Technical note

|  |  |  |
| --- | --- | --- |
|  | Name | Date |
| Written by : | COURTOIS Aurélie (Thales Services) | 14/06/2017 |
| Approved by : |  |  |
| For application : |  |  |

PAGE D’ANALYSE DOCUMENTAIRE

|  |  |  |  |
| --- | --- | --- | --- |
| CONFIDENTIALITY : | | | |
| KEYWORDS : Sen2cor, MACCS, Muscate | | | |
| Authors : COURTOIS Aurélie (Thales Services) | | | |
| Summary : This document describes tools to convert MACCS or Muscate product to Sen2cor format | | | |
| configuration management : | Date : | |  |
| Contract : | | | |
| host | | Number of suppl.pages : | |

HISTORIQUE DES MODIFICATIONS

|  |  |  |
| --- | --- | --- |
| Version | Date | Objet |
| 1.0 | 14/06/17 | Creation of document |

**DIFFUSION**

(dernière page du document)

| **Nom** | **Sigle/SOCIETE** | **Nb** | **Nom** | **Sigle/Société** | **Nb** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

**TABLE OF CONTENTS**

[1. Scope of the document 4](#_Toc511636946)

[2. Tools installation 4](#_Toc511636947)

[3. Product Format 4](#_Toc511636948)

[3.1. MACCS product 4](#_Toc511636949)

[3.2. Muscate product 6](#_Toc511636950)

[3.3. Sen2cor product 7](#_Toc511636951)

[4. Scope of delivery 8](#_Toc511636952)

[5. Description and use of tools 9](#_Toc511636953)

[5.1. Use of tools 9](#_Toc511636954)

[5.2. MACCS/Muscate product to Sen2cor format 9](#_Toc511636955)

[5.2.1. Masks and images 9](#_Toc511636956)

[5.2.2. Metadata file 10](#_Toc511636957)

# Scope of the document

The purpose of this document is to describe tools to convert MACCS or Muscate product to Sen2cor format.

These tools allow creating Sen2cor format product with:

* AOT image
* Water vapor image
* Reflectance images
* Mask: cloud, snow, water …
* Xml metadata

The document presents how to use the tools, and also MACCS, Muscate and Sen2cor products.

# Tools installation

To use the tools to convert MACCS or Muscate product to Sen2cor format you need to have python 2.7 and to add this library:

* Gdal version 1.11.2\_full: to read, write or translate JPEG2000 images.



Some gdal version haven’t JP2OpenJPEG driver. This driver is necessary for open JPEG2000 images.

Tools have been tested on Linux system.

# Product Format

## MACCS product

A MACCS product has the following architecture:

|  |
| --- |
| product\_directory/  ├── xxx\_xSC\_xxx.DBL => empty file  ├── xxx\_xSC\_xxx.HDR => xml header describe product  ├── xxx\_xSC\_xxx.DBL.DIR  │ ├── xxx\_xSC\_PDTANX\_xxx\_ATB.HDR  │ ├── xxx\_xSC\_PDTANX\_xxx\_ATB.DBL.TIF => AOT image  │ ├── xxx\_xSC\_PDTIMG\_xxx\_FRE.HDR  │ ├── xxx\_xSC\_PDTIMG\_xxx\_FRE.DBL.TIF => reflectance image  │ ├── xxx\_xSC\_PDTIMG\_xxx\_SRE.HDR  │ ├── xxx\_xSC\_PDTIMG\_xxx\_SRE.DBL.TIF => reflectance image  │ ├── xxx\_xSC\_PDTANX\_xxx\_CLD.HDR  │ ├── xxx\_xSC\_PDTANX\_xxx\_CLD.DBL.TIF => cloud mask  │ ├── xxx\_xSC\_PDTANX\_xxx\_MSK.HDR  │ ├── xxx\_xSC\_PDTANX\_xxx\_MSK.DBL.TIF => masks  │ ├── xxx\_xSC\_PDTANX\_xxx\_QLT.HDR  │ ├── xxx\_xSC\_PDTANX\_xxx\_QLT.DBL.TIF => quality mask  │ ├── xxx\_xSC\_PDTQLK\_xxx.HDR  │ ├── xxx\_xSC\_PDTQLK\_xxx.DBL.JPG => quicklook  │   └── PRIVATE  │  ├── xxx\_QCK\_xxx.HDR => xml header xml describe quicklooks  └── xxx\_QCK\_xxx.DBL.DIR  ├── xxx\_QL2\_1\_AOT.jpg => AOT quicklook  ├── xxx\_QL2\_1\_FRE.jpg => reflectance quicklook  ├── xxx\_QL2\_1\_SRE.jpg => reflectance quicklook  ├── xxx\_QL2\_1\_VAP.jpg => water vapor quicklook  ├── xxx\_GIP\_CKQLTL\_xxx.EEF  ├── xxx\_GIP\_CKEXTL\_xxx.EEF  └── xxx\_GIP\_L2COMM\_xxx.EEF |

