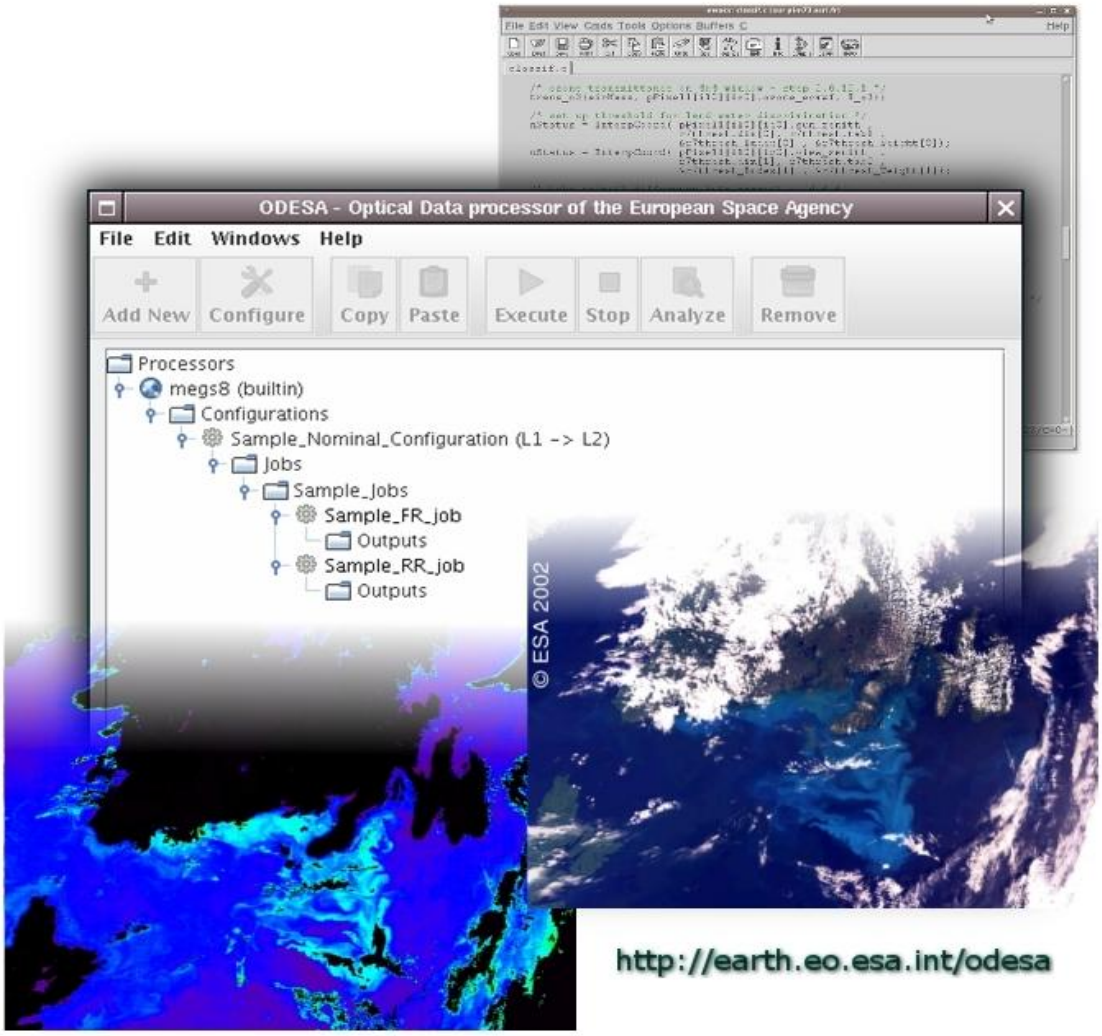
ODESA	**ODESA** **Software Distribution** **Quick Start Guide**	Ref: ODESA-ACR-QSG Date : March 5, 2012 Issue : version 1.2.4 Page : i

ACRI-ST

ODESA SOFTWARE DISTRIBUTION



Quick Start Guide

Reference:

Version 1.2.4

March 5, 2012

Change record

Issue	Date	Description	Change pages
1.0	16/06/2010	Quick Start Guide	Initial version
1.0.2	15/9/2010	Quick Start Guide, release	Sections 2.2, 3.3, 4.2.3 are updated
1.1	25/11/2010	Quick Start Guide, release	Sections 2.4, 2.5, 3.2 updated for new feature support.
1.2	02/12/2010	Quick Start Guide, release	Section 3.2.2 updated for new feature support.
1.2.1	07/01/2011	Quick Start Guide, release	Section 2.4 updated for new MEGS version.
1.2.2	03/02/2011	Quick Start Guide	Section 2.4 & 2.5 updated
1.2.4	05/03/2012	Quick Start Guide, release	Section 4 updated

Distribution List

Organisation	To	Nb. of copies
ESA	Philippe Goryl, Henri Laur	1
	ODESA user community	

Table of content

List of Figures

1 INTRODUCTION

1.1 Overview

This document is the version 1.2.4 of the Quick Start Guide (QSG) for the ODESA CFI Software Distribution. It has been prepared by the ODESA team at ACRI-ST. It describes the functionalities and characteristics of the application.

In case of any further enquiry, please send your request to:

service@odesa-info.eu

In case you would like to access other ODESA functionality, go and visit:

http://earth.eo.esa.int/odesa/

1.2 ODESA Background

ODESA is the answer to the requirements expressed by the MERIS Quality Working Group (QWG) and by the MERIS Validation Team (MVT), and more widely by the international user community of MERIS.

ODESA provides the users with a complete level 2 processing environment for the MERIS instrument.

ODESA supplies the user community with the MERIS Ground Segment development platform MEGS®, including software, embedded in an efficient framework for testing and for validation activities.

These validation facilities include match-up processing & analysis, data set selection & analysis, level 3 products generation & analysis and the possibility to perform remote processing, e.g. for testing purpose and for validation activities requiring large amounts of data.

ODESA is structured around the following modules:

MERIS on-line processing	Access MERIS data from remote processing facility available to qualified processors.
Software distribution	Download the MERIS level 2 processor (MEGS®) and its operation environment
Analysis tools	Download and install the ODESA analysis tools, including the BEAM toolbox
Validation & qualification	Validate your algorithm and get it qualified to access the MERIS on-line processing
Forum	All you want to discuss about ODESA and MERIS

1.3 License terms and conditions

The ODESA CFI software is provided under the following terms and conditions:

1. The software is the set of computer programs ODESA MEGS® packages regardless of the form in which you may subsequently use it, and regardless of any modification which you may make to it.
2. The license is a non-exclusive, non-transferable, limited license without fees, provided that
   - i. You shall not disclose or copy to a third party the software or any part thereof.
   - ii. You shall not remove or alter any proprietary legends or notices contained in the Software
   - iii. You commit to provide back any algorithm adaptation or modification should you believe it would benefit the user community and in such a case, submit it for its inclusion in the maintained version of ODESA MEGS®:
     - a. You agree that such modification or new algorithm shall undergo a validation/qualification process prior to its approval as an ESA supported algorithm
     - b. ACRI-ST and the European Space Agency shall be entitled to use and transfer to third parties any such qualified modification free of charge.
     - c. Such ESA supported algorithm shall be identified as such in the software with due acknowledgement to you.
   - iv. Neither the European Space Agency nor ACRI-ST shall be held liable for the consequences of any use of the software.
   - v. You agree to defend and indemnify the European Space Agency from and against all claims, proceedings, damages, costs and expenses arising from any infringement of third party rights resulting from the wrongful disclosure or use of the software contrary to the above terms.

1.4 Third Party Libraries

The ODESA software distribution makes use of a number of third party libraries, which are covered by different licenses.

1.4.1 The GDAL library

The core software library is covered by the X11/MIT License, however some of the modules used by the library come under slightly differing licenses. These are listed and included in the GDAL_LICENSE.txt file located in the *doc* folder of the ODESA distribution.

(http://trac.osgeo.org/gdal/wiki/FAQGeneral#WhatlicensedoesGDALOGRuse)

1.4.2 Libcsv

The libcsv library is covered by the LGPL version 2.1, which can also be found in the *doc* folder in the ODESA distribution.

1.4.3 Zlib

The zlib library has its own license, which can be found in the ZLIB_LICENSE.txt file in the *doc* folder.

(http://www.gzip.org/zlib/zlib_license.html)

1.4.4 NetCDF

The netCDF library comes under its own license, which can be found in the NetCDF_LICENSE.txt file in the *doc* folder.

(http://www.unidata.ucar.edu/software/netcdf/copyright.html)

1.4.5 HDF5

The HDF5 library comes under its own license, which can be found in the HDF5_LICENSE.TXT file in the *doc* folder.

(http://www.hdfgroup.org/HDF5/doc/Copyright.html)

1.4.6 libunwind

The libunwind library is covered by an X11 license, which can be found in the LIBUNWIND_LICENSE.txt in the *doc* folder.

1.5 Acronyms

The list of acronyms corresponds to names that are not necessarily cited in the text but are related to the field of interest of ODESA.

AATSR -----------Advanced Along Track Scanning Radiometer
AC-----------------Atmospheric Correction
AOT---------------Aerosol Optical Thickness
ATBD -------------Algorithm Theoretical Baseline Document
ADF ---------------Auxiliary Data Files
BEAM-------------Basic ERS and Envisat (A)ATSR and MERIS Toolbox
BOUSSOLE------Bouée pour l'acquisition de Séries Optiques à Long Terme
BPAC -------------Bright Pixel Atmospheric Correction
CDOM------------Coloured Dissolved Organic Matter
CEOS -------------The Committee on Earth Observation Satellites
CF------------------Climate and Forecast
CFI-----------------Customer-Furnished Item
CNES -------------Centre National d'Etudes Spatiales
COM --------------Coloured Organic Matter
COTS -------------Commercial Off-The-Shelf software
CTP----------------Cloud Top Pressure
CZCS--------------Coastal Zone Color Scanner
DPM --------------Detailed Processing Model
DUE---------------Data User Element of the ESA Earth Observation Envelope Programme II
ECSS --------------European Cooperation for Space Standardization
EO -----------------Earth Observation
EOSDIS-----------Earth Observing System Data and Information System
ESA----------------European Space Agency
ESL ----------------Expert Support Laboratories
EU -----------------European Union
EUMETSAT -----European Organisation for the Exploitation of Meteorological Satellites
GAC---------------Global Area Coverage (4 km sub-sampled SeaWiFS product)
GMES-------------Global Monitoring for Environment and Security
GNOME----------GNU Network Object Model Environment
GOCI--------------Geostationary Ocean Color Imager
GOOS-------------Global Ocean Observing System
GSC ---------------GMES Space Component
GSC CQC --------GMES Space Component Coordinated Quality Control
GSM---------------Garver, Siegel, Maritorena
GUI----------------Graphical User Interface
Hermes-----------Data portal for Ocean Colour Data Users http://hermes.acri.fr
IOCCG -----------International Ocean Colour Coordinating Group
IOCCP------------International Ocean Carbon Coordination Project
IODD -------------Input Output Data Definition

IOPs---------------Inherent Optical Properties
IWV ---------------Integrated Water Vapour
JDK----------------Java Development Kit
JRE ----------------Java Runtime Environment
JVM ---------------Java Virtual Machine
KDE ---------------K Desktop Environment
LOV ---------------Laboratoire d'Océanographie de Villefranche-sur-mer
LUT ---------------Look-Up Table
L1---------------- Level 1 (product)
L2---------------- Level 2 (product)
MERMAID ------MERIS Match-up In-situ Database
MERIS ------------Medium Resolution Imaging Spectrometer
MOBY ------------Marine Optical Buoy
MODIS -----------Moderate Resolution Imaging Spectrometer
MVT---------------MERIS Validation Team
N/A----------------Not Applicable
NASA-------------National Aeronautics and Space Administration
netCDF -----------Network Common Data Format
NIR----------------Near Infrared
NOMAD ---------NASA Bio-Optical Marine Algorithm Data Set
NOAA ------------National Oceanic and Atmospheric Administration
NRT ---------------Near-Real Time
OCL ---------------Open Control Loop
OCR ---------------Ocean Colour Radiance
OCTS -------------Ocean Color and Temperature Scanner
ODESA -----------Optical Data Processor of the European Space Agency
OLCI --------------Ocean and Land Colour Imager
PDF----------------Adobe Portable Document Format
POC ---------------Particulate Organic Carbon
POLDER ---------Polarization and Directionality of the Earth's Reflectances
PP-----------------Primary Production
QSG-------------- Quick Start Guide
QWG--------------Quality Working Group
RMS ---------------Root Mean Square
RT -----------------Radiative Transfer
S-3 -----------------Sentinel-3
SeaBASS----------SeaWiFS Bio-Optical Archive and Storage System
SeaDAS-----------SeaWiFS Data Analysis System
SeaWiFS----------Sea-Viewing Wide Field of View Sensor
SIMBIOS ---------Sensor Intercomparison for Marine Biological and Interdisciplinary Ocean Studies
SLSTR-------------Sea and Land Surface Temperature Radiometer
SPM --------------Suspended Particulate Matter

TOA---------------Top Of Atmosphere
YAML ------------YAML Ain't Markup Language

2 INSTALLATION

2.1 Minimum Requirements

The ODESA Software Distribution essentially consists of two parts: The graphical user interface, and the MEGS® software. The GUI is a Java application, and MEGS® is a C application. Both the GUI and MEGS® will run correctly on both 32 bit and 64 bit Linux machines, provided that MEGS® is compiled in 32 bit mode (the only supported and validated mode).

