EUROPEAN SOUTHERN OBSERVATORY
Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral
Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

VERY LARGE TELESCOPE

MUSE EDPS-GUI tutorial

2 rev. 20260612

Issue 2

Date June 12 2026

	Name	Date	Signature
Prepared:	Lodovico Coccato	June 12 2026	
Approved:			
Released:			

This page was intentionally left blank

Change record

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
1	15/05/2026	all	First public release for 2.11.0
2	12/06/2026	4.2.1	Add Section 4.2.1

This page was intentionally left blank

Contents

1 Introduction

1.1 Scope

This document describes how to reduce MUSE data with the edps-gui (Graphic User Interface), the dashboard of the ESO Data Processing System (EDPS), which the recommended interface to reduce data from ESO telescopes. Details on the MUSE data reduction stream and how to configure the reduction to meet specific scientific needs are also given.

For a more extensive documentation on the edps-gui itself, consult the dedicated manual here.

For a description of the MUSE pipeline itself, consult the pipeline manual available at:
https://www.eso.org/sci/software/pipe_aem_table.html.

Note: this tutorial refers to:

- MUSE instrument pipeline named muse, version 2.11.0.
- MUSE workflow: muse.muse_wkf
- EDPS version 1.5.7.
- edps-gui version 0.91.

1.2 What is EDPS?

The ESO Data Processing System (EDPS) is a framework to run ESO's data processing pipelines and it is meant to eventually replace the previous ESOReflex environment. The general principles of EDPS have been described by Freudling, Zampieri, Coccato et al. [2024, A&A, 681, A93]. Please refer to that paper if you have used EDPS for research resulting in a scientific publication.

Each of ESO's data processing pipeline consist of a series of standalone programs called *recipes*. Each recipe is designed to process certain type(s) of input data. The processing of these input data typically requires a range of auxiliary files such as calibration files. EDPS is designed to select appropriate input data for the different recipes of a pipeline, and execute them in sequence. This is done by specifying for each pipeline the workflow for organizing data and executing the recipes. This workflow can the used to process a set of data fully automatically.

1.3 Main concepts

EDPS is an environment designed to execute the recipes of an instrument pipeline according to a series of instructions. The main concepts in EDPS are:

- **Workflow** and **reduction cascades**. A workflow is a series of instructions designed to reduce data with an instrument pipeline in potentially multiple ways, by carrying on a sequence of tasks. Each workflow can define multiple reduction cascades, depending on the scientific needs. For example, the same workflow can be used to process data following different strategies that trigger different reduction steps (e.g. in one strategy flux calibration can be omitted) or different end-points (e.g., combine different science exposures, or stop after the reduction of individual exposures without combining them). Each of these "strategies" defines a "reduction cascade".

- **Task**, **jobs**, and **recipes**. A task is an element in the workflow that performs a given step of the data reduction cascade. Tasks are often associated to a recipe of the underlying instrument pipeline. A jobs is a work unit in a processing environment, that runs a recipe on a set of input data with a set of recipe parameters. A single task can generate several jobs: for example, a "bias" task, can generate multiple jobs, each of the running the bias recipe on a different set of input files.
- **Dataset**. A dataset is a collection of files, that are needed to perform the data reduction as specified by the workflow. It consists, for example, of one or more science files plus the calibrations needed to process them. In `EDPS`, datasets have an hierarchical structure, which highlights the connections between the various files and tasks (e.g., task A is an input to task B).
- **Target** and **Target category**. The "target", or the "target task" is the end point of the reduction cascade. When specifying a target, `EDPS` will process all and only the files needed to execute it. For example, if my target is "science", and the science files need the bias files, `EDPS` will process only the biases that have been selected to process those science files; then it processes the science using the product of the bias reduction. However, if my target is bias, then `EDPS` will process all and only the bias files, regardless they are not used by any science. In this case, `EDPS` does not processes the science, as it has already reached the end reduction point (e.g., process all biases). The "Target category" is a group of targets that have similar purposes. For example, the target category "science", includes all the tasks that deliver final scientific products, the target category "qc1calib" includes all and only the tasks that processes calibrations (e.g., bias, flat fields, standard stars).

1.4 Installation

1.4.1 Prerequisites

Prerequisites for a well functioning installation of `EDPS` and `EDPS`-gui are:

- Recent Firefox or Chrome browser, Python 3.11 or higher (but there are issues with Python 3.14).
- At least one ESO pipeline with EDPS workflow should be in your system. To install the desired ESO pipelines, follow the instructions in the ESO pipelines pages. NOTE: the `apptainer` installation method is currently not supported. After the installation, the esorex command must be in the path. To test whether the installation was successful, type

  ```
  esorex --recipes
  ```

  A list of available recipes should appear.
- Install `graphviz`, `fv`, and `ds9`, which have to be included in the system path (definying aliases not enough). On linux, `Graphviz` can be easily installed via:

```
sudo apt install graphviz (Debian, Ubuntu)
sudo dnf install graphviz (Fedora)
```

Check the , `Graphviz` webpage for installation instructions for other OS.

`fv` and `ds9`, are optional. To install them, follow the instructions in corresponding webpages. You can test whether these three packages are installed and their path are correctly set by typing on a terminal:

```
dot -V
fv  -version
ds9 -version
```

1.4.2 Installation steps

To install EDPS follow these steps:

- Create a new Python virtual environment and activate it:

```
python3 -m venv edpsgui
. edpsgui/bin/activate
```

  Make sure the python3 version is 3.11 or higher, but not 3.14.

- Install the required packages:

```
pip install --extra-index-url \
    https://ftp.eso.org/pub/dfs/pipelines/repositories/stable/src \
    edps edpsgui edpsplot adari_core
```

To run the `edps-gui` type from a terminal (with the active environment):

```
edps-gui
```

Important note. The first time `edps-gui` is executed, you will be asked to specify the directory where the reduction products (fits files and quality plots) will be stored. The default location is `$HOME/EDPS_data`. During the first execution, a configuration file named `application.properties` will also be saved in the directory (newly created) `$HOME/.edps`.

2 Reducing demo data

Follow this procedure to quickly reduce MUSE demo data. We assume that the `EDPS`, `edps-gui`, the MUSE pipeline and its associated demo data are installed in your system. For general instructions on how to install `EDPS` and the pipeline, see Section 1.4 or please visit: https://www.eso.org/sci/software/pipe_aem_main.html.

2.1 Setting the workflow

Proceed as follows:

1. If not done already, activate the `EDPS` virtual environment, defined during installation (Sect. 1.4).
2. Start the `edps-gui` dashboard by typing:

```
edps-gui
```

   The `edps-gui` dashboard will start in a browser window (Figure 1).

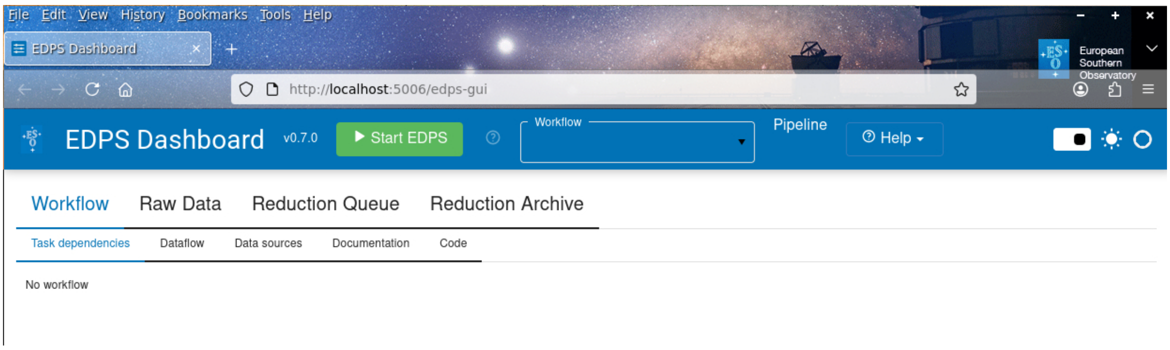


Figure 1: The empty `edps-gui` Dashboard; the underlying `EDPS` engine has not yet been started and no workflow has been loaded.

3. Optionally, before starting `EDPS`, one can specify new settings by pressing Help –> Settings (Figure 2).

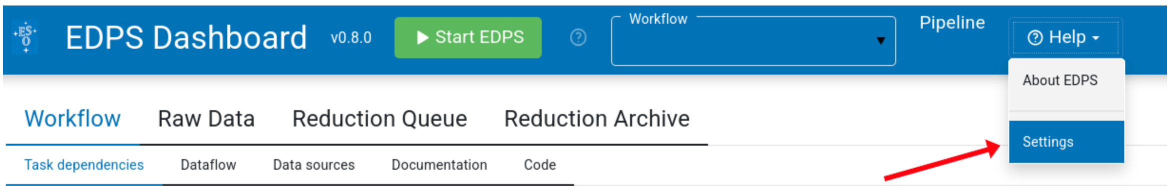


Figure 2: The “Help” –> “Settings” menu.

4. On the browser window with the dashboard, press the button ‘Start EDPS‘.
5. Choose the `muse.muse_wkf` workflow from the list in the ’Workflow’ field. The workflows offered in this selector depend on the installed pipelines. The graphic workflow representation will appear as in Figure 4.

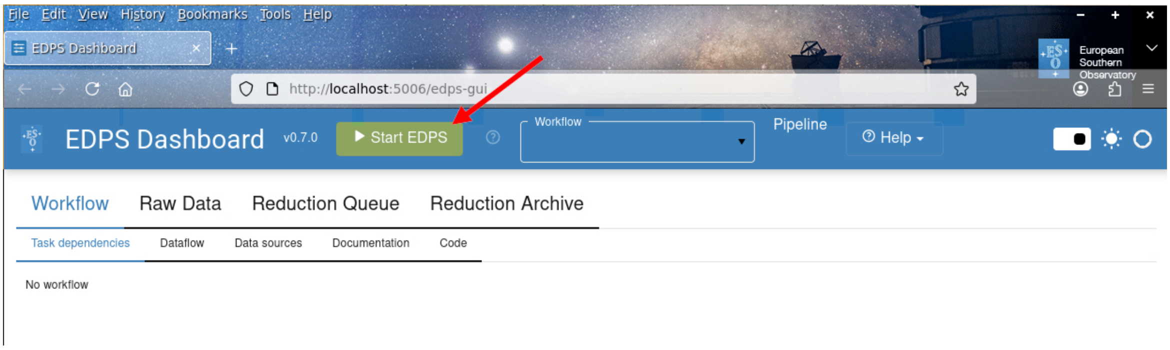


Figure 3: The “Start EDPS” button.

