<td>480</td>
<td>8</td>
</tr>
<tr>
<td>* Not relevant for 3 acquisition dates and done at the end of the season. The time will be multiplied with the number of months in one season</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crop Mask</td>
<td>68</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>5 days</td>
</tr>
<tr>
<td>Crop Type</td>
<td>68</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3 days</td>
</tr>
</tbody>
</table>

**Extrapolation for national site (~80 tiles) with data for 1 month on the Sen2-Agri production platform**

© UCL-Geomatics 2019

This document is the property of the Sen2-Agri partnership, no part of it shall be reproduced or transmitted without the express prior written authorisation of UCL-Geomatics (Belgium).
<table>
<thead>
<tr>
<th>Processor</th>
<th>Duration (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download from Amazon</td>
<td>13</td>
</tr>
<tr>
<td>MACCS</td>
<td>40</td>
</tr>
<tr>
<td>Composite</td>
<td>22.5</td>
</tr>
<tr>
<td>LAI Mono date with model generation</td>
<td>30</td>
</tr>
<tr>
<td>LAI Mono date without model generation</td>
<td>12.5</td>
</tr>
<tr>
<td>LAI Reprocessing 3 Dates</td>
<td>15</td>
</tr>
<tr>
<td>LAI Fitted *</td>
<td>20</td>
</tr>
</tbody>
</table>

* Not relevant for 3 acquisition dates and done at the end of the season. The time will be multiplied with the number of months in one season

<table>
<thead>
<tr>
<th>Type</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crop Mask</td>
<td>8</td>
</tr>
<tr>
<td>Crop Type</td>
<td>5</td>
</tr>
</tbody>
</table>
Appendix D - Building from source

In order to build the system, the script Sen2AgriBuildAll.sh (from the git sources sen2agri/packaging) can be used. This script presents to the user the following menu:

*--------------------------------------------------*
** 1) Install BUILD PREREQUISITES
** 2) BUILD Sen2AgriPlatform : OTB and GDAL
** 3) BUILD Sen2AgriProcessors, Downloaders and Demmaccs
** 4) BUILD Sen2AgriApplication
** 5) BUILD Sen2AgriWebsite
** 6) BUILD Selective: Enter list of options (ex: platform, processors, website, app)
** 7) BUILD Sen2Agri All Components
*--------------------------------------------------*

Please enter a menu option and enter or enter to exit.

The user can choose one of the option in order to build the desired component(s).

In the next 2 paragraphs are presented the options to build the system. Both of these options can be also implemented using the above interface and by selecting the corresponding menus.

**Building the core components**

For building the core components (used for manual execution) run the steps 1, 2 and 3 from the general installation script Sen2AgriBuildAll.sh.

**Building the components for automated usage**

Run step 7 from the general installation script Sen2AgriBuildAll.sh.

**Preparing the distribution**

In order to create a Sen2-Agri system distribution the following steps should be followed:

- Execute Sen2AgriBuildAll.sh

Execute install_platform/sen2agriCreateDistribution.sh by providing the previously created folder with the Sen2-Agri system RPM files.
Appendix E – How to add new users

The users can be added in the database by using a script launched from a terminal as follows:

```
psql -U <db_user> -d sen2agri -c "SELECT sp_adduser(<login>, <email>, <password>,
<roleId = {1|2} DEFAULT 2>, <siteId DEFAULT NULL>);"
```

For example:

```
psql -U <db_user> -d sen2agri -c "SELECT sp_adduser('anotheradmin',
'admin@domain.com', 'aDmInPwD', 1);"
```

To change user password:

```
psql -U <db_user> -d sen2agri -c "SELECT * FROM public.sp_changePassword
(CAST(<userId> AS smallint), <old_password>, <new_password>);"
```

```
psql -U <db_user> -d sen2agri -c "SELECT * FROM public.sp_changePassword (CAST(7 as
smallint), 'abcabc', 'abcabb');"
```
Appendix F – Advanced system configuration items

Appendix F.1 Processing System Configuration Utility

The Sen2-Agrí Configurator is a Linux desktop application used to configure the system parameters. In order to be launched, the X server graphical interface should have been installed during the operating system installation.

If CentOS is already installed without a graphical interface, this can be installed with the following commands:

```
$ sudo yum groupinstall "X Window System" "Desktop" "Desktop Platform"
$ sudo yum install gdm
```

If you would like this to be the default runlevel, the file “/etc/inittab” can be edited and set the default level to 5 instead of 3:

```
$ sudo vi /etc/inittab
```

Change:

```
id:3:initdefault:
To:
id:5:initdefault:
```

The sen2agri configurator can be launched with the following command:

```
$ sen2agri-config &
```

The parameters that will appear in the Sen2-Agrí Configurator are managed through the Sen2-Agrí database and, depending on their type, they can be modified either by an admin or non-admin user. All parameters values that are modified are validated by the configurator (see Figure 0-1).
If the user tries to save when an invalid parameter value was filled in, an error message will inform the user about the error and the values are not saved.
Some of the parameters are site specific while other ones are system specific. For the site specific parameters, the user has the possibility to select the site for which he is configuring the parameter (Figure 0-3).