Thus the minimum requirements for execution are:

- Linux kernel 2.4+ (32 or 64 bit)
- Java 1.6 (bundled with the ODESA source distribution)
- X Windowing System

For MEGS® development the following elements are additionally required:

- GCC 3.3.x+
- 32 bit development system libraries (if on a 64 bit machine)

It is also recommended to install the BEAM suite from Brockmann Consult (http://www.brockmann-consult.de/cms/web/beam/welcome) to view and analyze the products generated with the ODESA software distributions.

ODESA has been tested to work with BEAM 4.7 but should work equally well with older and newer versions. It is recommended to use the most up to date versions as they have better support for the output formats used by ODESA. (NOTE: Some file types that ODESA can process are not handled properly by BEAM yet, such as MERIS FSG files.)

2.2 Installation Procedure

The ODESA software distribution is provided as a tarball. All the necessary files (GUI, MEGS®, and relative ADFs) are provided for the processing of MERIS level 1 products.

The ODESA GUI requires the Java 6 JRE to run. To use a particular installation, the preferred method is to have the JAVA_HOME environment variable pointing to the directory where the JRE is installed, otherwise ODESA will use the version bundled with the distribution package.

To install the ODESA software distribution the tarball can be unzipped wherever desired, for example in */usr/local* .

If using GNU tar such as is found on GNU/Linux distribution, the command can be simply run as:

```
tar zxf odesa-$version.tar.gz
```

If this doesn't work (e.g. older or different versions of "tar") then one can fallback to the older method using the following formatted command:

```
gunzip -c odesa-$version.tar.gz | tar xf -
```

This will create the *odesa-$version* (eg */usr/local/odesa-1.2.4*) subdirectory, where all the necessary files will be found. It is then recommended to either:

1) Add the bin (eg */usr/local/odesa-$version/bin*) subdirectory to the PATH environment variable.
2) Create a softlink to the *odesa* script in the *bin* directory (eg */usr/local/odesa-$version/bin/odesa*) to one of the directories in the PATH variable (eg. */usr/local/bin*).

ODESA can then be launched by using the full path (eg */usr/local/odesa-$version/bin/odesa*) or simply typing '*odesa*' if the PATH variable is configured correctly.
Note: ODESA software creates in the user HOME directory a folder called ".odesa" to keep the user preferences.

2.3 Performance

Performance of the MEGS© processor depends of the size of the input product and the machine used. The table below provides some indicators (space/time).

	Size of level 1 - N1 (mega bytes)	Size of level 2 - N1 – (mega bytes)	Size of level2 for one intermediate variable (float) – NetCDF 3 (mega bytes)	Time of execution (minute)
RR (1121x15025, full swath)	536	601	64	37
RR (1121x97, provided with the software)	3.5	3.9	0.4	0.75
FR (2241x321, provided with the software)	23	25	2.7	6

Table 1: Some indicators of resources (space/time) required for execution of MEGS© (using Intel(R) Xeon(R) CPU X5460 @ 3.16GHz)

2.4 Known Issues

Some features used by ODESA are not functional or limited in certain environments. These are dependent on the window manager and desktop environment.

- Opening files with associated applications: Most desktop environments such as GNOME, KDE have a mechanism in place to open files with their associated programs; ODESA will attempt to use this if no specific view has been defined in the preferences.

- Window borders and resizing problems: Different window managers have different behaviours with certain window types. Some window managers may draw dialog windows without full borders (with no title bar for example), and sometimes their resizing is restricted.
- Although the default setting for ODESA is to create files read-and-writeable by everyone, and it is possible to point two ODESA users working directories to the same place, it is not recommended as there is no synchronization mechanism in place. To change the default file permission setting for ODESA one can modify the "*odesa*" script in the "bin" directory of the installation, changing the line near the top that reads "unmask 000" to something more appropriate, for example "umask 022" (owner only can write, rest of the world can read) or "umask 002" (owner and group can read and write, others can read). One must also take care when compiling MEGS as the default is to give all users the same permissions, this can be worked around by modifying the "_Install.sh" script in the particular processors' source directories at the lines reading "mode=XXX".

Beam 4.7.1 does not fully support yet the netCDF file produced by ODESA:

- netCDF signed byte are handled as unsigned byte (fixed in Beam 4.8).
- Flags stored in netCDF are not yet interpreted. (fixed in Beam 4.8)
- FRG, RRG and FSG level 2 N1 files are not supported.

The MEGS® reference version delivered with this version of ODESA is MEGS 8.0, which corresponds to MERIS 3rd reprocessing.

2.5 Bug Reports

In the case you would encounter a problem with the ODESA software, you should submit a bug report. This can be done by sending an email to service@odesa-info.eu. To aid in the resolving of any such problems, it is ideal for a bug report to include any information which may help, such as context, steps to reproduce the problem, or files which caused problems. Also the ODESA software keeps logs of its execution. These log files, named "odesa.log" (or "odesa.log.#" for older files) can be found in your ODESA configuration directory, the ".odesa" directory found in your home directory.

2.6 First Time Execution

You are ready to start using ODESA; just type:

odesa

When running ODESA for the first time, you will be presented with a dialog box informing you so, which will give you the opportunity to select a working directory. This will be the directory where all the configurations, jobs, and generated products will be saved.

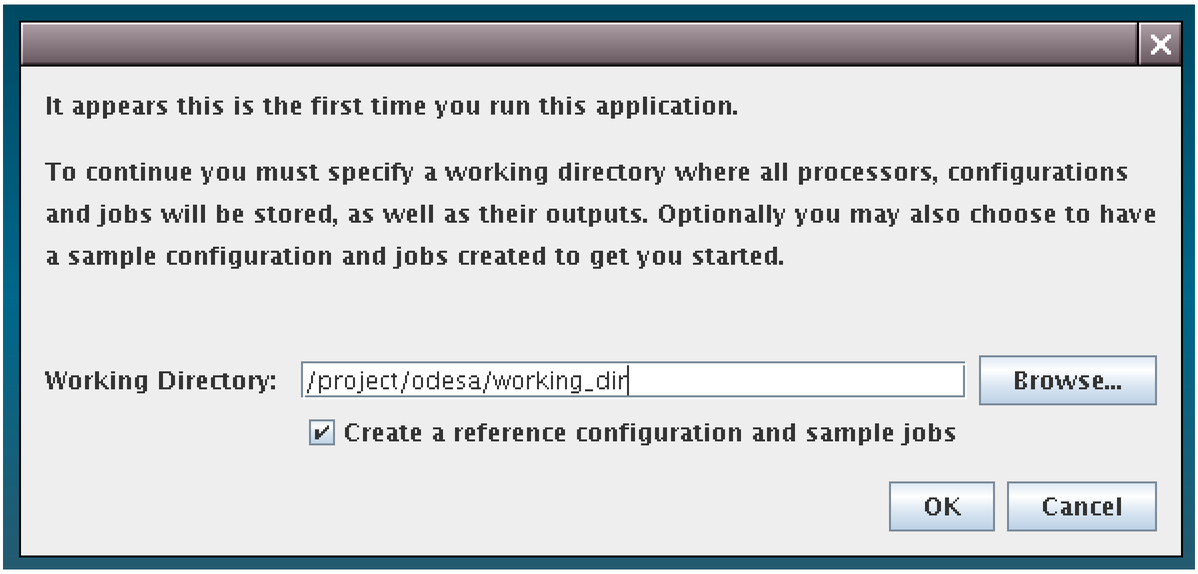


Figure 1: First execution window

Select the desired directory, either by typing in the path directly or using the browse button. If the selected working directory doesn't exist, ODESA will ask for confirmation in creating the directory.

Optionally (and it is strongly suggested to first time users) one may additionally create a couple of example jobs based on a reference configuration (corresponding to the nominal/standard MEGS® configuration) which may greatly help with understanding the workflow in ODESA.

For that purpose one MERIS RR level 1 product, one MERIS FR level 1 product, and one sample CSV file are delivered within the ODESA delivery package, found in the *resources/samples/images/* subdirectory:

MER_FR__1CNACR20080626_093818_000000992069_00437_33056_0000.N1
MER_RR__1CPACR20080626_093754_000002222069_00437_33056_0000.N1

sample.csv

Note: the FR sample product has been truncated and as such, is not a standard product in terms of size (2241x321 instead of 2241x2241).

3 GENERAL OPERATION

3.1 Main Window

After the initial working directory has been selected, and for each subsequent execution of ODESA, the user will be presented with the main screen:

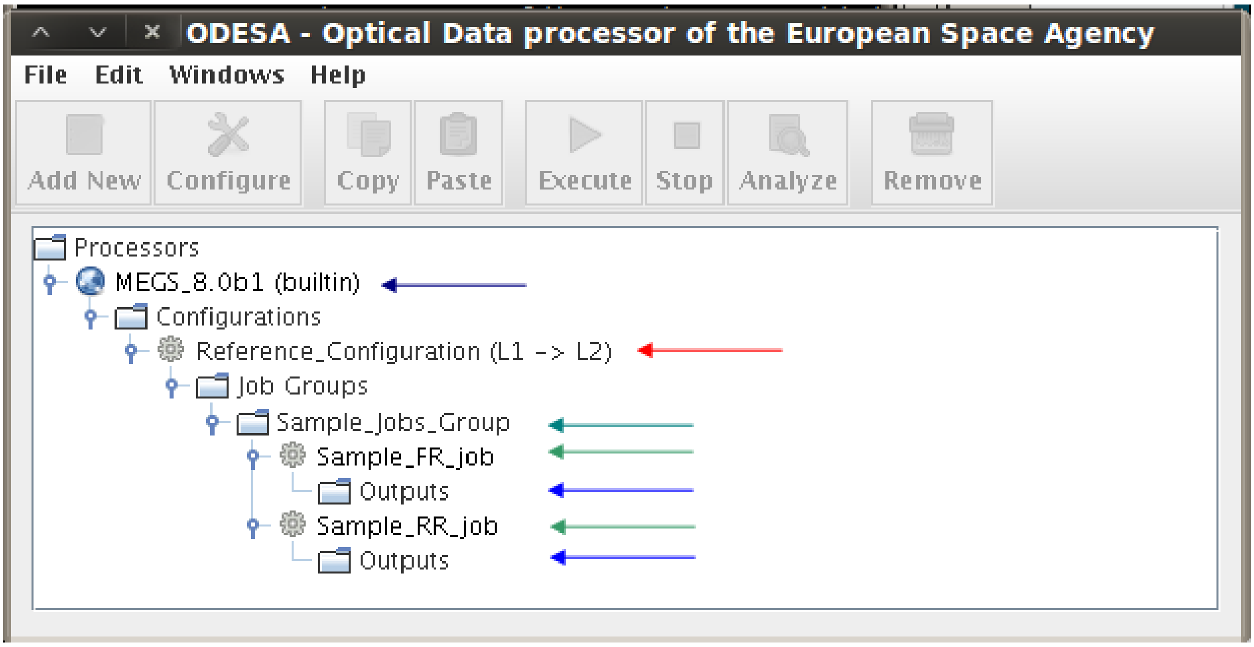


Figure 2: Main Window

After the first execution, if the creation of sample jobs based on the reference configuration was selected, the main screen will look as above. Visible are all the elements of an ODESA MERIS level 1 to level 2 processing. The logical hierarchy goes as follows:

Processor →	Configuration →	Job Group →	Job →	Outputs

Using the examples provided, it would correspond to:

Megs8 (built-in)→	Reference_Configuration	Sample_Jobs_Group→	Sample_FR_Job→	Outputs

Megs8 (built-in)→	Reference_Configuration	Sample_Jobs_Group→	Sample_RR_Job→	Outputs

- **Processor:** represents an instance of the processor. By default only the supplied MEGS® processor is available. In further releases of the software, it will be possible to create copies which can then be modified to implement new algorithms or other modifications.
- **Configuration:** is a general set of options with which to carry out MERIS product processing: Output formats (N1 and NetCDF) and the set of auxiliary data files to be used are defined at this level. Breakpoints are defined at this level too. Breakpoints also

include the alternative algorithms outputs (non ESA standard products) selected by the user.

- **Job**: A job is defined by a level 1 product to use as input, the desired region of interest over which to perform the processing, as well as its resulting outputs. A job is always defined within a job group.
- **Job Group**: is a logical grouping of jobs to facilitate the organization of jobs with respect to arbitrary parameters such as region of interest or a given thematic type.