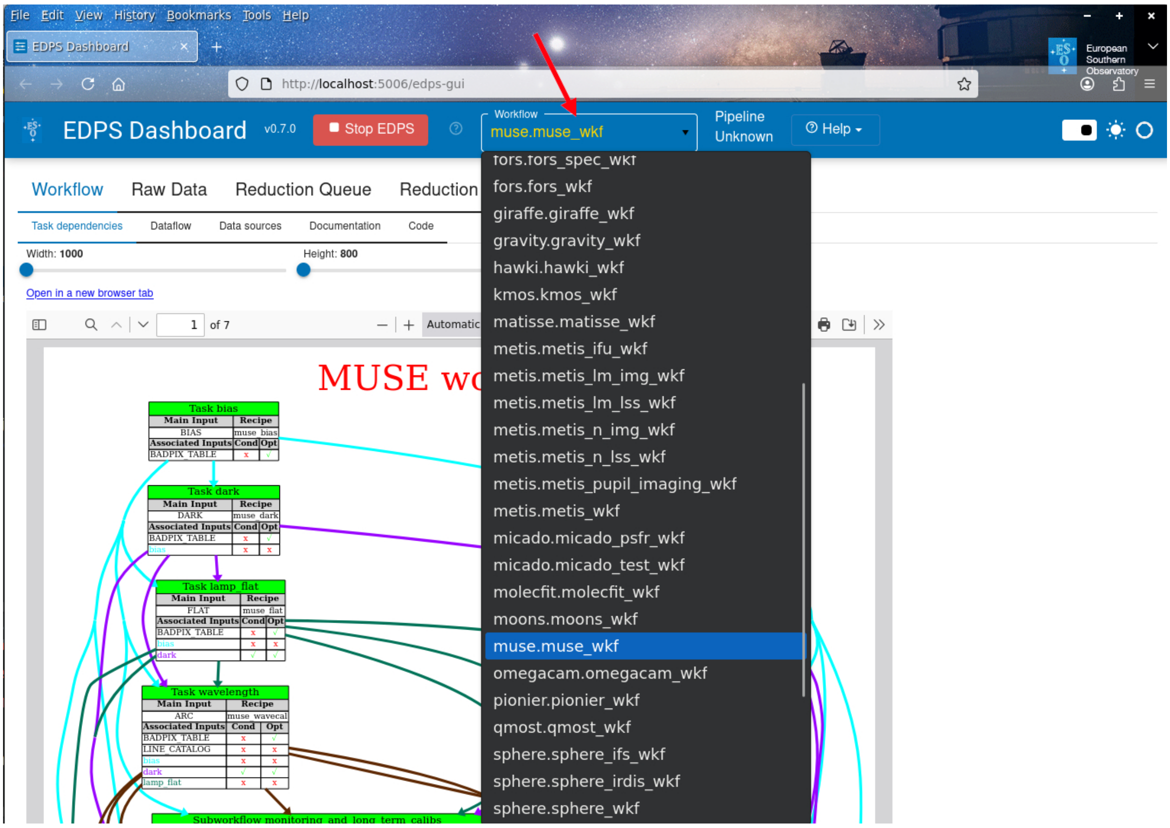


Figure 4: The `edps-gui` with the MUSE workflow loaded.

2.2 Selecting the input data

1. Press ‘Raw Data‘ to enter the corresponding tab, as in Figure 5.

2. Press ‘Select Inputs‘. A selection window will appear that allows to select data that are stored on a local disk (Figure 6).

3. (Optional). Select the reduction target, configure the workflow parameter and specify the association

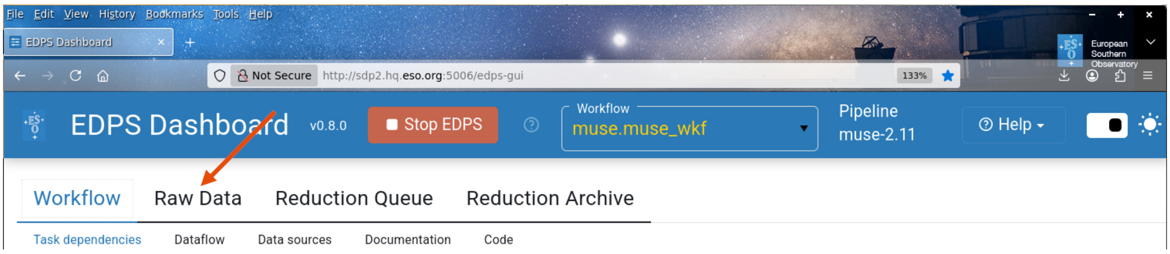


Figure 5: How to select RAW data Tab.

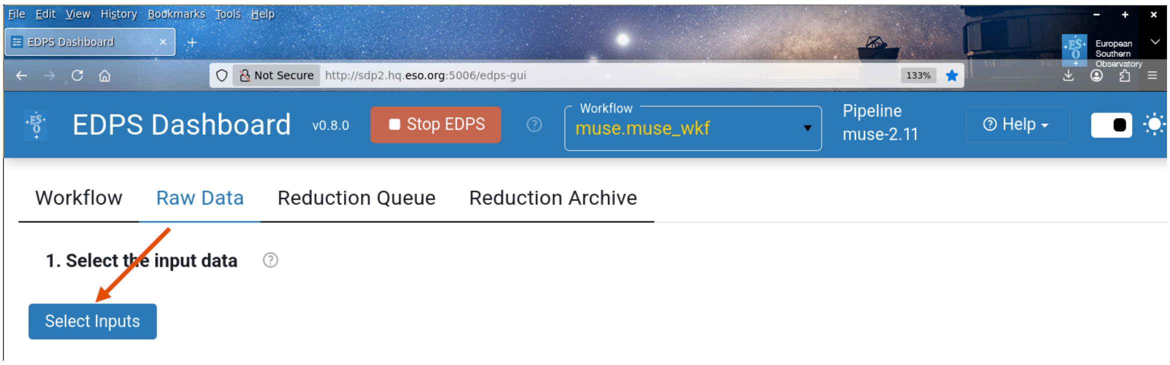


Figure 6: How to select input data.

preferences. These steps are optional. For more information see Section 4.

4. Press ‘Create Datasets‘. A list of datasets appears, one line for each set of science data (Figure 7).

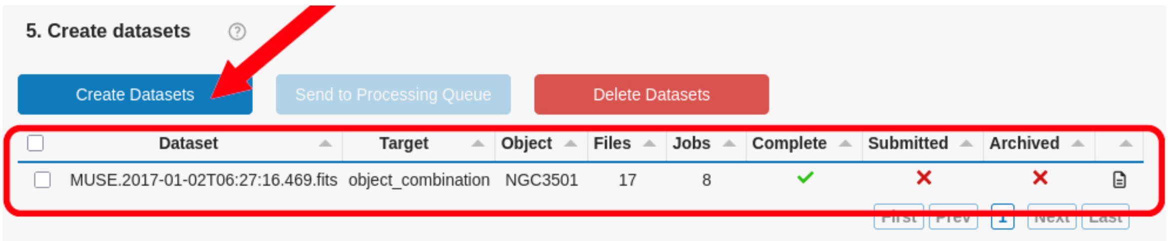


Figure 7: How to inspect the input data directory to create datasets.

5. Choose the datasets that should be processed (Figure 8)

   and send them to the data reduction queue by pressing ‘Submit to Reduction Queue‘. Note that this action does not start the reduction automatically.

2.3 Start the reduction

1. Press the ‘Reduction Queue‘ tab (Figure 9).

2. Select the datasets you’d like to reduce.

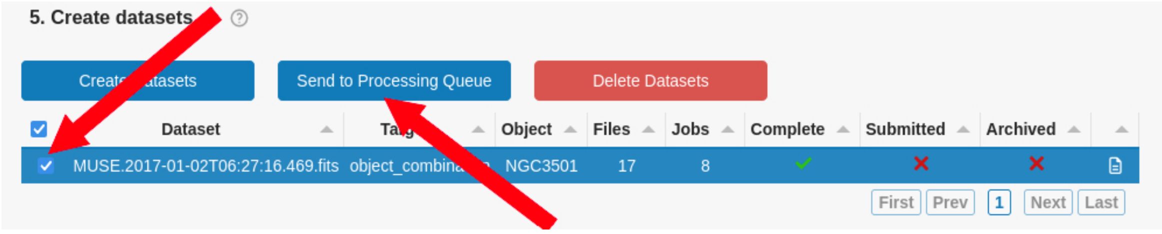


Figure 8: How to send the selected datasets to the Reduction Queue for processing.

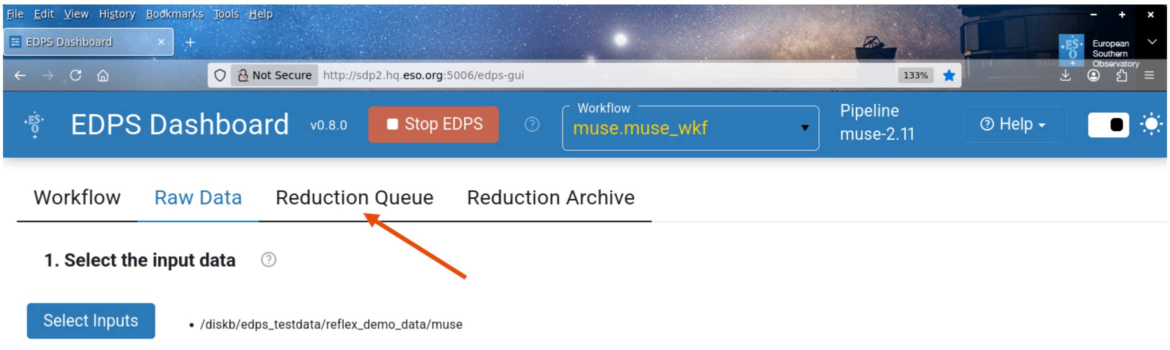


Figure 9: How to select Reduction Queue tab.

3. (Optional). Configure the workflow and recipe parameters by pressing the wheel button ⚙ to open the configuration editor. See Section 4.2 for more information on the configuration editor.
4. Press the ‘Reduce‘ button (Figure 10). The selected data will now be processed with the configured parameters.

