Supplement of Geosci. Model Dev., 11, 2419–2427, 2018
https://doi.org/10.5194/gmd-11-2419-2018-supplement

Geoscientific Model Development
Open Access

Supplement of

High-performance software framework for the calculation of satellite-to-satellite data matchups (MMS version 1.2)

Thomas Block et al.

Correspondence to: Thomas Block (tom.block@brockmann-consult.de)

FIDUCEO Multi-sensor Match up System (MMS) Manual

Tom Block, Sabine Embacher

Brockmann Consult GmbH

1/17/2017

FIDUCEO has received funding from the European Union's Horizon 2020 Programme for Research and Innovation, under Grant Agreement no. 638822

1 Content

2 Introduction

This document is the user manual for the Fiduceo Multisensor Matchup System software (MMS). It covers all tools, configuration files, supported data formats and plugin configurations used by the software. In addition, a description of the installation and operation procedures on the CEDA JASMIN cluster is given.

2.1 Scope

This document is thought as a living document, updates will be integrated whenever applicable or necessary. The contents of this version of the manual refers to the MMS software version 1.2.0.

2.2 Version Control

Version	Reason	Reviewer	Date of Issue
1.0	Initial version		
1.1	New minor version		
1.2	New minor version, added post-processing		2017-01-17

2.3 Software Changelog

This chapter lists the changes, updates and bugfixes applied to the MMS software.

2.3.1 Updates from version 1.2.0 to 1.2.1

- Updated BorderDistance condition to allow distinguishing between reference and secondary sensor
- Added RFI glint variables to AMSR-E reader
- Added post processing plugin to add ERA-interim NWP data to matchup datasets

2.3.2 Updates from version 1.1.3 to 1.2.0

- Implemented post processing engine
- Added spherical distance post-processing plugin
- Added AMSRE solar angles post-processing plugin
- Added SST in-situ time series extraction post-processing plugin
- Added WindowValue screening plugin

2.3.3 Updates from version 1.1.2 to 1.1.3

- Added reader cache size parameter to MMD writer configuration
- Updated MMDWriter to store CF conforming default fill value attributes

2.3.4 Updates from version 1.1.1 to 1.1.2

- Fixed bug in AVHRR reader that causes crashes when extracting 1x1 pixel windows

2.3.5 Updates from version 1.1.0 to 1.1.1

- Implemented full support for PostGIS databases

2.3.6 Updates from version 1.0.5 to 1.1.0

- Added condition plugin for overlap removal
- Added support for SST-CCI insitu data

2.3.7 Updates from version 1.0.4 to 1.0.5

- Migrated system configuration from properties to XML format
- Implemented configurable archiving rules

2.3.8 Updates from version 1.0.3 to 1.0.4

- Added MMD Writer configuration file
- Added MMD variable renaming engine

2.3.9 Updates from version 1.0.2 to 1.0.3

- Added reader for SSM/T-2 data
- Added full support for Apache H2 database
- Renamed parameter for AngularScreening plugin
- Added mmd writer configuration
- Fixed bug in HIRS reader plugin that prevented HIRS NOAA 18 data to be handled

2.3.10 Updates from version 1.0.1 to 1.0.2

- Added HIRS “lza” angular screening
- Added HIRS L1C reader
- Updated regular expression for AMSU-B/MHS reader
- Added reader for ATSR1, ATSR2 and AATSR L1B data in ENVISAT format
- Added optional calculation of matchup center distance variable
- Added PixelValue screening plugin
- Corrected fill value handling to follow CF conventions
- Added reader form AMSR-E L2A data in HDF format
- Added support for multiple processing version of sensor data

2.3.11 Updates from version 1.0.0 to 1.0.1

- Implemented cloud screening algorithm for AMSU-B, MHS and SSMIS according to University of Hamburg
- Bugfixes:
    - Fixed issue where input files were not closed correctly
    - Fixed performance bottleneck when adding secondary observation pixels

2.4 Applicable and Reference Documents

The following documents are applicable (AD) or references (RD) in this manual

RD 1	FIDUCEO_MMS_IP_v_1_1	Fiduceo Multisensor Matchup System Implementation Plan, version 1.1
RD 2	Buehler2007	S. A. Buehler, M. Kuvatov, T. R. Sreerekha, V. O. John, B. Rydberg, P. Eriksson, and J. Notholt: A cloud filtering method for microwave upper tropospheric humidity measurements, Atmos. Chem Phys. Discuss., 7, 7509–7534, 2007
RD 3	Hans2016	TN: Computing the Tb thresholds for cloud detection for the different viewing angles, Imke Hans, University of Hamburg
RD 4	SST_CCI-TN-UOL-003	ESA CCI Phase 2 (SST) - MMD Content Specification

2.5 Glossary

CDO	Climate Data Operators
CEDA	Centre for Environmental Data Analysis
CEMS	Climate and Environmental Monitoring from Space
JDBC	Java DataBase Connectivity
JDK	Java Development Kit
LZA	Local Zenith Angle
MMD	Multisensor Matchup Dataset
MMS	Multisensor Matchup System
NetCDF	Network Common Data Format
NWP	Numerical Weather Prediction
TN	Technical Note
VZA	Viewing Zenith Angle
XML	eXtensible Markup Language

3 Software Overview

The MMS is a software system that is composed of several interacting components. This chapter gives a high-level overview of the purpose and interactions of the components.

To operate the MMS tools, some external components need to be installed and accessible to the tools. This is mainly a unified archive of input data (see 4.1.1) and a database server engine (currently supported MongoDB, Apache H2 and PostGIS) with a user account accessible to the tools.

The calculation of matchup datasets is a three-step process:

1. Ingest satellite metadata into the database. This has to be run once for every input dataset that should be used for matchup analysis. This task is executed by the IngestionTool (chapter 3.1)
2. Find matchups and generate MMD files. This is done by the MatchupTool (chapter 3.2).
3. (optionally) Run a post-processing job on the generated MMDs. This is done by the PostProcessingTool (chapter 3.3).

3.1 Ingestion Tool

This tool extracts satellite metadata from the input files, pre-processes the metadata and stores the records to the database. The tool is invoked from the command line using the script *ingestion-tool.sh/bat* which is located in the *bin* directory of the MMS installation. The command line parameters can be plotted using the –h/--help option:

```
ingestion-tool version 1.1.1

usage: ingestion-tool <options>
Valid options are:
   -c,--config <arg>            Defines the configuration directory. Defaults to './config'.
   -end,--end-time <Date>       Define the ending time of products to inject.
   -h,--help                    Prints the tool usage.
   -s,--sensor <arg>            Defines the sensor to be ingested.
   -start,--start-time <Date>   Define the starting time of products to inject.
   -v,--version <arg>           Define the sensor data processing version.
```

--config: define the path to the configuration files directory

--sensor: define the sensor key to be ingested. Sensor keys are defined by the product reader modules available, see chapter 3.4.

--version: defines the sensor data processing version to be ingested. Must match one of the versions stored in the archive, see 4.1.1.

--start-date: defines the start date for the ingestion

--end-date: defines the end date for the ingestion

All satellite data file matching type and version with an acquisition time start that falls into the interval defined by *start-date* and *end-date* will be processed.

The ingestion tool uses the following configuration files:

- system-config.xml (chapter 4.1.1)
- database.properties (chapter 4.1.2)

3.2 Matchup Tool

The matchup tool calculates matchups and generates MMD files. The tool is invoked from the command line using the script *matchup-tool.sh/bat* which is located in the *bin* directory of the MMS installation. The command line parameters can be plotted using the –h/--help option:

```
matchup-tool version 1.1.1

usage: matchup-tool <options>
Valid options are:
   -c,--config <arg>            Defines the configuration directory. Defaults to './config'.
   -end,--end-time <arg>        Defines the processing end-date, format 'yyyy-DDD'
   -h,--help                    Prints the tool usage.
   -start,--start-time <arg>    Defines the processing start-date, format 'yyyy-DDD'
   -u,--usecase <arg>           Defines the path to the use-case configuration file. Path is
relative to the
                                configuration directory.
```

--config: define the path to the configuration files directory

--start-time: defines the start date for the ingestion

--end-time: defines the end date for the ingestion

--usecase: defines the usecase configuration file to use, see chapter 4.1.4

The matchup tool uses the following configuration files:

- system-config.xml (chapter 4.1.1)
- database.properties (chapter 4.1.2)
- mmd-writer-config.xml (chapter 4.1.3)

3.3 Post Processing Tool

The post processing tool allows to add/remove/modify variables in a set of MMD files already generated. It is a command line tool using the following parameters (the same printout can be achieved by running the tool using the –h parameter). The tool is started using shells scripts located in the bin directory of the installation named *post-processing-tool.sh/bat*.

```
post-processing-tool version 1.1.1

usage: post-processing-tool <options>
Valid options are:
   -c,--config <arg>            Defines the configuration directory. Defaults to './config'.
```

```
    -end,--end-date <arg>       Defines the processing end-date, format 'yyyy-DDD'.
                                DDD = day of year.
    -h,--help                   Prints the tool usage.
    -i,--input-dir <arg>        Defines the path to the input mmd files directory.
    -j,--job-config <arg>       Defines the path to post processing job configuration file.
                                Path is relative to the configuration directory.
    -start,--start-date <arg>   Defines the processing start-date, format 'yyyy-DDD'.
                                DDD = day of year.
```

--config: define the path to the configuration files directory

--job-config: define the job-configuration file to use; the format of this file is described in chapter 4.1.5.

