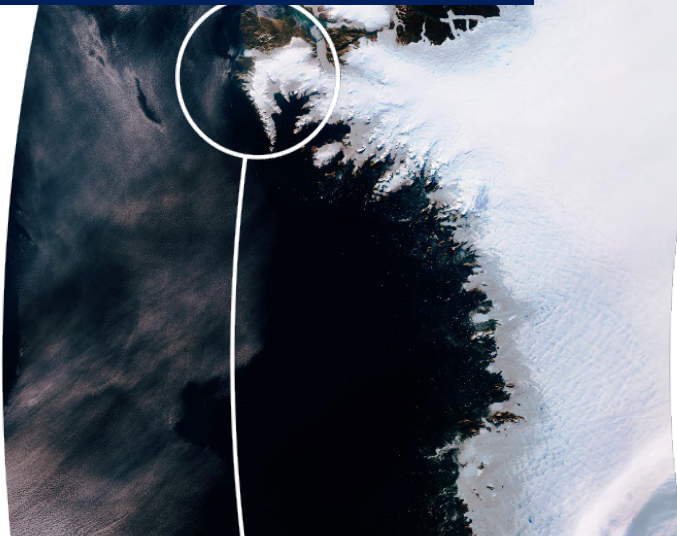




# Introduction to SNAP

Ana B. Ruescas  
*Remote Sensing Expert, Brockmann Consult GmbH*

*OSOW 2023*



# SNAP

## *Earth Observation Made Easy*

### SNAP is

- an ecosystem to analyse, process and communicate Earth Observation data
- an Open-Source Project - [github.com/senbox-org](https://github.com/senbox-org)
- scalable to run on notebooks up to large production clusters
- used for scientific analysis, operational production and training
- easy to use

### SNAP can

- access many satellite-based Earth Observation data products as well as generic raster formats directly in the cloud
- visualise the data in many ways
- analyse data using statistical functions, mathematical operations, correlation, comparison with point and vector data
- process satellite data with instrument specific as well as generic raster data operations
- save sessions and export results in various raster and non-raster formats
- be extended using Java and Python API

### SNAP has

- comprehensive documentation - [step.esa.int](https://step.esa.int)
- > 1 Million users and active community, > 10 000 forum users – [forum.step.esa.int](https://forum.step.esa.int)
- a long-term commitment of the European Space Agency



# SNAP Team and Heritage

- Development of SNAP and the Sentinel Toolboxes started in 2014
- BEAM and NEST, the predecessors, started in 2000
- More than 20 years available
- ESA is planning with SNAP beyond 2027



- Development consortium
  - Brockmann Consult, Germany
  - Skywatch, Canada
  - CS Romania
  - CS France

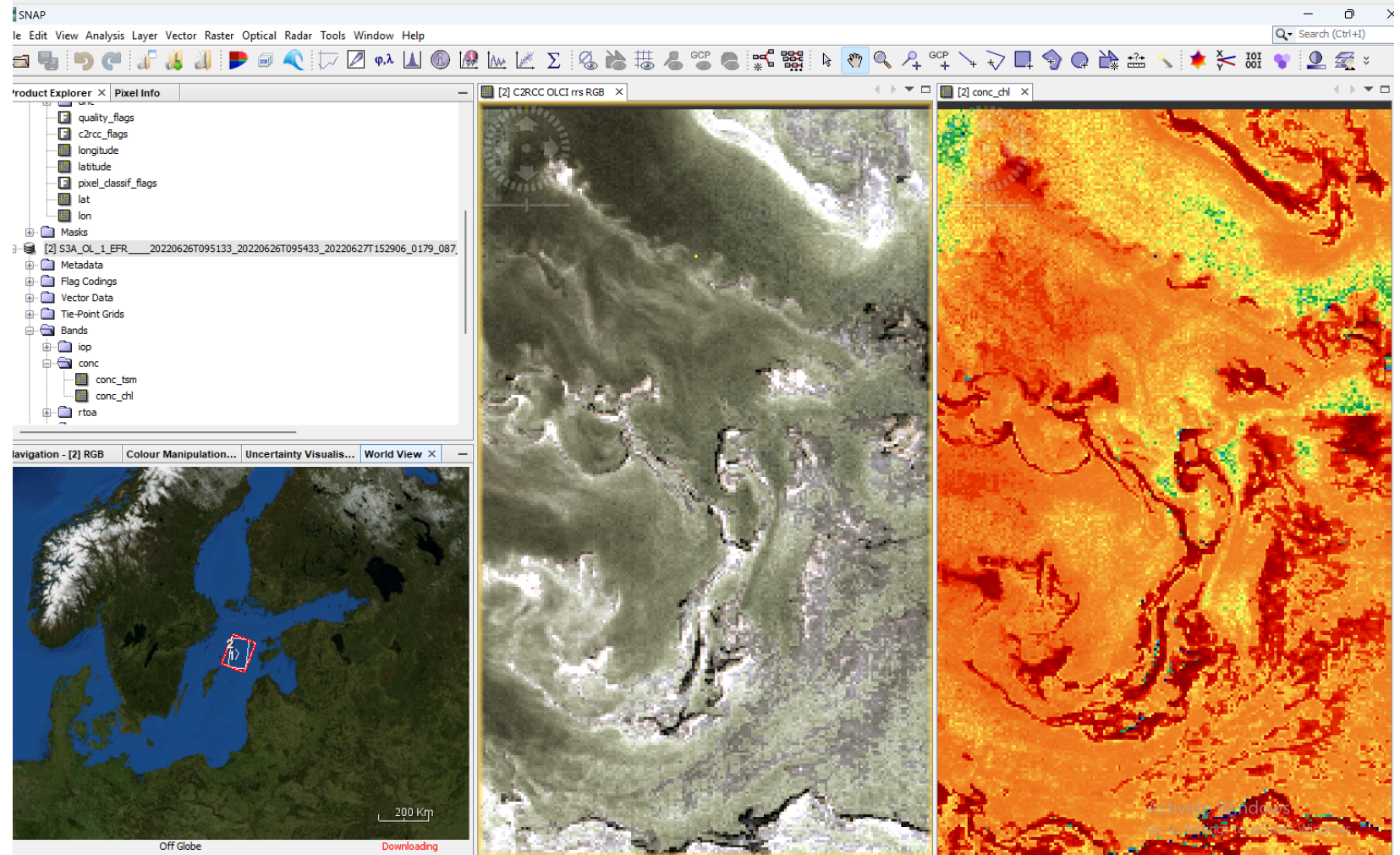


# Data Visualisation

SNAP Desktop is the GUI application which allows access to a large number of EO and generic raster data.

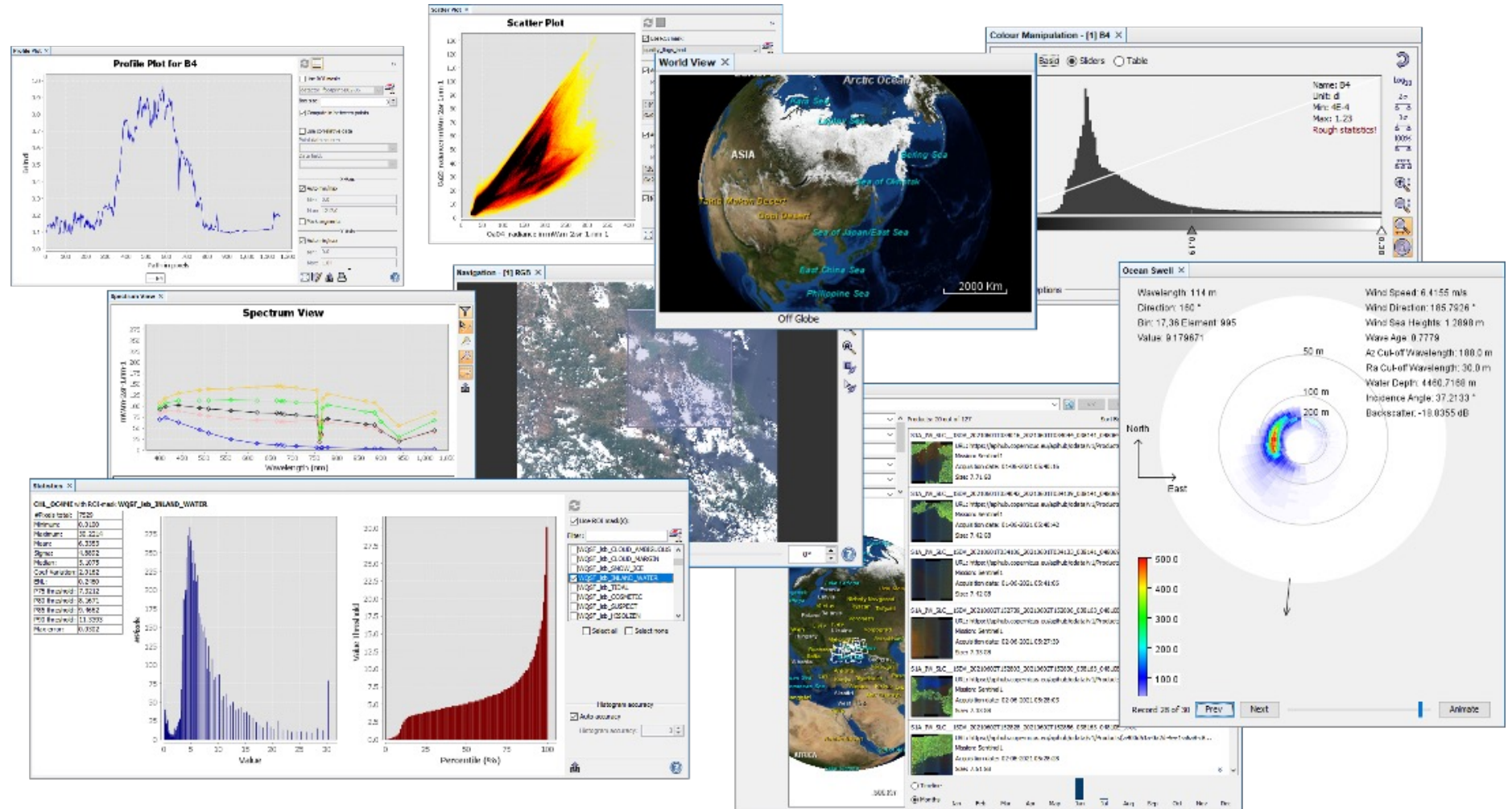
It provides various tools to display the data, and to visually analyse them.