Congratulations! You reduced your first data with the `EDPS` dashboard! All the reduced data are saved in the EDPS_data directory specified when executing `edps-gui` for the first time.

2.3.1 Quality plots

It is possible to inspect the information on each job, such as quality plots showing the products (for the most important jobs and products), the list of inputs and output files, the recipe parameters and logs. All the information on the job processing can be inspected from the ‘Reduction Queue‘ window. While quality plots are produced only for completed jobs, the other relevant information is available also for failed jobs.

The information associated for the main product can be inspected by pressing the magnifying glass symbol at the right side of each dataset. To inspect the information associated to other individual jobs (e.g., calibrations), proceed as follows:

- Expand the desired dataset by pressing the black arrow on its left. The list of jobs will appear with the associated status (COMPLETED, RUNNING, PENDING, MISSING, ABORTED, FAILED)

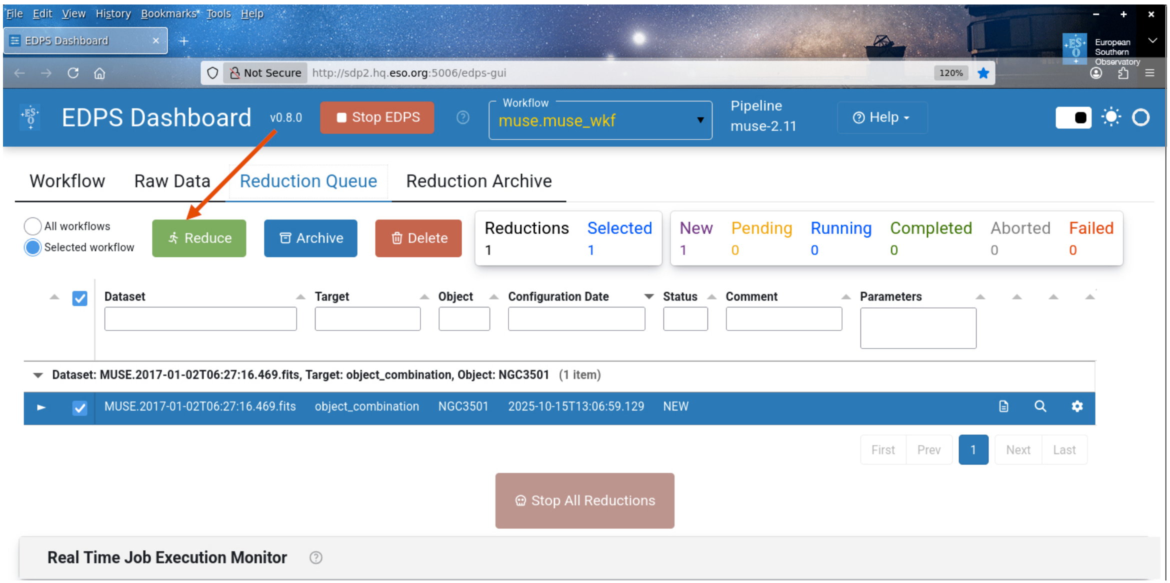


Figure 10: Reduce.

- Press the magnifying glass symbol at the right side of the job you want to inspect.

2.4 Exporting the final products

Completed reductions can be 'Archived' (i.e. declared 'completed' because no more work is needed) and removed from the Reduction Queue. Additionaly, even if all products for all tasks are saved in the EDPS_data directory, the most important producs can be 'exported' to a desired location.

To do so, proceed as follows:

1. In the 'Reduction Queue' tab, select the dataset and the dataset for which you want to export the final products, and press the 'Archive' button.
2. Go in the Reduction Archive tab and click on the 'Export' button. A new tab window appear where you can indicate the directory you want to copy your final products; finally press "Export" to copy the data.

Exported products are organized by 'DATASET' (named as the first scientific exposure of the dataset), and 'TIMESTAMP' (time of start of reduction)

The final products saved in the specified directory are:

- Datacubes and reconstructed images of mosaics, when combination of multiple exposures has been perfomed. Their name format are 'DATACUBE_COMBINED_' and 'IMAGE_COMBINED_' followed by the target name (as obtained from the header keyword 'OBS TARG NAME' of the first exposure)

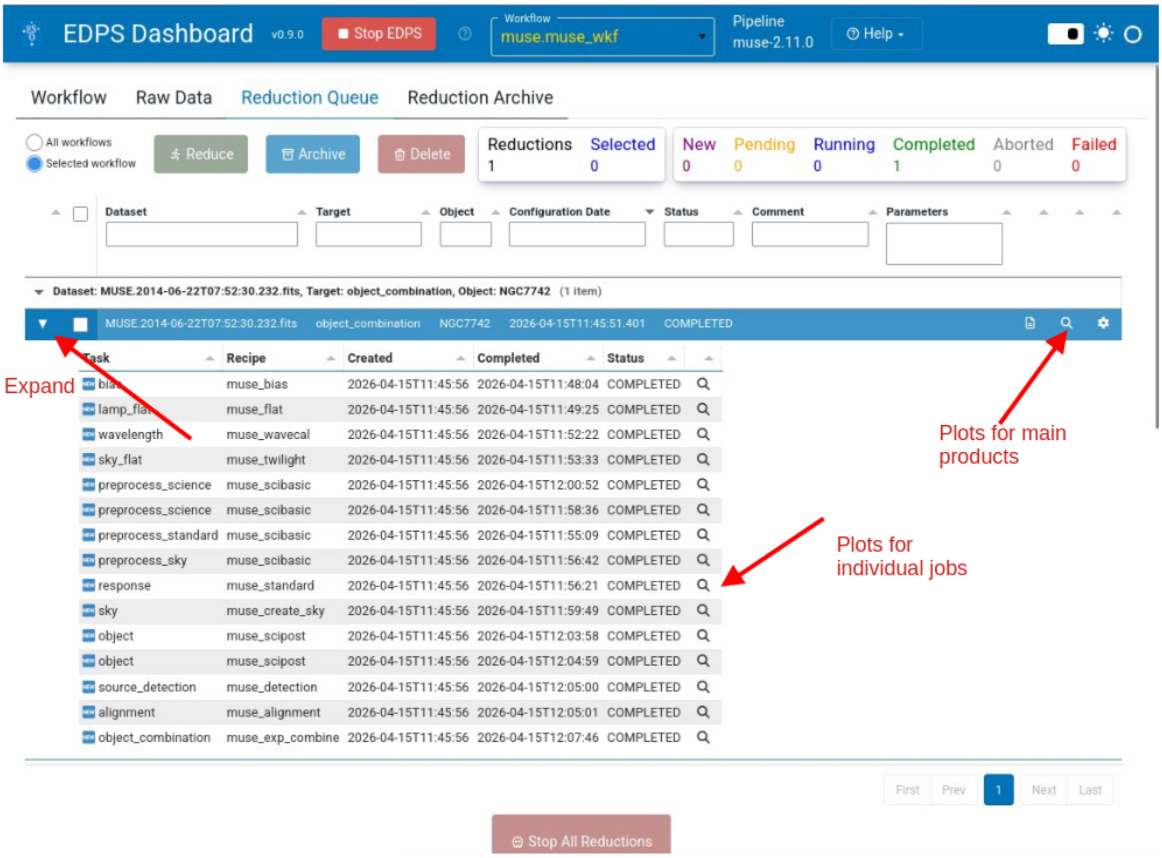


Figure 11: How to look for job information from the Reduction Queue tab.

- Datacubes and reconstructed images of individual exposures. Their name format are ‘DATACUBE_‘ and ‘IMAGE_‘ followed by the exposure identifier (header keyword ‘arcfile‘). They are stored in the subdirectory ‘individual_exposures‘

Additionally, if specified by the reduction target, also products of sky exposures (treated as regular science exposures) are saved. They product names have the same convention as regular science exposures.

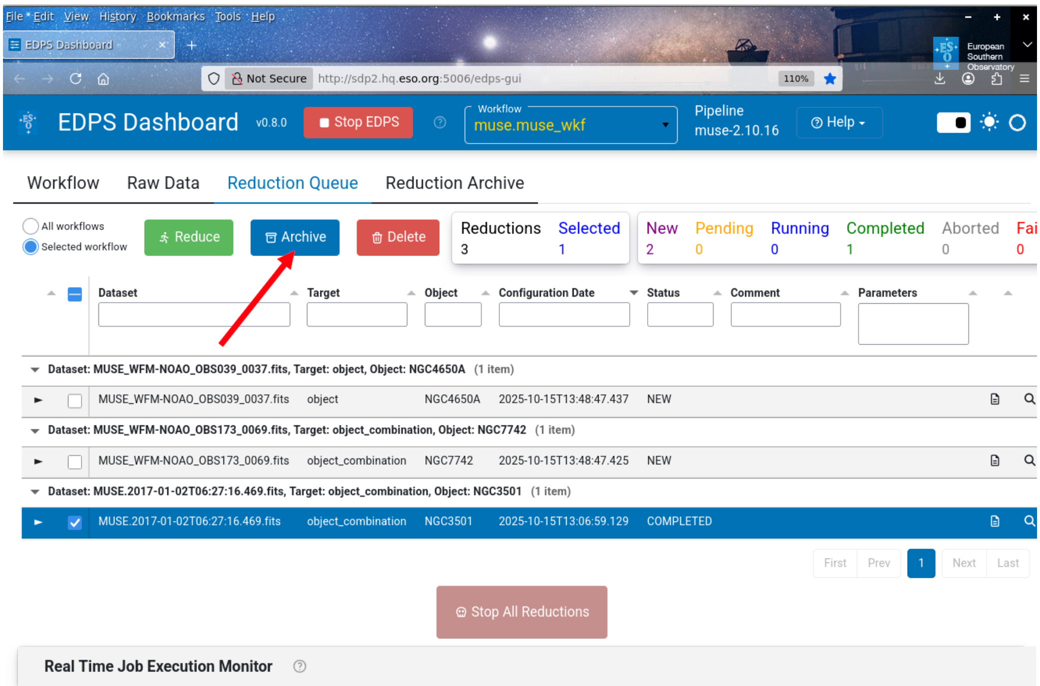


Figure 12: How to archive a completed reduction from the Reduction Queue tab.