--input-dir: defines the input data directory. All files inside this directory which match the file naming pattern "*mmd\\d{1,2}.*_.*_.*_\\d{4}-\\d{3}_\\d{4}-\\d{3}.nc*" are collected as input files for post processing. The directory search runs recursively through all subdirectories.

--start-date: defines the start date for the post-processing

--end-date: defines the end date for the post processing

All MMD files with a start date within the period defined by start-date and end-date will be taken into account for processing. If any severe error occurs while computing a post processing, the processing will report the error and continue processing with the next input file, if possible.

The post processing tool uses the following configuration files:

- system-config.xml (chapter 4.1.1)
- post processing configuration (chapter 4.1.5)

3.4 Supported Satellite Sensors

This chapter lists the input satellite data source types available for matchup processing. In subsequent chapters, the sensors, the data format and the available variables are described in more detail.

3.4.1 AMSR-E

AMSR-E L2A brightness temperatures in HDF4 format. This reader only handles the “Low_Res_Swath” data.

Sensor Key	Source
amsre-aq	AMSR-E data from Aqua platform

Variable Name	Description
Time	Acquisition time per scanline in TAI seconds since 1993
Latitude	Latitude of pixel in decimal degrees
Longitude	Longitude of pixel in decimal degrees
Earth_Incidence	Incidence angle of observation in degrees
Earth_Azimuth	Azimuth angle of observation in degrees

Sun_Elevation	Sun elevation angle in degrees
Sun_Azimuth	Sun azimuth angle in degrees
Land_Ocean_Flag_6	Land coverage flag for 6 GHz channel
X_XV_Res_1_TB	Brightness temperatures vertical polarisation in low resolution for channels X_X (6_9, 10_7, 18_7, 23_8, 36_5, 89_0)
X_XH_Res_1_TB	Brightness temperatures horizontal polarisation in low resolution for channels X_X (6_9, 10_7, 18_7, 23_8, 36_5, 89_0)
Scan_Quality_Flag	Quality flags per scan
Channel_Quality_Flag_XXV	Quality flags per vertical polarisation channel XX (6, 10, 18, 23, 36, 89)
Channel_Quality_Flag_XXH	Quality flags per horizontal polarisation channel XX (6, 10, 18, 23, 36, 89)
Res1_Surf	Land coverage percentage for resolution one
Geostationary_Reflection_Latitude	RFI sources glint angles latitude
Geostationary_Reflection_Longitude	RFI sources glint angles longitude

3.4.2 AMSU-B

AMSU-B L1c data from NOAA CLASS storage, converted to HDF using “atovin” and “convert_to_hdf5”.

Sensor Key	Source
amsub-n15	AMSU-B NOAA15
amsub-n16	AMSU-B NOAA16
amsub-n17	AMSU-B NOAA17

Variable Name	Description
btemps_ch1	Brightness temperature in channel 1 (2.9684 m^{-1})
btemps_ch2	Brightness temperature in channel 2 (5.0032 m^{-1})
btemps_ch3	Brightness temperature in channel 3 (6.1146 m^{-1})
btemps_ch4	Brightness temperature in channel 4 (6.1146 m^{-1})
btemps_ch5	Brightness temperature in channel 5 (6.1146 m^{-1})
chanqual_ch1	Channel quality for channel 1
chanqual_ch2	Channel quality for channel 2
chanqual_ch3	Channel quality for channel 3
chanqual_ch4	Channel quality for channel 4
chanqual_ch5	Channel quality for channel 5
instrtemp	Instrument temperature
qualind	Quality indicator
scanqual	Scan line quality
scnlin	Scan line number
scnlindy	Day of scan line acquisition
scnlintime	Time of scan line acquisition
scnlinyr	Year of scan line acquisition
Latitude	Latitude of pixel
Longitude	Longitude of pixel

Time	Acquisition time in TAI93 seconds
Satellite_azimuth_angle	The satellite azimuth angle
Satellite_zenith_angle	The satellite zenith angle
Solar_azimuth_angle	The solar azimuth angle
Solar_zenith_angle	The solar zenith angle

All per-scan line variables are extended to cover the full pixel grid.

3.4.3 (A)ATSR

ATSR1, ATSR2 and AATSR data in ENVISAT native format (*.N1)

Sensor Key	Source
atsr-e1	ATSR1 on ERS1
atsr-e2	ATSR2 on ERS2
aatsr-en	AATSR on ENVISAT

Variable Name	Description
btemp_nadir_1200	Brightness temperature, nadir view (11500-12500 nm)
btemp_nadir_1100	Brightness temperature, nadir view (10400-11300 nm)
btemp_nadir_0370	Brightness temperature, nadir view (3505-3895 nm)
reflec_nadir_1600	Reflectance, nadir view (1580-1640 nm)
reflec_nadir_0870	Reflectance, nadir view (855-875 nm)
reflec_nadir_0670	Reflectance, nadir view (649-669 nm)
reflec_nadir_0550	Reflectance, nadir view (545-565 nm)
confid_flags_nadir	Confidence flags, nadir view
cloud_flags_nadir	Cloud flags, nadir view
btemp_fward_1200	Brightness temperature, forward view (11500-12500 nm)
btemp_fward_1100	Brightness temperature, forward view (10400-11300 nm)
btemp_fward_0370	Brightness temperature, forward view (3505-3895 nm)
reflec_fward_1600	Reflectance, forward view (1580-1640 nm)
reflec_fward_0870	Reflectance, forward view (855-875 nm)
reflec_fward_0670	Reflectance, forward view (649-669 nm)
reflec_fward_0550	Reflectance, forward view (545-565 nm)
confid_flags_fward	Confidence flags, forward view
cloud_flags_fward	Cloud flags, forward view
lat_corr_nadir	Latitude corrections, nadir view
lon_corr_nadir	Longitude corrections, nadir view
sun_elev_nadir	Solar elevation, nadir view
view_elev_nadir	Satellite elevation, nadir view
sun_azimuth_nadir	Solar azimuth, nadir view
view_azimuth_nadir	Satellite azimuth, nadir view
lat_corr_fward	Latitude corrections, forward view
lon_corr_fward	Longitude corrections, forward view
sun_elev_fward	Solar elevation, forward view

view_elev_fward	Satellite elevation, forward view
sun_azimuth_fward	Solar azimuth, forward view
view_azimuth_fward	Satellite azimuth, forward view
latitude	Latitudes
longitude	Longitudes
altitude	Topographic altitude

All tie-point raster variables are interpolated to the full raster resolution.

3.4.4 AVHRR

AVHRR L1C GAC data in NetCDF format, converted using routines developed at University of Reading.

Sensor Key	Source
avhrr-n06	AVHRR NOAA06
avhrr-n07	AVHRR/2 NOAA07
avhrr-n08	AVHRR NOAA08
avhrr-n09	AVHRR/2 NOAA09
avhrr-n10	AVHRR NOAA10
avhrr-n11	AVHRR/2 NOAA11
avhrr-n12	AVHRR/2 NOAA12
avhrr-n13	AVHRR/2 NOAA13
avhrr-n14	AVHRR/2 NOAA14
avhrr-n15	AVHRR/3 NOAA15
avhrr-n16	AVHRR/3 NOAA16
avhrr-n17	AVHRR/3 NOAA17
avhrr-n18	AVHRR/3 NOAA18
avhrr-n19	AVHRR/3 NOAA19
avhrr-m02	AVHRR/3 METOP A
avhrr-m01	AVHRR/3 METOP B

Variable Name	Description
ch1	Channel 1 reflectance
ch2	Channel 2 reflectance
ch3	Channel 3 reflectance (only AVHRR/2)
ch3a	Channel 3a reflectance (only AVHRR/2 AVHRR/3)
ch3b	Channel 3b brightness temperature
ch4	Channel 4 brightness temperature
ch5	Channel 5 brightness temperature (only AVHRR/3)
cloud_mask	CLAVR-X cloud mask
cloud_probability	CLAVR-X cloud probability
dtime	Scanline time difference from start time
ict_temp	Temperature of internal calibration target
l1b_line_number	Level 1b line number
lat	Latitude coordinates

lon	Longitude coordinates
qual_flags	Quality Flags
relative_azimuth_angle	Relative Azimuth Angle
satellite_zenith_angle	Satellite zenith angle
solar_zenith_angle	Solar zenith angle

3.4.5 HIRS

HIRS L1C data converted to NetCDF using tools developed at University of Reading.

Sensor Key	Source
hirs-tn	HIRS/2 on Tiros-N
hirs-n06 – n12	HIRS/2 on NOAA 6 to NOAA 12
hirs-n14	HIRS/2 on NOAA 14
hirs-n15 – n17	HIRS/3 on NOAA 15 to NOAA 17
hirs-n18 – n19	HIRS/4 on NOAA 18 to NOAA 19
hirs-ma	HIRS/4 on Metop-A
hirs-mb	HIRS/4 on Metop-B

Variable Name	Description
time	Acquisition time per scanline in seconds since 1970
lat	Latitude of pixel
lon	Longitude of pixel
bt_ch01 – ch19	Brightness temperatures for channel 1 to 19
radiance_ch01 – ch19	TOA radiances for channel 1 to 19
counts_ch01 – ch19	Raw measurement counts for channels 1 to 19
lza	Local zenith angle
scanline	Original scanline number
scanpos	Original scan position number
scanline_type	Scanline type flag

3.4.6 MHS

MHS L1c data from NOAA CLASS storage, converted to HDF using “atovin” and “convert_to_hdf5”.