The figure on the right shows an RGB of a Sentinel 3 product together with a visualisation of a water quality parameter.



SNAP provides a rich suite of tools for data analysis, including profile and spectrum plots, statistical analysis, extraction of points through time series, and comparison with reference data (match-ups).

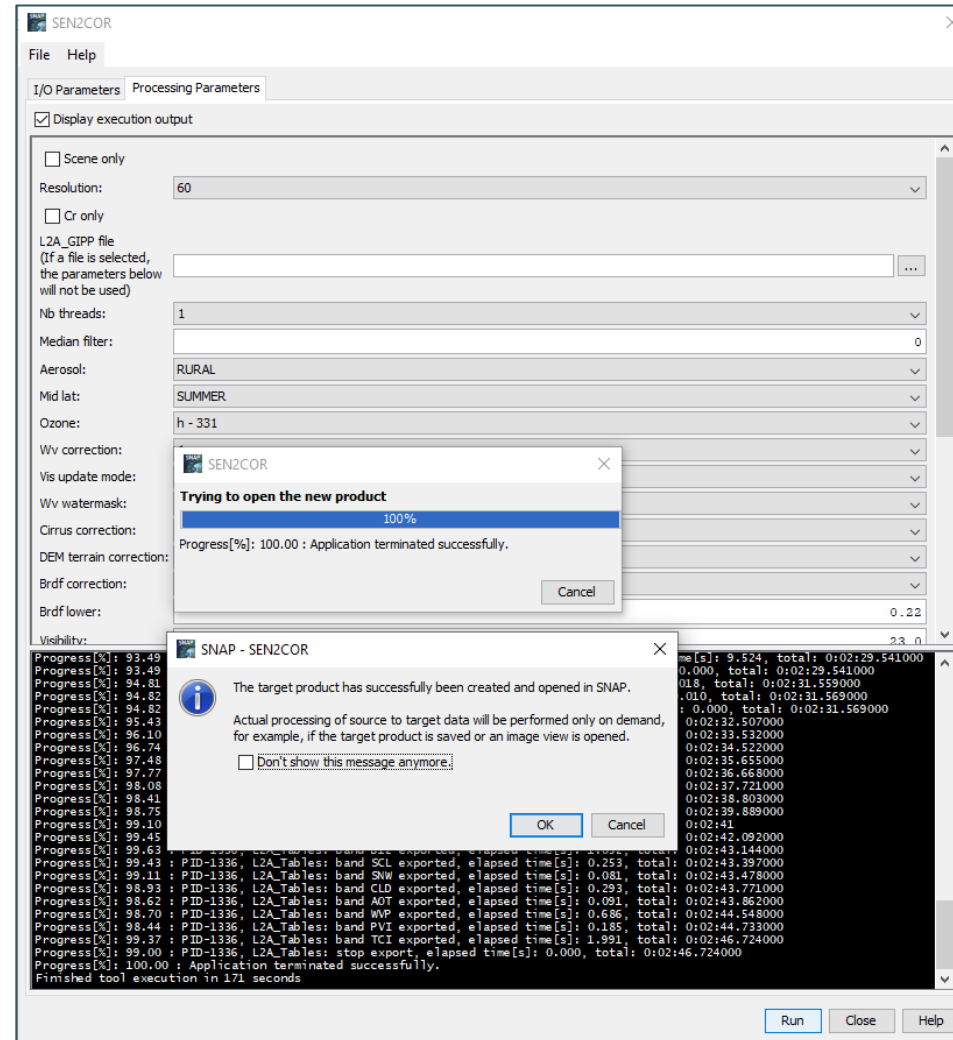
The figure shows some of the graphical analysis tools included in SNAP.

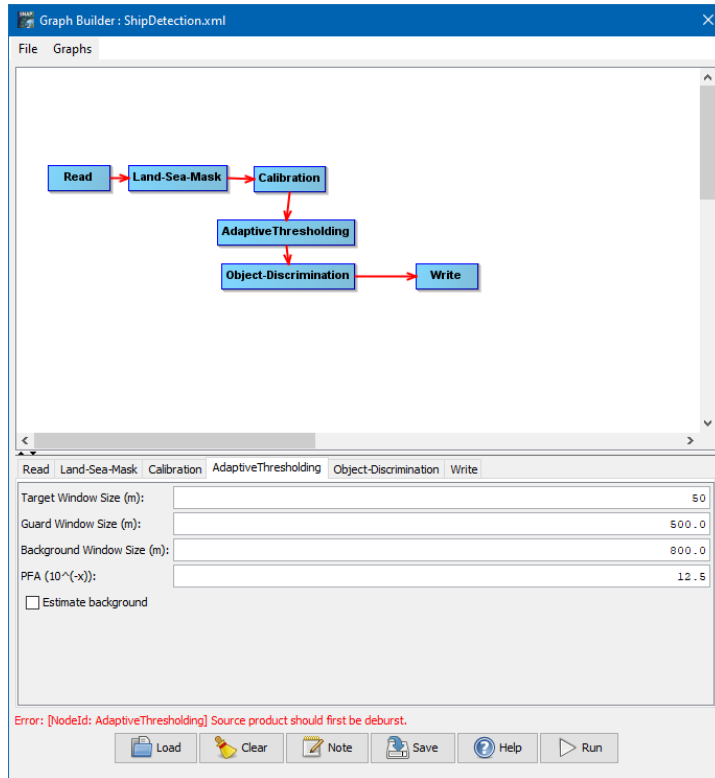


# Data Processing

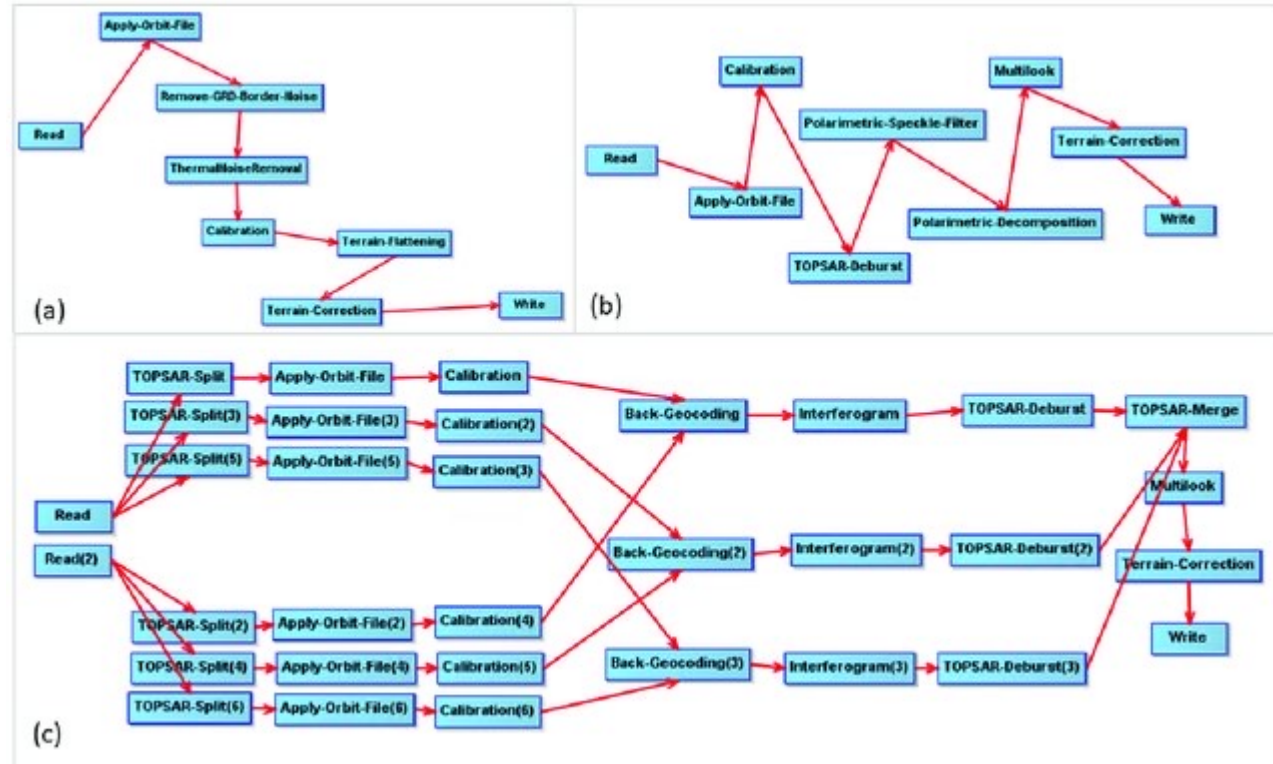
SNAP Data Processors analyse one or more input products and generate a new output product. Processors exist for generic operations such as band arithmetic, map projection or temporal aggregation. SNAP also provides a very large number of thematic processors, e.g. for atmospheric correction, biophysical indices calculation or retrieval of water quality. And SNAP supports special calibration of correction of satellite instruments with dedicated processors.

The figure shows the GUI for the Sentinel-2 Atmospheric Correction Processor `sen2cor`. The screenshot was taken when the processor was successfully executed. The black background shows the logging information during execution of the processor.





The SNAP graph builder allows to connect SNAP operators in processing graphs. These can be executed locally or in large clusters and cloud systems.



Example: Workflows in the SNAP graph builder tool for producing Synthetic Aperture Radar (SAR) analysis ready data (ARD) products. From Ticehurst, et al (2019). Building a SAR-Enabled Data Cube Capability in Australia Using SAR Analysis Ready Data. Data. 4. 100. 10.3390/data4030100.