The modification of the value of a parameter can be done either by site (by selecting a site and then modifying the value) or for all sites (by selecting Global and then modifying the value). In order to modify the value of a parameter only for a site, the Customize button will be used for values that were not set before for this site. To reset the value for the current site back to the global value, the Reset button can be used. This is illustrated in Figure 0-4.

If a parameter was already defined for a specific site, changing the parameter for Global will not affect the already customized specific parameter.

A non-admin user will only be able to see the information in the Sen2-Agri interface but will not be able to modify it (Figure 0-5).
Appendix F.2 Sen2-Agrí Archiver

The Sen2-Agrí Archiver is a system service which interrogates the execution database for products that need to be archived. It will receive a list of products, each product having a current path (source directory) and a destination path (destination directory).

The Configurator contains a tab dedicated to configure the Archiver (Figure 0-6):

For details about the parameters of Sen2-Agrí Archiver, see Appendix F.9 All advanced parameters table.
Appendix F.3 Sen2-Agli HTTP Listener

The Sen2-Agli HTTP Listener is a system service used to offer the information displayed in the Execution Dashboard and to communicate with the Monitor Agent to receive the information that it will send to the dashboard.

The Configurator contains a tab dedicated to the HTTP Listener configuration, called Dashboard (Figure 0-7).

Figure 0-7. Sen2-Agli Configurator – Dashboard tab

For details about the parameters of Sen2-Agli HTTP Listener, see Appendix F.9 All advanced parameters table.

Appendix F.4 Sen2-Agli Executor

The Sen2-Agli Executor is a system service that uses SLURM to execute the processors.

The Configurator contains a tab dedicated to the Executor configuration (Figure 0-8).

Figure 0-8. Sen2-Agli Configurator – Executor tab

The parameters used by the Executor are described in Appendix A.3 Sen2-Agli Executor parameters.

Appendix F.5 Sen2-Agli Monitor Agent

The Monitor Agent service is used to monitor the Sen2-Agli processing nodes and sends metrics to the HTTP Listener in order to be displayed in the Execution Dashboard.

The Configurator contains a tab dedicated to configure the Monitor Agent (Figure 0-9).

Figure 0-9. Sen2-Agli Configurator – Monitoring Agent tab
Find the parameters used by the Monitor Agent Service in Appendix A.4 Sen2-Agri Monitor Agent parameters.

**Appendix F.6 Sen2-Agri Persistence Manager**

The Persistence Manager exchanges information between the database and the other system components (Executor, Monitor Agent, Dashboard and Archiver).

The parameters used by the Persistence Manager are described in Appendix A.5 Sen2-Agri Persistence Manager parameters.

**Appendix F.7 Sen2-Agri Downloader**

The S2 L1C and L8 L1T products are downloaded automatically by the downloader processor installed and configured by the installation script described in section 3.3. This processor is scheduled to run periodically (hourly), as configured by the installation script. Its job is to download the available products/tiles from the ESA SciHub for S2 products and from [https://ers.cr.usgs.gov/login](https://ers.cr.usgs.gov/login) (for L8 products) according to the parameters described in Appendix A.6 Sen2-Agri Services download parameters.

The configuration parameters (from config table) can have values for each site id, and also a global value, for all sites. In the case the key is missing for a certain site, the global value is considered. Even if the database parameters are described here, they can be changed also from the Sen2-Agri Configurator application, so the direct database manipulation is not advised.

Some manual configuration should be performed in order to set up a proxy configuration if needed, please see 0.

To configure the application by using the Sen2-Agri Configurator, just launch the configurator as shown in Figure 0-10.

![Sen2-Agri Configurator](image)

**Figure 0-10. Sen2-Agri Configurator – Downloader tab**

When the downloader processor is launched, it interrogates the database to intersect the polygons of the sites with the shapes defined in the sentinel tiles (shape_tile_s2) table (see Appendix B - Sen2-Agri database tables for manual configuration for database tables). For the resulting tiles, the downloader interrogates scihub site [https://scihub.copernicus.eu/apihub/search?q=](https://scihub.copernicus.eu/apihub/search?q=) and the landsat site [https://ers.cr.usgs.gov/login](https://ers.cr.usgs.gov/login) about the available products for the defined season period for each configured site. From the available products, the downloader checks which of them were already downloaded (this
information is located in table **downloader_history**). After the new products are downloaded, they are added in the **downloader_history** table, with the status **not processed**. Later on, the demmaccs processor (see Appendix F.8 Sen2-Agri Demmaccs_launcher) will handle all the **not processed** products found in the **downloader_history** table.

**Appendix F.8 Sen2-Agri Demmaccs_launcher**

The demmaccs_launcher processor is also installed and configured by the installation script described in section 3.3. It handles all the fully downloaded products from the **downloader_history** table with the status **not processed** and its main goal is to check the database, to create a list with the unprocessed L1C and L1T products, to launch with the proper arguments the demmaccs processor which in its turn will launch the dem and maccs processors (see section 4.4.1) and to write the results in the database in both **downloader_history** and **product** tables. The demmaccs processor is able to launch the dem and maccs processors in parallel to minimize the processing time (see section 4.4.1.1 – **processes-number-dem** and – **processes-number-maccs** input arguments).