A MACCS product has these files:

* SRE: image in ground reflectance without the correction of slope effects
* FRE: image in ground reflectance with the correction of slope effects
* ATB: atmospheric and biophysical parameters with 2 bands :
* 1st band: water vapor content (WVC) coded over 8 bits
* 2nd band: aerosol optical thickness (AOT) coded over 8 bits
* CLD: cloud, cloud shadow and cirrus masks coded over 8 bits
* 1st bit: summary logical or of all cloud and shadow masks
* 2nd bit: logical or of all cloud masks
* 3rd bit: shadows mask from clouds within image
* 4th bit: shadows mask from clouds outside image
* 5th bit: reflectance threshold
* 6th bit: reflectance variation threshold
* 7th bit: extension of the cloud mask
* 8th bit: cirrus mask
* MSK: geophysical masks coded over 8 bits, 6 useful bits
* 1st bit: water mask
* 2nd bit: hidden surfaces
* 3rd bit: shadowed by topography mask
* 4th bit: sun too low flag
* 5th bit: tangent sun flag
* 6th bit: snow
* QLT: Quality masks coded over 8 bits
* 1st band: saturation mask
* 2nd band: aberrant pixels
* 3rd band:
* 1st bit: edge mask
* 2nd bit: AOT pixel mask (0 if computed, 1 if interpolated)
* 3rd bit: water vapor pixel mask (0 if computed, 1 if interpolated)

If the MACCS product is from Sentinel-2, there are 2 resolutions: R1 and R2 for every file.

## Muscate product

A Muscate product has the following architecture:

|  |
| --- |
| product\_directory/  ├── xxx\_MTD\_ALL.xml *🡪 metadata*  ├── xxx\_QKL\_ALL.jpg *🡪 quicklook*  ├── xxx\_ATB\_Rj.tif *🡪 atmospheric and biophysical parameters*  ├── xxx\_FRE\_Bi.tif *🡪 image in ground reflectance with the correction of slope effects*  ├── xxx\_SRE\_Bi.tif *🡪 image in ground reflectance without the correction of slope effects*  └── MASKS  ├── xxx\_CLM\_Rj.tif *🡪 cloud mask*  ├── xxx\_DFP\_Rj.tit *🡪 defective pixels mask*  ├── xxx\_DTF\_Rj-Dk.tif *🡪 detector footprint mask*  ├── xxx\_EDG\_Rj.tif *🡪 edge mask*  ├── xxx\_IAO\_Rj.tif *🡪 interpolated AOT pixels mask*  ├── xxx\_MG2\_Rj.tif *🡪 geophysical mask*     └── xxx\_SAT\_Rj.tif *🡪 saturation mask* |