Sensor Key	Source
mhs-n18	MHS on NOAA18
mhs-n19	MHS on NOAA19
mhs-ma	MHS on MetopA
mhs-mb	MHS on MetopB

Variable Name	Description
btemps_ch1 … ch5	Brightness temperatures for channel 1 to 5

chanqual_ch1 … ch5	Channel quality flags
instrtemp	Instrument temperature in Kelvin
qualind	Quality indicator
scanqual	Scan line quality
scnlin	Scan line number
scnlindy	Scan line acquisition day of year
scnlintim	Scan line acquisition seconds of day
scnlinyr	Scan line acquisition year
Latitude	Pixel latitude in degrees
Longitude	Pixel longitude in degrees
Satellite_azimuth_angle	Satellite azimuth angle
Satellite_zenith_angle	Satellite zenith angle
Solar_azimuth_angle	Sun azimuth angle
Solar_zenith_angle	Sun zenith angle

All per-scan line variables are extended to cover the full pixel grid.

3.4.7 SSMT2

SSMT2 L1C data in NetCDF format.

Sensor Key	Source
ssmt2-f11	SSMT/2 on DMSP F11
ssmt2-f12	SSMT/2 on DMSP F12
ssmt2-f14	SSMT/2 on DMSP F14
ssmt2-f15	SSMT/2 on DMSP F15

Variable Name	Description
ancil_data_Year_1	Scan information – year
ancil_data_DayofYear_1	Scan information – day of year
ancil_data_SecondsofDay_1	Scan information – seconds of day
ancil_data_SatLat	Satellite nadir latitude
ancil_data_SatLong	Satellite nadir longitude
ancil_data_SatAlt	Satellite altitude
ancil_data_SatHeading	Satellite heading
ancil_data_Year_2	Scan information – year
ancil_data_DayofYear_2	Scan information – day of year
ancil_data_SecondsofDay_2	Scan information – seconds of day
tb_ch1 … tb_ch5	Brightness temperatures in channels 1 to 5
lon	Pixel longitude
lat	Pixel latitude
channel_quality_flag_ch1 … ch5	Channel quality flags for channels 1 to 5
gain_control_ch1 … ch5	Gain control for channels 1 to 5
counts_to_tb_gain_ch1 … ch5	Conversion scale factors for channel 1 to 5
counts_to_tb_offset_ch1 … ch5	Conversion offset values for channels 1 to 5

Temperature_misc_housekeeping_thermistorcount01 … 18	Thermistor status information for thermistors 1 to 18
warm_counts_ch1_cal1 … ch5_cal4	Warm calibration counts for channels 1 to 5 and views 1 to 4
cold_counts_ch1_cal1 … ch5_cal4	Cold calibration counts for channels 1 to 5 and views 1 to 4
Satellite_zenith_angle	Artificial variable containing the satellite zenith angle derived from the scan position and the platform orbit altitude

3.5 Supported Insitu Data

This chapter lists the input in-situ data source types available for matchup processing. In subsequent chapters, the sensors, the data format and the available variables are described in more detail.

3.5.1 SST-CCI

In-situ data collected for SST-CCI, sea surface temperature. Formatted to NetCDF following the project specifications (RD 4)

Sensor Key	Source
drifter-sst	Drifter based SST data
ship-sst	Ship based SST data
gtmba-sst	Global Tropical Moored Buoy Array SST data
radiometer-sst	Radiometer based SST data
argo-sst	Argo floater based SST data
xbt-sst	Expendable Bathythermograph SST data
mbt-sst	MBT SST data
ctd-sst	CTD SST data
animal-sst	Animal based SST measurements
bottle-sst	Bottle SST data

Variable Name	Description
insitu.time	Acquisition time in seconds since 1978-01-01
insitu.lat	Latitude of measurement position
insitu.lon	Longitude of measurement position
insitu.sea_surface_temperature	SST in degrees Celsius
insitu.sst_uncertainty	Measurement uncertainty in degrees Celsius
insitu.sst_depth	Measurement depth in meters
insitu.sst_qc_flag	In situ sea surface temperature MOHC QC flag
insitu.sst_track_flag	In situ sea surface temperature MOHC track flag
insitu.mohc_id	Unique ID from MOHC database
insitu.id	Unique ID generated from acquisition year, month and mohc_id

4 Configuration

The Fiduceo MMS is a highly configurable software system that allows customisation wherever applicable.

4.1 Configuration Files

This chapter lists the available parameters in various configuration files that are used by the software. All configuration files should be located in a single configuration directory.

4.1.1 System Configuration

The system configuration file “system-config.xml” contains global settings that affect all components of the MMS. An example XML configuration is displayed below; each tag is explained later in the document.

```
<system-config>
  <geometry-library name="S2"/>

  <archive>
    <root-path>
      /usr/local/data/fiduceo
    </root-path>

    <rule sensors="drifter-sst, ship-sst, gtmba-sst, …">
      insitu/SENSOR/VERSION"
    </rule>
  </archive>
</system-config>
```

geometry-library: defines the geometry library to be used for geometric operations. Available values are

- “S2” – use the Google S2 spherical library – the default and recommended value
- “JTS” – use the Java Topology Suite library – experimental

archive: defines all settings related to the satellite data archive.

root-path: defines the root directory containing the input datasets used for matchup processing by default, the archive is organised as

<root>/<sensor-platform>/<version>/<year>/<month>(/<day>)

rule: this tag allows to define specific archiving rules that may differ from the default organisation. Any number of rules may be added to the configuration. A rule can apply to one or many sensors, defined by the “*sensors*” attribute of the rule tag. Multiple sensors are written as comma-separated list of sensor types. All types defined in 3.4 are valid entries for this tag.

A rule defines a general path into the archive. The elements of the path have to be separated by a forward slash “/”. Path elements can be either selected from a list of pre-defined entities or defined as custom elements. The pre-defined entities will be expanded dynamically during runtime; the custom elements are treated as constant path elements. Pre-defined path elements are:

- SENSOR – will be expanded to the sensor name being processed
- VERSION – will be expanded to the data version being processed
- YEAR – will be expanded to the currently processed year
- MONTH - will be expanded to the currently processed month
- DAY - will be expanded to the currently processed day

4.1.2 Database Configuration

The database configuration file “database.properties” defines the connection properties for the MMS database. It is a java properties file containing unstructured key/value pairs.

The following settings are available:

driverClassName: defines the database connection driver class name. The following values are available:

- “mongodb”: use the MongoDb database driver class – the default and recommended value
- “org.postgres,Driver”: use the PostGIS JDBC database driver
- “org.h2.driver”: use the H2 in memory database driver
- “com.mysql.jdbc.driver”: use the MySQL JDBC database driver - experimental

url: defines the database connection URL. The format depends on the driverClassName selected, please consult the documentation of the specific database vendor for details.

username: defines the database user name that is submitted during the database connection process.

password: defines the password that is submitted during the database connection process.

4.1.3 MMD Writer Configuration

The MMD output format and content can be configured using this file. In addition, it allows fine-tuning of the writer performance and behaviour. The configuration file is an XML file named “mmd-writer-config.xml” and located in the standard configuration directory.

An example configuration is displayed below; each tag is explained later in the document.

```
<?xml version="1.0" encoding="UTF-8"?>
<mmd-writer-config>
  <overwrite>false</overwrite>
  <cache-size>2048</cache-size>
  <reader-cache-size>8</reader-cache-size>
  <netcdf-format>N4</netcdf-format>
  <variables-configuration>
    …
```

```
    </variables-configuration>
</mmd-writer-config>
```

overwrite: this tag defines the MMD writer behaviour when encountering an already existing file in the target directory.

- false: software raises an exception and stops
- true: software deletes the existing file and writes the new one.

cache-size: defines the cache size in number of matchups that is used by the mmd-writer. Default value is 2048. Reduce this number when experiencing OutOfMemory errors on large subset window sizes.

reader-cache-size: defines the reader cache size in number of open file readers. The reader cache keeps frequently used input files open to increase the MMD writing performance. This setting defines the number of input files that are kept simultaneously open.

netcdf-format: defines the NetCDF file format version used for the MMD files. Available values are:

- “N3” - write MMDs in NetCDF version 3 format
- “N4” - write MMDs in NetCDF version 4 format – the default value

variables-configuration: allows to rename or exclude variables or attributes. A detailed description is given in the following chapter.

4.1.3.1 Variables Configuration

The variables configuration section of the MMD writer configuration allows fine-tuning the content of the final matchup data files. It is possible to

- rename a sensor
- define a separator per sensor namespace for name concatenation
- exclude variables
- rename variables
- rename attributes of variables

For a detailed description of the content and structure of an MMD file and the naming convention used, please refer to chapter 5.

4.1.3.1.1 Renaming a sensor

In the standard configuration the sensors reference are named as they appear in the descriptions in chapters 3.4 and 3.5. This behaviour can be changed using the sensor-rename tag. Multiple sensor-rename tags are allowed inside a variables configuration.

Example:

```
<mmd-writer-config>
  <variables-configuration>
    <sensor-rename source-name="amsub-n16" target-name="AMSU-B_NOAA16" />
  </variables-configuration>
</mmd-writer-config>
```

4.1.3.1.2 Define a separator per sensor namespace

The MMS stores input variables to the MMD using the pattern <sensor-name><separator><variable-name>, for example "ssmt2-f14_tb-ch4". This configuration aspect allows to change the separator character, which is set to the underscore character '_' by default. Separators can be associated to one or many sensors, which have to be added as a comma-separated list.