**Appendix F.9 All advanced parameters table**

The following configurations are used in the automatic mode of the Sen2-Agri System. They can be changed using the Processing System Configuration Utility described in paragraph Appendix F.1 Processing System Configuration Utility.

In Table 0-1, the advanced processor parameters are prefixed with:

- Composite: “processor.l3a.*”. These parameters can be changed in the ”L3A Composite” of the System Configuration Utility;
- LAI: “processor.l3b.*”. These parameters can be changed in the ”L3B Vegetation Status” of the System Configuration Utility;
- Phenological NDVI: “processor.l3e.*”. These parameters can be changed in the ”L3E Pheno NDVI metrics” of the System Configuration Utility;
- CropMask: “processor.l4a.*”. These parameters can be changed in the ”L4A Crop Mask” of the System Configuration Utility;
- CropType: “processor.l4b.*”. These parameters can be changed in the ”L4B Crop Type” of the System Configuration Utility;

<table>
<thead>
<tr>
<th>Parameter ID</th>
<th>Parameter description</th>
</tr>
</thead>
<tbody>
<tr>
<td>archiver.archive_path</td>
<td>Archive Path</td>
</tr>
<tr>
<td>archiver.max_age.l2a</td>
<td>L2A Product Max Age (days)</td>
</tr>
<tr>
<td>archiver.max_age.l3a</td>
<td>L3A Product Max Age (days)</td>
</tr>
<tr>
<td>archiver.max_age.l3b</td>
<td>L3B Product Max Age (days)</td>
</tr>
<tr>
<td>archiver.max_age.l4a</td>
<td>L4A Product Max Age (days)</td>
</tr>
<tr>
<td>archiver.max_age.l4b</td>
<td>L4B Product Max Age (days)</td>
</tr>
<tr>
<td>demmaccs.gips-path</td>
<td>path where the gips files are to be found</td>
</tr>
<tr>
<td>demmaccs.maccs-launcher</td>
<td>launcher for MACCS/MAJA within the keeping unit</td>
</tr>
<tr>
<td>demmaccs.output-path</td>
<td>path for l2a products</td>
</tr>
<tr>
<td>demmaccs.srtm-path</td>
<td>path where the srtm files are to be found</td>
</tr>
</tbody>
</table>