> This logical hierarchy corresponds to physical folders with same names as the nodes of the tree of figure 2.
>
> By this way, outputs are easily available for external usages.

3.2 Preparing/Adding Jobs

3.2.1 Adding the Job

A job can only be defined within a job group, thus if there are none, the creation of a new one is required. This is done by selecting the "Job Groups" node under the desired configuration, and either clicking the "Add New" toolbar button, or right clicking on the node and selecting "Add new job group…" from the popup menu. ODESA will prompt the user for the group's name.

This functionality is also used when the user wants to partition a set of jobs according to its own criteria (e.g. geographic).

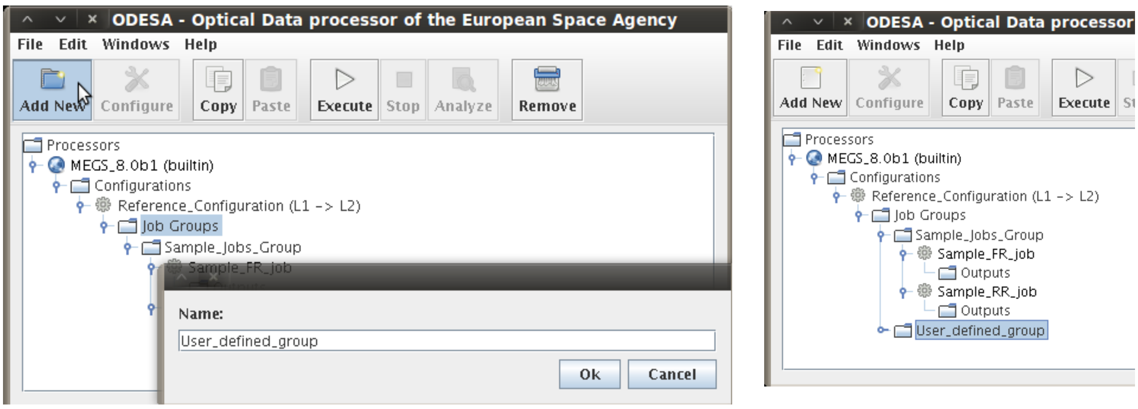


Figure 3: Creation of a job group

> Right clicking on a node makes appear a contextual popup menu with available functions.
>
> The functions are also available on the toolbar

Copy/paste functions are also available through Ctrl-C Ctrl-V short keys.

Selecting the newly created job group, it is then possible to create jobs, again either with the "Add New" toolbar button or the "Add new job…" entry in the popup menu.

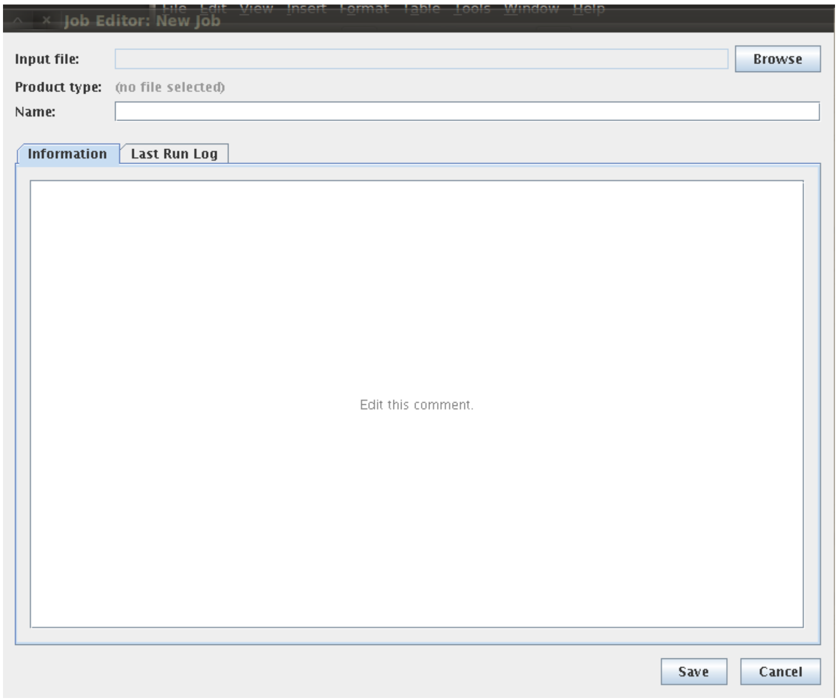


Figure 4: New Job

A job is defined by an input file, and a set of parameters defining the required outputs. The ODESA software can process two types of data, the standard MERIS L1 Envisat products, and data laid out in a CSV format (see CSV input format details in Annex 2). The input file is selected by using the browse button, and selecting an adequate file with the file chooser dialog. Once a file is selected, if no "id" has been given, the software will attempt to guess a useful default. For standard N1 product that will be the product's start date, for CSV files, the name of the CSV file itself.

Note: the input files are not copied into the working directory, but are accessed directly from their supplied path, they must therefore be accessible at the time a job is executed.

3.2.2 Job Configuration

3.2.2.1 Standard MERIS L1 Product

When the selected input file is a standard MERIS level 1 product, the available configuration options are the output formats and the region of interest. The output formats can be configured in the “Outputs” tab, which is selected by default once a valid input file as shown in Figure 5.

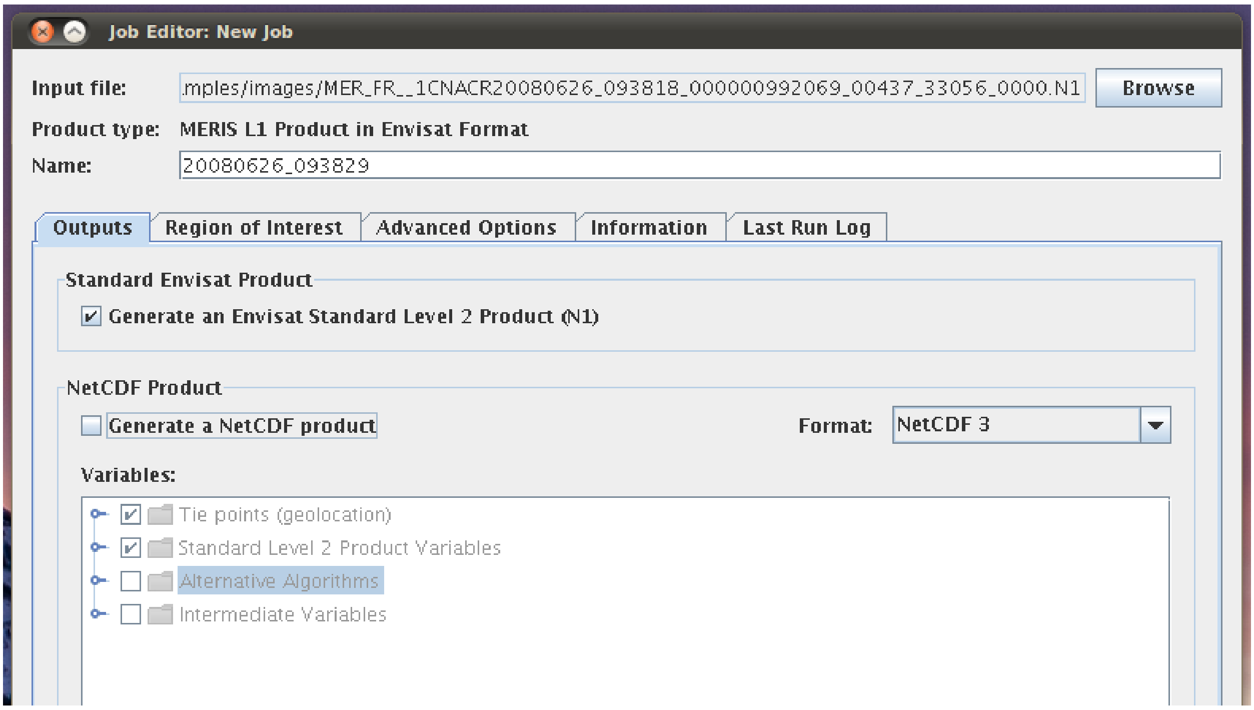


Figure 5: New Job after Image Selection

In this tab, two alternate output file formats can be selected, the standard N1 Level 2 product, and the NetCDF format.

- The N1 format is the standard nominal ESA ENVISAT MERIS Level 2 product format.

- The NetCDF format on the other hand is of particular interest because of its versatility. It is possible to configure the product variables that are included in the NetCDF output file. These are presented in a hierarchical list roughly grouped by thematic or other dominant characteristic. There is also a grouping for alternative algorithms which, when checked, will activate said algorithms. A list and description of all the variables available in the delivered ODESA software distribution can be found in Section 5. Additionally it is possible to select the actual NetCDF format to be used: either the NetCDF-3 format (the 64-bit offsets mode is used to accommodate large datasets), or the newer NetCDF-4 format (with an option for using NetCDF-4's internal GZIP compression method to reduce file size). When using NetCDF-4 files, they are created using the "classic model" for compatibility reasons, so it is always possible to translate one back to the NetCDF-3 format.

Note: all the ODESA outputs are supported by BEAM.

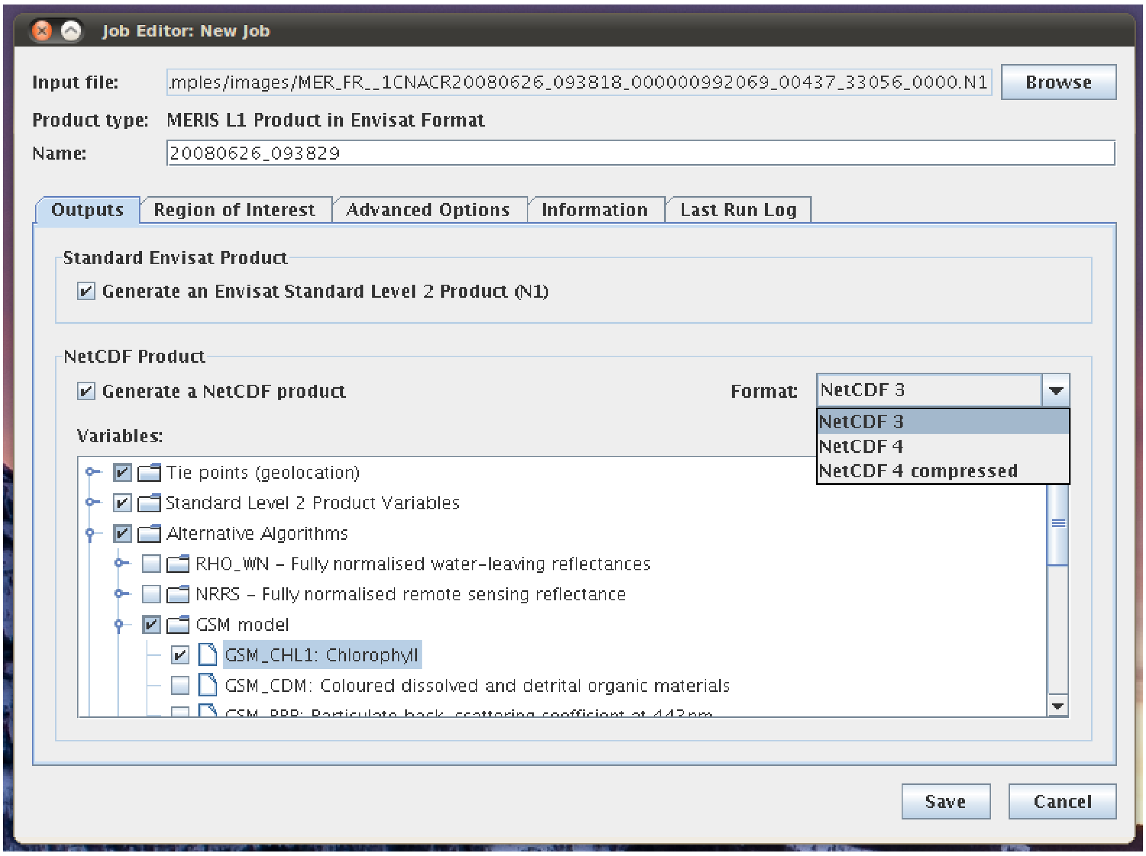


Figure 6: MERIS Output Products

When creating a new job the default configurations options are set to correspond to the standard nominal MERIS configuration: only the standard N1 formatted level 2 product will be generated.

Upon selecting the input file, its bounding box is also determined, and used by default, resulting in the whole file being processed if a smaller bounding box is not defined.