Example:

```
<mmd-writer-config>
  <variables-configuration>
    <separator sensor-names="sen1, sen2, ..." separator = "." />
  </variables-configuration>
</mmd-writer-config>
```

4.1.3.1.3 Exclude variables

It is possible to exclude variables of the input data from being written to the MMD file. The exclude tag allows to exclude any number of variables from a given sensors variables list. The names appearing in the exclude tag must exactly match the variable names of the input products as listed in chapters 3.4 and 3.5.

Example:

```
<mmd-writer-config>
  <variables-configuration>
    <sensors names="hirs-n17, hirs-n16">
      <exclude source-name="not_needed" />
      <exclude source-name="useless" />
    </sensors>
  </variables-configuration>
</mmd-writer-config>
```

4.1.3.1.4 Rename variables

Using this tag, it is possible to rename variables, i.e. to define a different name for the variable in the MMD as it originally appeared in the input data. This tag only changes the variable name, the sensor and separator parts of the final composite name are not affected, these can be changed separately. The names of the source variables appearing in the rename tag must exactly match the variable names of the input product as listed in chapters 3.4 and 3.5.

Example:

```
<mmd-writer-config>
```

```
  <variables-configuration>
    <sensors names="hirs-n17, hirs-n16">
      <rename source-name="stupid" target-name="cool" />
      <rename source-name="boring" target-name="exciting" />
    </sensors>
  </variables-configuration>
</mmd-writer-config>
```

4.1.3.1.5 Rename attributes of variables

It is possible to adapt the names of attributes of any variable in the MMD. The names of the source variables and attribute names appearing in the rename-attribute tag must exactly match the variable and attribute names of the input product as listed in chapters 3.4 and 3.5.

```
<mmd-writer-config>
 <variables-configuration>
  <sensors names="hirs-n17, hirs-n16">
   <rename-attribute source-name="attN1" target-name="attN1_r" />
   <rename-attribute variable-names="var1" source-name="attN2" target-name="attN2_r" />
   <rename-attribute variable-names="var1, var2" source-name="attN3" target-name="attN3_r" />
  </sensors>
 </variables-configuration>
</mmd-writer-config>
```

4.1.4 UseCase Configuration

This configuration file defines the parameter-set to be used for a matchup processing. It is an XML file that has to be located in the configuration directory used for processing.

An example configuration is displayed below; each tag is explained later in the document.

```
<?xml version="1.0" encoding="UTF-8"?>
<use-case-config name="mmd02">
  <sensors>
    <sensor>
      <name>avhrr-n17</name>
      <primary>false</primary>
    </sensor>
    <sensor>
      <name>avhrr-n18</name>
      <primary>true</primary>
    </sensor>
  </sensors>

  <dimensions>
    <dimension name="avhrr-n17">
      <nx>5</nx>
      <ny>5</ny>
    </dimension>
```

```
    <dimension name="avhrr-n18">
      <nx>5</nx>
      <ny>5</ny>
    </dimension>
  </dimensions>

  <write-distance>true</write-distance>
  <output-path>/tmp/fiduceo_test/usecase-02</output-path>

  <conditions>
    <time-delta>
      <time-delta-seconds>10800</time-delta-seconds>
    </time-delta>
    <spherical-distance>
      <max-pixel-distance-km>3.0</max-pixel-distance-km>
    </spherical-distance>
  </conditions>

  <screenings>
    <angular>
      <primaryVZAVariable name="satellite_zenith_angle" />
      <secondaryVZAVariable name="satellite_zenith_angle" />
      <maxAngleDelta>10.0</maxAngleDelta>
    </angular>
  </screenings>

  <num-random-seed-points>20000</num-random-seed-points>
</use-case-config>
```

Each usecase configuration file starts with an XML version and character encoding statement. The enclosing <use-case-config name="mmd02"> tag defines the content of the configuration. Each usecase configuration has to be given a name that is set in the “name” attribute as shown above. This name is used during MMD file generation to create the result file names.

sensors: this tag encloses a list of sensors to be processed. Each sensor element in this list must contain a valid <name> tag that defines the senor name. The sensor names must match one name of the list of supported sensors, see chapters 3.4 and 3.5. One of the sensors must be tagged as being the primary (i.e. reference) sensor in this list of sensors. This is accomplished by adding a primary tag with the value “true”:

```
<primary>true</primary>
```

The primary tag with a value of “false” can be omitted, the software automatically assumes this when the tag is missing.

Optionally, a sensor data can be restricted to a single data/processing version. This can be expressed by adding a version tag:

```
<data-version>v2.08</data-version>
```

When the version tag is omitted, the MMS uses all available data versions for this sensor for processing.

dimensions: enclosed by this tag is a list of dimension tags. These define the output window size for each matchup, i.e. the size of the data extract stored to the MMD with the matchup x/y location in the centre. It is required that for each sensor defined in the sensor list, an associated dimension is being defined. The association is realised by matching the names, stored as dimension attribute “name”.

The tags “nx” and “ny” define the extent of each dimension, they define the complete width and height of the extract. Following the requirement of a central matchup pixel, both values have to be odd numbers.

write-distance: specifies whether the MMD file shall contain an additional variable that stores the matchup centre distance in kilometres. Valid values are “true” and “false”, defaults to “false”.

output-path: defines the target directory for the processing. All MMD files generated will be stored in the directory denoted in this tag.

conditions: this tag encloses a list of condition plugin configurations. The values in each configuration are depending on the plugin and are explained in more detail in chapter 4.1.4.1.

screenings: this tag encloses a list of screening plugin configurations. The values in each configuration are depending on the plugin and are explained in more detail in chapter 4.1.4.2.

num-random-seed-points: specifies the number of global generated pseudo random seed points (sobol generated points). The points are nearly equidistant distributed on the globe and also nearly equidistant distributed in the time range defined by the matchup tool command line arguments. (See 3.2 Matchup Tool)

If number of random seed points is bigger than 0 the “Seed Point Matchup Strategy” is used to find matchup pixels between primary and secondary sensors.

The “Seed Point Matchup Strategy” is helpful to avoid that all the pixels of the entire observations are matchups, e.g. if the primary and secondary observation geometries are congruent, because they are on the same platform.

4.1.4.1 Conditions

This chapter lists the configuration settings for all condition plugins available for the MatchupTool. A condition plugin serves for checking that a matchup conforms to the condition defined. A condition plugin operates solely on the raw matchup information, i.e.

- Primary and secondary pixel
  - Geo-location longitude and latitude
  - Pixel raster location x and y
  - Pixel acquisition time in seconds since epoch
- Primary and secondary product raster width and height
- Primary and secondary extraction window sizes

- The processing time-range

When one or more of the subsequent plugin configurations are present in the configuration, the corresponding plugin is loaded and configured using the values supplied. During processing, the chain of plugins is applied to the matchups. The order of the condition configurations defines the processing order.

4.1.4.1.1 Border Distance Condition

This condition ensures that the matchup location within the data file raster is within a defined distance from the raster borders. The condition can check primary and secondary acquisition positions using different parameters, according to the configuration. If the pixel is outside the defined boundaries, the matchup is rejected. Example:

```
<border-distance>
  <primary>
    <nx>22</nx>
    <ny>6</ny>
  </primary>

  <secondary>
    <nx>5</nx>
    <ny>8</ny>
  </secondary>
</border-distance>
```

“primary”: if tag is present, the primary (i.e. reference) sensor position is checked

“secondary”: if tag is present, the secondary sensor position is checked

“nx” defines the minimal distance of the matchup centre location x coordinate to the left or right data raster boundary.

“ny” defines the minimal distance of the matchup centre location y coordinate to the upper or lower data raster boundary.

4.1.4.1.2 Spherical Distance Condition

This condition ensures that the primary and secondary matchup location are within a given spherical distance. If the matchup pixel centre locations (in lon/lat) are further apart than the threshold value, the matchup is rejected. Example:

```
<spherical-distance>
  <max-pixel-distance-km>6</max-pixel-distance-km>
</spherical-distance>
```

4.1.4.1.3 Time Delta Condition

This condition verifies that the pixel acquisition time difference of the contributing primary and secondary products do not exhibit the configured threshold. All pixels with an acquisition time difference above the threshold are rejected. Example:

```
<time-delta>
   <time-delta-seconds>10800</time-delta-seconds>
</time-delta>
```

4.1.4.1.4 Overlap Remove Condition

This condition plugin removes overlapping extraction windows. The plugin can operate either on the primary or on the secondary sensor extracts. It checks all matchup pixels for overlapping extraction windows and rejects all appropriate, so that the remaining list of matchups does not have overlapping areas. Example:

```
<overlap-remove>
   <reference>PRIMARY</reference>
</overlap-remove>
```

“reference”: denotes the sensor that shall be used for overlap removal. Valid choices are “PRIMARY” and “SECONDARY”.

4.1.4.2 Screenings

This chapter lists the configuration settings for all screening plugins available for the MatchupTool. A screening plugin serves for checking that a matchup conforms to the condition defined. A screening plugin operates on the raw matchup information (see chapter 4.1.4.1) and additionally has access to the product data readers.

When one or more of the subsequent plugin configurations are present in the configuration, the corresponding plugin is loaded and configured using the values supplied. During processing, the chain of plugins is applied to the matchups. The order of the screening configurations defines the processing order.

4.1.4.2.1 Angular Screening

This plugin performs a number of screenings on the satellite or viewing zenith or azimuth angles. The names of the variables can be configured, as well as the screenings that shall be applied and the associated thresholds. A screening is executed only when the associated tag is present in the configuration. Example:

```
<angular>
   <primary-vza-variable name=”zenith_angle”/>
   <secondary-vza-variable name=”satellite_zenith_angle”/>
   <max-primary-vza>10.0</max-primary-vza>
   <max-secondary-vza>15.0</max-secondary-vza>
```

```
   <max-angle-delta>17.0</max-angle-delta>
</angular>
```

“primary-vza-variable” denotes the variable name containing the angle to be screened for the primary sensor.