## User Forum

The screenshot shows the 'step forum' interface. At the top, there's a navigation bar with 'Categories', 'Latest', and 'Top' options. Below this, there are three main categories listed:

- s1tbx** (29 / month): The S1 Toolbox category regroups all threads about the Sentinel-1 Toolbox, as SAR readers or processors. It includes sub-categories like Problem Reports, Interferometry, Polarimetry, StaMPS, PyRate, and snaphu.
- s2tbx** (15 / month): The S2 Toolbox category regroups all threads about the Sentinel-2 Toolbox as Sentinel-2 product readers and product manipulation, Sentinel-2 processors as L2A processor for atmospheric correction, L3 processor for temporal synthesis, etc. It includes sub-categories like sen2cor, sen2three, Problem Reports, and sen2like.
- s3tbx** (2 / month): The S3 Toolbox category regroups all threads about the Sentinel-3 Toolbox as readers and processors for Sentinel-3 OLCI & SLSTR L1 & L2. Useful information about Sentinel-3 and the data can be found at the S3VT Documentation page. It includes sub-categories like Problem Reports.

## Issue Tracker

The screenshot shows the 'SNAP' Issue Tracker landing page. The header includes navigation options like 'Ihre Aufgaben', 'Projekte', 'Filter', and 'Dashboard'. The main content is titled 'Landing Page' and lists several projects with their respective managers:

- CHRIS-Box (CHRIS)**: Leitung: Marco Peters
- Sentinel-1 Toolbox (SITBX)**: Leitung: Luis Veci
- Sentinel-2 Toolbox (SIITBX)**: Leitung: Florian Douzich
- Sentinel-3 Toolbox (SIITBX)**: Leitung: Marco Peters
- Sentinel Application Platform (SNAP)**: Leitung: Marco Peters
- SMOS Toolbox (SMOSTBX)**: Leitung: Tom Block
- SNAP Requirements Monitoring (SRM)**: Leitung: Marco Peters
- SNAP User Feedback (SUF)**: Leitung: Oana Hogoiu

## Tutorials

The screenshot shows the 'SNAP Tutorials' page. The header includes a 'DOCUMENTATION' link. The main content is titled 'Tutorials' and states 'Found 70 tutorials'. Below this, there is a list of tutorial categories:

- SNAP (GENERAL TOOLBOX USAGE)
- SENTINEL-1 TOOLBOX (SAR APPLICATIONS)
- SENTINEL-2 TOOLBOX (HIGH RESOLUTION OPTICAL APPLICATIONS)
- SENTINEL-3 TOOLBOX (MEDIUM RESOLUTION OPTICAL APPLICATIONS)
- ESA TRAINING COURSES (ESA TRAINING COURSES)
- EXTERNAL RESOURCES (EXTERNAL RESOURCES)
- OTHER (OTHER TUTORIALS)
- ALL (ALL TUTORIALS)

## For Developers

The screenshot shows the 'SNAP Developers' page. The header includes a 'COMMUNITY' link. The main content is titled 'Developers' and lists several resources for developers:

- Source code**: We are using Git to keep track of code changes, and the source code is available on... We highly encourage fixes and new features made to the code be submitted to the...
- SNAP API Documentation**: For developers who want to extend or patch SNAP we provide the Javadoc here:
  - SNAP Engine API Documentation
  - SNAP Desktop API Documentation
- License**: The Sentinel Toolboxes and full source code is open-source software and is distributed under the... license.
- Wiki**: The [Developers Wiki](#) contains valuable resources for getting familiar with the software.
- Forum**: There is a dedicated section in the [Forum](#) for development-oriented topics.



# SNAP Community & Tutorials



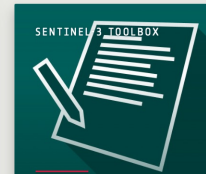
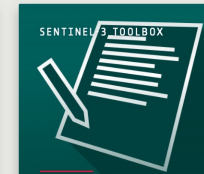
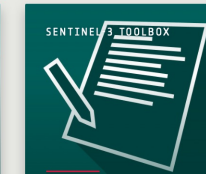
<http://forum.step.esa.int/c/s3tbx>  
<http://step.esa.int/main/doc/tutorials/>  
<https://senbox.atlassian.net/wiki/spaces/SNAP/pages/1898053693/SNAP+FAQs>

**SENTINEL-3 TOOLBOX (MEDIUM RESOLUTION OPTICAL APPLICATIONS)**

- ESA TRAINING COURSES (ESA TRAINING COURSES)
- EXTERNAL RESOURCES (EXTERNAL RESOURCES)
- OTHER (OTHER TUTORIALS)
- ALL (ALL TUTORIALS)

Showing [1 ... 6] from 6 Sort By (Tutorial)

Search for specific tutorials...

 <p><b>DOCUMENT</b></p> <p><b>Data conversion and export for Sentinel-3</b></p> <p>A short guide on converting and exporting Sentinel-3 data to GeoTIFF format for use in GIS software. This guide is kindly provided by our users lenkafonkova and hek17 in the forum.</p> <p>JUNE 22, 2018 MARPET</p> <p>READ →</p>	 <p><b>VIDEO</b></p> <p><b>Download &amp; Visualise Sentinel-3 Data</b></p> <p>EUMETSAT shows how to download and visualise their provided Sentinel-3 data with the Sentinel-3 Toolbox.</p> <p>MARCH 1, 2017 ADMIN</p> <p>PLAY →</p>	 <p><b>DOCUMENT</b></p> <p><b>Introduction to Sentinel-3 Toolbox</b></p> <p>This presentation gives a general introduction to the usage of the Sentinel-3 Toolbox.</p> <p>JUNE 1, 2015 ADMIN</p> <p>READ →</p>	 <p><b>DOCUMENT</b></p> <p><b>Rayleigh Correction Tutorial (S3 OLCI, MERIS, S2 MSI)</b></p> <p>Introduction to the Rayleigh correction provided by the Sentinel-3 Toolbox. The document gives information on the collocation of Sentinel-3, Sentinel-2 and Sentinel-1.</p> <p>JUNE 16, 2021 ANA B. RUESCAS, DAGMAR MÜLLER</p> <p>READ →</p>	 <p><b>DOCUMENT</b></p> <p><b>S3TBX Collocation Tutorial</b></p> <p>The tutorial explains how to collocate satellite data and which technical and scientific considerations need to be made. Even the examples focus on the collocation of Sentinel-3, Sentinel-2 and Sentinel-1.</p> <p>OCTOBER 7, 2022 ANA B. RUESCAS, MARCO PETERS</p> <p>READ →</p>
---	--	--	--	---





Dear SNAP users, ✎ Edit ✕

We would love to have your feedback on your SNAP experience. It would help us to know which are the things that should be improved and taken into consideration for the future, so that you will be satisfied when using SNAP.

Please take an anonymous survey in order to help us helping you:  
[SNAP User Survey](#)

Many thanks from SNAP Team!

all categories ▾ **Categories** Latest New (2) Unread (275) Top + New Topic

Category	Topics	Latest
<b>s1tbx</b>  The S1 Toolbox category regroups all threads about the Sentinel-1 Toolbox, as SAR readers or processors. 5.6k 60 unread ■ Problem Reports 12 unread ■ Interferometry 2 unread ■ Polarimetry ■ StaMPS 3 unread ■ PyRate ■ snaphu 2 unread	<b>Failure to import ICEYE H5 file and solution</b> ● ■ Problem Reports 0 10h	
	<b>Error while running snaphu-unwrapping</b> ● ■ snaphu 0 11h	
	<b>COSMO-SkyMed Coreg_ifg_subset Error</b> ● ■ s1tbx 10 20h	
	<b>Snap2stamps error</b> ● ■ StaMPS 239 20h	
	<b>Mosaicking of SAR SLC images</b> ● ■ Interferometry 0 22h	
	<b>NESZ of Radarsat-2</b> ● ■ s1tbx 0 22h	
	<b>Atmospheric Correction for InSAR</b> ● ■ s1tbx 119 23h	
	<b>Uav_sar_snap</b> ● ■ snap 0 1d	
	<b>ps_plot velocity</b> ● ■ StaMPS 12 1d	
	<b>Phase to displacement theory</b> ● ■ Show Room 0 1d	
	<b>SNAP software Back-Geocoding Error</b> ● ■ Problem Reports 0 1d	
	<b>No_intial_PS_candidites (sentinel-1)</b> ● ■ Problem Reports 0 1d	
	<b>UNITS of sentinel 3</b> ● ■ s3tbx 0 1d	
<b>s2tbx</b>  The S2 Toolbox category regroups all threads about the Sentinel-2 Toolbox as Sentinel-2 product readers and product manipulation, Sentinel-2 processors as L2A processor for atmospheric correction, L3 processor for temporal synthesis, etc. 2.5k 61 unread ■ sen2cor 16 unread ■ sen2three ■ Problem Reports 7 unread ■ sen2like		
<b>s3tbx</b>  The S3 Toolbox category regroups all threads about the Sentinel-3 Toolbox as readers and processors for Sentinel-3 OLCI & SLSTR L1 & L2. Useful information about Sentinel-3 and the data can be found at the <a href="#">S3VT Documentation</a> page. 579 21 unread 1 new ■ Problem Reports 3 unread		
<b>snap</b>  This category contains all topic about the Sentinel Toolbox Application (SNAP) not related to a specific Sentinel Toolbox. 2.2k 114 unread 1 new		