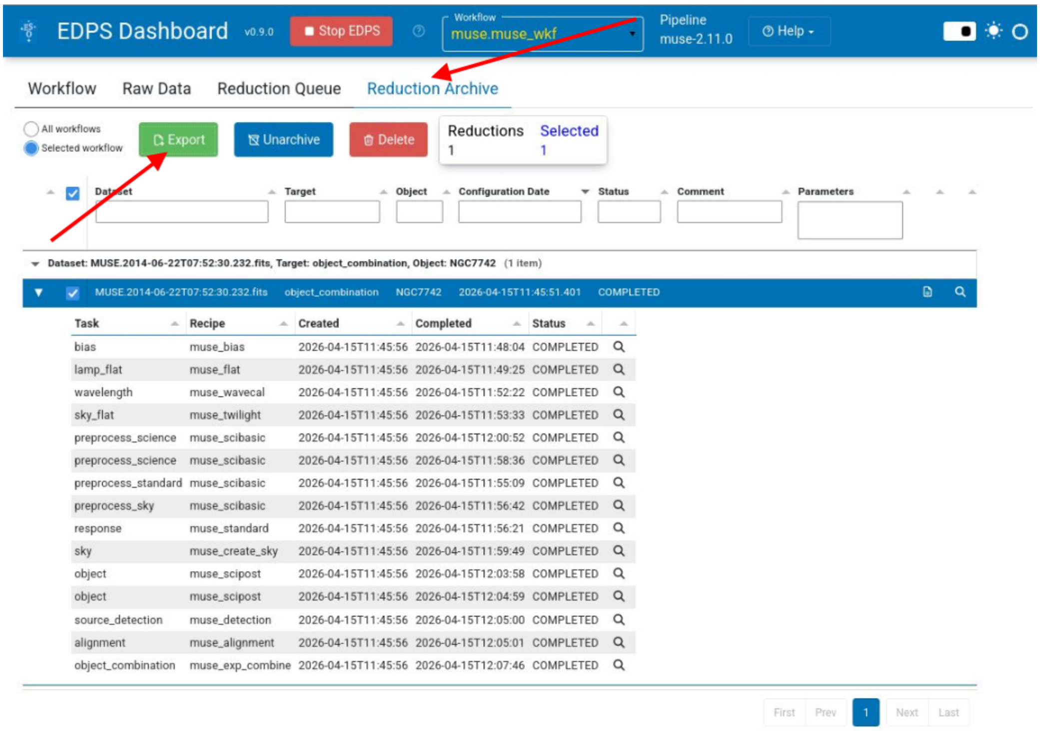


Figure 13: The reduction archive tab. This table contains all the different configurations of datasets that are declared "finished" and removed from the Reduction Queue. From this page, the user can export the most important files into a desired local directory.

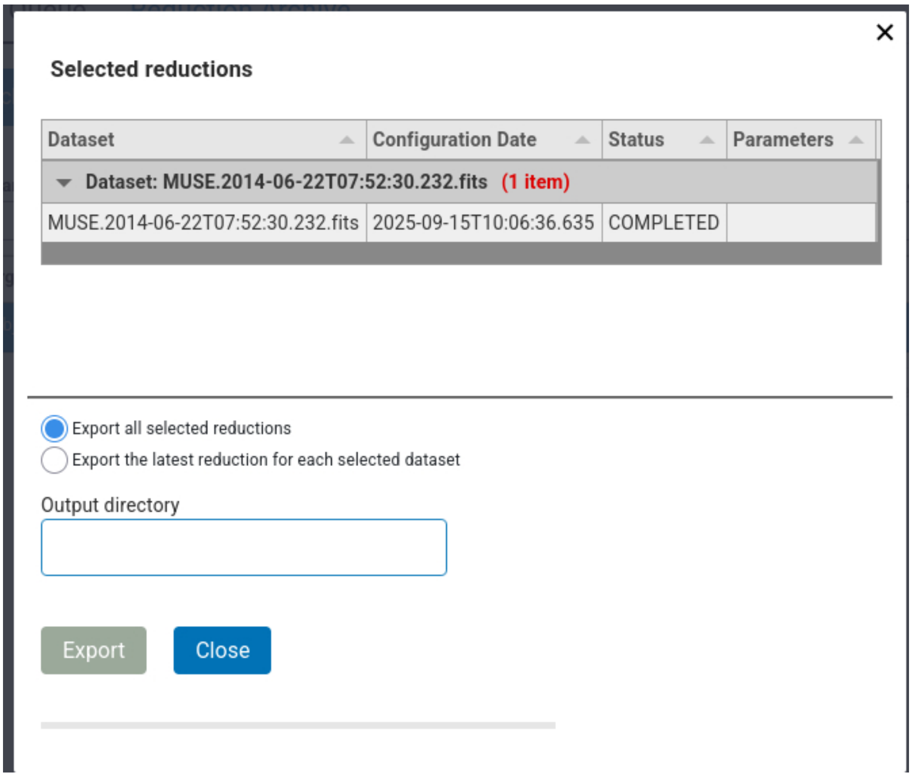


Figure 14: The EXPORT dialogue window, where the user can decide which reduced configuration to save and where.

3 The MUSE data reduction flow.

The overall data flow of the MUSE pipeline is displayed in Figure 15.

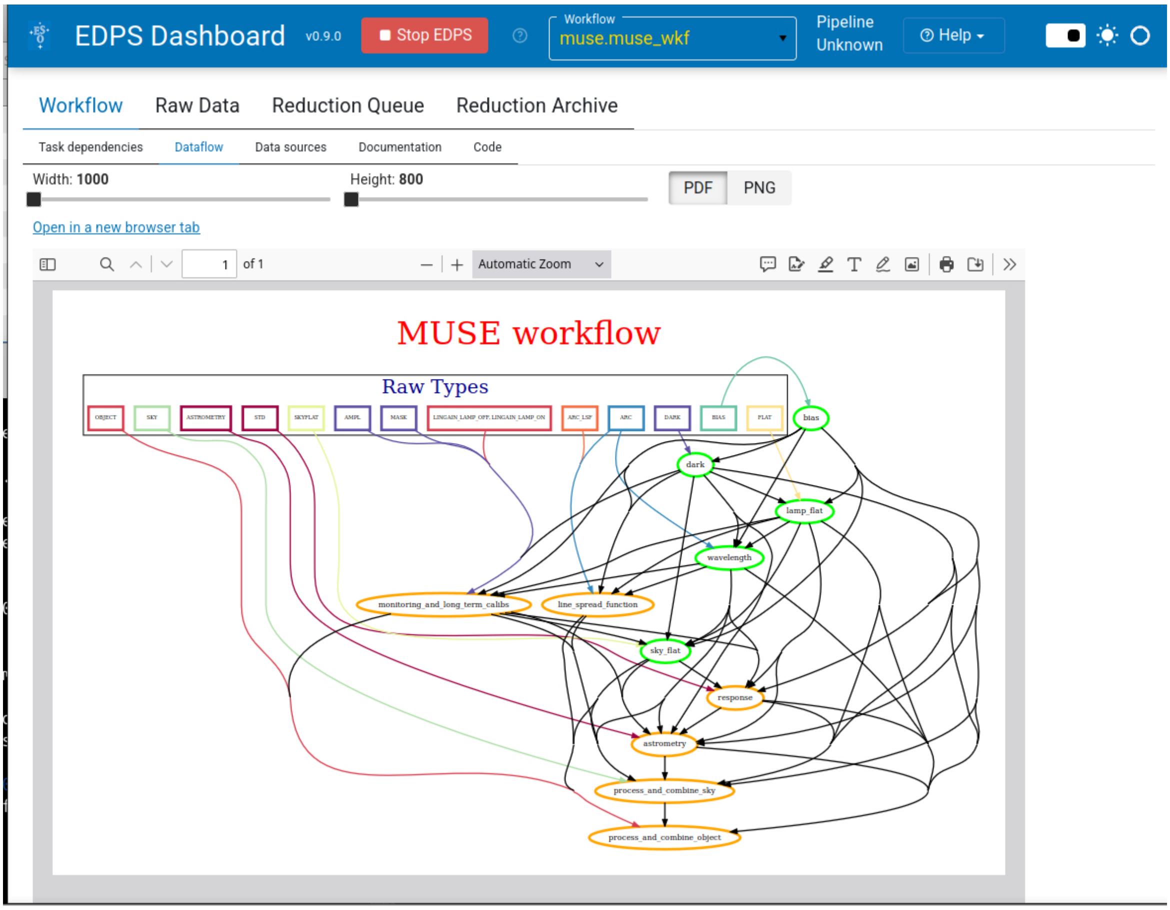


Figure 15: The data reduction cascade of the MUSE workflow.

The reduction cascade is organized in tasks, which represent well-defined steps in the process. Tasks can be grouped inside sub-workflows. Each task runs a recipe; the detailed description of the algorithms, input, outputs and recipe parameters used in each recipe are available in the pipeline manual. Here, we present only the description of most important features.

The `muse.muse_wkf` `EDPS` workflow is designed to execute the tasks that deliver the final reduced data cube for each dataset. It can be either the product of a single exposure, or the combination of multiple exposures. Only calibrations needed by the selected the scientific exposures are processed.

It is possible to set `EDPS` to perform the data reduction until a certain step of the reduction chain (e.g. to reduce only standar stars, or only flat fields). This is done by specifying the desired tasks in the field **Select reduction target** of the **Raw Data** tab.

The reduction steps of the `muse.muse_wkf` workflow are listed below. Before starting the reduction, the parameters of the recipes associated to each task can be configured by pressing the button close to each dataset configuration. See for more info on the configuration editor 4.2

3.1 Bias subtraction

The task **bias** reduces raw bias frames to generate a combined masterbias. It runs the recipe **muse_bias**.

3.2 Computation of dark current

The task **dark** runs the recipe **muse_dark**. If raw dark frames are present, the actor processes them and creates a master dark. This product is currently used only for instrument monitoring but not for scientific reduction. Dark frames are taken monthly, therefore they might not represent the dark current at the time of the observations, and they add noise to the final products. If needed, darks can be used in the reduction cascade by setting the workflow parameter **use_darks** to "yes" in the **science_parameters** parameter set.

3.3 Flat field correction

The task **lamp_flat** runs the recipe **muse_flat**. It processes the raw flat-fields exposures, producing a master flat and a trace table.

3.4 Wavelength correction

The task **arc** executes the recipe **muse_wavecal**. It processes the raw arc frames, producing a table with the wavelength solution.