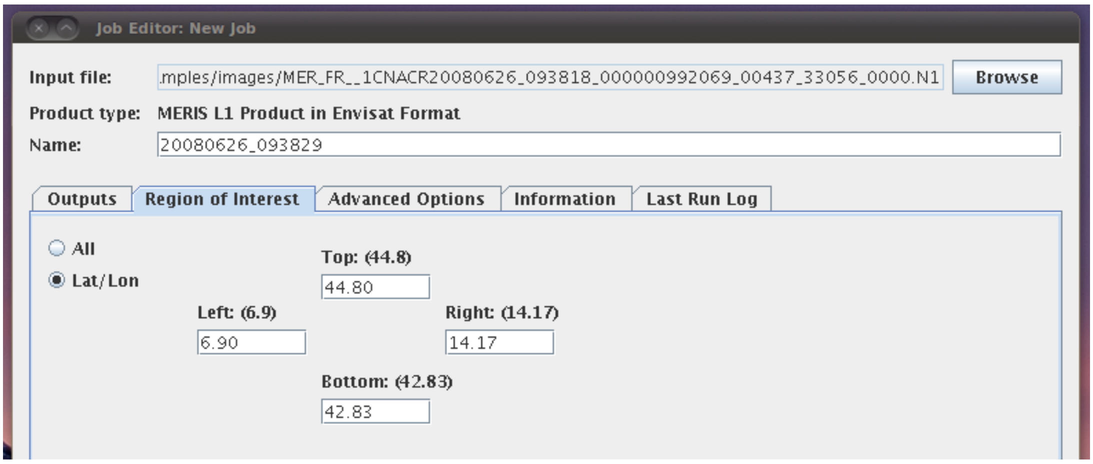


Figure 7: Output Products

The bounding box has no effect on the size of the level2 product stored in the N1 ESA format (the level2 will have the same pixel size as the level1 but output is computed only inside the bounding box, the rest of the file is filled with no-data). This limitation does not exist for products in netCDF format whose size matches the area selected by the user.

3.2.2.2 Starting from a text file (L1 CSV Job)

ODESA allows to process Level 1 data stored in text file, e.g. an extraction over a MERMAID matchup, or simulated data. The input file must be in CSV format, as detailed in Annex 2.

The corresponding output configuration tab differs slightly from a standard job, with two possible modes:

- Either a configurable CSV format, with total flexibility in the choice of output fields.
- Or a standard MERMAID CSV files, with fixed output field as defined in the MERMAID project (see http://hermes.acri.fr/mermaid).

Figure 8 is an example of the customized configuration mode.

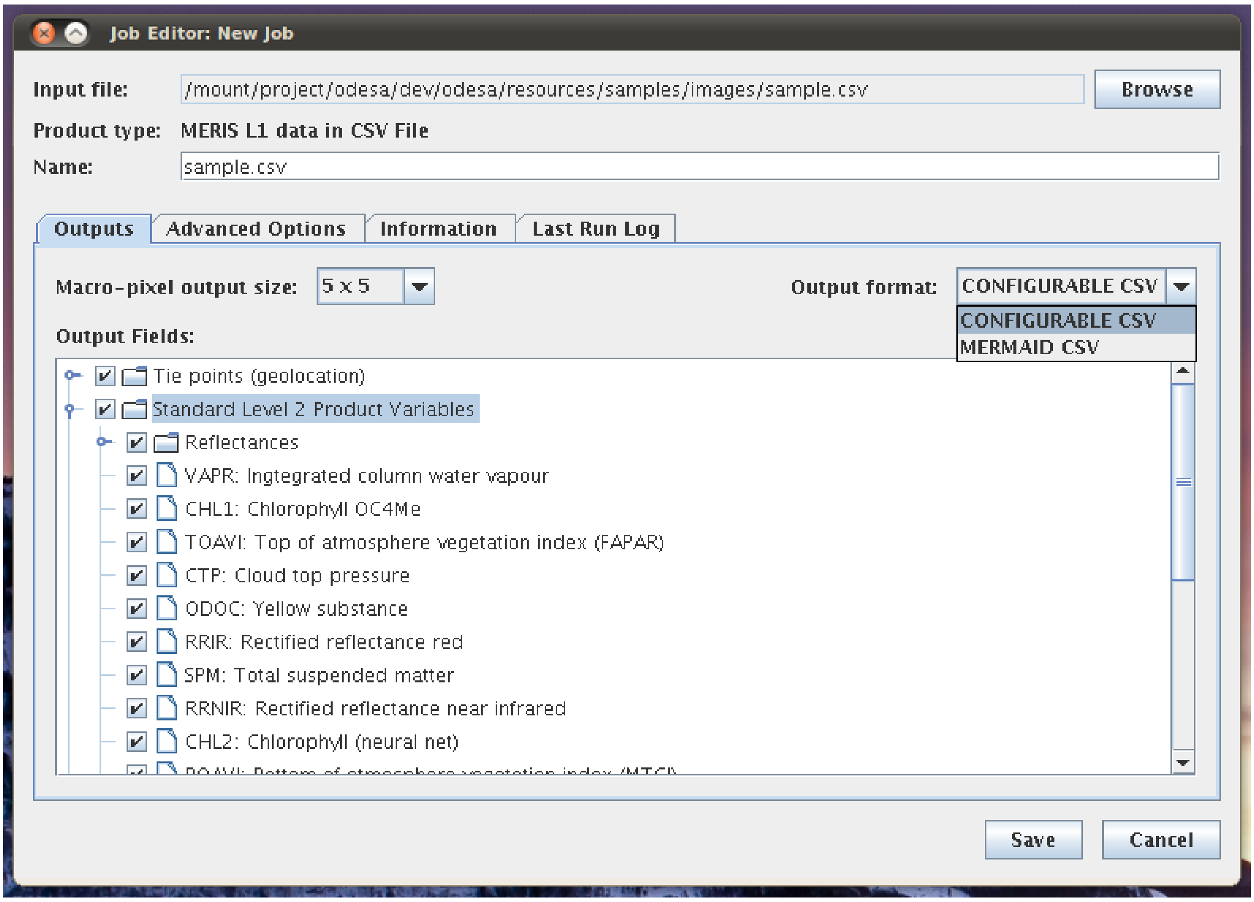


Figure 8: Selection of output products in a Level 1 CSV job

In the CSV format's output configuration tab it is also possible to choose the macro-pixel's output size.

3.2.2.3 Advanced Options

In the advanced options tab, it is possible to select low level options that will affect the output of a MEGS processing. Currently the only possible choices are which processing branches to enable. By default all three, land, water, and cloud branches are selected. When a branch is disabled, all product calculations specific to that pixel classification will be skipped. If these products are selected in the outputs tab, they will still be present in the file, but will contain "no-data" values.

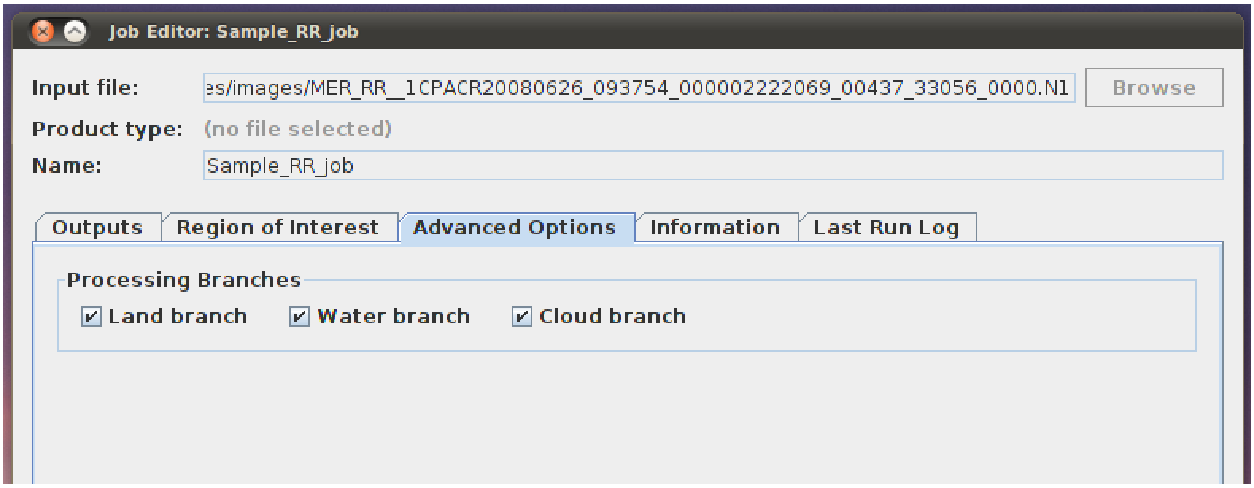


Figure 9: Advancted options tab

3.2.2.4 Information tab

This section allows the user to input a completely free description associated with the configuration. This extra information is displayed in the "Information window" when it is activated.

3.2.2.5 Last Run Log tab

In this tab it is possible to see the error messages that may have occurred during the processing phase. Obviously this section will only contain information after (or in certain cases during) execution of the job.

3.3 Running a Processing Job

This is done by selecting the job, and pressing the "Execute" tool button, or the contextual menu's "Execute" item. When doing this, the job will be queued to run. When a job is queued, its colour in the tree changes to blue, if it is currently being processed, it is coloured orange. If all goes well its colour will be green upon completing the processing, else it will be coloured red to indicate an error occurred.

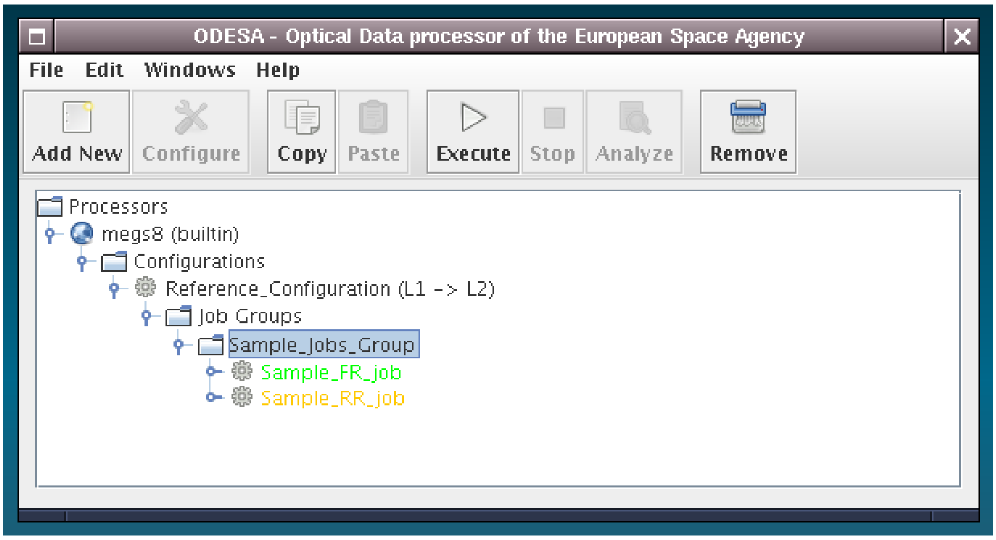


Figure 10: Main Window with Running and Queued jobs

It is possible to queue all the jobs in a job group or all the job groups within a configuration by selecting the parent nodes (the job group itself or "Job Groups") and pressing the execute button.

When a job has been executed, any change of its parameters done after the execution, will be warned by an “outdated” annotation as well as red warning icon. This annotation is completed by letters indicating the origin of the warning:

- J: means a parameter of the job has been changed
- C: means a parameter of the configuration has been changed
- P: the processors software has been updated

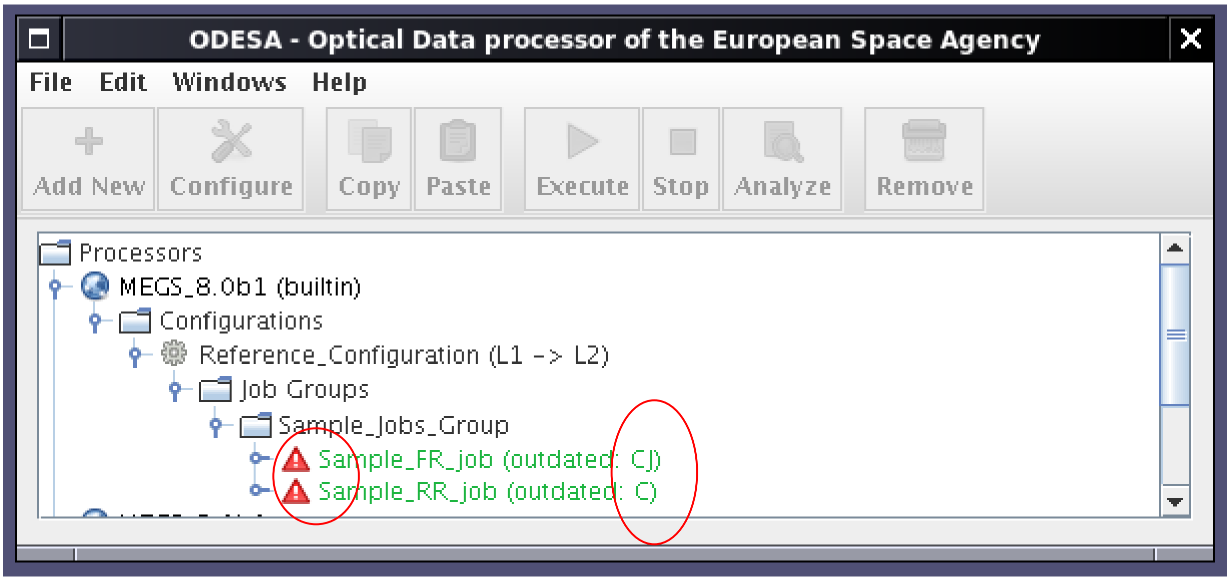


Figure 11: Main Window with outdated warning about jobs executed

3.3.1 Monitoring Execution

It is possible to keep an eye on the execution queue and the currently executing process through the use of the "Execution Manager" window, which is reachable from the main window's menu: Windows → Execution Manager.