© UCL-Geomatics 2019

This document is the property of the Sen2-Agri partnership, no part of it shall be reproduced or transmitted without the express prior written authorisation of UCL-Geomatics (Belgium).
<table>
<thead>
<tr>
<th>Ref</th>
<th>Sen2-Agri_SUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issue</td>
<td>Page</td>
</tr>
<tr>
<td>3.2</td>
<td>138</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>demmacs.swbd-path</td>
<td>path where the swbd files are to be found</td>
</tr>
<tr>
<td>demmacs.working-dir</td>
<td>working directory for demmacs</td>
</tr>
<tr>
<td>downloader.l8.max-retries</td>
<td>Maximum retries for downloading a product</td>
</tr>
<tr>
<td>downloader.l8.write-dir</td>
<td>Write directory for Landsat8</td>
</tr>
<tr>
<td>downloader.max-cloud-coverage</td>
<td>Maximum Cloud Coverage (%)</td>
</tr>
<tr>
<td>downloader.s2.max-retries</td>
<td>Maximum retries for downloading a product</td>
</tr>
<tr>
<td>downloader.s2.write-dir</td>
<td>Write directory for Sentinel2</td>
</tr>
<tr>
<td>downloader.start.offset</td>
<td>MAJA initialization interval (months before start of season)</td>
</tr>
<tr>
<td>executor.listen-ip</td>
<td>Executor IP Address</td>
</tr>
<tr>
<td>executor.listen-port</td>
<td>Executor Port</td>
</tr>
<tr>
<td>executor.module.path.bands-extractor</td>
<td>BandsExtractor Path</td>
</tr>
<tr>
<td>executor.module.path.color-mapping</td>
<td>Color Mapping Path</td>
</tr>
<tr>
<td>executor.module.path.composite-mask-handler</td>
<td>Composite Mask Handler path</td>
</tr>
<tr>
<td>executor.module.path.composite-preprocessing</td>
<td>Composite Mask Handler path</td>
</tr>
<tr>
<td>executor.module.path.composite-splitter</td>
<td>Composite Mask Handler path</td>
</tr>
<tr>
<td>executor.module.path.composite-total-weight</td>
<td>Composite Mask Handler path</td>
</tr>
<tr>
<td>executor.module.path.composite-update-synthesis</td>
<td>Composite Mask Handler path</td>
</tr>
<tr>
<td>executor.module.path.composite-weigh-aot</td>
<td>Composite Mask Handler path</td>
</tr>
<tr>
<td>executor.module.path.composite-weigh-on-clouds</td>
<td>Composite Mask Handler path</td>
</tr>
<tr>
<td>executor.module.path.compression</td>
<td>Compression Path</td>
</tr>
<tr>
<td>executor.module.path.compute-confusion-matrix</td>
<td>Compute Confusion Matrix Path</td>
</tr>
<tr>
<td>executor.module.path.compute-image-statistics</td>
<td>Compute image statistics</td>
</tr>
<tr>
<td>executor.module.path.compute-images-statistics</td>
<td>Compute Images Statistics Path</td>
</tr>
<tr>
<td>executor.module.path.concatenate-images</td>
<td>Concatenate Images Path</td>
</tr>
<tr>
<td>executor.module.path.crop-mask-features-with-insitu</td>
<td>CropMask features with insitu path</td>
</tr>
<tr>
<td>executor.module.path.data-smoothing</td>
<td>Data smoothing</td>
</tr>
<tr>
<td>executor.module.path.dummy-module</td>
<td>Dummy module path</td>
</tr>
<tr>
<td>executor.module.path.erosion</td>
<td>Erosion</td>
</tr>
<tr>
<td>executor.module.path.feature-extraction</td>
<td>Feature Extraction Path</td>
</tr>
<tr>
<td>executor.module.path.features-without-insitu</td>
<td>Features without insitu</td>
</tr>
<tr>
<td>executor.module.path.gdalwarp</td>
<td>gdalwarp Path</td>
</tr>
<tr>
<td>executor.module.path.image-classifier</td>
<td>Image Classifier Path</td>
</tr>
<tr>
<td>executor.module.path.image-compression</td>
<td>Image compression</td>
</tr>
<tr>
<td>executor.module.path.lai-bv-err-image-inversion</td>
<td>Applies the error regression model to the set of input reflectances</td>
</tr>
<tr>
<td>executor.module.path.lai-bv-image-inversion</td>
<td>Applies the regression model to the set of input reflectances</td>
</tr>
<tr>
<td>executor.module.path.lai-bv-input-variable-generation</td>
<td>BV input variables generator</td>
</tr>
<tr>
<td>executor.module.path.lai-err-time-series-builder</td>
<td>Builds a raster with all error images time series</td>
</tr>
<tr>
<td>executor.module.path.lai-fitted-reproc-splitter</td>
<td>Splits the fitted LAI as image for each date</td>
</tr>
<tr>
<td>executor.module.path.lai-fitted-reprocessing</td>
<td>Executes the fitted reprocessing</td>
</tr>
<tr>
<td>executor.module.path.lai-inverse-model-learning</td>
<td>Inverse model learning</td>
</tr>
<tr>
<td>executor.module.path.lai-local-window-reproc-splitter</td>
<td>Splits the local window LAI as image for each date</td>
</tr>
<tr>
<td>executor.module.path.lai-local-window-reprocessing</td>
<td>Executes the local window reprocessing</td>
</tr>
<tr>
<td>executor.module.path.lai-models-extractor</td>
<td>Determines the model to be used for the current execution</td>
</tr>
<tr>
<td>executor.module.path.lai-mono-date-mask-flags</td>
<td>Extracts the mask flags for the monodate LAI</td>
</tr>
<tr>
<td>executor.module.path.lai-msk-flags-time-series-builder</td>
<td>Builds a raster with all masks from the time series</td>
</tr>
<tr>
<td>executor.module.path.lai-ndvi-rvi-extractor</td>
<td>LAI NDVI and RVI features extractor</td>
</tr>
<tr>
<td>executor.module.path.lai-prosail-simulator</td>
<td>Prosail simulator</td>
</tr>
<tr>
<td>executor.module.path.lai-time-series-builder</td>
<td>Builds a raster with all time series</td>
</tr>
<tr>
<td>executor.module.path.lai-training-data-generator</td>
<td>Training data generator</td>
</tr>
<tr>
<td>executor.module.path.lsms-segmentation</td>
<td>LSMS segmentation</td>
</tr>
<tr>
<td>executor.module.path.lsms-small-regions-merging</td>
<td>LSMS small regions merging</td>
</tr>
<tr>
<td>executor.module.path.majority-voting</td>
<td>Majority voting</td>
</tr>