* MTD: metadata
* QKL: quicklook
* SRE: image in ground reflectance without the correction of slope effects
* FRE: image in ground reflectance with the correction of slope effects
* ATB: atmospheric and biophysical parameters with 2 bands :
* 1st band: water vapor content (WVC) coded over 8 bits
* 2st band: aerosol optical thickness (AOT) coded over 8 bits
* SAT: saturation mask coded over 8 bits, 1 bit per spectral band (number of useful bits = number of spectral bands)
* DFP (optional): defective pixels mask coded over 8 bits, 1 bit per spectral band (number of useful bits = number of spectral bands)
* CLM: cloud mask coded over 8 useful bits:
* 1st bit (CM1): cloud\_mask\_all, result of a “logical OR” for all the cloud and shadow masks
* 2nd bit (CM2): cloud\_mask\_all\_cloud, result of a “logical OR” for all the cloud masks (except extension mask)
* 3rd bit (CM3): cloud\_mask\_refl, cloud mask identified by a reflectance threshold
* 4th bit (CM4): cloud\_mask\_refl\_var, cloud mask identified by a threshold on reflectance variance
* 5th bit (CM5): cloud\_mask\_extension, cloud mask identified by the extension of cloud masks
* 6th bit (CM7): cloud\_mask\_shadow, shadow mask of clouds inside the image
* 7th bit (CM8): cloud\_mask\_sahdvar, shadow mask of clouds outside the image
* 8th bit (CM9): cloud\_mask\_cirrus, cloud mask identified with the cirrus spectral band
* MG2: geophysical mask of level 2, made of 1 band coded over 8 useful bits:
* 1st bit (WTR): water mask
* 2nd bit (CM2) : cloud\_mask\_all\_cloud, result of a “logical OR” for all the cloud masks (except extension mask)
* 3rd bit (SNW): snow mask
* 4th bit (logical OR between CM7 and CM8): shadow masks of clouds
* 5th bit (SHD): topographical shadows mask
* 6th bit (HID): hidden areas mask
* 7th bit (STL): sun too low mask
* 8th bit (TGS): tangent sun mask
* EDG: edge mask coded over 8 bits, 1 useful bit
* IAO: interpolated AOT pixels mask
* DTF : Detectors FootPrint mask coded over 8 bits, 1 bit per spectral band (number of useful bits = number of spectral bands)

## Sen2cor product

A Sen2cor product has the following architecture:

|  |
| --- |
| product\_directory/  ├── AUX\_DATA  ├── DATASTRIP  ├── GRANULE  │ ├── AUX\_DATA  │ ├── IMG\_DATA  │ │ └──Rjm  │ │ ├── xxx\_AOT\_Rjm.jp2 *🡪 AOT image*  │   │ ├── xxx\_Bi\_Rjm.jp2 *🡪 reflectance image*  │ │ ├── xxx\_SCL\_Rjm.jp2 *🡪 masks*  │ │ ├── xxx\_TCI\_Rjm.jp2  │ │ ├── xxx\_VIS\_Rjm.jp2  │ │ └── xxx\_WVP\_Rjm.jp2 *🡪 water vapor image*  │ │  │ ├── MTD\_TL.xml *🡪 xml metadata*  │ └── QI\_DATA  ├── HTML  ├── INSPIRE.xml  ├── L2A\_Manifest.xml  ├── manifest.safe  ├── MTD\_MSIL2A.xml  ├── rep\_info  └── xxx\_report.xml |

All masks are in file xxx\_SCL\_Rjm.jp2, where Rj is the mask resolution. At every mask is associate one or more value:

* Edge mask: 0
* Cloud mask:
* Medium probability: 8, 9 and 10
* High probability: 9 and 10
* Shadow mask: 3
* Snow mask: 11
* Water mask: 6

# Scope of delivery

We describe in this section source code of the delivery witch is in Python language. There are some directories:

* AOTMap : with scripts to translate AOT image from MACCS or Muscate format to Sen2cor format,
* demo : directory with inputs, outputs and an example of shell script to run tools,
* dimap: with script to read MACCS or Muscate metadata and to write necessary information in Sen2cor metadata format,
* masks: with scripts to find and translate masks,
* product: with scripts to get product type and transform input product to Sen2cor format,
* refleImgs: with scripts to translate reflectance images from MACCS or Muscate format to Sen2cor format,
* waterVapor: with scripts to translate water vapor image from MACCS or Muscate format to Sen2cor format.

The Table 1 gives the files in delivery and their function.