“secondary-vza-variable” denotes the variable name containing the angle to be screened for the secondary sensor.

“max-primary-vza” denotes the threshold for the primary angular variable in decimal degrees. All matchups with a value higher than the threshold value are rejected. Requires “primary-vza-variable” to be set. If tag is not present, the screening is not applied.

“max-secondary-vza” denotes the threshold for the secondary angular variable in decimal degrees. All matchups with a value higher than the threshold value are rejected. Requires “secondary-vza-variable” to be set. If tag is not present, the screening is not applied.

“max-angle-delta” denotes the maximal angular difference absolute value in decimal degrees. Matchups with an angular difference higher than the threshold are rejected. Requires “primary-vza-variable” and “secondary-vza-variable” to be set. If tag is not present, the screening is not applied.

4.1.4.2.2 Angular Cosine Proportion Screening

This plugin performs a screening on satellite viewing zenith angles as defined by the University of Hamburg (see RD 1), following the equation:

$$\left|\frac{\cos(primaryVZA)}{\cos(secondaryVZA)} - 1\right| < \varepsilon_1$$

All matchups with a proportion value exceeding the threshold are rejected. Example configuration:

```
<angular-cosine-proportion>
   <primary-variable name=”zenith_angle”/>
   <secondary-variable name=”satellite_zenith_angle”/>
   <threshold>0.02</threshold>
</angular-cosine-proportion>
```

“primary-variable” denotes the variable name containing the angle to be screened for the primary sensor.

“secondary-variable” denotes the variable name containing the angle to be screened for the primary sensor.

“threshold” denotes the threshold value (i.e. the epsilon$_1$ in the equation above).

4.1.4.2.3 HIRS LZA Angular Screening

The LZA variable in the HIRS L1C data describes the local zenith angle for each scanline x-position. It does not distinguish left and right nadir positions. To implement a correct angular difference screening, the scanposition dataset has to be taken into account.

This plugin implements a correct angular difference screening for HIRS data following the algorithm:

$$sigma_{primary} = scanpos_{primary} < 28 \ ? -1.0 : 1.0$$

$$sigma_{secondary} = scanpos_{secondary} < 28 \ ? -1.0 : 1.0$$

$$angDelta = abs(sigma_{primary} * lza_{primary} - sigma_{secondary} * lzas_{econdary})$$

All matchups with an angDelta above the “max-lza-delta” threshold are rejected. Values are in decimal degrees. Example configuration:

```
<hirs-lza-delta>
   <max-lza-delta>10.0</max-lza-delta>
</hirs-lza-delta>
```

4.1.4.2.4 Buehler Cloud Screening

This plugin performs a cloud screening for microwave sensors based on the algorithm described in RD 2 and RD 3. It is applicable to AMSU-B, MHS and SSM/T-2. It can be applied either to primary, secondary or both sensor data. The screening is only executed when all three configuration values for the sensors are set. To switch the screening for a sensor off, remove the associated tags from the configuration. All matchups detected as cloudy are rejected.

Example configuration:

```
<buehler-cloud>
   <primary-narrow-channel name=”btemp_ch18”/>
   <primary-wide-channel name=”btemp_ch20”/>
   <primary-vza name=”Satellite_zenith_angle”/>

   <secondary-narrow-channel name=”btemp_ch3”/>
   <secondary-wide-channel name=”btemp_ch4”/>
   <secondary-vza name=”Satellite_zenith_angle”/>
</buehler-cloud>
```

“primary-narrow-channel” denotes the variable name containing the narrow bandwidth channel for the primary sensor.

“primary-wide-channel” denotes the variable name containing the wide bandwidth channel for the primary sensor.

“primary-vza” denotes the variable name containing the satellite zenith angle values for the primary sensor.

"secondary-narrow-channel" denotes the variable name containing the narrow bandwidth channel for the secondary sensor.

"secondary-wide-channel" denotes the variable name containing the wide bandwidth channel for the secondary sensor.

"secondary-vza" denotes the variable name containing the satellite zenith angle values for the secondary sensor.

4.1.4.2.5 Pixel Value Screening

This screening plugin allows selecting matchups using a mathematical expression consisting of any combination of input variables. The plugin allows entering different expressions for primary and secondary sensor. Mathematical expressions can be arbitrarily complex but must evaluate to a Boolean value. The mathematical expressions are evaluated using the matchup location only. All matchups where either the primary or the secondary expression evaluate to false are rejected.

Mathematical expressions can be composed of variables and functions.

Variables available are all variables contained in the input satellite or in-situ data using the name as it appears in the file and as documented in section chapters 3.4 and 3.5, e.g. "btemps_ch5" for AMSU-B brightness temperature in channel 5. In addition to this, the constants "PI" and "E" are available denoting pi and e. And of course any number typed into the expression is treated as numerical constant.

Mathematical operations and functions available are:

Function	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
==	Equality comparison
!=	Inequality comparison
>	Greater than
>=	Greater than or equal
<	Less than
<=	Less than or equal
abs	Calculates absolute value of argument
acos	Arc cosine of argument
ampl	Calculates amplitude of arguments ($a^2 + b^2$)
asin	Arc sine of argument
atan	Arc tangent of argument
atan2	The theta component of the point (r, theta) in polar coordinates that corresponds to the point (x, y) in Cartesian coordinates. (https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#atan2(double,%20double))
avg	Mean value of arguments

bit_set	Checks whether a bit is set or not, first argument is the value to check, second argument is the bit index (e.g. bit_set(cloud_flags_fward , 1))
ceil	Round upwards to next integer value
cos	Cosine of argument
cosech	Hyperbolic cosecant of argument
cosh	Hyperbolic cosine of argument
deg	Converts argument from radians to decimal degrees
distance	Calculates the distance (sqrt of sum of squares)
exp	Exponential of argument
exp10	Ten to the power of argument
feq	Fuzzy equality (1e-6 accuracy) or when used with three arguments, third argument supplies maximal delta.
floor	Round downwards to next integer value
fneq	Fuzzy inequality (1e-6 accuracy) or when used with three arguments, third argument supplies maximal delta.
inf	Check argument for infinity. Returns 1 if argument is positive or negative infinite, returns 0 otherwise
log	Natural logarithm of argument
log10	Logarithm to the base of 10 of argument
max	Maximum value of arguments
min	Minimum value of arguments
nan	Checks argument for “not a number”. Returns 1 if argument is not a number, 0 otherwise
phase	Calculates phase of arguments (atan2(b,a))
pow	General power function a^b
rad	Converts argument in decimal degrees to radians
random_gaussian	Generates a random number from a gaussian distributed process
random_uniform	Generates a random number from a uniformly distributed process
round	Round to closes integer value
sech	Hyperbolic secant of argument
sign	Compute signum, returns +1 for positive, -1 for negative and 0 for zero argument
sin	Sine of argument
sinh	Hyperbolic sine of argument
sq	Square of argument
sqrt	Square root of argument
tan	Tangent of argument
tanh	Hyperbolic tangent of argument

All matchups where the mathematical expression for either primary or secondary sensor evaluates to false are rejected. Example configuration:

```
<pixel-value>
  <primary-expression>btemp_ch4 >= 180.5</primary-expression>
  <secondary-expression>cloud_probability < 0.25</secondary-expression>
</pixel-value>
```

4.1.4.2.6 Window Value Screening

This screening plugin allows selecting matchups using a mathematical expression consisting of any combination of input variables. The plugin allows entering different expressions for primary and secondary sensor. Mathematical expressions can be arbitrarily complex but must evaluate to a Boolean value. The mathematical expressions are evaluated for each pixel of a window defined by the matchup location and an interval defined by the use case configuration.

Additional to the mathematical expression two more properties must be defined. The percentage and the evaluate property. The evaluate property decide whether pixels with fill value should be ignored or all pixels of the window are used to calculate the percentage pixel count.

All matchups where either the primary or the secondary percentage evaluate to less pixels are valid are rejected.

Mathematical expressions can be composed of variables and functions.

Variables available are all variables contained in the input satellite or in-situ data using the name as it appears in the file and as documented in section chapters 3.4 and 3.5, e.g. “btemps_ch5” for AMSU-B brightness temperature in channel 5. In addition to this, the constants “PI” and “E” are available denoting pi and e. And of course any number typed into the expression is treated as numerical constant.

Mathematical operations and functions available are the same as in 4.1.4.2.5.

Example configuration:

```
<window-value>
   <primary>
      <expression>btemp_ch4 >= 180.5</expression>
      <percentage>22.3</percentage>
      <evaluate>EntireWindow</evaluate>  // second option is “IgnoreNoData”
   </primary>
   <secondary>
      <expression> cloud_probability < 0.25</expression>
      <percentage>50.0</percentage>
      <evaluate>IgnoreNoData</evaluate>
   </secondary >
</window-value>
```

4.1.5 Post Processing Configuration

The post processing configuration file stores the parameters and configuration used by the post-processing tool. It is an XML file that has to be located in the general configuration files directory, see chapter 4.1. It contains the main tag *<post-processing-config>* and a list of *<post-processings>* which contain the configurations of the plugins. A list of implemented post processing plugins is described in 4.1.5.1.