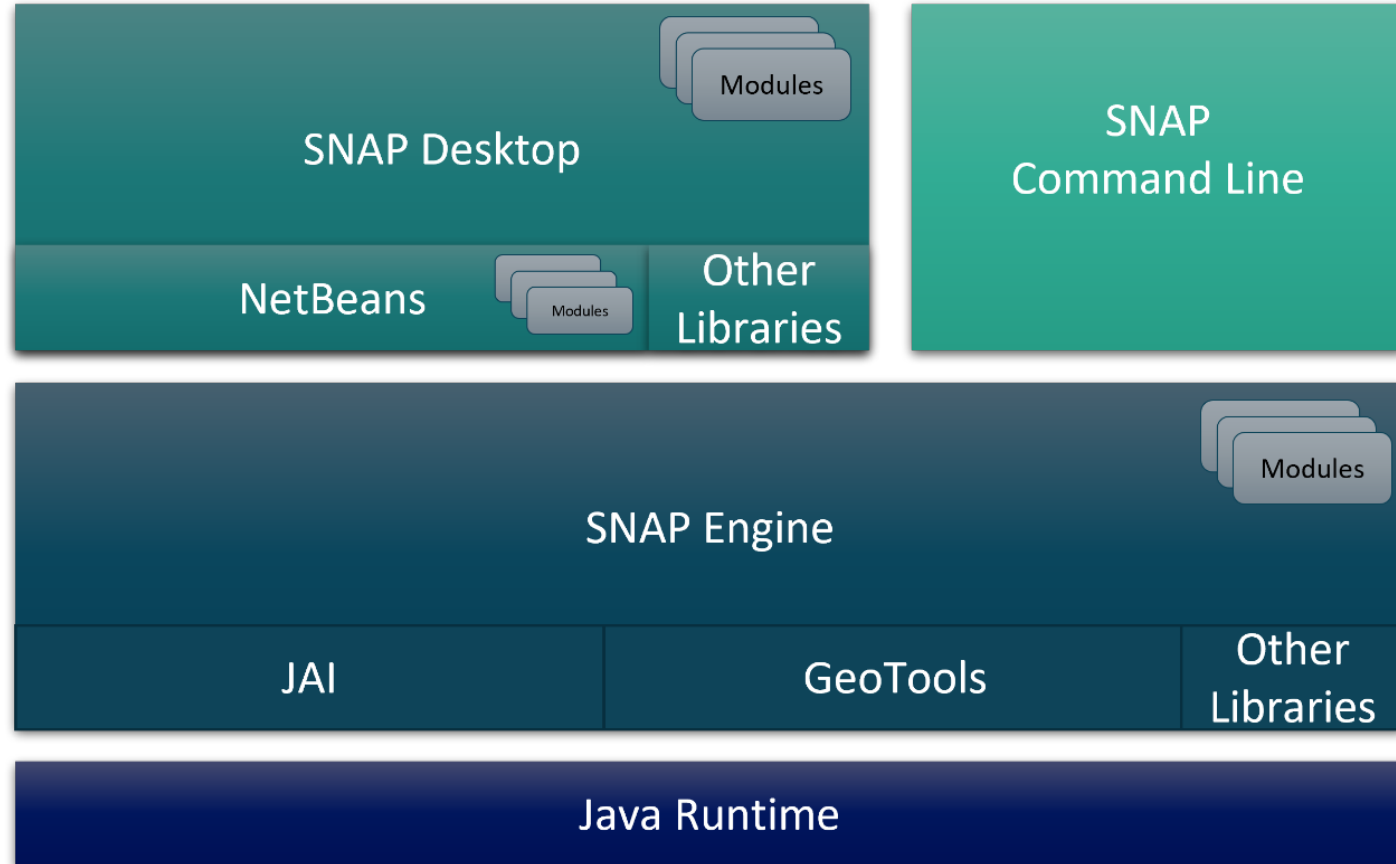
# System Architecture

SNAP system architecture is structured into four layers.

At the bottom, the JDK provides the runtime environment and abstracts from the operating system.

On top of the JDK sits the SNAP engine. It provides the common functionality, the product data model, the processing framework, and the API for using and extending SNAP.

SNAP Desktop and Command Line interface are based on top of SNAP Engine and add the user interfaces.

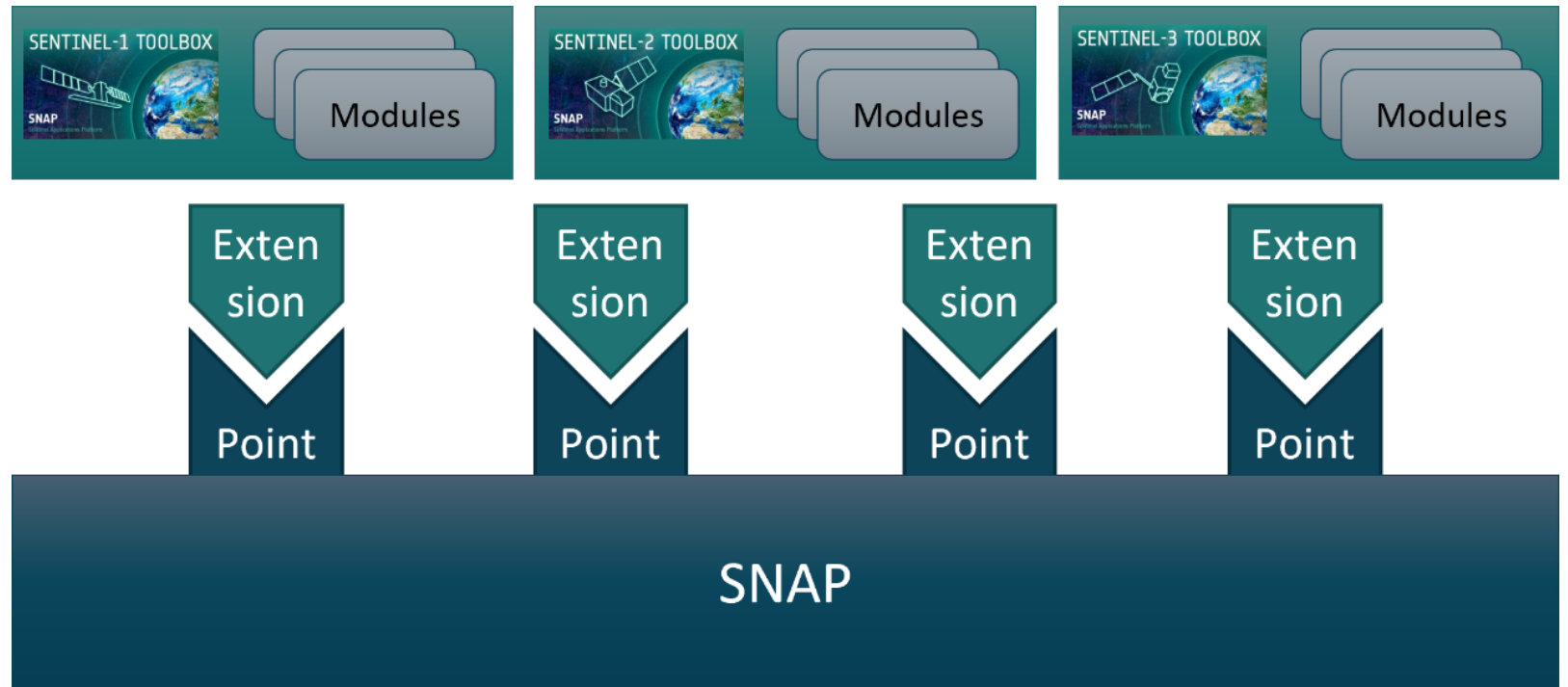


# Toolbox Concept (Plug-ins)

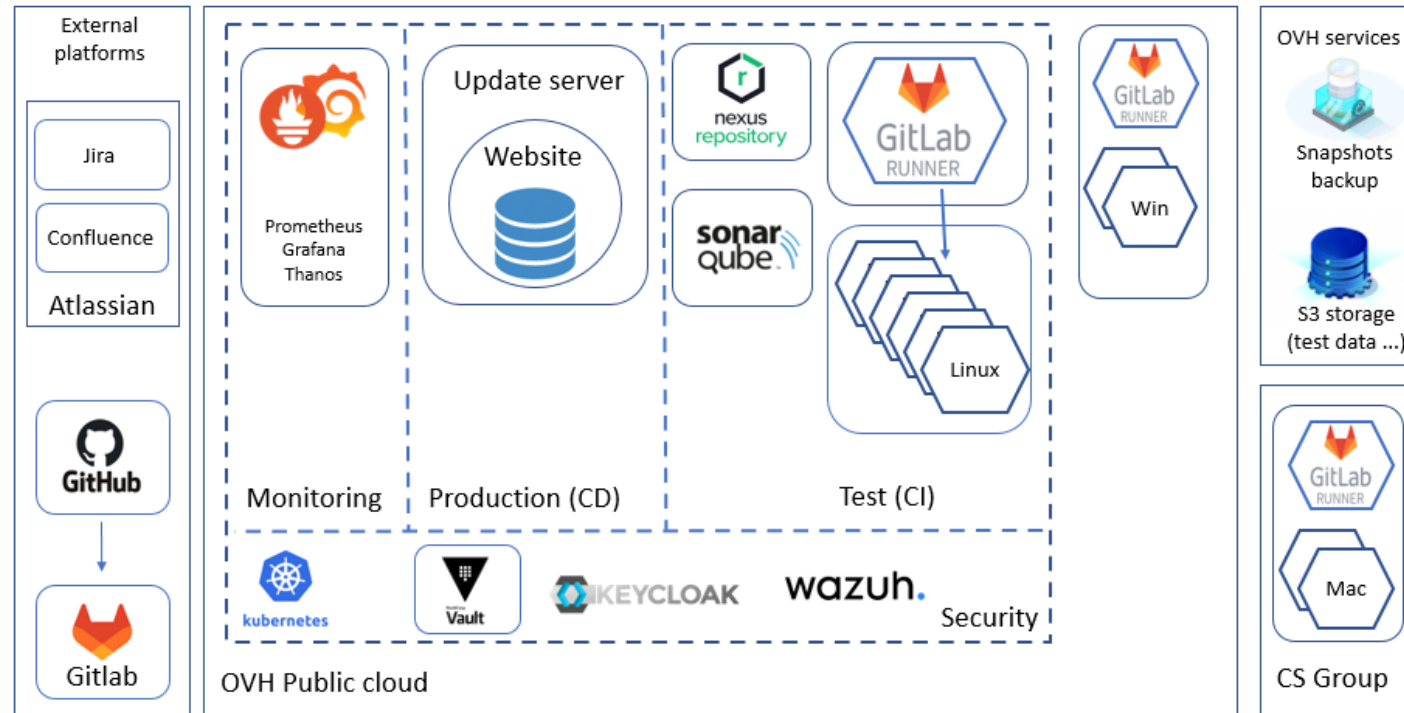
SNAP is made up by modules.  
The whole SNAP source code is split into numerous modules.