<tr>
<td>executor.module.path.mean-shift-smoothing</td>
<td>Mean shift smoothing</td>
</tr>
<tr>
<td>executor.module.path.ogr2ogr</td>
<td>ogr2ogr Path</td>
</tr>
<tr>
<td>executor.module.path.pheno-ndvi-metrics</td>
<td>Phenological NDVI metrics path</td>
</tr>
<tr>
<td>executor.module.path.pheno-ndvi-metrics-splitter</td>
<td>Phenological NDVI metrics splitter path</td>
</tr>
<tr>
<td>executor.module.path.principal-component-analysis</td>
<td>Principal component analysis</td>
</tr>
<tr>
<td>executor.module.path.product-formatter</td>
<td>Product Formatter Path</td>
</tr>
<tr>
<td>executor.module.path.random-selection</td>
<td>Random Selection Path</td>
</tr>
<tr>
<td>executor.module.path.sample-selection</td>
<td>Sample Selection Path</td>
</tr>
<tr>
<td>executor.module.path.statistic-features</td>
<td>Statistic Features Path</td>
</tr>
<tr>
<td>executor.module.path.temporal-features</td>
<td>Temporal Features Path</td>
</tr>
<tr>
<td>executor.module.path.temporal-resampling</td>
<td>Temporal Resampling Path</td>
</tr>
<tr>
<td>executor.module.path.train-images-classifier</td>
<td>Train Images Classifier Path</td>
</tr>
<tr>
<td>executor.module.path.train-images-classifier-new</td>
<td>TrainImagesClassifierNew</td>
</tr>
<tr>
<td>Ref</td>
<td>Sen2-Agri_SUM</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Issue</td>
<td>Page</td>
</tr>
<tr>
<td>3.2</td>
<td>140</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>executor.module.path.trimming</th>
<th>Trimming</th>
</tr>
</thead>
<tbody>
<tr>
<td>executor.module.path.xml-statistics</td>
<td>XML Statistics</td>
</tr>
<tr>
<td>executor.processor.l2a.name</td>
<td>L2A Processor Name</td>
</tr>
<tr>
<td>executor.processor.l2a.path</td>
<td>L2A Processor Path</td>
</tr>
<tr>
<td>executor.processor.l3a.keep_job_folders</td>
<td>Keep L3A temporary product files for the orchestrator jobs</td>
</tr>
<tr>
<td>executor.processor.l3a.name</td>
<td>L3A Processor Name</td>
</tr>
<tr>
<td>executor.processor.l3a.path</td>
<td>L3A Processor Path</td>
</tr>
<tr>
<td>executor.processor.l3a.Slurm_qos</td>
<td>Slurm QOS for composite processor</td>
</tr>
<tr>
<td>executor.processor.l3b.keep_job_folders</td>
<td>Keep L3B/C/D temporary product files for the orchestrator jobs</td>
</tr>
<tr>
<td>executor.processor.l3b.name</td>
<td>L3B Processor Name</td>
</tr>
<tr>
<td>executor.processor.l3b.path</td>
<td>L3B Processor Path</td>
</tr>
<tr>
<td>executor.processor.l3e.keep_job_folders</td>
<td>Keep L3E temporary product files for the orchestrator jobs</td>
</tr>
<tr>
<td>executor.processor.l3b_lai.Slurm_qos</td>
<td>Slurm QOS for LAI processor</td>
</tr>
<tr>
<td>executor.processor.l3e_pheno.Slurm_qos</td>
<td>Slurm QOS for Pheno NDVI processor</td>
</tr>
<tr>
<td>executor.processor.l4a.keep_job_folders</td>
<td>Keep L4A temporary product files for the orchestrator jobs</td>
</tr>
<tr>
<td>executor.processor.l4a.name</td>
<td>L4A Processor Name</td>
</tr>
<tr>
<td>executor.processor.l4a.path</td>
<td>L4A Processor Path</td>
</tr>
<tr>
<td>executor.processor.l4a.Slurm_qos</td>
<td>Slurm QOS for CropMask processor</td>
</tr>
<tr>
<td>executor.processor.l4b.keep_job_folders</td>
<td>Keep L4B temporary product files for the orchestrator jobs</td>
</tr>
<tr>
<td>executor.processor.l4b.name</td>
<td>L4B Processor Name</td>
</tr>
<tr>
<td>executor.processor.l4b.path</td>
<td>L4B Processor Path</td>
</tr>
<tr>
<td>executor.processor.l4b.Slurm_qos</td>
<td>Slurm QOS for CropType processor</td>
</tr>
<tr>
<td>executor.shapes_dir</td>
<td>Tiles shapes directory</td>
</tr>
<tr>
<td>executor.wrapper-path</td>
<td>Processor Wrapper Path</td>
</tr>
<tr>
<td>general.scratch-path</td>
<td>Default path for temporary files</td>
</tr>
<tr>
<td>general.scratch-path.l3a</td>
<td>Path for L3A temporary files</td>
</tr>
<tr>
<td>general.scratch-path.l3b_lai</td>
<td>Path for L3B/L3C/L3D temporary files</td>
</tr>
<tr>
<td>general.scratch-path.l3e_pheno</td>
<td>Path for L3E temporary files</td>
</tr>
<tr>
<td>general.scratch-path.l4a</td>
<td>Path for L4A temporary files</td>
</tr>
<tr>
<td>general.scratch-path.l4b</td>
<td>Path for L4B temporary files</td>
</tr>
<tr>
<td>http-listener.listen-port</td>
<td>Dashboard Listen Port</td>
</tr>
<tr>
<td>http-listener.root-path</td>
<td>Document Root Path</td>
</tr>
<tr>
<td>monitor-agent.disk-path</td>
<td>Disk Path To Monitor For Space</td>
</tr>
<tr>
<td>monitor-agent.scan-interval</td>
<td>Measurement Interval (s)</td>
</tr>
<tr>
<td>processor.l3a.bandsmapping</td>
<td>Bands mapping file for S2</td>
</tr>
<tr>
<td>Ref</td>
<td>Sen2-Agri_SUM</td>
</tr>
<tr>
<td>-----</td>
<td>---------------</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Issue</strong></td>
<td><strong>Page</strong></td>
</tr>
<tr>
<td>3.2</td>
<td>141</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>processor.l3a.generate_20m_s2_resolution</td>
<td>Specifies if composite for S2 20M resolution should be generated</td>
</tr>
<tr>
<td>processor.l3a.half_synthesis</td>
<td>Half synthesis interval in days</td>
</tr>
<tr>
<td>processor.l3a.lut_path</td>
<td>L3A LUT file path</td>
</tr>
<tr>
<td>processor.l3a.preproc.scatcoeffs_10m</td>
<td>Scattering coefficients file for S2 10 m</td>
</tr>
<tr>
<td>processor.l3a.preproc.scatcoeffs_20m</td>
<td>Scattering coefficients file for S2 20 m</td>
</tr>
<tr>
<td>processor.l3a.sched_wait_proc_inputs</td>
<td>L3A Composite scheduled jobs wait for products to become available</td>
</tr>
<tr>
<td>processor.l3a.synth_date_sched_offset</td>
<td>Difference in days between the scheduled and the synthesis date</td>
</tr>