Also the configuration file contains information whether the input mmd files should be overwritten in place

```
<overwrite>
```

or new extended files should be written to the output directory

```
<create-new-files>
  <output-directory><!—the existing output directory path --></output-directory>
</create-new-files>
```

<u>Example Post Processing Configuration File:</u>

```
<post-processing-config>
  <create-new-files>
    <output-directory><!—the existing output directory path --></output-directory>
  </create-new-files>
  <post-processings>
    ... <!-- List of plugin configurations -->
  </post-processings>
</post-processing-config>
```

Plugins can be chained and are processed in the order they appear in the configuration file.

4.1.5.1 Post Processing Plugins

Each post processing plugin has its own XML signature.

4.1.5.1.1 Spherical Distance Plugin

The spherical distance plugin is a post processing plugin, which can be used to add a spherical distance variable to a matchup dataset i.e. the distance of the matchup acquisitions for primary and additional sensor, calculated as geodesic distance in kilometers.

Requirements:

- The matchup dataset must contain latitude and longitude information for primary and secondary sensor.
- This geo information can be organized in 1 by 1, 3 by 3, 5 by 5 or each other odd manner.
- The matchup dataset must not contain a variable with the name, defined in **<var-name>** tag.

```
<spherical-distance>
  <target>
    <data-type>Float</data-type>
    <var-name>post_dist</var-name>
    <dim-name>matchup_count</dim-name>
  </target>

  <primary-lat-variable scaleAttrName="Scale">amsub-n16_Latitude</primary-lat-variable>
  <primary-lon-variable scaleAttrName="Scale">amsub-n16_Longitude</primary-lon-variable>
  <secondary-lat-variable>ssmt2-f14_lat</secondary-lat-variable>
  <secondary-lon-variable>ssmt2-f14_lon</secondary-lon-variable>
</spherical-distance>
```

If the geo information variable contains values which must be scaled to get the real lat/lon information, in the plugin XML the names of scaling (*scaleAttrName*) and offset (*offsetAttrName*) attributes have to be supplied. The plugin scales the variables accordingly before calculating the geodesic distance. If scaling or offset attributes are defined but not available, a runtime exception will be thrown.

The primary and secondary lat/lon value will always be extracted from the center pixel.

4.1.5.1.2 SST Insitu Timeseries Plugin

This plugin extracts a time series from in the situ data files and appends the data to the mmd file.

Requirements:

- An input mmd file must contain variables containing "_insitu." in its name. The part before "_insitu." is interpreted as the insitu sensor type.
  e.g. "gtmba-sst_insitu.id" resolves to insitu sensor type "gtmba-sst"
- An input mmd file must contain a variable named
  "<insitu-sensor-type>_file_name"
  which contains the original insitu data file name per matchup
- An input mmd file must contain a dimension named "file_name" which defines the field size of a file_name field in the "<insitu-sensor-type>_file_name variable.
- An input mmd file must contain a dimension named "matchup_count" which defines the number of matchups in this file.

Example sst time series post processing plugin xml part:

```
<sst-insitu-time-series>
  <version>v03.3</version>
  <time-range-in-seconds>129600</time-range-in-seconds>
  <time-series-size>220</time-series-size>
  <secondary-sensor-matchup-time-variable>
    amsre.acquisition_time
  </secondary-sensor-matchup-time-variable>
</sst-insitu-time-series>
```

"version": defines the processing version of the insitu data files. Also encodes the data archive path.

"time-range-in-seconds": defines the complete time range extract range from the insitu acquisitons

"time-series-size": defines the size of the extract – should match the maximum number of acquisitions possible within the extraction time range

"secondary-sensor-matchup-time-variable": variable name for the reference times

The plugin expects that the insitu file archive is organized without year/month/day paths. This means that all the insitu files of same sensor type and version should be in one directory.

Example archive sniped from system-config.xml

```
<system-config>
   . . .
   <archive>
      <root-path><!-- archive root path --></root-path>
      <rule sensors="animal-sst, gtmba-sst">insitu/SENSOR/VERSION</rule>
   </archive>
   . . .
</system-config>
```

Time series variables created by the plugin:

- insitu.dtime [int] (time delta between secondary matchup time and insitu time)
- insitu.id [long] (this a unique id generated by combining YEAR, MONTH and mohc_id)
- insitu.lat [float]
- insitu.lon [float]
- insitu.mohc_id [int]
- insitu.sea_surface_temperature [float]
- insitu.sst_depth [float]
- insitu.sst_qc_flag [short]
- insitu.sst_track_flag [short]
- insitu.sst_uncertainty [float]
- insitu.time [int]
- insitu.y [int]

4.1.5.1.3 NPW plugin

This plugin adds numerical weather prediction (NWP) data to matchup datasets. The NWP data is extracted from ERA-Interim data (http://www.ecmwf.int/en/research/climate-reanalysis/era-interim). The plugin allows two different processes to be executed, a time series extraction for single pixel matchups (like insitu data) or an n x m pixel extract with profile data for atmospheric parameters for satellite-sensor extracts. The plugin relies on the publicly available CDO software (https://code.zmaw.de/projects/cdo) to perform the time-series aggregation and projection tasks.

The parametrization for the time-series and sensor-extracts is configured in specific sub-tags, see example below.

Configuration example:

```
<nwp>
```

```
<cdo-home>/usr/local/bin/cdo</cdo-home>
<nwp-aux-dir>/the/auxiliary/files</nwp-aux-dir>
<delete-on-exit>true</delete-on-exit>
<temp-dir>/temp/dir</temp-dir >

<time-series-extraction>
   <analysis-steps>19</analysis-steps>
   <forecast-steps>33</forecast-steps>

   <time-variable-name>acquisition-time</time-variable-name>
   <longitude-variable-name>animal-sst_inistu.lon</longitude-variable-name>
   <latitude-variable-name>animal-sst_inistu.lat</latitude-variable-name>

   <analysis-center-time-name>matchup.nwp.an.t0</analysis-center-time-name>
   <forecast-center-time-name>matchup.nwp.fc.t0</forecast-center-time-name>

   <an-ci-name>an_sea-ice-fraction</an-ci-name>
   <an-sstk-name>an_sea-surface-temperature</an-sstk-name>
   <fc-sstk-name>fc_sea-surface-temperature</fc-sstk-name>
   <an-u10-name>an_east_wind</an-u10-name>
   <an-v10-name>an_north_wind</an-v10-name>
   <fc-u10-name>10m_east_wind_component</fc-u10-name>
   <fc-v10-name>10m_north_wind_component</fc-v10-name>
   <fc-msl-name>mean_surface_pressure</fc-msl-name>
   <fc-t2-name>2m-temperature</fc-t2-name>
   <fc-d2-name>2m_dew_point</fc-d2-name>
   <fc-tp-name>total_precipitation</fc-tp-name>
   <an-clwc-name>cloud_liquid_water_content</an-clwc-name>
   <fc-clwc-content-name>cloud_liquid_water_content</fc-clwc-name>
   <an-tcwv-name>an_total_column_water_vapour</an-tcwv-name>
   <fc-tcwv-name>fc_total_column_water_vapour</fc-tcwv-name>
   <fc-sshf-name>fc_surface_sensible_heat_flux</fc-sshf-name>
   <fc-slhf-name>surface_latent_heat_flux</fc-slhf-name>
   <fc-blh-name>boundary_layer_height</fc-blh-name>
   <fc-ssrd-name>fc_downward_surface_solar_radiation</fc-ssrd-name>
   <fc-strd-name>fc_downward_surface_thermal_radiation</fc-strd-name>
   <fc-ssr-name>fc_surface_solar_radiation</fc-ssr-name>
   <fc-str-name>fc_surface_thermal_radiation</fc-str-name>
   <fc-ewss-name>fc_turbulent_stress_east_component</fc-ewss-name>
   <fc-nsss-name>fc_turbulent_stress_north_component</fc-nsss-name>
   <fc-e-name>fc_evaporation</fc-e-name>
</time-series-extraction>

<sensor-extraction>
   <time-variable-name>acquisition_time</time-variable-name>
   <longitude-variable-name>amsre.longitude</longitude-variable-name>
   <latitude-variable-name>amsre.latitude</latitude-variable-name>

   <x-dimension>5</x-dimension>
```

```
        <x-dimension-name>nwp_x</x-dimension-name>
        <y-dimension>5</y-dimension>
        <y-dimension-name>nwp_y</y-dimension-name>
        <z-dimension>60</z-dimension>
        <z-dimension-name>nwp_z</z-dimension-name>

        <an-ci-name>seaice_fraction</an-ci-name>
        <an-asn-name>snow_albedo</an-asn-name>
        <an-sstk-name>sea_surface_temperature</an-sstk-name>
        <an-tcwv-name>total_column_water_vapour</an-tcwv-name>
        <an-msl-name>mean_sea_level_pressure</an-msl-name>
        <an-tcc-name>total_cloud_cover</an-tcc-name>
        <an-u10-name>10m_east_wind_component</an-u10-name>
        <an-v10-name>north_wind</an-v10-name>
        <an-t2-name>2m_temperature</an-t2-name>
        <an-d2-name>2m_dew_point</an-d2-name>
        <an-al-name>albedo</an-al-name>
        <an-lnsp-name>log_surface_pressure</an-lnsp-name>
        <an-skt-name>skin_temperature</an-skt-name>
        <an-t-name>temperature_profile</an-t-name>
        <an-q-name>water_vapour_profile</an-q-name>
        <an-o3-name>ozone_profile</an-o3-name>
        <an-clwc-name>cloud_liquid_water</an-clwc-name>
        <an-ciwc-name>cloud_ice_water</an-ciwc-name>
        <an-tp-name>total_precipitation</an-tp-name>
    <sensor-extraction>

</nwp>
```

Configuration parameters are explained in detail below:

- “cdo-home”: denotes the absolute path to the root directory of the binary CDO operators installation.
- “nwp-aux-dir”: defines the directory where the ERA-Interim auxiliary files are located.
- “delete-on-exit”: Set whether to delete all temporary files after processing or not. Default value is: true
- “temp-dir”: sets the temporary directory to be used. The directory must exits and be writable. If not set, the standard system temporary directory is used

Configuration parameter for the time-series extraction process:

- “analysis-steps”: Defines the number of time steps around the matchup time for NWP analysis data (6 hr time resolution). Default is: 17.
- “forecast-steps”: Defines the number of time steps around the matchup time for NWP forecast data (3 hr time resolution). Default is: 33.
- “time-variable-name”: Defines the name of the time variable to use as reference time. Time variables are expected to store data in seconds since 1970 format.