3.5 Characterization of the line spread function

This subworkflow is designed to run the recipe **muse_lsf** to produce a calibration that characterize the line spread function. In MUSE, the generation of the line spread function can be done by processing raw arc frames (done by the task **line_spread_function_arc**) or by dedicated calibrations (done by the task **line_spread_function_lsf**). Alternatively, a static calibration is used. The behavior is determined by the workflow parameter **lsfmode**, which is described in Section 4.3.

3.6 Monitoring calibrations, geometry and astrometry calibrations

The subworkflow **monitoring_and_long_term_calibs** contains tasks aimed at creating long term calibrations, such as geometry calibrations (task **geometry**, recipe **muse_geometry**), astrometric calibrations (tasks **preprocess_astrometry**, **astrometry**, recipes **muse_scibasic**, **muse_astrometry**). These calibrations are distributed by the ESO archive together with the scientific data. It is anyway possible to recreate these calibrations from raw data by setting the workflow parameters **recompute_geometry** and **recompute_astrometry** to "yes".

Additionally, the subworkflow contains tasks aimed at monitoring the instrument performance such as the throughput (task throughput, recipe **muse_ampl**), the instrument linearity and gain (task **linearity_and_gain**, recipe **muse_lingain**). These tasks however, are executed only at the observatory, and they are not used in the data reduction flow.

3.7 Sky subtraction

The subworkflow **Process and combine sky** contains the tasks in charge of evaluating the sky background from dedicated sky exposures.

They are:

- **preprocess_sky**. It runs the recipe **muse_scibasic** on dedicated sky exposures.
- **sky**. It runs the recipe **muse_create_sky** to generate SKY_LINES and SKY_CONTINUUM calibrations.

Sky subtraction can also be done by evaluating the sky contribution directly on regions in the target field of view that is free from contamination from astronomical sources. The most important products to check, regardless of the method used, are the

- SKY_CONTINUUM. It should not contain features that are typical of astronomical objects. For example, if the sky region contains HII regions, the Halpha emission lines are not registered as a sky line in the input list of sky lines. As a consequence, it will be present in the SKY_CONTINUUM and therefore subtracted from the target spectrum.
- SKY_MASK. A comparison between the image field of view and the sky mask should confirm that astronomical sources are not included as sky regions.

If the SKY_MASK overlap with some source or if SKY_CONTINUUM contains astrophysical signal such as nebular emission lines, consider to decrease the recipe parameter **skymodel-fraction** or to provide a custom SKY_MASK in the input directory.

SKY_CONTINUUM and SKY_MASK products are available from the tasks sky (if dedicated sky observations are used) and object (if the sky was evaluated on the target field of view). Click on the magnifying glass symbol close to the appropriate task, and select the tab **Output Files**. Products can be inspected via the `fv` fits viewer (if available).

More information on the sky background removal are given in Section 4.4.

This subworkflow contains also the tasks that consider dedicated sky exposures as self-contained science exposures and processes and combine them as such (tasks **preoprocess_sky_sky**, **object_sky**, **alignment_sky**, and **combine_sky**). The strategy to combine sky exposures is the same as for the regular science exposures.

3.8 Flux calibration

The subworkflow **response** contains the tasks to process standard stars and create the response and telluric corrections. They are:

- **preprocess_standard**. It runs the recipe **muse_scibasic** on raw standard stars
- **response**. It runs the recipe **muse_standard** and generate STD_RESPONSE and STD_TELLURIC calibrations. To disable telluric correction on your data, set the workflow parameter **telluric_correction** to 'FALSE', see Section 4.9.

3.9 Reduction and combination of scientific exposures

The subworkflow **Process and combine object** contains the tasks in charge of aligning and combining scientific exposures. They are:

- **preprocess_science**. This task runs the recipe **muse_scibasic** on scientific exposures.
- **autocalibration_from_object_sky**. This tasks select autocalibration coefficients computed from sky exposures, if requested. It is not directly associated to any recipe, see Section 4.8.
- **object**. This task runs the recipe **muse_scipost** to reduce a single scientific exposure.
- **source_detection**. This task runs the recipe **muse_detection** to detect sources in the datacube of each individual exposure and create a source catalog. The source catalogs are used for the alignment of the exposures.
- **alignment**. This tasks runs the recipe **muse_alignment** to correct for coordinate offsets the exposures that are meant to be combined together. It uses sources detected in the previous step to compute the offsets.
- **object_combination**. This tasks runs the recipe **muse_exp_combine** and combines individual exposures to produce the final datacube (see Section 4.6 on how to specify exposures to be combined).

The most important thing to check in this step is whether or not the images to combine were properly aligned. In the Reduction Queue tab, locate and expand the reduced dataset you want to check, and click on the magnifying glass symbol close to the alignment task as shown in Figure 16.

The following interactive plot will appear, showing the preview of the combination of the field of views after alignment correction (Figure 17). In the case of good alignment, there are no duplicated sources.

In case of bad alignment, here we provide some tipps that can help in improving the alignment. Re-configure the reduction of the dataset by pressing the ⚙ button and set the recipe parameter accordingly in the configuration window (Sect. 4.2).

- **Task source_detection**.
  - In the case of too many or too few detected sources (e.g. in crowded fields), decrease or increase the parameters **source_min** and **source_max** (minimum and maximum number of detections, respectively). Sometimes it helps to change the initial threshold level, via the parameter **start_threshold_sigma**. To prevent the threshold to go too low during iterations, one can set the minimum accepted threshold via the parameter **threshold_limit**.

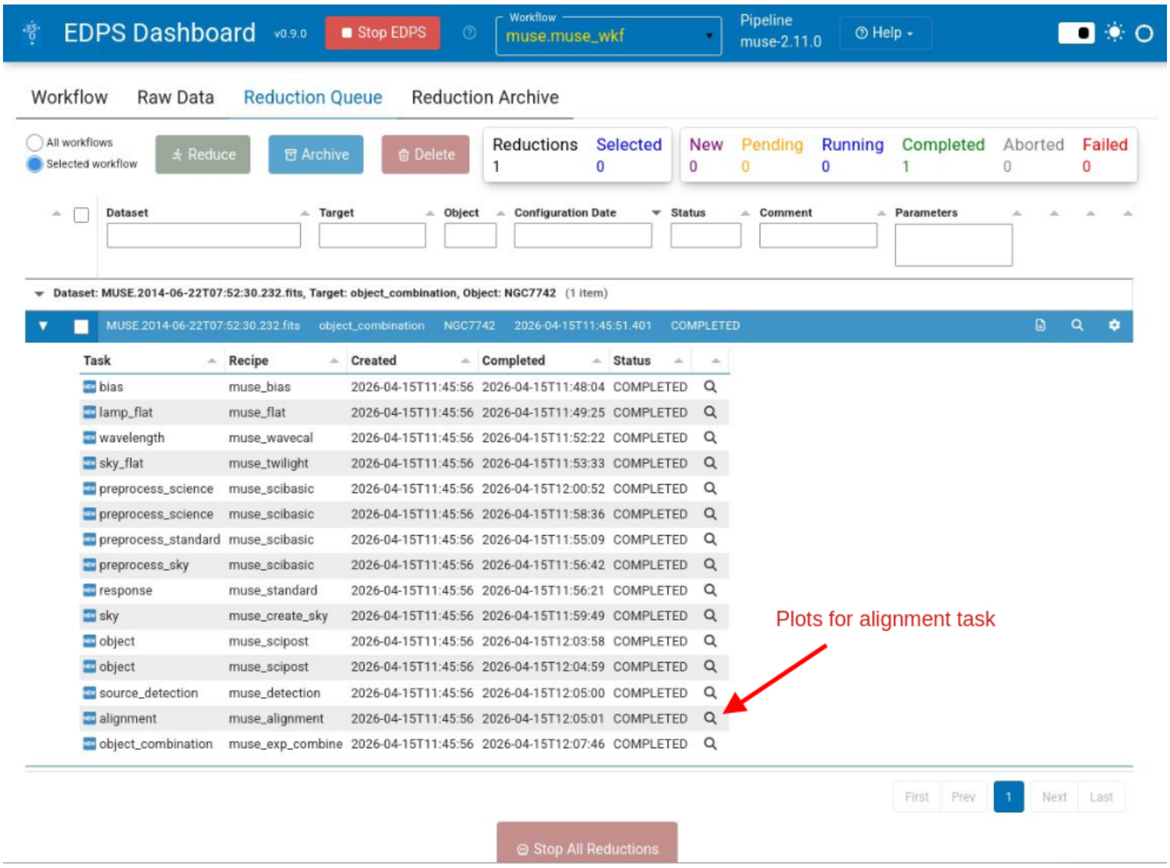


Figure 16: How to display the quality plots for the alignment task.

- A lot of sources are detected in the edges of the field of view. This can be solved by increasing the parameter **border_distance_pxl**.

- **Task alignment**.
  - The maximum number of allowed stars is detected, but the alignment is not correct. This can happen in crowded fields, and the alignment algorithm does not find the correct matching between the sources detected in different frames. There are two tricks to overcome this issue. First, one could either decrease the number of maximum sources allowed in the detection. However, this solution is not advisable for mosaicing, because the number of reference sources in the overlapping regions could be too small. Second, if the offsets are known to be small, the user can decrease value of the first search radius and set the corresponding recipe parameter (**search_radii**) to, for example, '10.,4.,2.,0.8'. The first entry (10 in this example) forces the matching algorithm to find offsets of at most 10 arcseconds.
  - Avoid source misidentification across exposures. To avoid to associate the wrong sources, one can specify the maximum difference in magnitude or in fwhm between sources in different exposures. This can be done by setting the recipe parameters **threshold_on_mag** and **threshold_on_fwhm**, respectively.