Open-Source Developers can extend SNAP by writing their own module and use plug-in points to link with SNAP.

Toolboxes are just a set of modules which are distributed and installed all together. This makes it convenient for users to install the Sentinel Toolboxes. Toolboxes docking with their extensions at the extension points



# Continuous Integration and Quality Assurance



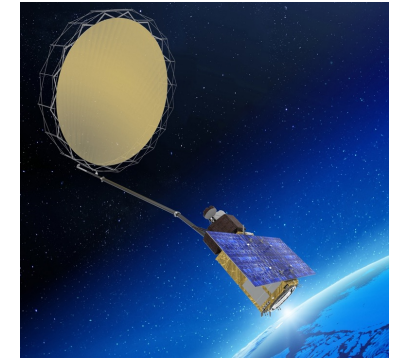
SNAP developers are using state-of-the-art systems for continuous integration, fully automated tests with large test coverage, tools for system monitoring, and a security layer including secured storage of credentials, AntiVirus & AntiMalware. SNAP ecosystem is fully GDPR compliant.

- **06/2023 – SNAP 10**
  - Optical and Microwave Toolboxes
  - Large software renovation
  - Making it technologically future proof
- **12/2023 – SNAP 11**
  - Product Groups
  - STAC support (Spatio-Temporal Asset Catalogue)
  - Preparing for hyperspectral CHIME and microwave CIMR
- **06/2024 – SNAP 12**
  - Change detection Toolbox
  - S1 ETAD improvements
  - Generic cloud detection operators
- **12/2024 – SNAP 13**
  - Support NISAR, BIOMASS
  - Time series tools
  - Optical and hyperspectral synergistic tools

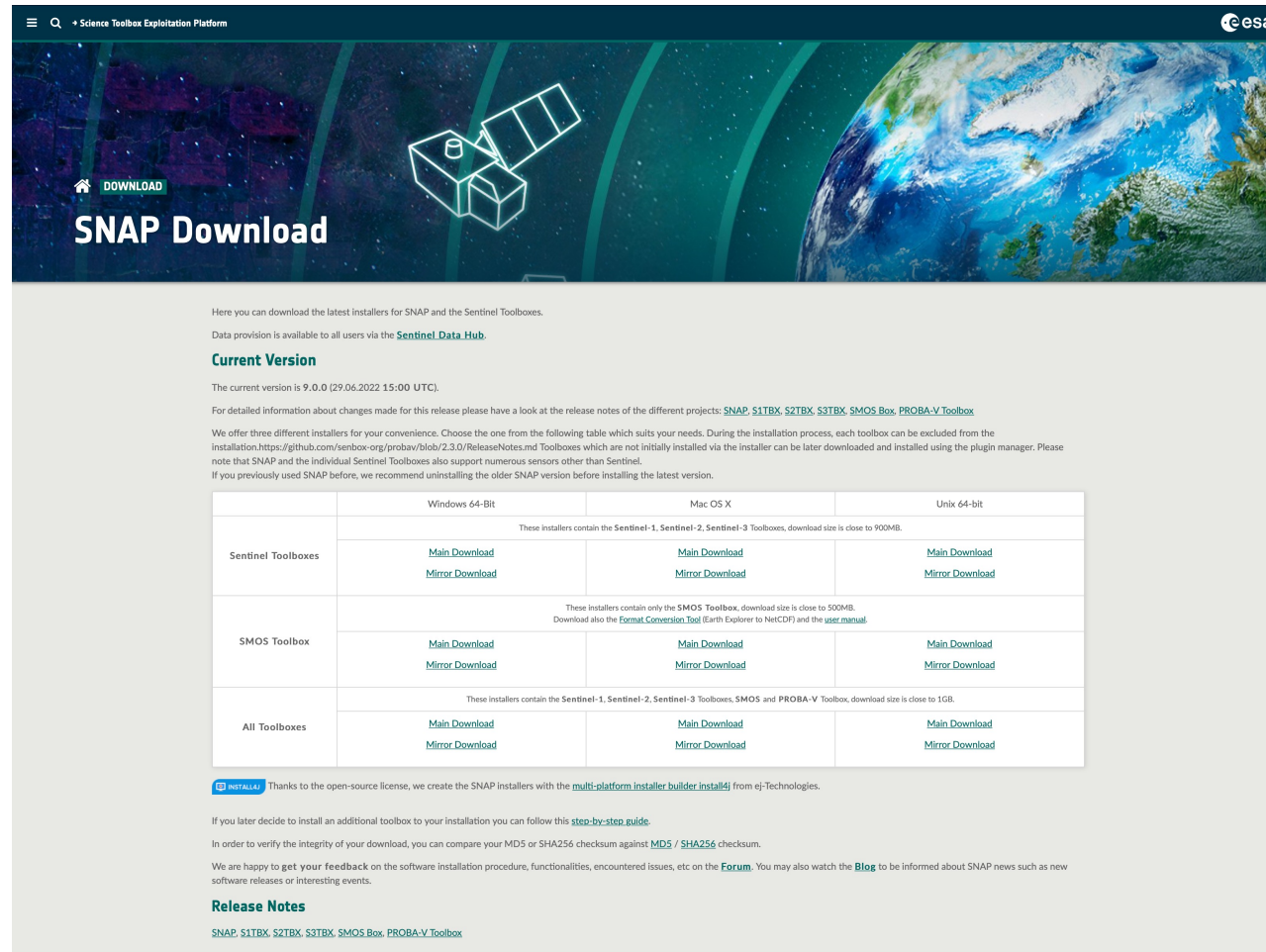
Optical Toolbox  
with future CHIME support



Microwave Toolbox  
with future CIMR support



## https://step.esa.int/main/download/snap-download/



Here you can download the latest installers for SNAP and the Sentinel Toolboxes.  
Data provision is available to all users via the [Sentinel Data Hub](#).


### Current Version

The current version is 9.0.0 (29.06.2022 15:00 UTC).

For detailed information about changes made for this release please have a look at the release notes of the different projects: [SNAP](#), [S1TBX](#), [S2TBX](#), [S3TBX](#), [SMOS Box](#), [PROBA-V Toolbox](#)

We offer three different installers for your convenience. Choose the one from the following table which suits your needs. During the installation process, each toolbox can be excluded from the installation. <https://github.com/senbox-org/probav/blob/2.3.0/ReleaseNotes.md> Toolboxes which are not initially installed via the installer can be later downloaded and installed using the plugin manager. Please note that SNAP and the individual Sentinel Toolboxes also support numerous sensors other than Sentinel.  
If you previously used SNAP before, we recommend uninstalling the older SNAP version before installing the latest version.

	Windows 64-Bit	Mac OS X	Unix 64-bit
	These installers contain the Sentinel-1, Sentinel-2, Sentinel-3 Toolboxes, download size is close to 900MB.		
Sentinel Toolboxes	<a href="#">Main Download</a>	<a href="#">Main Download</a>	<a href="#">Main Download</a>
	<a href="#">Mirror Download</a>	<a href="#">Mirror Download</a>	<a href="#">Mirror Download</a>
	These installers contain only the SMOS Toolbox, download size is close to 500MB. Download also the <a href="#">Format Conversion Tool</a> (Earth Explorer to NetCDF) and the <a href="#">user manual</a> .		
SMOS Toolbox	<a href="#">Main Download</a>	<a href="#">Main Download</a>	<a href="#">Main Download</a>
	<a href="#">Mirror Download</a>	<a href="#">Mirror Download</a>	<a href="#">Mirror Download</a>
	These installers contain the Sentinel-1, Sentinel-2, Sentinel-3 Toolboxes, SMOS and PROBA-V Toolbox, download size is close to 1GB.		
All Toolboxes	<a href="#">Main Download</a>	<a href="#">Main Download</a>	<a href="#">Main Download</a>
	<a href="#">Mirror Download</a>	<a href="#">Mirror Download</a>	<a href="#">Mirror Download</a>

 thanks to the open-source license, we create the SNAP installers with the [multi-platform installer builder install4j](#) from ej-Technologies.

If you later decide to install an additional toolbox to your installation you can follow this [step-by-step guide](#).