- “longitude-variable-name”: Defines the name of the longitude variable to use as reference. Data is expected in decimal degrees east.
- “latitude-variable-name”: Defines the name of the longitude variable to use as reference. Data is expected in decimal degrees north.
- “analysis-center-time-name”: Defines the name of the variable for analysis center times. Values are in seconds since 1970-01-01. Default: matchup.nwp.an.t0.
- “forecast-center-time-name”: Defines the name of the variable for forecast center times. Values are in seconds since 1970-01-01. Default: matchup.nwp.fc.t0.
- “an-ci-name”: Defines the name of the target variable for analysis sea-ice-fraction (CI). Default: matchup.nwp.an.sea_ice_fraction.
- “an-sstk-name”: Defines the name of the target variable for analysis sea surface temperature (SSTK). Default: matchup.nwp.an.sea_surface_temperature.
- “fc-sstk-name”: Defines the name of the target variable for forecast sea surface temperature (SSTK). Default: matchup.nwp.fc.sea_surface_temperature.
- “an-u10-name”: Defines the name of the target variable for analysis 10m east wind component (U10). Default: matchup.nwp.an.10m_east_wind_component.
- “an-v10-name”: Defines the name of the target variable for analysis 10m north wind component (V10). Default: matchup.nwp.an.10m_north_wind_component.
- “fc-u10-name”: Defines the name of the target variable for forecast 10m east wind component (U10). Default: matchup.nwp.fc.10m_east_wind_component.
- “fc-v10-name”: Defines the name of the target variable for forecast 10m north wind component (V10). Default: matchup.nwp.fc.10m_north_wind_component.
- “fc-msl-name”: Defines the name of the target variable for forecast mean sea level pressure (MSL). Default: matchup.nwp.fc.mean_sea_level_pressure.
- “fc-t2-name”: Defines the name of the target variable for forecast 2m temperature (T2). Default: matchup.nwp.fc.2m_temperature.
- “fc-d2-name”: Defines the name of the target variable for forecast 2m dew point (D2). Default: matchup.nwp.fc.2m_dew_point.
- “fc-tp-name”: Defines the name of the target variable for forecast total precipitation (TP). Default: matchup.nwp.fc.total_precipitation.
- “an-clwc-name”: Defines the name of the target variable for analysis cloud liquid water content (CLWC). Default: matchup.nwp.an.cloud_liquid_water_content
- “fc-clwc-name”: Defines the name of the target variable for forecast cloud liquid water content (CLWC). Default: matchup.nwp.fc.cloud_liquid_water_content
- “an-tcwv-name”: Defines the name of the target variable for analysis total column water vapour (TCWV). Default: matchup.nwp.an.total_column_water_vapour
- “fc-tcwv-name”: Defines the name of the target variable for forecast total column water vapour (TCWV). Default: matchup.nwp.fc.total_column_water_vapour
- “fc-sshf-name”: Defines the name of the target variable for forecast surface sensible heat flux (SSHF). Default: matchup.nwp.fc.surface_sensible_heat_flux
- “fc-slhf-name”: Defines the name of the target variable for forecast latent sensible heat flux (SLHF). Default: matchup.nwp.fc.surface_latent_heat_flux.
- “fc-blh-name”: Defines the name of the target variable for forecast boundary layer height (BLH). Default: matchup.nwp.fc.boundary_layer_height.
- “fc-ssrd-name”: Defines the name of the target variable for forecast downward surface solar radiation (SSRD). Default: matchup.nwp.fc.downward_surface_solar_radiation.
- “fc-strd-name”: Defines the name of the target variable for forecast downward surface thermal radiation (STRD). Default: matchup.nwp.fc.downward_surface_thermal_radiation.

- “fc-ssr-name”: Defines the name of the target variable for forecast surface solar radiation (SSR). Default: matchup.nwp.fc.surface_solar_radiation.
- “fc-str-name”: Defines the name of the target variable for forecast surface thermal radiation (STR). Default: matchup.nwp.fc.surface_thermal_radiation.
- “fc-ewss-name”: Defines the name of the target variable for forecast turbulent stress east component (EWSS). Default: matchup.nwp.fc.turbulent_stress_east_component.
- “fc-nsss-name”: Defines the name of the target variable for forecast turbulent stress north component (NSSS). Default: matchup.nwp.fc.turbulent_stress_north_component.

Configuration parameter for the sensor extraction process:

- “time-variable-name”: Defines the name of the time variable to use as reference time. Time variables are expected to store data in seconds since 1970 format.
- “longitude-variable-name”: Defines the name of the longitude variable to use as reference. Data is expected in decimal degrees east.
- “latitude-variable-name”: Defines the name of the longitude variable to use as reference. Data is expected in decimal degrees north.
- “x-dimension”: defines the extract window extent in x direction.
- “x-dimension-name“: defines the name of the NetCDF dimension for the x direction extent of the NWP extract.
- “y-dimension”: defines the extract window extent in y direction.
- “y-dimension-name“: defines the name of the NetCDF dimension for the y direction extent of the NWP extract.
- “z-dimension”: defines the extract window extent in z direction (i.e. atmospheric layers).
- “z-dimension-name“: defines the name of the NetCDF dimension for the z direction extent of the NWP extract.
- “an-ci-name”: Defines the name of the target variable for analysis sea-ice-fraction (CI). Default: amsre.nwp.seaice_fraction.
- “an-asn-name”: Defines the name of the target variable for analysis snow albedo (ASN). Default: amsre.nwp.snow_albedo.
- “an-sstk-name”: Defines the name of the target variable for analysis sea surface temperature (SSTK). Default: amsre.nwp.sea_surface_temperature.
- “an-tcwv-name”: Defines the name of the target variable for analysis total column water vapour content (TCWV). Default: amsre.nwp.total_column_water_vapour.
- “an-msl-name”: Defines the name of the target variable for analysis mean sea level pressure (MSL). Default: amsre.nwp.mean_sea_level_pressure.
- “an-tcc-name”: Defines the name of the target variable for analysis total cloud coverage (TCC). Default: amsre.nwp.total_cloud_cover.
- “an-u10-name”: Defines the name of the target variable for analysis 10m east wind component (U10). Default: amsre.nwp.10m_east_wind_component.
- “an-v10-name”: Defines the name of the target variable for analysis 10m north wind component (V10). Default: amsre.nwp.10m_north_wind_component.
- “an-t2-name”: Defines the name of the target variable for analysis 2m temperature (T2). Default: amsre.nwp.2m_temperature.
- “an-d2-name”: Defines the name of the target variable for analysis 2m dew point (D2). Default: amsre.nwp. 2m_dew_point.

- “an-al-name”: Defines the name of the target variable for analysis albedo (AL). Default: amsre.nwp.albedo.
- “an-lnsp-name”: Defines the name of the target variable for analysis logarithmic surface pressure (LNSP). Default: amsre.nwp. log_surface_pressure.
- “an-skt-name”: Defines the name of the target variable for analysis skin temperature (SKT). Default: amsre.nwp.skin_temperature.
- “an-t-name”: Defines the name of the target variable for analysis temperature profile (T). Default: amsre.nwp.temperature_profile.
- “an-q-name”: Defines the name of the target variable for analysis water vapour profile (Q). Default: amsre.nwp.water_vapour_profile.
- “an-o3-name”: Defines the name of the target variable for analysis ozone profile (O3). Default: amsre.nwp.ozone_profile.
- “an-clwc-name”: Defines the name of the target variable for analysis cloud liquid water (CLWC). Default: amsre.nwp.cloud_liquid_water.
- “an-ciwc-name”: Defines the name of the target variable for analysis cloud ice water (CIWC). Default: amsre.nwp.cloud_ice_water.
- “an-tp-name”: Defines the name of the target variable for analysis total precipitation (TP). Default: amsre.nwp.total_precip.

4.1.5.1.4 Cloud Flagging for HIRS

This plugin computes and adds cloud flag data to a matchup dataset containing HIRS data.