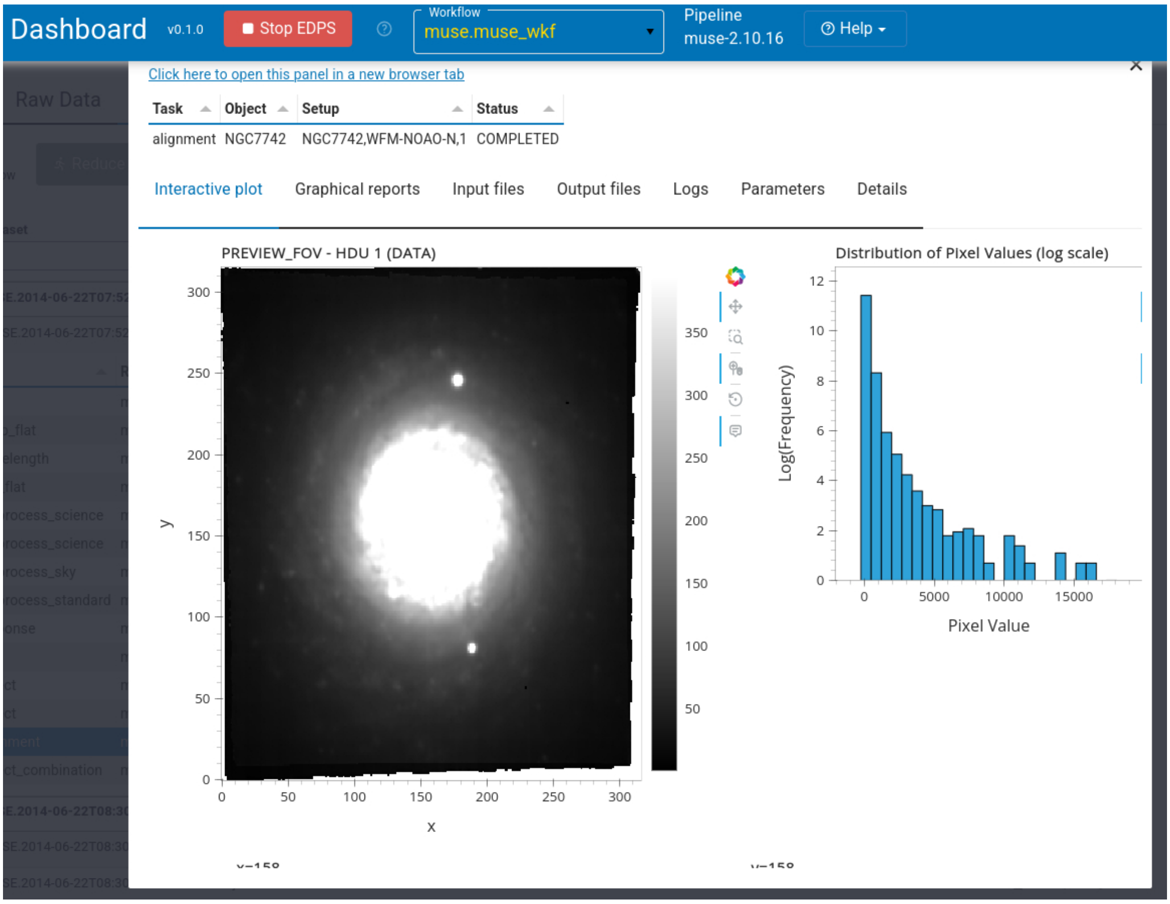


Figure 17: Quality plots for the alignment. On the left, the overview of the combined field of view after alignment, on the right an histogram of the pixel counts. Other information such as the list of inputs and outputs, recipe logs, recipe parameters, and job technical info can be accessed from the various tabs.

Tipp: In case of complicated alignment, it might be advisable to specify **alignment** as reduction target when creating the datasets in the **Raw Data** tab. In this way the reduction stops at this step, without wasting resources to combine the exposures. After the correct recipe parameters set has been found, the full reduction cascade can be triggered.

4 Overview of all the data reduction configuration options

4.1 Selection of most appropriate calibrations

By default, EDPS associates raw calibrations to the reduction process. It is also possible to use pre-processed calibrations (a.k.a. master calibrations) if available, in order to speed up the reduction. The preference can be specified in the Raw Data tab, before creating the datasets.

Possible values of the Calibration Preferences are:

- **raw_per_quality_level**: At equal quality of reduction, association of raw calibrations is preferred. This is the default.
- **master_per_quality_level**: At equal quality of reduction, association of master calibrations is preferred.
- **raw**. Association of raw calibration is preferred, despite the quality of results.
- **master**. Association of master calibration is preferred, despite the quality of results.

When master calibrations are used, the reduction step needed to process raw calibrations are not executed. The reduction then moves directly to the process of scientific exposures.

For example, if reduction speed for a quick check is preferred over a high quality reduction, one can select "master". In this case, old master calibrations are associated even if there are raw calibrations closer in time (and therefore more likely to ensure better quality products).

The quality level that the selected calibrations deliver is indicated close to each dataset in the ‘Raw input‘ tab, under the colum ‘CalibLevel‘. CalibLevel=0 indicates that calibrations that follow the rules of the instrument calibration plans have been selected. The higher the number, the poorer the quality of the products.

4.2 Configuration of parameters: the configuration editor

The data reduction of each dataset can be configured according to the scientific needs using an appropriate configuration editor.

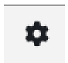
The EDPS workflows contain two types of parameters and they both have default values that can be modified to improve the data reduction.

- **Workflow parameters** (for some workflows only) are global and they are applied to the entire workflow. They are accessible both in the ‘Raw Data‘ tab, prior to the creation og a dataset, and in the ‘Reduction Configuration‘ editor, in the ‘Reduction queue‘ tab. Note: some workflow parameters were already configured before creating the dataset and sending it to the reduction queue. Here, they can be changed again. Please, note that the parameters have an effect only on the files that are already in the dataset. If one specifies a parameter that should include extra files in the dataset (e.g., the inclusion of more calibrations), files are not added and the reduction might fail. If you need to change a parameter that modifies the dataset content, please go back to the Raw data tab and create a new dataset.

- **Recipe parameters** are specific to the individual recipes and can be configured per task. They are accessible in the 'Reduction Configuration' editor, in the 'Reduction queue' tab.

This editor allows to configure the data reduction for a given dataset by specifying workflow and recipe parameters.

Note: some workflow parameters were already configured before creating the dataset and sending it to the reduction queue. Here, they can be changed again. Please, note that the parameters have an effect only on the files that are already in the dataset. If one specifies a parameter that should include extra files in the dataset (e.g., the inclusion of more calibrations), files are not added and the reduction might fail. If you need to change a parameter that modifies the dataset content, please go back to the Raw data tab and create a new dataset.

To open the editor, click on the wheel button  next to the dataset you desire to configure the reduction for. A window with the configuration editor appears as shown Figure 18.

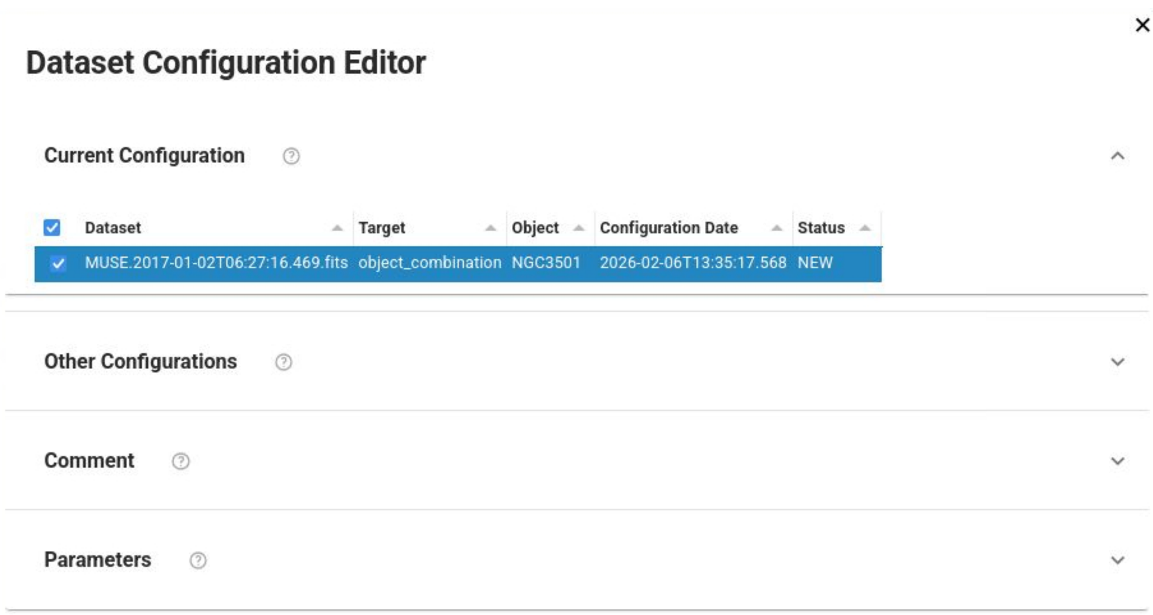


Figure 18: The Reduction Configuration editor. It contains 4 sections, that indicate the current configuration, list of other configurations to set, comments to insert, and the parameters to modify.

The editor is divided into 4 parts, which can be accessed pressing the corresponding expansion arrow.

- **Current configuration.** It indicates the name of the selected configuration for a given dataset (Figure 19).
- **Other configurations.** It allows to specify other configurations, to which the changes shall be copied to (Figure 20.
- **Comment** It allows to specify a comment to describe the configuration. It is possible to append or replace a comment (Figure 21). Comments can be changed on all configurations. It is possible to save the comment for the current configuration only, or for all the selected configurations.

- **Parameters.** A window as in Figure 22 appears.

  The window allows to:

  - Select the parameter set. A pre-determined list of workflow parameters and recipe parameters for a given use case. For the majority of the cases, the "science" parameter set can be used.
  - Edit the workflow parameters. These are parameters that regulates the reduction strategy, e.g. whether to use a given calibration or not, or to trigger a certain reduction step. Note that if the changes imply that some files not in the dataset are needed, the reduction might fail. In case, go back to the raw data tab, edit the workflow parameters there, and recreate the datasets.
  - Edit the recipe parameters. These are parameters associated to the recipe of a given task. Note: the same recipe parameters can be configured differently for the tasks that run the same recipe. Default parameters are shown (albeit some parameters can be dynamic, e.g. ‘EDPS‘ changes their value depending on the type of input data).

  Change the values according to the needs and then select whether to save it to the current or the selected configurations. Note, complete configurations cannot be modified, new configurations will be automatically created instead.

Figure 19: The first part of the Reduction Configuration Editor, that indicates the selected configuration.

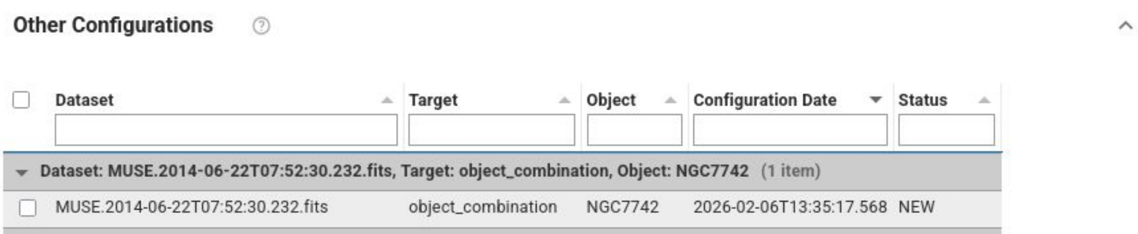


Figure 20: The second part of the Reduction Configuration Editor, that indicates other configurations for which we’d like to apply the changes.