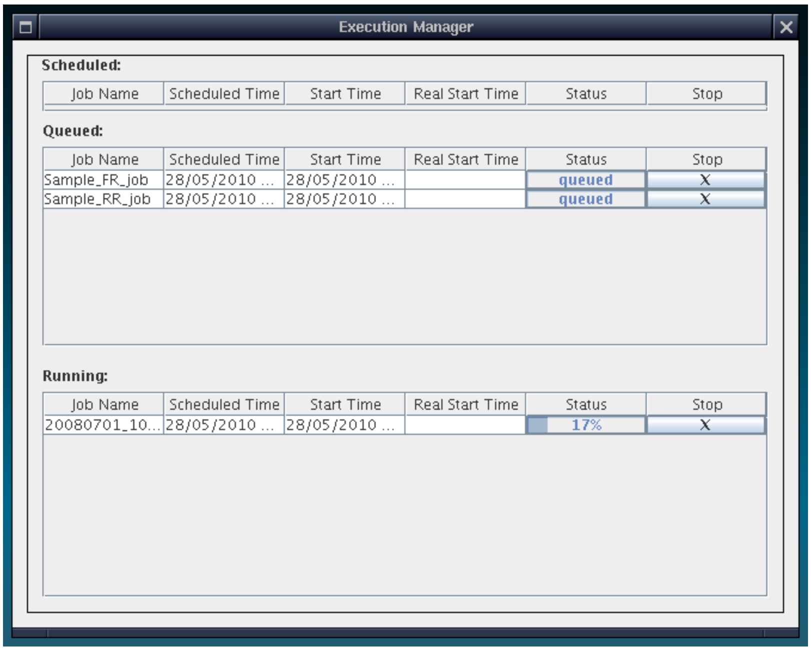


Figure 12: Execution Manager Window

3.3.2 Opening Product Files

Once a job's execution has terminated, and no errors occurred, the list of output products is shown in the job's outputs' node. The generated files are placed in the working directory within the job's folder.

As a convenience, product files may be opened directly by double clicking on the product file, or selecting it and clicking on the "Analyze" button on the toolbar.

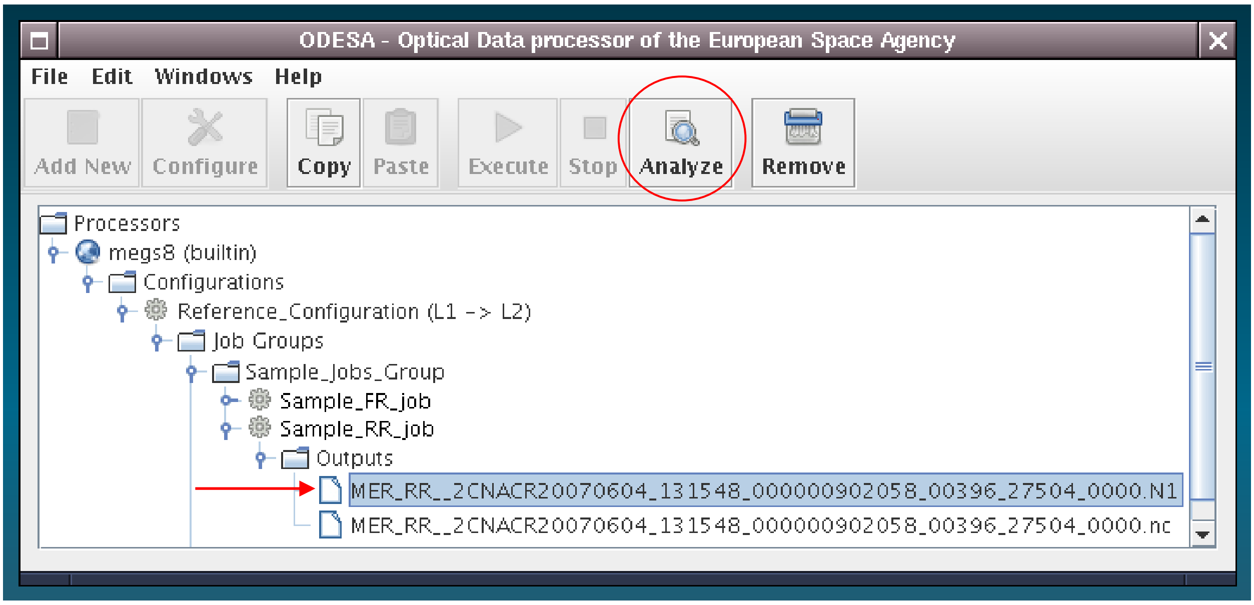


Figure 13: Analysing outputs products

ODESA will attempt to open the products with the executable defined in the preferences (See below for details), and if no such executable is defined, it will attempt to open the product with the associated application if possible.

NOTE: Opening files with the associated application is dependent on the desktop environment in use, most of the widely used ones such as GNOME, KDE, Xfce and the likes will support this feature.

3.3.3 Setting a viewing application

Specifying the viewing application to be used to open generated products can be done in the preferences window. This window is reachable from the File menu: File → Preferences.

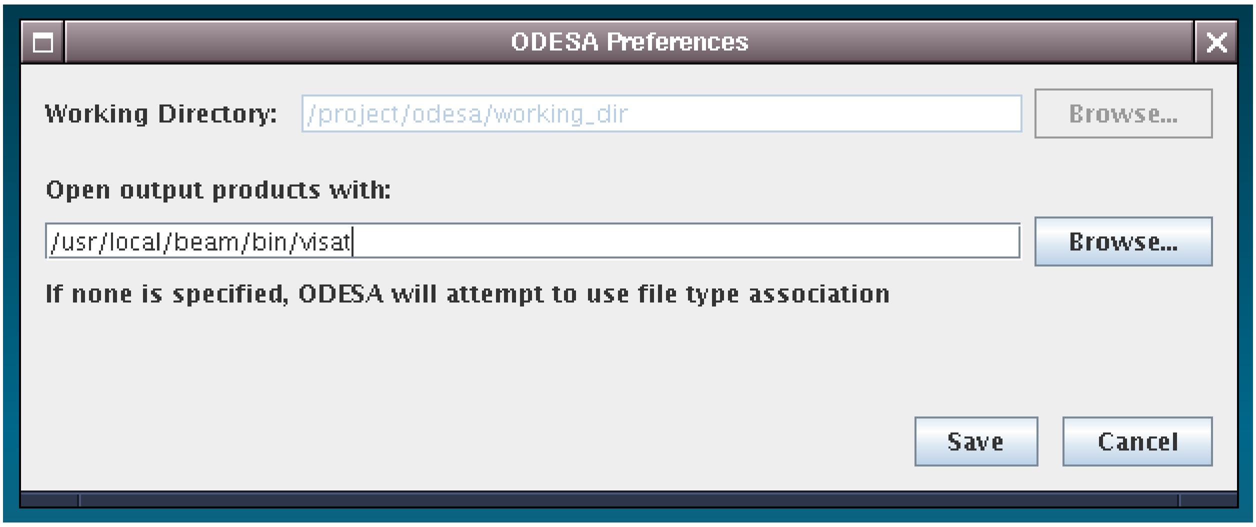


Figure 14: Setting preferences

On this preferences window, simply set the path to the executable to use for viewing applications (it must be able to accept the file name as the only parameter), either by typing it in, or using the browse button.

By removing the path and leaving it blank, the default action of attempting to use the associated application for a give file type will be restored.

3.4 Adding a new Configuration

To create a new configuration, simply select the "configurations" node under the desired processor, and either click the "Add New" button (far left), or right click on the node and select "Add new configuration … " from the pop up menu.

The "configuration Editor" window will appear. It will be possible for the user to specify a name for the new configuration file, but the mode drop down menu is automatically set to "L1 →L2" the only mode available presently.

Adding a new configuration or copying the "Reference_Configuration" doesn't automatically create new Auxiliary Data Files. The ADF file needs to be duplicated to be edited as explained in section 3.4.4.

3.4.1 Auxiliary Data Files

A number of parameters pertinent to the processing of MERIS products are specified in what are known as Auxiliary Data Files, or ADFs for short. Data in these files range from simple threshold values and band indices to use for certain algorithms, up to large Look-Up-Tables such as climatology data.

The reference ADF files used in MERIS processing are distributed along with the ODESA software distribution. Some users may want to view or edit these values to understand or fine tune them in order to verify or enhance the behaviour of a certain algorithms. To this end the ODESA software contains a tool to be used for this purpose.

3.4.2 ADF Applicability

Out of the eleven ADF files applicable to any MERIS level 2 processing, there are two that will differ when processing products before and after 2002-12-24 12:14:45. To handle this, each ADF is given a validity period. In the current case for MERIS, the two ADFs in question are "atmosphere" and "lv2conf." In the ADF configuration screen the different applicable ADF files can be distinguished by looking at the Validity Period date.

3.4.3 Viewing ADF Data

When editing a configuration, one can view the ADFs selected from the configurations tab:

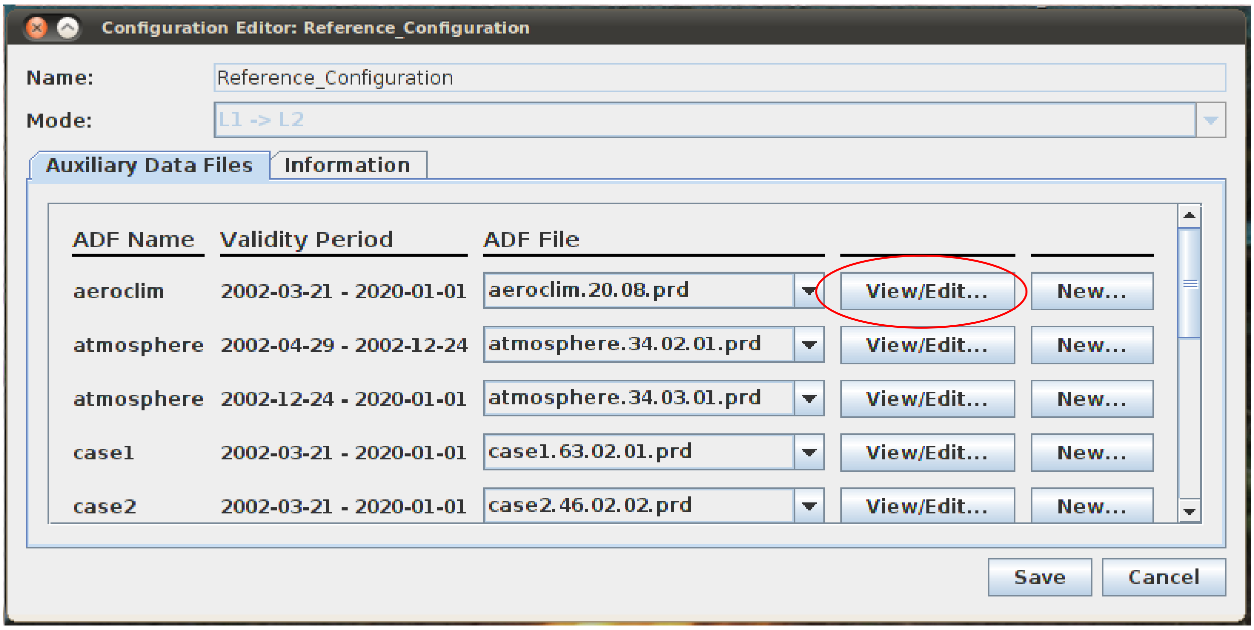


Figure 15: Configuration ADF tab – View/Edit

Clicking on a View / Edit button will open an ADF editor window for the corresponding selected ADF file:

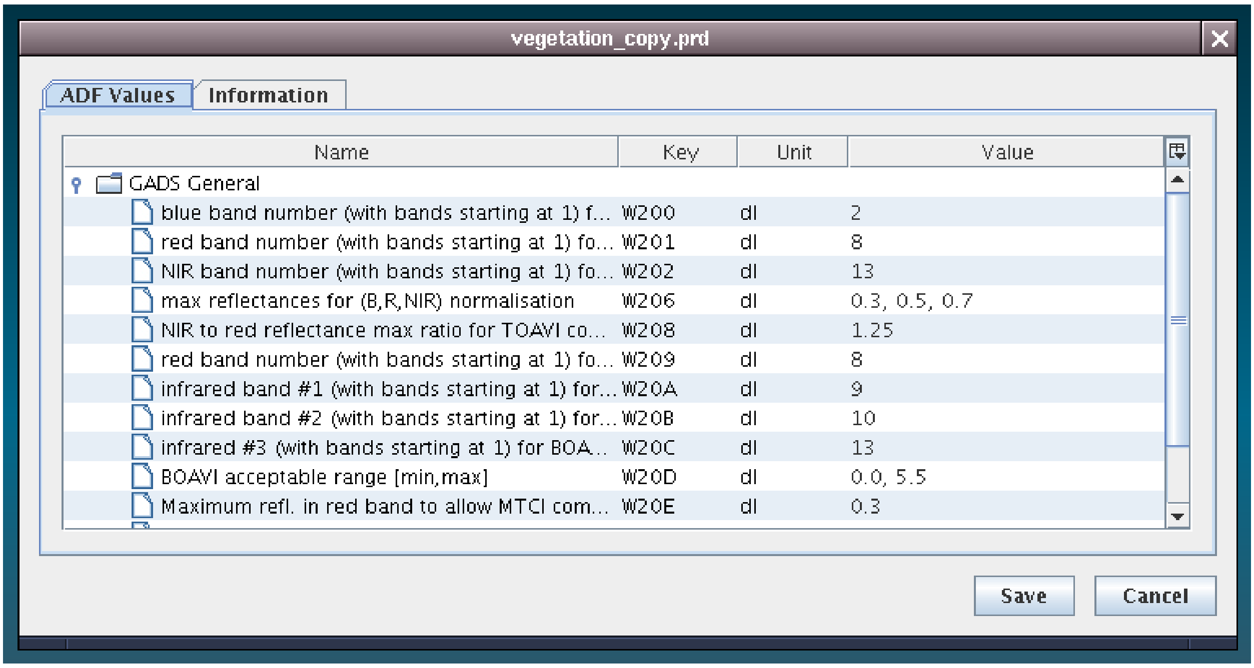


Figure 16: Viewing an ADF file

From within this window, one can easily navigate the ADF and view the values of interest. Note however that only scalar and one-dimensional arrays of up to 20 elements are shown, as well as repeated records with the same properties.

3.4.4 Editing ADF Data

While the supplied reference ADFs cannot be edited, it is possible to create user defined files for this very purpose. To create a new ADF, one needs simply to select the desired ADF in the ADFs tab of the configuration window and click the "New" button.

New ADFs created and modified by the user are placed in the working directory.

The general directory structure is as follows:

$WORKING_DIRECTORY/auxdatafiles/<processor_type>/<adf_format>/<adf_type>

For example a new ADF for the atmosphere products using the default name (atmosphere_copy.prd) would be found under:

$WORKING_DIRECTORY/auxdatafiles/megs/20/atmosphere_copy.prd

Limitations:

- This is the user responsibility not to modify an ADF (previously created by the user) when a processing is running.
- In case of modification of an ADF, this is the user responsibility to know which jobs are attached to this ADF and to run/update the outputs of the jobs.

NOTE: ADFs are shared for a given processor type across all configurations.

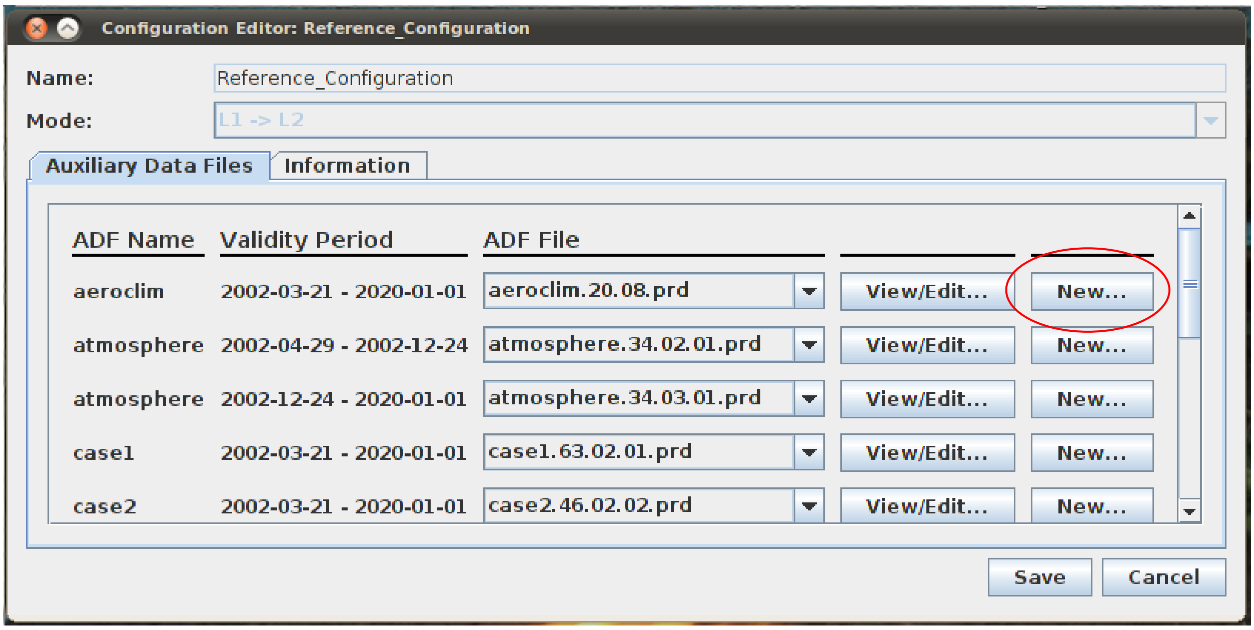


Figure 17: Configuration ADF tab – New

At this point ODESA will prompt for the new ADF's file name:

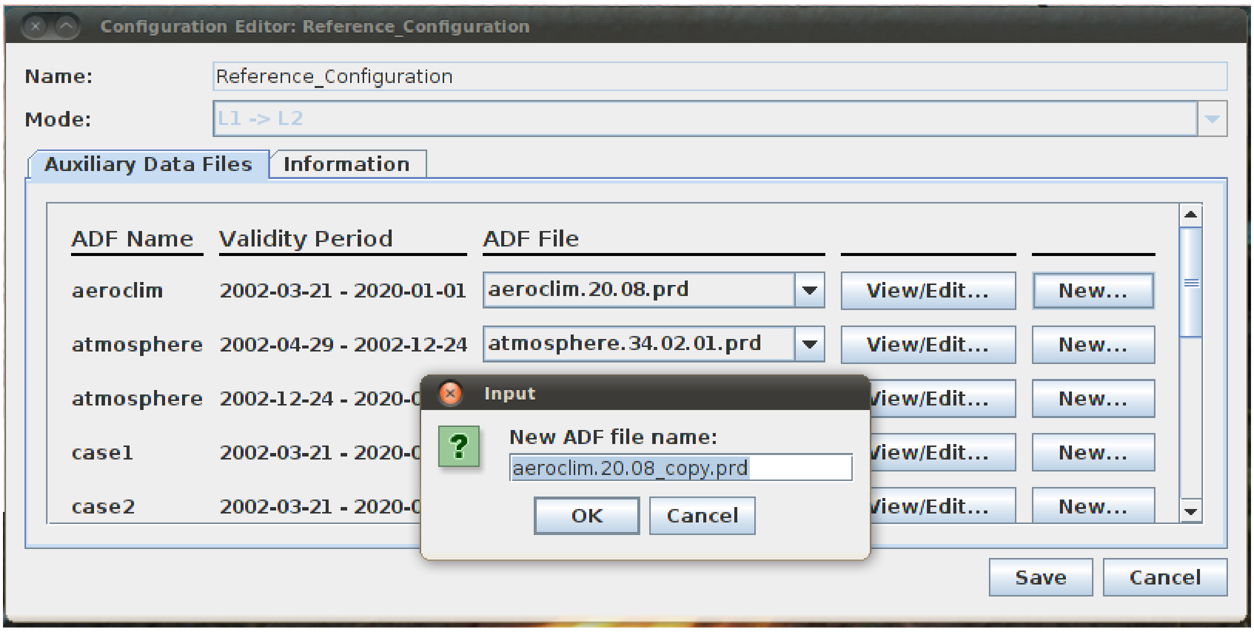


Figure 18: Prompting for new ADF file name

When this is done the newly created ADF file will be selectable in the list of available files:

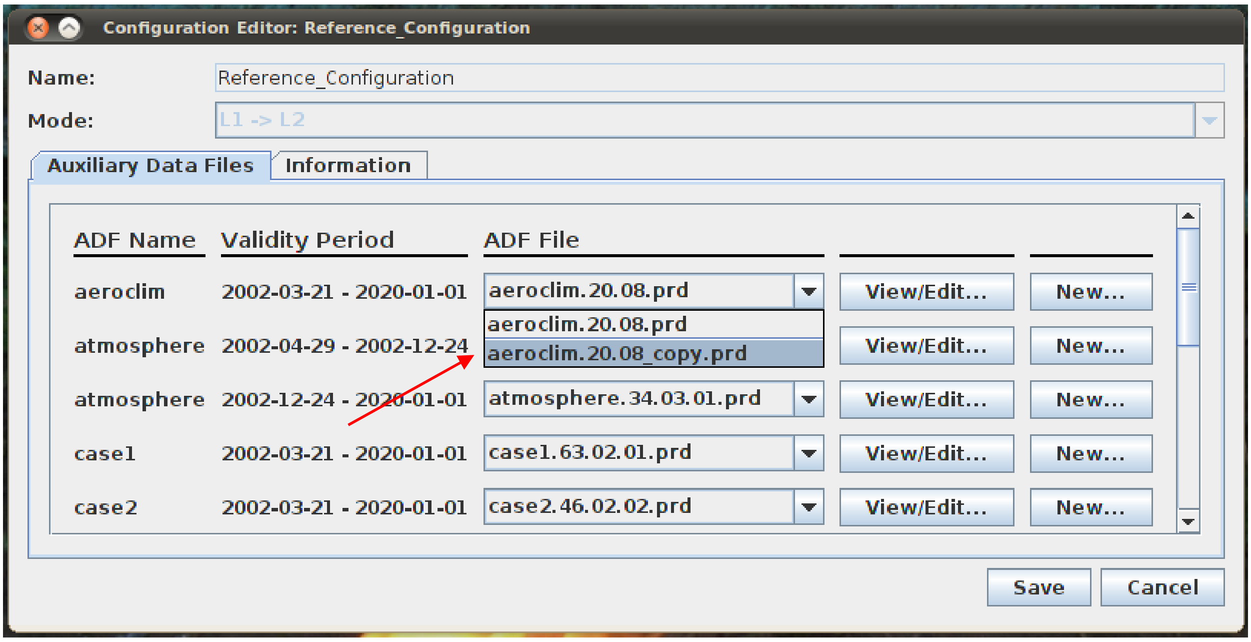


Figure 19: User defined ADF file

By selecting a user defined ADF file and clicking on the "View/Edit" button, the editor window appears, however the fields are now editable. To Change a value of an ADF field, simply double clicking on the value column will start the editing mode, and the values can be changed.

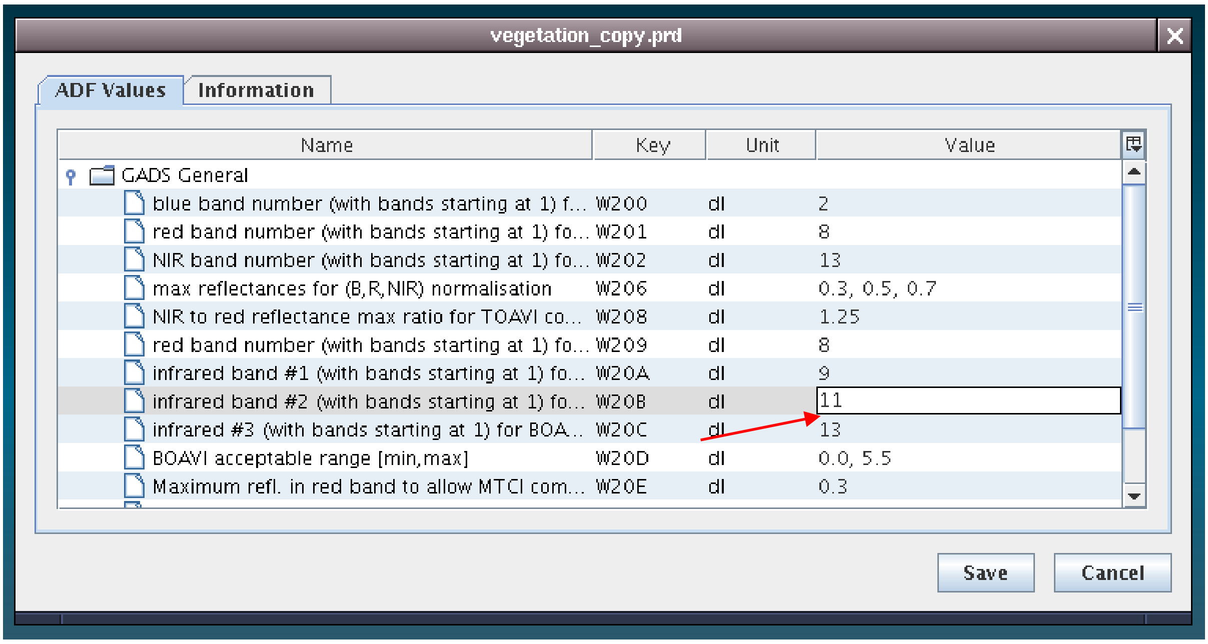


Figure 20: Editing an ADF file

4 ALGORITHM DEVELOPMENT USING MEGS®

4.1 Copying a Processor

The MEGS® processor distributed with ODESA is to be kept as a reference and cannot be modified. To make changes such as implementing new algorithms to test, one must first create a copy, which will be placed in ODESA's working directory. This is done by selecting a processor to copy, such as the default distributed one, and either clicking on the "Duplicate" button in the tool bar or the "Duplicate…" item in the popup menu reachable by right clicking on the processor's node.