Example plugin xml part:

```
<hirs-l1-cloudy-flags>
   <hirs-sensor-name>hirs-n18</hirs-sensor-name>
   <hirs-var-name-source-file-name>hirs-n18_file_name</hirs-var-name-source-file-name>
   <hirs-var-name-source-x>hirs-n18_x</hirs-var-name-source-x>
   <hirs-var-name-source-y>hirs-n18_y</hirs-var-name-source-y>
   <hirs-var-name-processing-version>hirs-n18_processing_version</hirs-var-name-processing-version>
   <hirs-var-name-source-11_1-um>bt_ch08</hirs-var-name-source-11_1-um>
   <hirs-var-name-cloud-flags>hirs-n18_cloudy_flags</hirs-var-name-cloud-flags>
   <hirs-var-name-latitude>hirs-n18_lat</hirs-var-name-latitude>
   <hirs-var-name-longitude>hirs-n18_lon</hirs-var-name-longitude>
   <hirs-var-name-11_1-um>hirs-n18_bt_ch08</hirs-var-name-11_1-um>
   <hirs-var-name-6_5-um>hirs-n18_bt_ch12</hirs-var-name-6_5-um>
   <distance-product-file-path>/path/to/the/distance-NetCDF-file.nc</distance-product-file-path>
</hirs-l1-cloudy-flags>
```

The plugin differentiates three matchup scenes:

- “water”
- “ice covered water”
- “land”

In the case of a "water" scene, it is needed to reopen the source orbit and extract a 45 x 45 pixels area around the center position of the current matchup.

Therefor this list of parameters is needed:

- `<hirs-sensor-name>`
  The sensor name, which was used to ingest the input product to the database.
- `<hirs-var-name-source-file-name>`
  The name of the MMD variable, which contains the source filename of the input file
- `<hirs-var-name-source-x>`
  The name of the MMD variable, which contains the source center pixel **x** of the matchup
- `<hirs-var-name-source-y>`
  The name of the MMD variable, which contains the source center pixel **y** of the matchup
- `<hirs-var-name-processing-version>`
  The processing version, which was used to ingest the input product to the database.
- `<hirs-var-name-source-11_1-um>`
  The name of the 11,1µm brightness temperature variable of the source file

In the cases "ice covered water" or "land" the plugin can compute the flags without extraction of data from the source product.

The following list of parameters are effective for all cases.

```
<hirs-var-name-cloud-flags>hirs-n18_cloudy_flags</hirs-var-name-cloud-flags>
<hirs-var-name-latitude>hirs-n18_lat</hirs-var-name-latitude>
<hirs-var-name-longitude>hirs-n18_lon</hirs-var-name-longitude>
<hirs-var-name-11_1-um>hirs-n18_bt_ch08</hirs-var-name-11_1-um>
<hirs-var-name-6_5-um>hirs-n18_bt_ch12</hirs-var-name-6_5-um>
<distance-product-file-path>/path/to/the/distance-NetCDF-file.nc</distance-product-file-path>
```

5 MMD file format

TODO

5.1 Global Attributes

5.1.1 sensor-name

A comma separated list of sensor names. The first name is the name of the primary sensor.

5.2 Extra MMD Variables

5.2.1 <sensor-name>_x

5.2.2 <sensor-name>_y

5.2.3 <sensor-name>_file_name

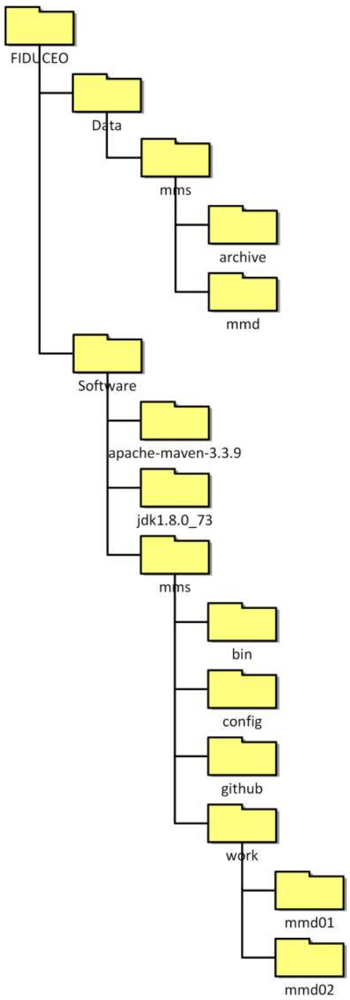
5.2.4 <sensor-name>_processing_version

Contains the processing version of the original sensor data file.

5.2.5 <sensor-name>_acquisition_time

6 Installation and Operation on CEMS

This chapter covers the installation and operation of the Fiduceo MMS on the parallel processing facility JASMIN/CEMS supplied by CEDA. The Fiduceo project has a specific virtual machine assigned on the CEMS system for operations.



6.1 Workspace Structure

This chapter describes the Fiduceo workspace directory structure and the way the MMS in embedded into it. The sketches omit all non-MMS-related directories for a more clear view of the workspace structure.

The directory branch below "Data" contains a subdirectory for the "mms" that contains MMS related data. The "archive" directory contains all sensor and in-situ input data ordered in the directory structure described in the MMS Implementation Plan (see RD 1) whereas the directory "mmd" serves as target directory for the processing results. Subdirectories for each use-case will be generated during the processing of matchups.

The directories below "Software" contain all software comprising the MMS and also all third-party software used to build and run the MMS. Subdirectories "apache-maven-3.3.9" and "jdk1.8.0_73" contain the apache maven build manager and the Java JDK/runtime required to compile and run the MMS.

The subdirectory "mms" contains all MMS related software, configuration, a build directory and a set of working directories. The current operating version of the MMS is contained in the directory "bin". The configuration files used by the MMS are stored in the directory "config".

The subdirectory "github" is the build directory. It contains the complete source code of the MMS and is configured to be able to fetch code changes from github and to build and deploy the software.

In the "work" directory contained are the working directories for processing. Any processing job needs to be operated from a dedicated sub-directory of "work". Each subdirectory for a specific use-case processing contains status files, log-directories and all other files required by PMonitor to operate.

6.2 Building the MMS on CEMS

The build and deployment process is mostly automated. To build and deploy "master" branch (i.e. the latest development version) of the github repository, change to the directory "FIDUCEO/Software/mms/github/fiduceo" and invoke the script "build_and_install_on_cems.sh". This script performs the following steps:

- Checkout the latest version of the code from github
- Invoke maven to build the code and run all tests
- Clean up the "bin" directory
- Invoke maven to build an assembly of the binaries and deploy it to the "bin" directory.

When these steps have been run successfully, the MMS software is ready to operate.

6.3 Operating the MMS on CEMS

This chapter covers all information with respect to operations of the MMS on the CEMS environment and specific configuration and set-up tasks.

6.3.1 Setting up the environment

Before the MMS is able to operate on the LOTUS cluster, a number of environment variables and paths need to be set. This is done by simply sourcing a pre-configured script.

Assuming that you are in a use-case subdirectory e.g. in "FIDUCEO/Software/mms/work/mmd08" sourcing the environment file is accomplished by executing:

```
[tblock01@lotus mmd08]$. ../../bin/mms-env.sh
```

The script file is automatically installed to the directory "FIDUCEO/Software/mms/bin" by the maven deployment script.

6.3.2 CEMS Specific Configuration

The CEMS environment requires a number of specific configuration settings that ensure the proper operation on the cluster. All configuration files accessed by the operational MMS are located in the directory “FIDUCEO/Software/mms/config”.

IMPORTANT: The configuration files in this directory will not be updated by the MMS build and deploy procedure. Any changes in the file content, e.g. additional fields or field renaming, have to be inserted manually into the operational configuration.

6.3.2.1 System Configuration

The MMS system configuration is set up to pick up the correct input data archive path for the CEMS installation. The following setting is mandatory on CEMS:

archive-root: /group_workspaces/cems2/fiduceo/Data/mms/archive

For a description of the system configuration file, please refer to chapter 4.1.1.

6.3.2.2 Database Configuration

The database installed on the Fiduceo workspace is a MongoDB server installation, version 3.2. This database requires a fixed set of access configuration parameters for the database configuration:

driverClassName: mongodb

url: mongodb://172.26.69.130:27017/FIDUCEO

username: xxxx

password: xxxx

Username and password are available on request. For a description of the database configuration file, please refer to chapter 4.1.2.

6.3.3 Operating the MMS

On the parallel processing environment on the LOTUS cluster, the MMS software is controlled by an additional Python layer, PMonitor. This software layer controls the deployment and parallel execution of MMS-Tool processes (Ingestion, Matchup, Post Processing) to the cluster nodes. It ensures that processing can be paused and re-invoked and that on e.g. hardware or network failures, large processing jobs can safely continue where they have been interrupted.

PMonitor is invoked from the console on the lotus VM using three shell scripts that are located in the directory “FIDUCEO/Software/mms/bin” – and that are automatically available on the path when the environment is set up correctly following the instructions in chapter 6.3.1.

pmstart.sh: this command is used to submit a parallel processing task to the cluster. It assumes a single cmd-line argument, the name of the workflow-file that contains the process description, e.g.

“pmstart.sh ingest_avhrr_n07.py”.

pmstop.sh: this command stops the execution of parallel tasks and kills the currently running processes on LOTUS. PMonitor ensures that the processing can later pick up processing at the point where the job has been stopped. It assumes a single cmd-line argument, the name of the workflow-file that contains the process description, e.g.

“pmstop.sh ingest_avhrr_n11.py”.

pmclean.sh: this command cleans up all temporary files that have been generated during the processing execution. It assumes a single cmd-line argument, the name of the workflow that has been processed, e.g.

“pmclean ingest_avhrr_m02”.

The shell scripts are designed in a way that it is safe to close the console after a processing job has been started. Processing will keep on running in the background.

6.3.4 Workflow Files

The definition of the jobs to be executed by PMonitor is stored in workflow files. These are Python files that create and configure a workflow object, which is using PMonitor to execute the parallel processing on the cluster. Workflow files are available from and have to be stored into the directory “FIDUCEO/Software/mms/bin/python”.