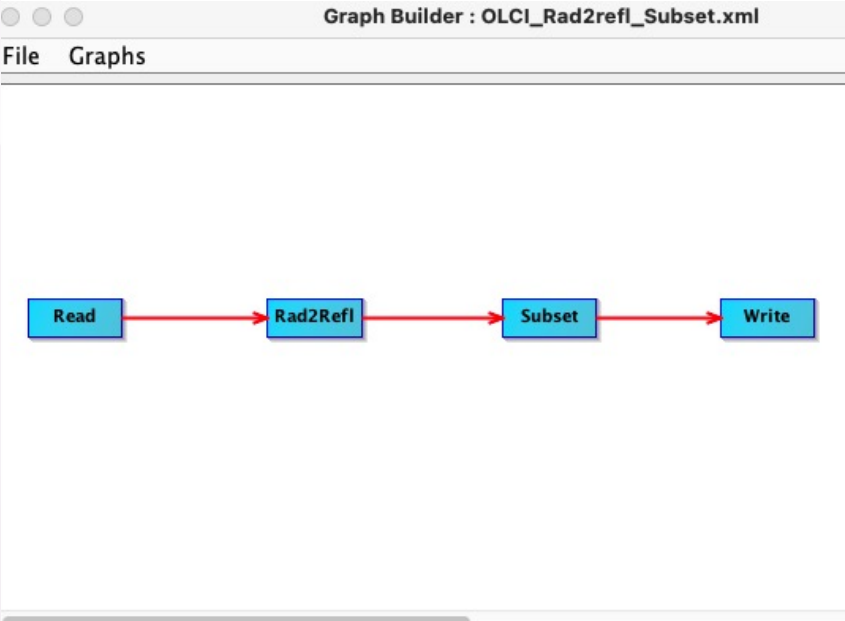
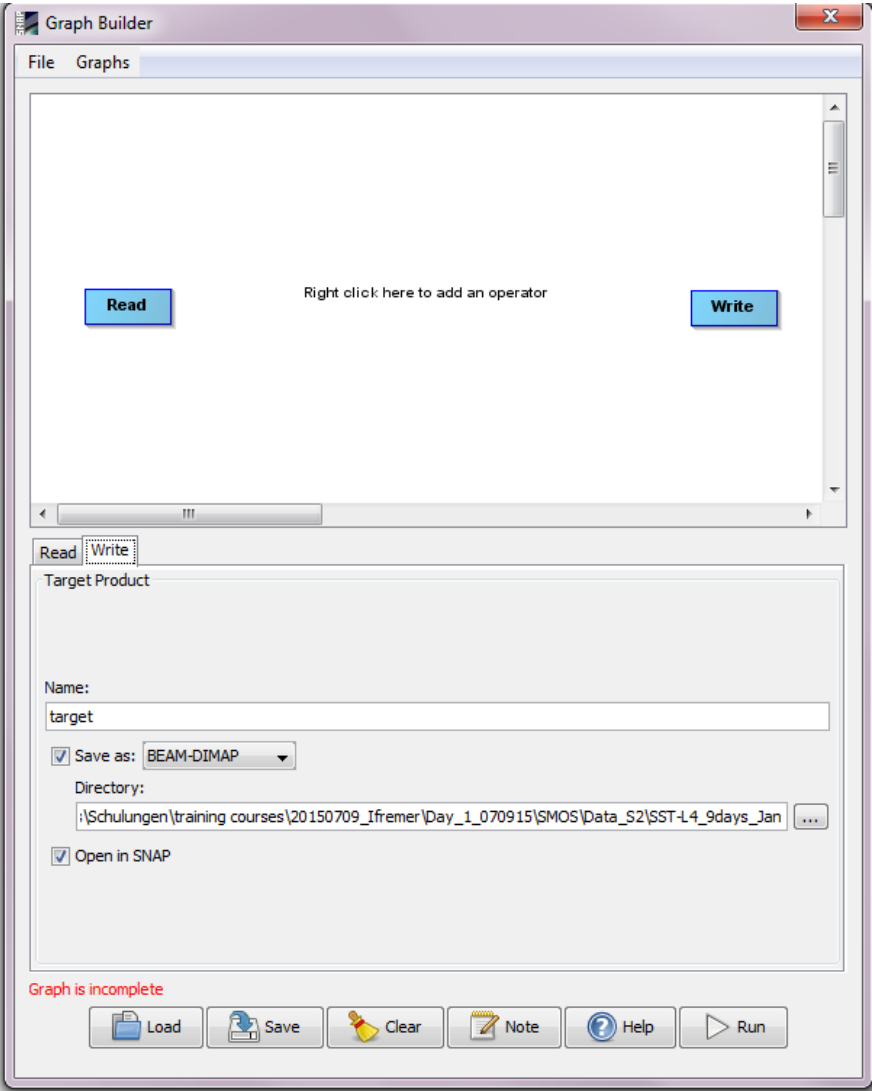
In order to verify the integrity of your download, you can compare your MD5 or SHA256 checksum against [MD5](#) / [SHA256](#) checksum.

We are happy to get your [feedback](#) on the software installation procedure, functionalities, encountered issues, etc on the [Forum](#). You may also watch the [Blog](#) to be informed about SNAP news such as new software releases or interesting events.

### Release Notes

[SNAP](#), [S1TBX](#), [S2TBX](#), [S3TBX](#), [SMOS Box](#), [PROBA-V Toolbox](#)

# Batch processing: Graph Builder



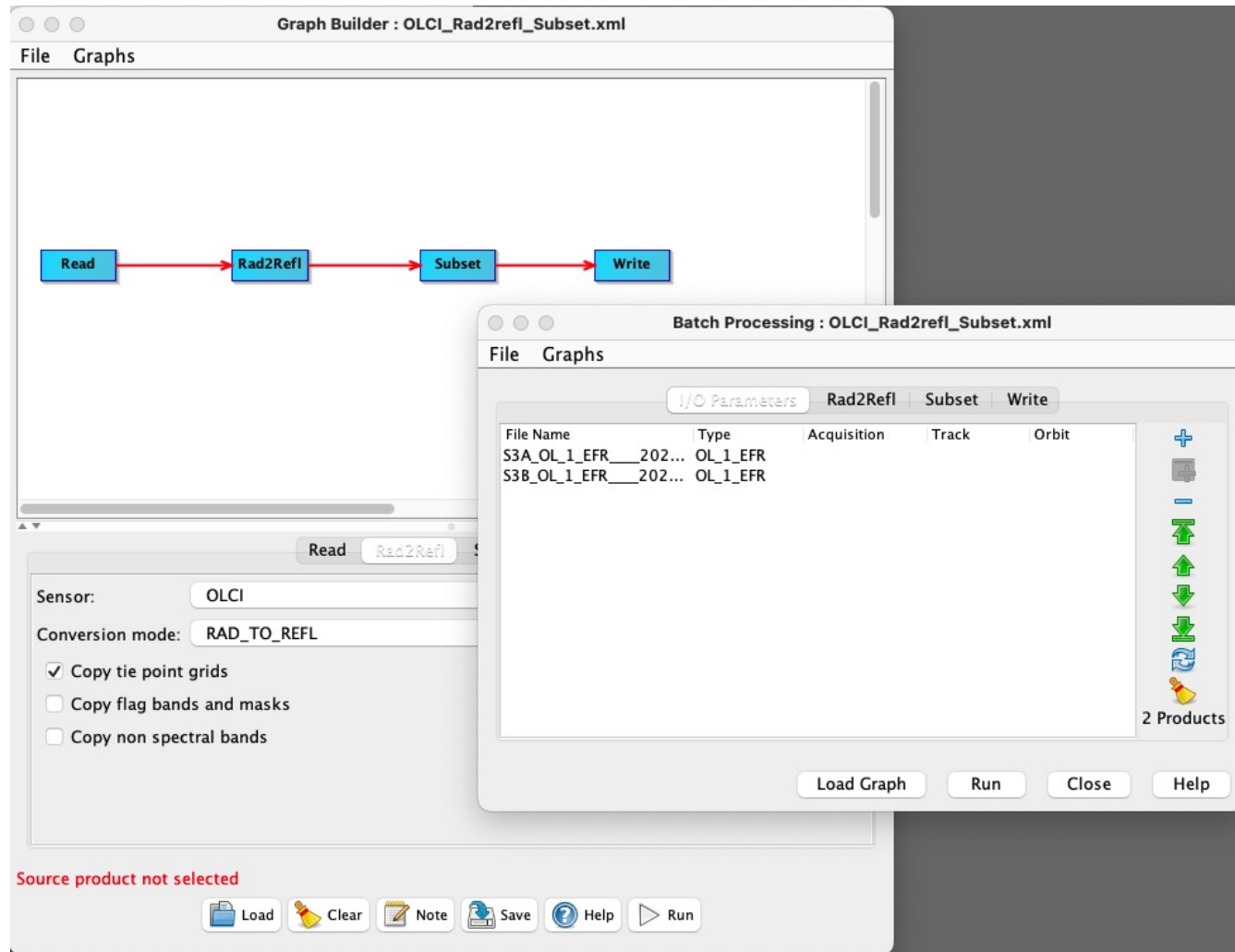
## Not all processors are available on Graph Builder

The screenshot shows the Graph XML editor window. The XML code defines a graph with four nodes: 'Read', 'Rad2Refl', 'Subset', and 'Write'. The 'Rad2Refl' node is highlighted in blue. The XML code is as follows:

```
<graph id="Graph">
  <version>1.0</version>
  <node id="Read">
    <operator>Read</operator>
    <sources />
    <parameters class="com.bc.ceres.binding.dom.XppDomElement">
      <useAdvancedOptions>>false</useAdvancedOptions>
      <copyMetadata>>true</copyMetadata>
      <bandNames />
      <pixelRegion>0,0,2147483647,2147483647</pixelRegion>
      <maskNames />
    </parameters>
  </node>
  <node id="Rad2Refl">
    <operator>Rad2Refl</operator>
    <sources>
      <sourceProduct refid="Read" />
    </sources>
    <parameters class="com.bc.ceres.binding.dom.XppDomElement">
      <sensor>OLCI</sensor>
      <conversionMode>RAD_TO_REFL</conversionMode>
      <copyTiePointGrids>true</copyTiePointGrids>
      <copyFlagBandsAndMasks>false</copyFlagBandsAndMasks>
      <copyNonSpectralBands>false</copyNonSpectralBands>
    </parameters>
  </node>
  <node id="Subset">
    <operator>Subset</operator>
    <sources>
      <sourceProduct refid="Rad2Refl" />
    </sources>
    <parameters class="com.bc.ceres.binding.dom.XppDomElement">
      <sourceBands />
      <tiePointGrids />
      <region>0,0,1,1</region>
      <referenceBand />
      <geoRegion />
      <subSamplingX>1</subSamplingX>
      <subSamplingY>1</subSamplingY>
      <fullSwath>false</fullSwath>
      <copyMetadata>true</copyMetadata>
    </parameters>
  </node>
  <node id="Write">
    <operator>Write</operator>
    <sources>
      <sourceProduct refid="Subset" />
    </sources>
  </node>
</graph>
```

# Batch processing: Graph Builder

www.eumetsat.int



- The Batch Processing tool available via the DAT allows you to execute a single reader/writer graph for a set of products.
- Select the Batch Processing tool from the Graphs menu and then press the "Load" button to browse for a previously saved graph.
- Next, add products in the IO tab by pressing the "Add" button or dragging and dropping a ProductSet or Products from the Project or Products views.
- Set the target folder where the output will be written to and then press "Run".