<tr>
<td>processor.l3a.weight.aot.maxaot</td>
<td>Maximum value of the linear range for weights w.r.t. AOT</td>
</tr>
<tr>
<td>processor.l3a.weight.aot.maxweight</td>
<td>Maximum weight depending on AOT</td>
</tr>
<tr>
<td>processor.l3a.weight.aot.minweight</td>
<td>Minimum weight depending on AOT</td>
</tr>
<tr>
<td>processor.l3a.weight.cloud.coarseresolution</td>
<td>Coarse resolution for quicker convolution</td>
</tr>
<tr>
<td>processor.l3a.weight.cloud.sigmalarge</td>
<td>Standard deviation of gaussian filter for distance to large clouds</td>
</tr>
<tr>
<td>processor.l3a.weight.cloud.sigmasmall</td>
<td>Standard deviation of gaussian filter for distance to small clouds</td>
</tr>
<tr>
<td>processor.l3a.weight.total.bandsmapping</td>
<td>The bands mapping for the master and secondary product types</td>
</tr>
<tr>
<td>processor.l3a.weight.total.weightdatemin</td>
<td>Minimum weight at edge of the synthesis time window</td>
</tr>
<tr>
<td>processor.l3b.fitted</td>
<td>Specifies if fitting reprocessing (end of season) should be performed for LAI</td>
</tr>
<tr>
<td>processor.l3b.generate_models</td>
<td>Specifies if models should be generated or not for LAI</td>
</tr>
<tr>
<td>processor.l3b.lai.localwnd.bwr</td>
<td>Backward radius of the window for the local algorithm</td>
</tr>
<tr>
<td>processor.l3b.lai.localwnd.fwr</td>
<td>Forward radius of the window for the local algorithm</td>
</tr>
<tr>
<td>processor.l3b.lai.lut_path</td>
<td>L3B LUT file path</td>
</tr>
<tr>
<td>processor.l3b.lai.modelsfolder</td>
<td>Folder where the models are located</td>
</tr>
<tr>
<td>processor.l3b.lai.rsrcfgfile</td>
<td>L3B RSR file configuration for ProsailSimulator</td>
</tr>
<tr>
<td>processor.l3b.mono_date_lai</td>
<td>Specifies if monodate processing should be performed for LAI</td>
</tr>
<tr>
<td>processor.l3b.production_interval</td>
<td>The backward processing interval from the scheduled date for L3B products</td>
</tr>
<tr>
<td>processor.l3b.reprocess</td>
<td>Specifies if N-Day reprocessing should be performed for LAI</td>
</tr>
<tr>
<td>processor.l3b.reproc_production_interval</td>
<td>The backward processing interval from the scheduled date for L3C products</td>
</tr>
<tr>
<td>processor.l3b.sched_wait_proc_inputs</td>
<td>L3B/L3C/L3D LAI scheduled jobs wait for products to become available</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>processor.l3e.sched_wait_proc_inputs</td>
<td>L3E PhenoNDVI scheduled jobs wait for products to become available</td>
</tr>
<tr>
<td>processor.l4a.classifier</td>
<td>Random forest classifier / SVM classifier choices=[rf, svm]</td>
</tr>
<tr>
<td>processor.l4a.classifier.field</td>
<td></td>
</tr>
<tr>
<td>processor.l4a.classifier.rf.max</td>
<td>maximum depth of the trees used for Random Forest classifier</td>
</tr>
<tr>
<td>processor.l4a.classifier.rf.min</td>
<td>minimum number of samples in each node used by the classifier</td>
</tr>
<tr>
<td>processor.l4a.classifier.rf.nbtrees</td>
<td>The number of trees used for training</td>
</tr>
<tr>
<td>processor.l4a.erode-radius</td>
<td>The radius used for erosion</td>
</tr>
<tr>
<td>processor.l4a.mahalanobis-alpha</td>
<td>The parameter alpha used by the mahalanobis function</td>
</tr>
<tr>
<td>processor.l4a.min-area</td>
<td>The minimum number of pixel in an area where for an equal number of crop and nocrop samples the crop decision is taken</td>
</tr>
<tr>
<td>processor.l4a.mission</td>
<td>The main mission for the time series</td>
</tr>
<tr>
<td>processor.l4a.nbcomp</td>
<td>The number of components used by dimensionality reduction</td>
</tr>
<tr>
<td>processor.l4a.radius</td>
<td>The radius used for gapfilling in days</td>
</tr>
<tr>
<td>processor.l4a.random_seed</td>
<td>The random seed used for training</td>
</tr>
<tr>
<td>processor.l4a.range-radius</td>
<td>The range radius defining the radius (expressed in radiometry unit) in the multispectral space</td>
</tr>
<tr>
<td>processor.l4a.reference_data_dir</td>
<td>CropMask folder where insitu data are checked</td>
</tr>
<tr>
<td>processor.l4a.sample-ratio</td>
<td>The ratio between the validation and training polygons</td>
</tr>
<tr>
<td>processor.l4a.sampling-rate</td>
<td>The sampling rate</td>
</tr>
<tr>
<td>processor.l4a.sched_wait_proc_inputs</td>
<td>L4A Crop Mask scheduled jobs wait for products to become available</td>
</tr>
<tr>
<td>processor.l4a.segmentation-minsize</td>
<td>Minimum size of a region (in pixel unit) in segmentation.</td>
</tr>
<tr>
<td>processor.l4a.segmentation-spatial-radius</td>
<td>The spatial radius of the neighborhood used for segmentation</td>
</tr>
<tr>
<td>processor.l4a.smoothing-lambda</td>
<td>The lambda parameter used in data smoothing</td>
</tr>
<tr>
<td>processor.l4a.temporal_resampling_mode</td>
<td>The temporal resampling mode choices=[resample, gapfill]</td>
</tr>
<tr>
<td>processor.l4a.training-samples-number</td>
<td>The number of samples included in the training set</td>
</tr>
<tr>
<td>processor.l4a.window</td>
<td>The window expressed in number of records used for the temporal features extraction</td>
</tr>
<tr>
<td>processor.l4b.classifier</td>
<td>Random forest classifier / SVM classifier choices=[rf, svm]</td>
</tr>
<tr>
<td>processor.l4b.classifier.field</td>
<td>Training samples feature name</td>
</tr>
<tr>
<td>processor.l4b.classifier.rf.max</td>
<td>maximum depth of the trees used for Random Forest classifier</td>
</tr>
<tr>
<td>Ref</td>
<td>Sen2-Agri_SUM</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Issue</td>
<td>Page</td>
</tr>
<tr>
<td>3.2</td>
<td>143</td>
</tr>
</tbody>
</table>