Table 1 : Description of files in dellivery

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| MaccsMuscate2Sen2cor.py | PY | Executable script to run tools |
| XmlTools.py | PY | Give tools to read/write xml |
| MACCSAOTMap.py | PY | Script to translate AOT image from MACCS to Sen2cor. |
| MuscateAOTMap.py | PY | Script to translate AOT image from Muscate to Sen2cor. |
| MACCS\_20170410 | Dir | Example of MACCS product (input) |
| Muscate\_20170410 | Dir | Example of Muscate product (input) |
| run.sh | SH | Example of shell script to run tools |
| Sen2cor\_MACCS\_20170410 | Dir | Sen2cor product from MACCS product (output) |
| Sen2cor\_Muscate\_20170410 | Dir | Sen2cor product from Muscate product (output) |
| MACCSDimap.py | PY | Script to read/write metadata for MACCS product |
| MACCSS2Dimap.py | PY | Script to read/write metadata for MACCS product, specific to Sentinel-2 |
| MuscateDimap.py | PY | Script to read/write metadata for Muscate product |
| MACCSMasks.py | PY | Script to get MACCS masks and transform in Sen2cor mask |
| MuscateMask.py | PY | Script to get Muscate masks and transform in Sen2cor mask |
| MACCSProduct.py | PY | Script to get product type and transform MACCS product in Sen2cor product |
| MACCSS2Product.py | PY | Script to get product type and transform MACCS product in Sen2cor product, specific to Sentinel-2 |
| MuscateProduct.py | PY | Script to get product type and transform Muscate product in Sen2cor product |
| MACCSReflectImgs.py | PY | Script to translate reflectance image from MACCS to Sen2cor |
| MuscateReflectImgs | PY | Script to translate reflectance image from Muscate to Sen2cor |
| MACCSWaterVapor.py | PY | Script to translate water vapor image from MACCS to Sen2cor |
| MuscateWaterVapor.py | PY | Script to translate water vapor image from Muscate to Sen2cor |

# Description and use of tools

## Use of tools

An example of how to use the tools is in demo/run.sh. You can convert a MACCS/Muscate product by running the command line:

|  |
| --- |
| python MaccsMuscate2Sen2cor.py –p <MACCS product> -w <workspace> |

You can also execute the shell script run.sh but you need to modified product and workspace directory before.

You have the description of all options in Table 2.

Table 2 : Option to convert MACCS/Muscate product in Sen2cor format

|  |  |  |  |
| --- | --- | --- | --- |
| **Option** | **Description** | **Optional (True/False)** | **Default value** |
| --product | Directory for product to convert | False | -- |
| --reflType | Type of reflectance image to use: FRE or SRE | True | FRE |
| --workspace | Location of workspace | False | -- |

It creates a directory with name: Sen2cor\_NameProduct. NameProduct is the name of input product directory. If the output directory already exists, then it creates a new name. Moreover, there is text file (Convert2Sen2cor\_date.log) write at end of execution with some information and/or warnings. These information and/or warnings are written in the console during the execution.

## MACCS/Muscate product to Sen2cor format

### Masks and images

To create a Sen2cor product from MACCS or Muscate product, tools translate some masks and images and a metadata file. To do this, they need first to get type of input product. A MACCS product is a little different for Sentinel-2 that’s why tools separate these two cases.

In terms of type product, tools find masks and image to translate. For AOT, reflectance and water vapor images they just translate tif image to jp2 image and save with a new name whose correspond to Sen2cor product. For mask, they have to concatenate some images in one. For edge, snow and water, they take the MACCS or Muscate masks and create a new mask with the correct value (0 for edge, 11 for snow and 6 for water). For shadows, they take masks for shadow of clouds inside the image and clouds outside the image and put 3 in the mask.

Finally, for clouds and MACCS product they take the 2nd and 7th bits of CLD mask for medium probability cloud mask and put 8, and take the 2nd bit of CLD mask for high probability cloud mask and put 9. For clouds and Muscate product, they take the 2nd and the 5th bits of CLM mask for medium probability cloud mask and put 9, and take the 2nd bit of CLM mask for high probability cloud mask and put 9.

Same masks are taken to calculate a scene classification for metadata.

To use converted product in SNAP, images need to be in:

* Unsigned int for reflectance, AOT and water vapor
* Byte for mask

### Metadata file

For a MACCS product, the metadata is the .HDR file in the product directory. For a Muscate product, the metadata is the MTD\_ALL.xml file.

A Sen2cor metadata file is created (MTD\_TL.xml) with information in MACCS or Muscate metadata file. If a value is unknown then the value write in Sen2cor metadata file is N/A.

Moreover, for MACC Sentinel-2 product and Muscate product, another file is created (MTD\_MSIL2A.xml). This file is necessary to open tiles with SNAP (Sentinel Application Platform).