- Use SNAP's command-line tools:
  - From a command-line shell (v1)
  - From shell scripts (v2)
  - From Python, IDL, MatLab...using dedicated systems (v3)
- Many advantages
  - ✓ No intermediate files written, no I/O overhead
  - ✓ Reusability of processing chains
  - ✓ Simple and comprehensive operator configuration
  - ✓ Reusability of operator configurations



- Look into `$(SNAP-HOME)/bin` directory:  
gpt (.exe in Windows) → used to execute various SNAP operators and chain of operators
- ESA SNAP/SNAP command line help>gpt -h
- pconvert → used to convert product files into other data and image formats (quick-look generation)
- snappy-conf → SNAP application configuration launcher for snap-py

```

bin -- zsh -- 80x24
Last login: Wed Nov 16 09:37:56 on console
[(base) abruescas@eduroamnat108-136 ~ % cd ../../
[(base) abruescas@eduroamnat108-136 / % cd Applications/snap/bin
[(base) abruescas@eduroamnat108-136 bin % ls
C2RCC_S3A_OL_1_EFR____20200718T090634.data
C2RCC_S3A_OL_1_EFR____20200718T090634.dim
ESA SNAP Uninstaller.app
OLCI_gpt_v1.sh
Test1.sh
gpt
gpt.vmoptions
pconvert
pconvert.vmoptions
smos-grid-point-exporter
smos-grid-point-exporter.vmoptions
snap
snap-conf-optimiser.app
snap.app
snappy-conf
target.data
target.dim
(base) abruescas@eduroamnat108-136 bin %

```

```

bin -- zsh -- 84x55
usage is to provide an additional context to be used
from within the Velocity templates. See option -v.
Displays version and diagnostic information.

--diag
Operators:
Aatsr.SST                               Computes sea surface temperatu
re (SST) from (A)ATSR products.
AATSR.Ungrid                             Ungrids (A)ATSR L1B products a
nd extracts geolocation and pixel field of view data.
AdaptiveThresholding                     Detect ships using Constant Fa
lse Alarm Rate detector.
AddElevation                             Creates a DEM band
AddLandCover                             Creates a land cover band
ALOS-Deskewing                           Deskewing ALOS product
Apply-Orbit-File                          Apply orbit file
Arc.SST                                   Computes sea surface temperatu
re (SST) from (A)ATSR and SLSTR products.
ArviOp                                    Atmospherically Resistant Vege
tation Index belongs to a family of indices with built-in atmospheric corrections.
Azimuth-Shift-Estimation-ESD             Estimate azimuth offset for th
e whole image
AzimuthFilter                             Azimuth Filter
Back-Geocoding                            Bursts co-registration using o
rbit and DEM
BandMaths                                 Create a product with one or m
ore bands using mathematical expressions.
BandMerge                                  Allows copying raster data fro
m any number of source products to a specified 'master' product.
BandPassFilter                             Creates a basebanded SLC based
on a subband of 1/3 the original bandwidth
BandsDifferenceOp                           No description available.
BandSelect                                  Creates a new product with onl
y selected bands
BandsExtractorOp                           Creates a new product out of t
he source product containing only the indexes bands given

```

**Go to your SNAP folder and look  
Call gpt  
(if only one SNAP instance on your computer, just type gpt on the command line; on Mac you need to type the full path in your terminal)**

```

BiophysicalLandsat80p                    The 'Biophysical Processor' op
erator retrieves LAI from atmospherically corrected Sentinel-2 products
BiophysicalLandsat80p                    The 'Biophysical Processor' op
erator retrieves LAI from atmospherically corrected Landsat8 products
BiophysicalOp                             The 'Biophysical Processor' op
erator retrieves LAI from atmospherically corrected Sentinel-2 products
c2rcc.landsat8                             Performs atmospheric correctio
n and IOP retrieval with uncertainties on Landsat-8 L1 data products.
c2rcc.meris                                 Performs atmospheric correctio
n and IOP retrieval with uncertainties on MERIS L1b data products.
c2rcc.meris4                               Performs atmospheric correctio
n and IOP retrieval with uncertainties on MERIS L1b data products from the 4th repro

```



- Most important SNAP batch-mode tool
- Usage:
  - *gpt <op>|<graph-file> [options] [<source-file-1> <source-file-2> ...]*
- Which operator are available?
  - *gpt -h* or *./gpt -h*
- List of operators may vary depending on installed SNAP plug-ins

# Call gpt from command line

```
S2Resampling_subset_c2rcc_msi.xml X
Users > abruescas > Dropbox > Mac > Desktop > ESA_Academy_final > Hands_on > S2DATA > S2Resampling_subset_c2rcc_msi.xml
1  <!--> ./gpt /Users/yourpathtoxml/S2Resampling_subset_c2rcc_msi.xml -Ssource=/Users/yourpathtoinputimage/S2B_MS
2  -->
3  <graph id="Graph">
4    <version>1.0</version>
5    <node id="S2Resampling">
6      <operator>S2Resampling</operator>
7      <sources>
8        <sourceProduct>${source}</sourceProduct>
9      </sources>
10     <parameters>
11       <resolution>60</resolution>
12       <upsampling>Bilinear</upsampling>
13       <downsampling>Mean</downsampling>
14       <flagDownsampling>First</flagDownsampling>
15       <resampleOnPyramidLevels>true</resampleOnPyramidLevels>
16     </parameters>
17   </node>
18
19   <node id="Subset">
20     <operator>Subset</operator>
21     <sources>
22       <source>S2resampling</source>
```

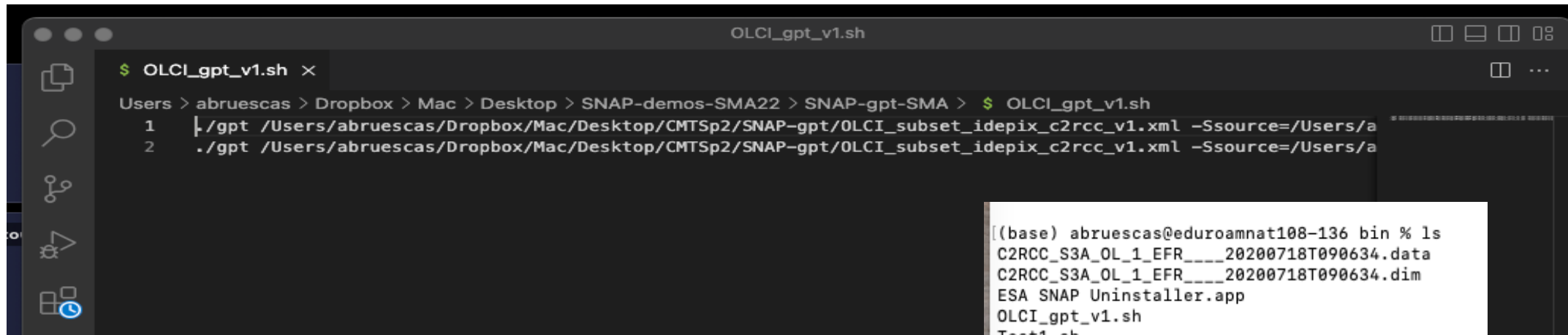
Mac/Linux

```
./gpt /Users/yourpathtoxml/S2Resampling_subset_c2rcc_msi.xml -
Ssource=/Users/yourpathtoinputimage/S2B_MSIL1C_20220818T095549_N0400_R122_T33UXS_20220818T104434.SAFE -t
/Users/yourpathtooutputimage/S2B_MSIL1C_20220818T095549_N0400_R122_T33UXS_20220818T104434_res_sub_c2rcc.dim
```

```
29     <fullSwath>false</fullSwath>
30     <tiePointGridNames/>
31     <copyMetadata>true</copyMetadata>
32   </parameters>
33 </node>
34
35 <node id="c2rcc">
36   <operator>c2rcc.msi</operator>
37   <sources>
38     <sourceProduct>Subset</sourceProduct>
39   </sources>
40   <parameters>
41     <validPixelExpression>B8 > 0</validPixelExpression>
42     <salinity>35.0</salinity>
43     <temperature>15.0</temperature>
```

# Batch mode: Using gpt XML with shell (v1)

- Running on several images (batch mode): use Excel or other spreadsheet/editor for compiling a list with the names of all products to be processed. Then save as .sh (linux, mac) or .bat (windows).
- For running the script, open a terminal and give the order: OLCI\_gpt\_v1.sh



```
OLCI_gpt_v1.sh
$ OLCI_gpt_v1.sh x
Users > abruecas > Dropbox > Mac > Desktop > SNAP-demos-SMA22 > SNAP-gpt-SMA > $ OLCI_gpt_v1.sh
1 | /gpt /Users/abruecas/Dropbox/Mac/Desktop/CMTSp2/SNAP-gpt/OLCI_subset_idepix_c2rcc_v1.xml -Ssource=/Users/a
2 | ./gpt /Users/abruecas/Dropbox/Mac/Desktop/CMTSp2/SNAP-gpt/OLCI_subset_idepix_c2rcc_v1.xml -Ssource=/Users/a
```

```
[(base) abruecas@eduroamnat108-136 bin % ls
C2RCC_S3A_OL_1_EFR____20200718T090634.data
C2RCC_S3A_OL_1_EFR____20200718T090634.dim
ESA SNAP Uninstaller.app
OLCI_gpt_v1.sh
Test1.sh
gpt
gpt.vmoptions
pconvert
pconvert.vmoptions
smos-grid-point-exporter
smos-grid-point-exporter.vmoptions
snap
snap-conf-optimiser.app
snap.app
snappy-conf
target.data
target.dim
```