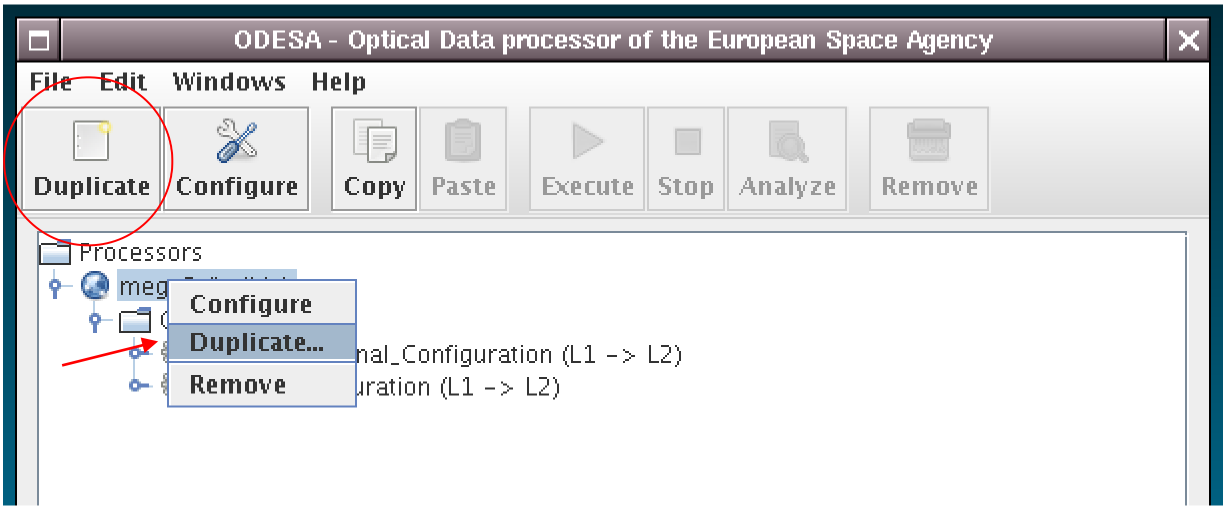


Figure 21: Copying a processor

At this point ODESA will prompt the user for a new name used to identify the processor

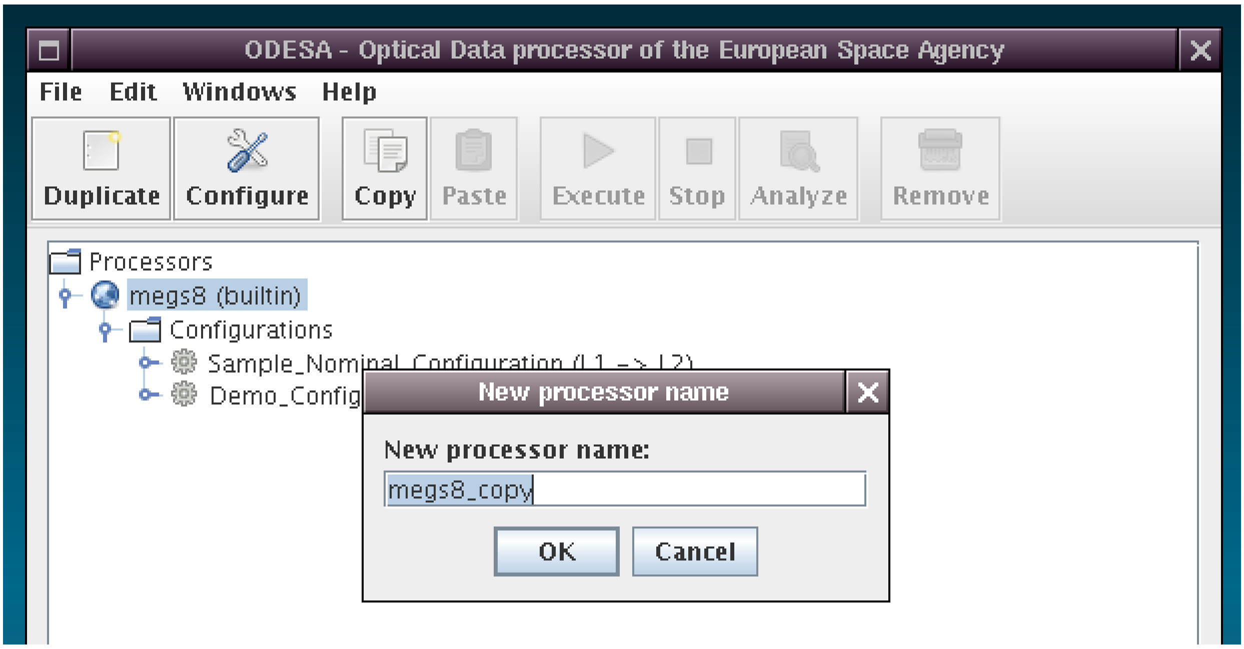


Figure 22: Naming a new processor

And will then proceed to duplicate the selected processor, creating this new one, which can then be modified at will.

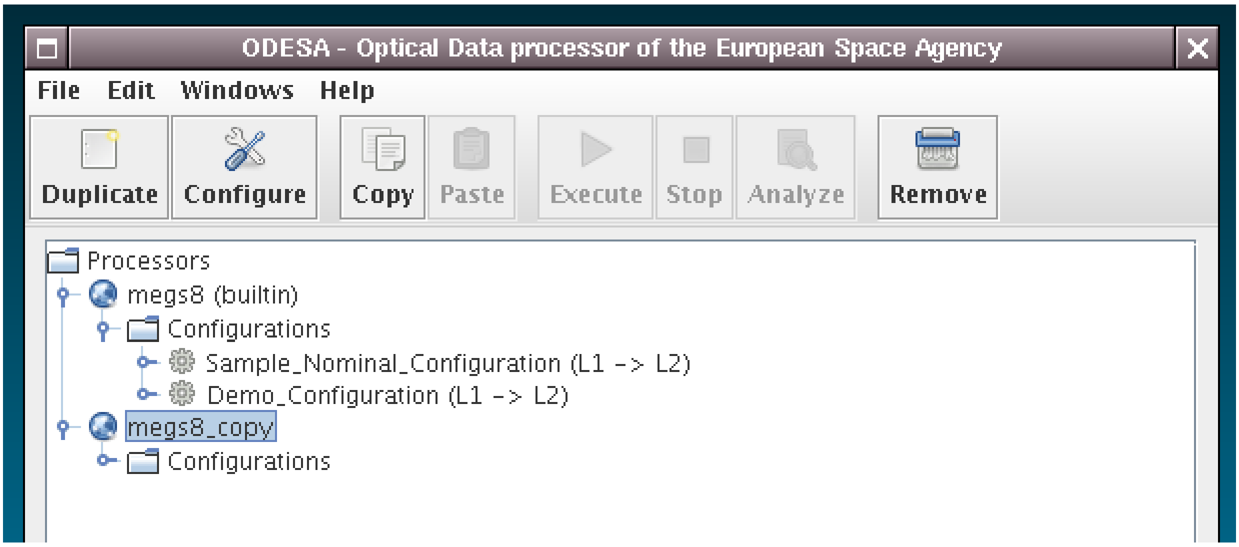


Figure 23: New processor available

4.2 Developing and Compiling MEGS®

After creating a copy, the MEGS® source code can be found in ODESA's working directory. For example, if the working directory was */project/odesa/working_dir* and the new copy was named as above ("megs8_copy") the new code will be found under */project/odesa/working_dir/processors/megs8_copy/src*. This directory contains the necessary files for development/modification of algorithms in the level 2 processing software. The core files where development should take place are found in the *level2* subdirectory.

The MEGS® code is distributed as an autotools package, and compiling MEGS from source should be as simple as following the well known common procedure:

```
./configure
make
```

and to finally put the executable in place:

```
make install
```

4.2.1 General Program Structure

The general architecture of the MEGS® software is rather simple. It can loosely be separated in the 3 stages: Initialization, Processing, and Finalization. A brief description of these stages follows.

- **Initialization** – Input product information, read the needed ADFs, and prepare the output product.
- **Processing (main loop)**:
  - **Read block** – Read a 33x65 block from the L1 input product. Blocks are read at 32x64 intervals however and one pixel column will overlap with the next block's when not on the edge of the image. This is done sequentially across the width of the product, then down its length.
  - **Process block** – This is where the core of MEGS work happens. The various steps described in the DPM (pre-processing, classification, pressure, cloud, land, water…) are each applied to the whole block in sequence, roughly following the order described in the DPM.
  - **Write block** – Write the processed block to each of the required output files (N1, NetCDF).
- **Finalization** – Close the output files and clean-up.

The **Processing** stage is the one of interest here. Most of the algorithms are defined in their own source file (and function), and special tags are interspersed in the code comments to identify the code that matches a given MERIS DPM step or equation. These tags are *@mstep* and *@meq* respectively.

In the main file "level2.c" is where the main processing loop is found, inside the main() function. From this main loop the bigger steps identified in the DPM block diagrams are called sequentially, which in turn call other sub-routines to execute the sub-step.

Using the @mstep @meq tags make it relatively easy to find where in the code modifications have to be made to implement changes in existing algorithms, by simply searching the files for a matching tag followed by the unique number as identified in the DPM. They can also be helpful in locating where particular algorithms are executed so that they can be replaced with another algorithm, or to insert some pre- or post-processing code before or after the algorithm is called.

4.2.2 Pixel and Block structures

As stated above, the basic processing unit in MEGS® is a block of 33x65 pixels. In the code this is represent as a two dimensional array. It is actually declared as:

```
S_Pixel **Pixel
```

and dynamically allocated, that way its contents can always be accessed as

```
Pixel[line][col]
```

Note that some functions in MEGS® which work on the whole block of pixels and will take the whole block pointer as a parameter while others apply to single pixel and require only a pointer to a single pixel.

The standard values that are affected to a pixel are described in the pixel.h file, which describes the S_Pixel structure. Each field of the pixel structure is name as closely as possible to variables in the DPM, and conventionally the Greek letters are spelled out. In the pixel.h file, each field is also described by an accompanying comment which describes the value it will hold. These comments include some special tagging used in generating the code documentation with Doxygen. In fact, it is much easier to inspect the pixel structure from the generated documentation.

4.2.3 Adding Outputs and Intermediate Values

Most values of interest (level 1 radiance input, level 2 outputs and intermediate values) are already represented in the pixel structure described above and new code will most likely operate within that domain. There are however a few cases where "new" values are desired, such as a new product, and possibly new set of intermediate values.

The internal structure containing pixel information (in pixel.h)
must not be modified (as this is used by the precompiled binary-only code).
This remark applies also to any structure defined in the *.h files.

To overcome this general purpose an array of 50 elements (of type 'float') has been affixed to the pixel structure. For example, if a new ocean color algorithm is developed, given 'pix' as a point to an S_Pixel structure,

```
pix->user_bp[0] = new_ocean_color_algo(pix);
```

would put the result in the first breakpoint array cell.

All that is needed to be able to visualize the values is to enable the NetCDF output in a job, and select the corresponding breakpoint in the ODESA (under "User Breakpoints"). Keep in mind that in the C code the array is indexed starting from 0, while in the interface they are indexed from starting from 1.

Adding a new configuration or copying the "Reference_Configuration" doesn't automatically create new Auxiliary Data Files. The ADF file needs to be duplicated to be edited as explained in section 3.4.4.

4.2.4 Adding new Source Files

When new source files are added to MEGS®, the Makefile.am file (found in *./src/level2/src/*) must be modified to reflect this change. The entry starting with "level2_SOURCE =" is followed by a list of all file to be include: simply add any new source file to this list.