4.2.1 Apply the same parameters to multiple datasets and configurations

A common user case is to apply the same set of optimized parameters to other datasets or other configurations. This is possible with the **Other configurations** tab of the configuration editor. To do so, proceed as follows:

- From the Reduction Queue tab, select the dataset/configuration that has the parameters you want to adopt for other reductions.

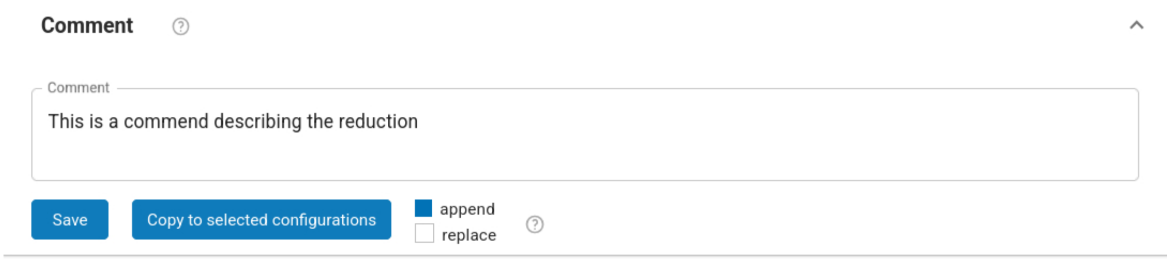


Figure 21: The third part of the Reduction Configuration Editor, that allows to specify a comment to the selected configurations.

- Open the configuration editor by clicking on the weel-button on the right-hand side, at the end of the line of the selected configuration.
- If necessary, edit the parameters.
- On the **Other configurations** tabs, select the datasets or configurations you want to reduce with the same parameters of the "Current configuration". Only datasets/configurations present in the Reduction Queue can be selected.
- Go on the Parameters tab of the configuration editor and click on the button **Copy to selected configurations**. Selected configurations that were labelled with "NEW" will be updated, selected configurations that were labelled with "COMPLETED/FAILED/ABORTED" will not be updated, but "NEW" configurations will be created instead.
- If you have edited paramters on the 3rd step, please note that the original "Current configuration" will not be updated by pressing the button **Copy to selected configurations**; only those selected in the "Other configurations" tab will be updated. If you want the edited parameters to be copied also to the "Current configuration", then press the "SAVE" button (available only if the Current configuration is labelled as NEW) or "Create new configuration" button (in this case the Current configuration is not updated, but a "NEW" configuration is created instead).

The newly created or updated configurations are now ready for reduction.

4.3 Parametrization of the Line Spread Function

The MUSE pipeline uses a model of the Line Spread Function (LSF) to model emission sky lines for the sky subtraction. An accurate knowledge of the LSF is crucial for a good removal of sky lines. The strategy for computing the LSF is set by the workflow parameter **lsfmode**. The best results are generally obtained by setting it to 'arc', which uses the arc lines of the wavelength calibrations taken the same day to model the line spread function.

If the model of the line spread function is not accurate, residual sky lines have to be expected. In particular, residuals with Pcygni profiles, or wings. In this case, it could be worth trying other methods Possible values of **lsfmode** are:

- **any**. Characterize the LSF from dedicated lsf raw exposures. If not available, use arc raw exposures. If not available, use static calibration from the pipeline distribution.
- **lsf**. Characterize the LSF from dedicated lsf raw exposures. If not available, use static calibration from the pipeline distribution. Dedicated lsf calibrations usually provide a better characterization of the LSF with respect to the regular arc calibrations, because they are more numerous. However, they are taken every few months, therefore they might not describe exaclty the LSF profile of the night where the observations were taken.
- **arc**. Characterize the LSF from dedicated arc raw exposures. If not available, use static calibration from the pipeline distribution.
- **static**. Use static calibration from the pipeline distribution. Use this option in combination with master calibration for fast, but potentially less accurate, reduction.

4.4 Sky subtraction

A crucial aspect of the MUSE data reduction is the sky subtraction. The workflow supports several strategies, which are determined by the value of the workflow parameter **skysubtraction**. By default **skysubtraction** is set to **model**.

In this configuration the MUSE workflow uses dedicated sky exposures (if available) to compute the SKY_CONTINUUM and the SKY_LINES calibrations. The SKY_CONTINUUM is the directly subtracted from the target field of view, whereas the intensities of the sky emission lines are refitted (using a small portion of the target field of view) to compensate the time variation between scientific exposure and dedicated sky observations. In some rare cases, this strategy could lead to over-subtracted sky lines, especially for objects that have intense spectral features that coincide with some isolated sky lines. In this case, it might be worth setting the parameter **skysubtraction** to **subtract-model**. Note that **skysubtraction=subtract-model** fails if no dedicated sky observations are present.

If dedicated sky exposures are not present, the sky (continuum and emission lines) is evaluated directly on the target field.

If the region used to compute the sky (either in the dedicated sky exposure or in the target field) contain astronomical objects, you might notice spurious absorption lines or distorted or surprisingly weak emission lines in your final spectrum. The region where the sky has to be computed can be set by dedicated recipe parameters (**fraction**, and **fraction_ignore**, see below) or by including a SKY_MASK among input files.

Possible values of **skysubtraction** are:

- **model** (default). It uses fluxes indicated in an input SKY_LINES table (either static calibration, or coming from a dedicated sky exposures) as starting estimates, but re-fits them on the global sky spectrum created from the science exposure. If a SKY_CONTINUUM is computed from dedicated sky exposures, it is directly subtracted from the data. Otherwise, the sky continuum is computed directly from teh science exposure. The portion of the field of view of the science exposures it uses to fit the sky lines and compute the continuum is regulated by the recipe parameters **skymodel_fraction** and **skymodel_ignore**. Select this option if no dedicated sky exposures are available (sky lines and sky continuum will be computed on the target field of view), or if dedicated sky exposures are present and enough empty sky regions are available in the target field of view (sky continumm is computed on dedicated sky exposures, sky lines are re-fitted on the target field of view). To discard completely dedicated sky exposures, remove them from the pool of input data.

- **subtract-model**. It uses sky emission lines and sky continuum determined on the field of view of dedicated sky exposures. Select this option only if **model** fails. This method works only if dedicated sky exposures are present.
- **auto**. It automatically set the parameter **skysubtraction** to **model** if no dedicated sky exposures are present in the dataset, and to **subtract-model** if dedicated sky exposures are present.
- **simple**. It creates a sky spectrum from the science data, and directly subtracts it, without taking the LSF into account (LSF_PROFILE and input SKY files are ignored).
- **none**. It does not perform a sky subtraction.

The following recipe parameters of the tasks **object** and **object_sky**, also have an impact on the quality of the sky subtraction. They can be accessed by selecting the corresponding task from the 'Recipe parameters' section of the configuration window.

- **skymodel_fraction**. It specifies the fraction of the field of view (or of the SKY_MASK, if present among the inputs) to be used when calculating the sky continuum and sky emission lines.
- **skymodel_ignore**. It specifies the fraction of the field of view (or of the SKY_MASK, if present among the inputs) to exclude when calculating the sky continuum and sky emission lines. These parameters are applied to the histogram of the flux values of the pixels. For example, if **skymodel_ignore=0.05** and **skymodel_fraction=0.8**, then the brighter 80% of the pixels, with the exclusion of the fainter 5%, are included in the computation. They refer to the full field of view of the input frame, or only to the part specified by the input sky mask.

The following recipe parameters of the task **sky**, also have an impact on the quality of the sky subtraction. They can be accessed by selecting the corresponding task from the 'Recipe parameters' section of the configuration window.

- **fraction**. As **skymodel_fraction**, but it is applied to the dedicated sky exposure.
- **ignore**. As **skymodel_ignore**, but it is applied to the dedicated sky exposure.

Important note: The MUSE `EDPS` workflow automatically sets **skymode-fraction** and **skymodel-ignore** according to the following schema.

- **skymodel-ignore=0.05** always.
- If no dedicated sky exposures are present: **skymodel-fraction=0.4**.
- If **skysubtraction=model** (or **skysubtraction=auto** but no dedicated sky exposures are present): **skymodel-fraction=0.75** (i.e. the majority of the field of view is supposed to be free from object contamination).

- For all other cases, **skymodel-fraction=0.2** (only 20% of the field of view is supposed to be free from object contamination).

To override these values, specify the values in the 'Override value' field of the configuration window.

4.5 Creation of a SKY_MASK

It is possible to specify a sky mask, e.g. a fits file per exposure that specifies the regions in the field of view to consider as pure sky. The best way for doing it is to start the reduction without sky mask and specify the targets **object** and **sky**. The product folder will contain the SKY_MASK automatically computed by the pipeline. Edit them with your favourite fits editor and copy them in the input directory. Restart the reduction. `EDPS` will associate to each target and dedicated sky frame the SKY_MASK with the same mjd-obs (so, make sure you have the correct mjd-obs correspondence). To use the entire SKY_MASK set **skymodel_fraction** and **fraction** to 1, and **skymodel_ignore** and **ignore** to 0 in the desired actor (e.g.: **object** for regular science frames, **sky** for the dedicated sky exposures), of the configuration panel.

4.6 Combination of exposures

The MUSE pipeline allows to align and combine scientific exposures belonging to the same target. It is possible to specify the method how to identify the exposures that are meant to be combined, this is regulated by the workflow parameters **combine_science**, **max_diameter_WFM**, **max_separation_WFM**, **max_diameter_NFM**, **max_separation_NFM**.

`EDPS` first groups the scientific exposures present in the input directories by the header keyword specified by the workflow parameter **combine_science**. Most useful values are:

- **obs.targ.name** groups all those exposures that have the same target name and instrumental setup.
- **obs.id** groups all those exposures that have the same OB identification and instrumental setup.
- **tpl.start** groups all those exposures that have the same template start and instrumental setup.
- **exptime** groups all those exposures that have the same exposure time and instrumental setup.
- **night** groups all exposures of the same night and the same instrumental setup.
- **instrume** groups all exposures of the same instrumental setup.