# Batch mode: Using gpt XML with shell (v2)

## OLCI\_subset\_idepix\_c2rcc.xml

```
1 <graph id="Graph">
2   <version>1.0</version>
3   <node id="Read">
4     <operator>Read</operator>
5     <sources/>
6     <parameters>
7       <file>/Users/abruescas/Dropbox/Mac/Desktop/CMTSp2/SNAP-gpt/OLCI_gpt_v2/S3A_OL_1_EFR_____20200725T0925
8     </parameters>
9   </node>
10
11  <node id="Subset">
12    <operator>Subset</operator>
13    <sources>
14      <source>Read</source>
15    </sources>
16    <parameters class="com.bc.ceres.binding.dom.XppDomElement">
17      <sourceBands>0a01_radiance,0a02_radiance,0a03_radiance,0a04_radiance,0a05_radiance,0a06_radiance,0a07_
18      <geoRegion>POLYGON((15.836 54.278, 15.836 59.178, 21.103 59.178, 21.103 54.278, 15.836 54.278))</geoRe
19      <subSamplingX>1</subSamplingX>
20      <subSamplingY>1</subSamplingY>
21      <fullSwath>false</fullSwath>
22      <tiePointGridNames/>
23      <copyMetadata>true</copyMetadata>
24    </parameters>
25  </node>
26
27
28  <node id="Idepix.Olci">
29    <operator>Idepix.Olci</operator>
30    <sources>
31      <sourceProduct>Subset</sourceProduct>
32    </sources>
33    <parameters>
34      <outputSchillerMNValue>false</outputSchillerMNValue>
35      <computeMountainShadow>true</computeMountainShadow>
36      <mntShadowExtent>0.9</mntShadowExtent>
37      <computeCloudShadow>true</computeCloudShadow>
38      <outputCtp>false</outputCtp>
39      <computeCloudBuffer>true</computeCloudBuffer>
40      <cloudBufferWidth>2</cloudBufferWidth>
41      <useSrtmLandWaterMask>false</useSrtmLandWaterMask>
42    </parameters>
43  </node>
44
45  <node id="c2rcc">
46    <operator>c2rcc.olci</operator>
47    <sources>
48      <sourceProduct>Subset</sourceProduct>
49    </sources>
50    <parameters>
51      <validPixelExpression>!quality_flags.invalid and (!quality_flags.land or quality_flags.fresh_inland_
```

## OLCI\_gpt\_v2.sh

```
~/Dropbox/Mac/Desktop/SNAP-gpt/SNAP-gpt_v2/S3A_OL_1_EFR_____20200725T0925
'bash
folder=/yourdirectory/OLCI_gpt_v2/
-t=C2RCC_S3_
=.dim
ls -d -a /yourdirectory/*.SEN3)

product_basename=$(basename $i)
ac_date=$(echo $product_basename | cut -d '_' -f 8)
Output_pathname=$output_folder$nameStart$ac_date$nameEnd
/Applications/snap/bin/gpt OLCI_subset_idepix_c2rcc.xml -Source=$i -t $Output_pathname
```



- OLCI\_CMTS\_processing.py  
or
- Python-GPT.ipynb  
(OLCI\_subset\_idepix\_c2rcc.xml)

```
jupyter Python-GPT Last Checkpoint: Last Tuesday at 13:57 (unsaved changes) Logout
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)
In [ ]: 1 from os import system
        2 from os.path import exists, basename
        3 from glob import glob

In [ ]: 1 input_path = "/Users/abruecas/Dropbox/Mac/Desktop/CMTSp2/SNAP-gpt/"
        2 output_path = "/Users/abruecas/Dropbox/Mac/Desktop/CMTSp2/SNAP-gpt/Python-gpt/"

In [ ]: 1 ProcessGraph = '/Users/abruecas/Dropbox/Mac/Desktop/CMTSp2/SNAP-gpt/Python-gpt/OLCI_subset_idepix_c2rcc.xml'

In [ ]: 1 inputFile_list = glob(input_path + "/*.SEN3")
        2 inputFile_list

In [ ]: 1 GptProcessor = "/Applications/snap/bin/gpt"

In [ ]: 1 inputFile_list.sort()

In [ ]: 1 for file in inputFile_list:
        2     print(file)
        3     # S3A_OL_1_EFR____20200725T092516_20200725T092816_20200726T153823_0179_061_036_1980_MAR_0_NT_002.SEN3
        4     output_file = output_path + 'C2RCC_S3_' + basename(file)[16:24] + '.dim'
        5     print(output_file)
        6     exit(123)
        7     if exists(output_file):
        8         continue
        9     print(output_file)
        10    gpt_call = GptProcessor + " -c 12G " + ProcessGraph + " --sourceProduct=" + file + " -t " + output_file
        11    print(gpt_call)
        12    print("Processing Request...")
        13    system(gpt_call)
        14
```

SNAP / User Guide / Processing



## Bulk Processing with GPT



Created by Marco Peters  
Last updated: 2021-03-26 · 4 min read

### Bulk Processing with GPT

This little tutorial gives an introduction on bulk processing with the command shell on Windows and Unix systems. The provided scripts try to stay very generic in order to serve multiple processing requirements. However, not every edge case can be covered. The intention is to cover at least the main use cases. The scripts can probably be improved at multiple points but they can give you a starting point to write your own scripts. If you know improvements to the scripts or have questions regarding the usage of the script you are kindly invited to the [SNAP Forum](#). A general introduction to GPT and graphs can be found at [Creating a GPF Graph](#).

The four files mentioned below are attached for download.

- [processDataset.bat](#) (Windows)
- [processDataset.bash](#) (Unix)
- [resample\\_s2.xml](#)
- [resample\\_20m.properties](#)

#### Table of Contents

- [The Windows Script](#)
- [The Unix Script](#)
- [Description of the Scripts](#)
- [Known Limitations of the Scripts](#)
- [Example Usage](#)
  - [XML Graph File for S2 Resampling \(resample\\_s2.xml\)](#)
  - [Parameters File \(resample\\_20m.properties\)](#)
- [Helpful Links](#)

Add inline comment





- Bridge between SNAP/Java and Python
- Allows invocation of SNAP functionality from Python and vice versa
- Data provided by SNAP can be used with e.g., numpy, matplotlib, etc.
- Processing operators can be implemented in Python
- GPF processing graphs can be created and executed with Python if preferred over command line

```
from snappy import ProductIO
import numpy as np
import matplotlib.pyplot as plt

p = ProductIO.readProduct('snappy/testdata/MER_FRS_L1B_SUBSET.dim')
rad13 = p.getBand('radiance_13')
w = rad13.getRasterWidth()
h = rad13.getRasterHeight()
rad13_data = np.zeros(w * h, np.float32)
rad13.readPixels(0, 0, w, h, rad13_data)
p.dispose()
rad13_data.shape = h, w
imgplot = plt.imshow(rad13_data)
imgplot.write_png('radiance_13.png')
```

1. Product is read using SNAP
2. Getting a band
3. Reading the data of the band
4. Dispose product (clean memory)
5. Re-shape the data array
6. Show image
7. Write as png



```
from snappy import ProductIO
from snappy import GPF
from snappy import Hashmap

p = ProductIO.readProduct('eo/S2A_MSIL1C_20201229<...>T144625.zip')
parameters = Hashmap()
parameters.put('resolution', '10')
parameters.put('upsampling', 'Nearest')
parameters.put('downsampling', 'Mean')

target_product = GPF.createProduct('S2Resampling', parameters, p)

ProductIO.writeProduct(target_product, <'your/out/directory'>, 'BEAM-DIMAP')
```

1. Read the input product
2. Provide processing parameters
3. Call the operator 'Resample'
4. Write the processed product



- Based on SNAPPY
- Dedicated to create and run GPF graphs in a pythonic way
- Provided as conda package for Linux
- On Win and Mac a docker container needs to be used
- Currently still an external tool, but will be integrated into the SNAP ecosystem in the future



```
from snapista import Graph
from snapista import Operator, OperatorParams

g = Graph()

productPath = 'path to S1 file'

g.add_node(
    operator=Operator(
        "Read",
        formatName="SENTINEL-1",
        file=productPath,
    ),
    node_id="read",
)

g.add_node(
    operator=Operator("Apply-Orbit-File", continueOnFail="true"),
    node_id="apply-orbit-file",
    source="read",
)

g.add_node(
    operator=Operator(
        "Remove-GRD-Border-Noise", borderLimit="2000", trimThreshold="0.2"
    ),
    node_id="noise-removal",
    source="apply-orbit-file",
)
```

1. Add a read operation (node)
2. Add several operations more



```
g.add_node(  
    operator=Operator("Calibration"),  
    node_id="calibration",  
    source="noise-removal",  
)  
  
g.add_node(  
    operator=Operator("LinearToFromdB"),  
    node_id="linear",  
    source="calibration",  
)  
  
g.add_node(  
    operator=Operator(  
        "Terrain-Correction",  
        pixelSpacingInMeter="20.0",  
        demName="SRTM 3Sec",  
        mapProjection="AUTO:42001",  
    ),  
    node_id="terrain-correction",  
    source="linear",  
)  
  
g.add_node(  
    operator=Operator("Write", file='result.tif', formatName="GeoTIFF-BigTIFF"),  
    node_id="write",  
    source="terrain-correction",  
)  
  
g.run()
```

3. Add a write operation
4. Finally execute the graph



## More examples and how to configure snappy

- Configure Python to use the snappy  
<https://senbox.atlassian.net/wiki/x/BQAI Aw>
- How to use the SNAP API from Python  
<https://senbox.atlassian.net/wiki/x/CoAmAQ>
- What to consider when writing an Operator in Python  
<https://senbox.atlassian.net/wiki/x/AoCBAg>



## More examples and how to configure SNAPISTA

- **Installation Guide**

<https://snap-contrib.github.io/snapista/installation.html>

- **Getting Started**

<https://snap-contrib.github.io/snapista/gettingstarted.html>

- **More examples**

<https://snap-contrib.github.io/snapista/examples/operator.html>





**Thank you!**  
Questions are welcome.