4.2.5 Code Documentation

As mentioned above, most of the code is documented with comments often containing special tags which make references to particular parts of the DPM. These tags are used to enhance the output of Doxygen generated documentation which is used by MEGS®.

To regenerate this documentation simply run

```
make doc
```

from the *./src/level2* directory, and the documentation will be generated in the *./doc* subdirectory. To generate the documentation a couple of tools are required:

- **Doxygen** – This is the main program that parses the source code and generates documentation.
- **Graphviz** – This is used by Doxygen to generate graphics such as call graphs.
- **LaTeX** – This is used by Doxygen to render special tags like are used for pixel variables' descriptions

> All these tools must be found in the PATH in your environment. They are readily and easily available in most Linux distributions' package system. For example in Ubuntu they are the *doxygen*, *graphviz*, and *texlive* packages.

This documentation is also very helpful in navigating around the MEGS® source code as it generates a hyperlinked version of the files and cross-referenced documentation. It is very easy to follow the interactions and it is a simple matter of clicking on various objects to get more information on how they are used.

4.2.6 Compiling MEGS®

When ODESA detects that an instance of MEGS has had it source modified (i.e. some source files appear to be newer than the binary itself), it will inform the user by annotating the processor instance with "recompilation needed," as well as red warning icon as show below:

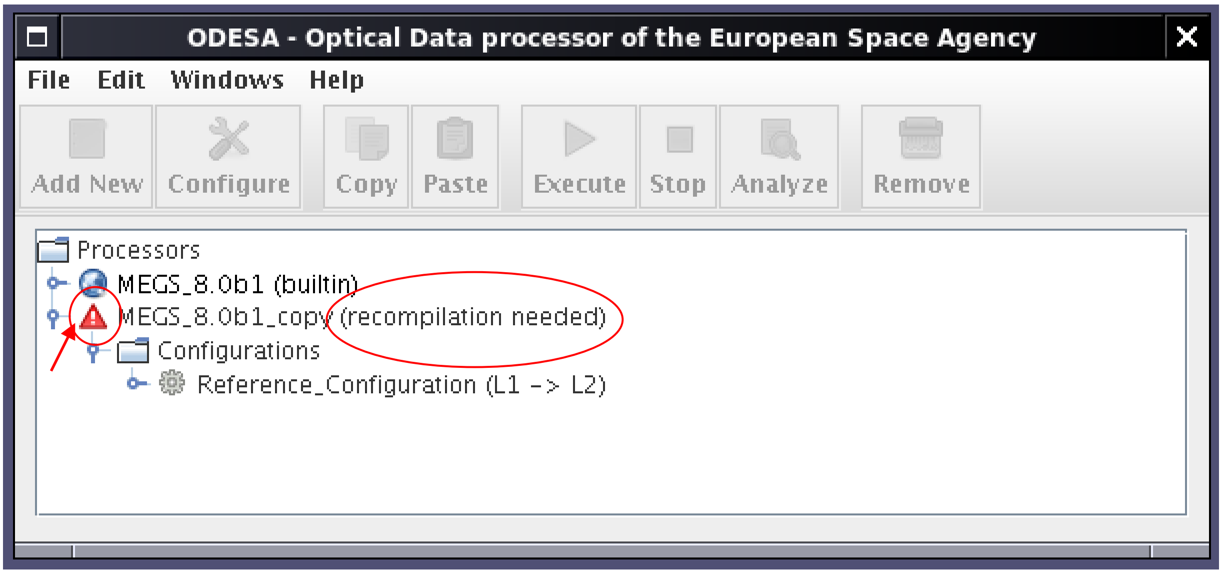


Figure 24: ODESA detected a modified MEGS source code

From within the *src/level2* directory, running the command "*make install*" will build MEGS® and then install it in the *bin* directory (sibling to the *src* directory).

The "make install" command is required at the top level of the src/level2 directory to properly update the binary code used by the interface.

4.2.7 Requirements

To successfully compile MEGS®, a proper development environment must be present on the machine. This consists of the usual tools such as:

- make
- GCC C compiler
- Autotools: autoconf automake

Because MEGS® is compiled as a 32 bit application, when compiling on a 64 bit machine, there are a few additional elements that have to be present. On most 64 bit Linux distributions, the 32 bit development environment isn't installed by default, thus must be also be installed.

As an example, on recent Ubuntu distributions, the necessary packages to be installed are *libc6-dev-i386 and gcc-multilib.*

All the non standard libraries (hdf5, netcdf4, zlib, unwind, etc) required by MEGS® are included precompiled in the distribution

4.3 Running New Code

User defined copies of MEGS® processors behave exactly like the built-in one, using the same system of configurations, job groups and jobs. In fact it is possible to copy and paste configurations across processors, which facilitates the testing of new modifications since the output products of the two can then be compared.

5 ANNEX 1: BREAKPOINTS

It is possible to configure the product variables that are included in the NetCDF output file (see section 3.4). A list and description of all the variables available in the delivered ODESA software distribution is detailed below.

5.1 Tie points

- Latitude
- Longitude
- Viewing zenith angle
- Sun zenith angle
- Azimuth difference angle
- Sun azimuth angle
- Altitude
- ECMWF zonal wind
- ECMWF meridional wind
- ECMWF wind speed modulus
- ECMWF pressure
- ECMWF ozone
- ECMWF relative humidity

5.2 Standard Level 2 Products

- Reflectances
- Integrated column WV
- Chlorophyll (OC4Me)
- Top of atmosphere vegetation index (FAPAR)
- Cloud top pressure
- Yellow substance
- Rectified reflectance red
- Total suspended Matter
- Rectified reflectance near infrared
- Chlorophyll (Neural Net)
- Bottom of atmosphere vegetation index (MTCI)
- Surface pressure
- Photosynthetically active radiation

- Cloud albedo
- Aerosol Angstrom exponent
- Cloud type:
- Aerosol optical thickness
- Cloud optical thickness
- Level 2 Flags: These standard level 2 flags correspond to the flag bytes found in the N1 level 2 products. In the N1 they are coded on 3 bytes (24 bits), however in the NetCDF file they are stored in a 32bit integer, where the top byte is unused. The table below describes what flag each bit represents, numbered from least significant to most significant bit.

BIT	Meaning
0	Water pixels: White scatter Land pixels: Low pressure
1	Low sun – The sun is too low on the horizon (zenith)
2	Water pixels: High glint Land pixels: TOAVI Inval Rec
3	Water pixels: BPAC on Land pixels: DDV
4	Water pixels: Medium glint Land pixels: TOAVI class WS
5	Water pixels: Ice haze Land pixels: TOAVI class CSI
6	Water pixels: CASE2_Y Land pixels: TOAVI class BAD
7	Water pixels: CASE2_ANOM Land pixels: TOAVI class bright
8	Water pixels: CASE2_S Land pixels: SNOW_ICE
9	ABSOA_DUST
10	O_AERO_DB
11	SUSPECT
12	COSMETIC
13	COASTLINE
14	PCD_19 Invalid Water or Land: ORINP0 or OROUT0 or ACFAIL Water: Whitecaps or UNCGLING or ICE_HIGHAERO or HIINLD or ACLIM or SATURATED(b775, b865) or (NOT(CASE2_S or CASE2_Y

	or CASE2_A) and DROUT) Cloud: ORINP2 or SATURATED(b753, b760) Land: DDV and SATURATED(b412, 442, 490, 665, 865)
15	PCD_18 Invalid Water: Land: Cloud:
16	PCD_17
17	PCD_16
18	PCD_15
19	PCD_14
20	PCD_1_13
21	WATER
22	CLOUD
23	LAND

5.3 Alternative Algorithms

- RHO_WN - Fully normalised water-leaving reflectances
- NRRS – Fully normalised water-leaving reflectances (GSM input)
- GSM Model
  - Chlorophyll
  - Coloured dissolved and detrital organic materials
  - Particulate back-scattering coefficient at 443nm
  - Chlorophyll error
  - Coloured dissolved and detrital organic materials error
  - Particulate back-scattering coefficient at 443nm error
  - CHI2
  - Number of iterations
  - CHL1 and CDM covariance
  - CHL1 and BBP covariance
  - CDM and BBP covariance
  - NRRS – Full normalised water-leaving reflectances (GSM output)
- Case 2 NN BOA
  - Chlorophyll 2 concentration
  - Suspended Partical Matter

        - Yellow Substance

5.4 Intermediate Variables

- General
    - Separated L1 flags
    - Separated L2 flags
    - Separated intermediary L2 flags
    - Separated annotation flags
    - Separated band saturation flags
    - Detector index
    - Apparent pressure
    - Surface albedo
    - Aerosol optical thickness
    - Aerosol reflectance
    - Level 2 flags 1
    - Level 2 flags 2
    - Annotation flags for the quality of the atmospheric corrections/Negative corrected reflectance
    - Rayleigh reflectance corrected for pressure (water)
    - Coarse Rayleigh reflectance (land, water, cloud)
    - Band saturation flags
- TOA Radiances and Reflectances
    - TOA radiance, correct for stratospheric aerosol
    - TOA reflectance
    - Stratospheric aerosol corrected reflectance
    - TOA reflectance, corrected for stratospheric aerosol contribution and gaseous absorption
    - TOA reflectance, corrected for aerosol, gaseous absorption and smile effect
- Land
    - Stratospheric aerosol index
    - Top of aerosol reflectance
    - DDV model number
    - DDV reflectance and coupling term correction factors
    - Aerosol ground BRDF coupling terms
    - Rayleigh aerosol BRDF coupling terms
    - Rayleigh ground BRDF coupling terms
    - Corrected surface reflectances

        - MERIS differential snow index
        - Atmospherically robust vegetation index
        - ARVI threshold
    - Water
        - Mixing ratio
        - Bracketing Aerosol model 1
        - Bracketing Aerosol model 2
        - Sediment load retrieved by the bright pixel method
        - Marine reflectance retrieved by the BPAC
        - Final bbp estimate at band 775 retrieved by the BPAC
        - Final reflectance at b775 retrieved by the BPAC
        - Final Angstrom exponent retrieved by BPAC
        - Annotation flags for the quality of the BPAC
        - Separated annotation flags for the quality of the BPAC
        - Glint reflectance
        - Aerosol single scattering albedo
        - Aerosol forward scattering probability
        - Downward diffuse total transmittance (Rayleigh + aerosol)
        - Upward diffuse total transmittance (Rayleigh + aerosol)
        - Glint corrected reflectance
        - Sun glint reflectance at TOA
        - Normalised water-leaving reflectance
        - C2R retrieval
            - Tau443
            - Tau550
            - Tau778
            - Tau865
            - Angstrom
            - Glintrat
            - Chi_sum_atm
            - Chi_sum_wat
            - RLpath
            - RLw
            - RLw_a
            - RL_tosa
            - Trans_ed

6 ANNEX 2: CSV INPUT FORMAT

The CSV format used by the MEGS software has the following specifications.

It is a standard CSV formatted file using a semi-colon (';') as a separator. The first line must contain a "header" line, which contains the column names; thus field ordering is unimportant, as each column can be identified by the corresponding name in the "header" line. Each of the following lines describes a single pixel.

Pixels can be grouped and processed together as a "macro-pixel" by giving a YxY set of pixels, with the same "matchup id". Y must be an odd number between 1 and 19. Pixels belonging to the same matchup must be consecutive and ordered North-East to South-West in line with the MERIS convention.

The mandatory fields are a matchup id which uniquely identifies a set of points, and the Level1 data, as described below:

MATCHUP_ID	A unique name that groups a set of pixels together.
TIME	A UTC time stamp (format: YYYYMMDDTHHMMSSZ) representing the pixel acquisition time.
RESOLUTION	MERIS Resolution mode to which the pixels correspond (RR:~1km pixels, or FR: ~300m pixels)
COSMETIC DUPLICATED GLINT_RISK SUSPECT LAND_OCEAN BRIGHT COASTLINE INVALID	These correspond to the MERIS Level1 flags, taking the value 0 (flag not set) or 1 (flag set)

LAT LON VIEW_ZENITH SUN_ZENITH VIEW_AZIMUTH SUN_AZIMUTH DELTA_AZIMUTH SCATT_ANGLE ALTITUDE WINDU WINDV WINDM PRESS_ECMWF OZONE_ECMWF	These are various geophysical parameters of the pixel on the ground. These correspond to the data described by the tie points in the MERIS Level1 product.
DETECTOR	The detector index corresponding to the detector on the MERIS instrument.
TOAR_01 TOAR_02 TOAR_03 TOAR_04 TOAR_05 TOAR_06 TOAR_07 TOAR_08 TOAR_09 TOAR_10 TOAR_11 TOAR_12 TOAR_13 TOAR_14 TOAR_15	These are the Level 1 radiances for each of the MERIS bands.

For a sample file one can take a look at the supplied "sample.csv" file, which has a sample job created upon first execution (if desired) or in the "resources/samples/images" subdirectory of the ODESA installation.