- `processor.l4b.classifier.rf.min`: minimum number of samples in each node used by the classifier
- `processor.l4b.classifier.rf.nbtrees`: The number of trees used for training
- `processor.l4b.classifier.svm.k`: Type of kernel
- `processor.l4b.classifier.svm.opt`: Automatic optimisation of the parameters
- `processor.l4b.crop-mask`: Crop mask file path or product folder to be used
- `processor.l4b.mission`: The main mission for the time series
- `processor.l4b.random_seed`: The random seed used for training
- `processor.l4b.sample-ratio`: The ratio between the validation and training polygons
- `processor.l4b.sampling-rate`: 
- `processor.l4b.sched_wait_proc_inputs`: L4B Crop Type scheduled jobs wait for products to become available
- `processor.l4b.temporal_resampling_mode`: The temporal resampling mode choices=[resample, gapfill]
- `processor.l4b.training-samples-number`: The number of samples included in the training set
- `resources.working-mem`: OTB applications working memory (MB)
- `site.upload-path`: Upload path
- `processor.l3b.lai.laibandscfgfile`: Configuration of the bands to be used for LAI

**Appendix F.10 Advanced useful scripts**

- `delete_site.py` – allows deleting an existing site from the system

Examples:

```
delete_site.py -s site_short_name
```

Or

```
delete_site.py -s site_short_name -d false -e false -a false -l false -m false -t false
```

The “site_short_name” is the short name of the site from the “Sites” page of the Sen2-Agri GUI. The first example above removes only the site from the database without removing the files on disk. The second example removes also the files from the disk (downloaded or processed).

Use `delete_site.py -help` to see the list of options and details.

- `filter_site_download_tiles.py` – allows filtering, enabling or disabling the downloads for a certain site.

Example 1 to disable the download for all L8 product tiles for the site “test”. The value `-t 2` is for L8, if you want to disable S2 use `-t 1`:

```
filter_site_download_tiles.py -t 2 -s test -e false
```

Example 2 to enable the download for all L8 product tiles that intersect the area defined for the site “test”.

© UCL-Geomatics 2019

This document is the property of the Sen2-Agri partnership, no part of it shall be reproduced or transmitted without the express prior written authorisation of UCL-Geomatics (Belgium).
**filter_site_download_tiles.py** -t 2 -s test -e true

Example 3 to filter during downloading the S2 tiles for the site “test” to only the two specified by the parameter “–l” (they need to be comma followed by a space separated).

**filter_site_download_tiles.py** -t 1 -s test -e true -l "34RFP, 43FP"

**IMPORTANT NOTE:** The specified tiles need to be tiles that intersect the site extent otherwise the downloader will download nothing. This is a filter parameter and not a manner to replace the site extent. It should be used only to eliminate the tiles that have only small areas intersecting with your site and to keep only the relevant ones.

The script allows to the user controlling individually the download of each site. For example, you can disable the download of L8 for 2 sites but not for the 3rd. Nevertheless, after configuring these filters (especially if you enable per site downloads), you need eventually to restart the downloaders using “sudo systemctl restart sen2agri-landsat-downloader” or “sudo systemctl restart sen2agri-sentinel-downloader”.

**IMPORTANT NOTE:** The scripts should use them with care, only if you really need them and you fully understood how they are functioning (see also the help for the parameters). If incorrectly used, you might delete your site and your products or your downloader will not work anymore.

Use `filter_site_download_tiles.py -help` to see the list of options.

- **insert_l2a_product_to_db.py** – allows inserting into a site the L2A products processed in another location. See the help of the script for details.

- **import_theia.py** – allows inserting into a site the L2A products downloaded from THEIA or produced with a MAJA outside the Sen2Agri system. The downloaded products should be in the zip form and located into the same directory. The script allows to import all the THEIA products in that location to the specified site:

  ```bash
  import_theia.sh <INPUT_PRDS_DIR> <SITE_SHORT_NAME>
  ```

  where:
  - `<INPUT_PRDS_DIR>` is the directory where the products to be imported are located
  - `<SITE_SHORT_NAME>` – the short name of the site where the products will be imported

  The imported products are unzipped and a new level of directories is added in order to obtain the correct Sen2Agri MACCS/MAJA product structure. This is needed as the internal structure of the L2A products in the Sen2Agri is slightly different that the default MAJA generated products (an additional level of directory exists in the Sen2Agri, the rest of the product being identical).
## Appendix G – ESA CCI LC legend

<table>
<thead>
<tr>
<th>Label</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Global</strong></td>
<td><strong>Regional</strong></td>
</tr>
<tr>
<td>No Data</td>
<td>0</td>
</tr>
<tr>
<td>Cropland, rainfed</td>
<td>10</td>
</tr>
<tr>
<td>Cropland, rainfed, herbaceous cover</td>
<td>11</td>
</tr>
<tr>
<td>Cropland, rainfed, tree or shrub cover</td>
<td>12</td>
</tr>
<tr>
<td>Cropland, irrigated or post-flooding</td>
<td>20</td>
</tr>
<tr>
<td>Mosaic cropland (&gt;50%)/natural vegetation (tree, shrub, herbaceous cover) (&lt;50%)</td>
<td>30</td>
</tr>
<tr>
<td>Mosaic natural vegetation (tree, shrub, herbaceous cover) (&gt;50%)/cropland (&lt;50%)</td>
<td>40</td>
</tr>
<tr>
<td>Tree cover, broadleaved, evergreen, closed to open (&gt;15%)</td>
<td>50</td>
</tr>
<tr>
<td>Tree cover, broadleaved, deciduous, closed to open (&gt;15%)</td>
<td>60</td>
</tr>
<tr>
<td>Tree cover, broadleaved, deciduous, closed (&gt;40%)</td>
<td>61</td>
</tr>
<tr>
<td>Tree cover, broadleaved, deciduous, open (15-40%)</td>
<td>62</td>
</tr>
<tr>
<td>Tree cover, needleleaved, evergreen, closed to open (&gt;15%)</td>
<td>70</td>
</tr>
<tr>
<td>Tree cover, needleleaved, evergreen, closed (&gt;40%)</td>
<td>71</td>
</tr>
<tr>
<td>Tree cover, needleleaved, evergreen, open (15-40%)</td>
<td>72</td>
</tr>
<tr>
<td>Tree cover, needleleaved, deciduous, closed to open (&gt;15%)</td>
<td>80</td>
</tr>
<tr>
<td>Tree cover, needleleaved, deciduous, closed (&gt;40%)</td>
<td>81</td>
</tr>
<tr>
<td>Tree cover, needleleaved, deciduous, open (15-40%)</td>
<td>82</td>
</tr>
<tr>
<td>Tree cover, mixed leaf type (broadleaved and needleleaved)</td>
<td>90</td>
</tr>
<tr>
<td>Mosaic tree and shrub (&gt;50%)/herbaceous cover (&lt;50%)</td>
<td>100</td>
</tr>
<tr>
<td>Mosaic herbaceous cover (&gt;50%)/tree and shrub (&lt;50%)</td>
<td>110</td>
</tr>
<tr>
<td>Shrubland</td>
<td>120</td>
</tr>
<tr>
<td>Evergreen shrubland</td>
<td>121</td>
</tr>
<tr>
<td>Deciduous shrubland</td>
<td>122</td>
</tr>
<tr>
<td>Grassland</td>
<td>130</td>
</tr>
<tr>
<td>Lichens and mosses</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>150</td>
</tr>
<tr>
<td>Ref</td>
<td>Sen2-Agri_SUM</td>
</tr>
<tr>
<td>-----</td>
<td>---------------</td>
</tr>
<tr>
<td>Issue</td>
<td>Page</td>
</tr>
<tr>
<td>3.2</td>
<td>146</td>
</tr>
</tbody>
</table>

| Sparse vegetation (tree, shrub, herbaceous cover) (<15%) | Sparse tree (<15%) | 151 |
| Sparse shrub (<15%) | 152 |
| Sparse herbaceous cover (<15%) | 153 |
| Tree cover, flooded, fresh or brakish water | 160 |
| Tree cover, flooded, saline water | 170 |
| Shrub or herbaceous cover, flooded, fresh/saline/brakish water | 180 |
| Urban areas | 190 |
| Bare areas | 200 |
| Consolidated bare areas | 201 |
| Unconsolidated bare areas | 202 |
| Water bodies | 210 |
| Permanent snow and ice | 220 |