Then, within each group, `EDPS` creates sub-groups on the basis of their sky coordinates. Exposures in the same sub-group are aligned and combined together. The keywords that regulates how exposures in each group are clustered by sky position are: **max_diameter_WFM** and **max_separation_WFM**, for the Wide Field Mode, and **max_diameter_WFM** and **max_separation_WFM**, for the Narrow Field Mode.

- **max_diameter_WFM**. It specifies the maximum projected diameter of the sub-group of exposures to combine, in degrees. If an exposure would make the sub-group bigger than this value, it is not included in this sub-group.

- **max_separation_WFM**. Is specifies the maximum separation on the sky from an exposure and a sub-group, in degrees. If the exposure is more distant than this threshold, it is not included in the sub-group. If it is closer, and if its inclusion does not make the sub-group size to exceed **max_diameter_WFM**, then it is added to the sub-group.

The default values ensures that exposures with an overlap of about 1/4 of the field of view are grouped together, and that the resulting mosaic does not exceed 3 times the MUSE field of view.

Note: To process individual exposures without combining them together, specify the reduction target **object** and remove **object_combination** when creating the datasets in the 'Raw Data' tab.

4.7 Sky flats

By default, the MUSE pipeline uses sky flats exposures to account for large scale illumination variations within the field of view. The workflow parameter **use_skyflats** specifies whether or not to use skyflats in the data reduction cascade. Possible values are:

- **no**. Do not use sky flats in the reduction cascade. Set the parameter to 'no' only if the skyflats are not available or if they have bad quality.
- **yes** (default). Use sky flats in the reduction cascade. This option requires the skyflats to be present. It is the recommended strategy."

4.8 Autocalibration

The MUSE pipeline has an algorithm (autocalibration) designed to minimize the non-homogeneities in flux calibration between different IFUs and slices. It works under the assumption that the vaste majority of the field of view is dominated by the background sky (e.g., only few nearly point-like sources are present in the field) and that the sky bakground is constant within the field of view.

The workflow supports several autocalibration strategies, that are defined by the workflow parameter **autocalibration**. Possible values are:

- **none**. Does not apply self-calibration (default).
- **deepfield**. Applies the self-calibration on the target field itself, under the condition that the sky background is constant and each IFU/slice is dominated by sky background (exposures of small and sparse objects).
- **closest_sky**. It applies the self-calibration algorithm to a sky exposure of the same night and instrument setup. The self-calibration coefficients determined on the sky exposure are then applied to the scientific target exposure. If not available, the dataset is reduced without autocalibration. Results are often sub-optima, hence inspect the resulting field of view to assess the outcome.
- **user** It uses the closest in time user-provided calibration table (from the same night). If not available among the input data directory, the dataset is incomplete.

4.9 Telluric correction

.

The MUSE pipeline derives the atmospheric transmission (category: STD_TELLURIC) from the standard star. The correction is then applied to the entire datacube. This is enabled by default. If telluric lines are still present in the final product, one can skip the telluric correction (set the workflow parameter **telluric_correction** to 'FALSE') and do it with external tool.

4.10 Wavelength range

It is possible to specify the wavelength range of the output by setting the parameters **wavelength_min** (default 4000 Å) and **wavelength_max** (default 10000 Å). If the full wavelength range is not needed for scientific analysis, one can speed up the reduction process by specifying a shorter wavelenght range , e.g. by concentrating only on the spectra features of scientific interest.

4.11 Geometric and astrometric calibrations

By default, the ESO archive provides the closer in time geometric and astrometric calibrations. Generally, they ensure the best reduction and there is no need to change them. However, if one wants to reprocess them, it has to download the corresponding raw calibrations (DPR.TYPE = WAVE,MASK and ASTROMETRY, respectively) from the archive and set the workflow parameters **recompute_geometry** and **recompute_astrometry** to 'yes'. Note that the computation of geometry calibration is expensive in terms of computation and RAM.

4.12 Removal of DARK currents

The DARK current in MUSE detector is very low. Dark calibrations are taken as part of the instrument monitoring and calibration plan, but it is not advisable to process and remove a master dark from the scientific data, as it increase the noise in the product. Therefore, darks are not used in scientific reduction. To reduce raw dark frames, specify the reduction target "dark" in the 'Raw Input' tab, before creating the datasets.

In case one wants to test the effects of including darks in the scientic reduction, set the workflow parameter **use_darks** to 'yes'.

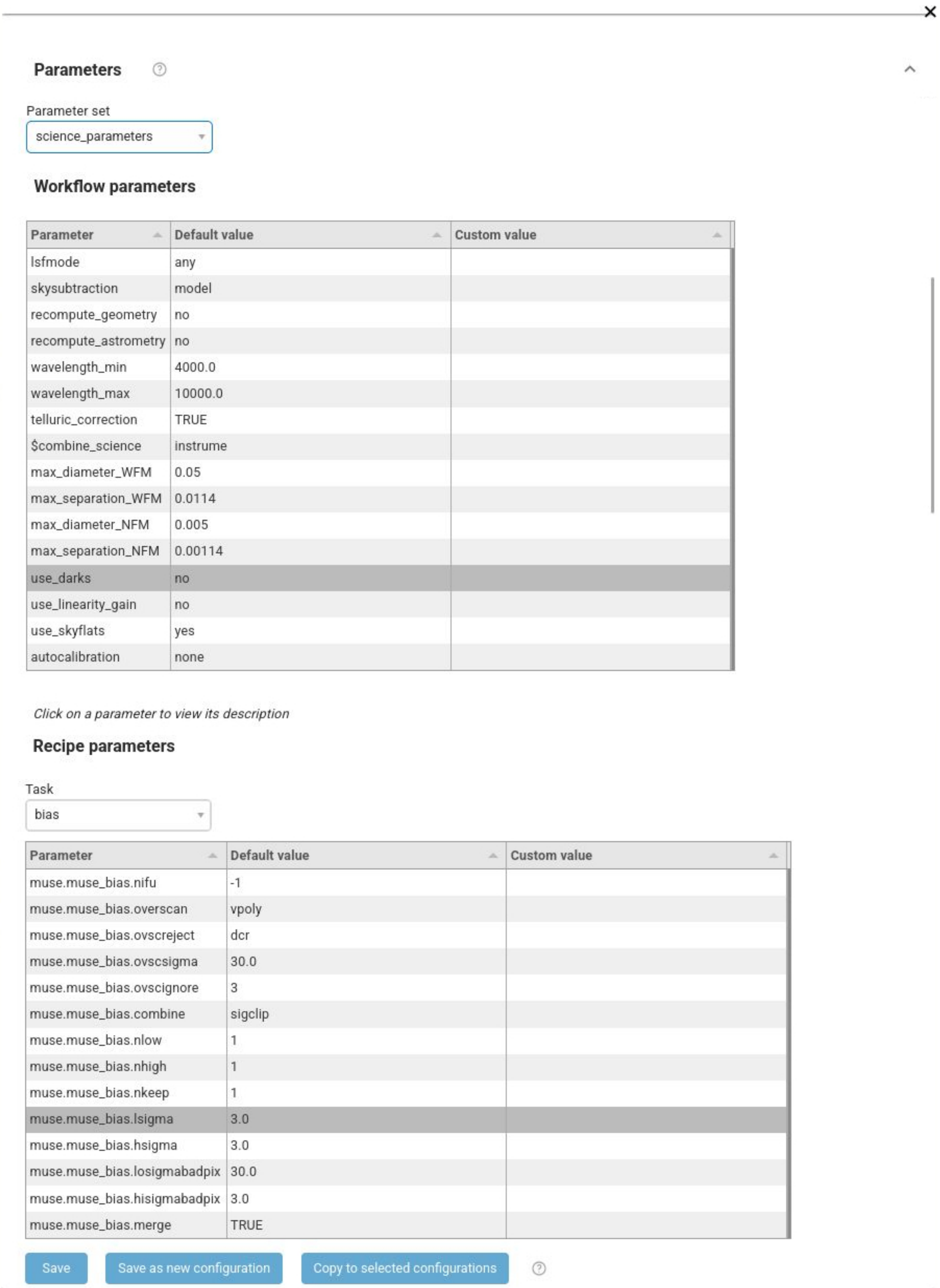


Figure 22: The fourth part of the Reduction Configuration Editor, that allows to specify the parameters sets and the recipe parameter per task. These settings can be applied to the "Selected Configuration" (Fig. 19) or to the "Other Configurations (Fig. 20).

5 List of workflow tasks

This is the list of all the tasks and associated recipes in the MUSE workflow. Only some of them are needed for scientific reduction, they are indicated by the flag "yes" (triggered by default) or "optional" (triggered only if requested by a workflow parameter). Other tasks are not used for scientific reduction (they are indicated by the flag "no"), they are mainly used for instrument monitoring and they can be executed only by specifying them as target. Note that, when a task is specified as target, all the tasks that generate the calibrations needed for it are automatically executed.

TASK	RECIPE	Used in science reduction	Notes
alignment	muse_alignment	yes	Aligns a set of exposures for combination using source list.
alignment_sky	muse_alignment	no	Aligns a set of sky exposures for combination using source list.
astrometry	muse_astrometry	no	Computes astrometry.
autocalibration_from_object_sky	none	no	Selects sky exposures for additional slice-to-slice flat fielding.
bias	muse_bias	yes	Computes the master bias.
dark	muse_dark	optional	Computes the master dark.
geometry	muse_geometry	optional	Generates geometry calibrations.
lamp_flat	muse_flat	yes	Computes master flat field.
line_spread_function_arc	muse_lsf	yes	Computes LSF from arc exposures.
line_spread_function_lsf	muse_lsf	optional	Computes LSF from dedicated calibrations.
linearity_and_gain	muse_lingain	no	Computes linearity and gain correction.
object	muse_scipost	yes	Creates a datacube from individual scientific observations.
object_combination	muse_exp_combine	yes	Creates a datacube from combined scientific observations.
object_sky	muse_scipost	no	Processes sky exposures as they were science exposures.
preprocess_astrometry	muse_scibasic	optional	Reduces astrometric calibrations.
preprocess_science	muse_scibasic	yes	Reduces science exposures.
preprocess_sky	muse_scibasic	yes	Reduces sky exposures.
preprocess_sky_sci	muse_scibasic	no	Reduces sky exposures as they were science exposures.
preprocess_standard	muse_scibasic	yes	Reduces standard star observations.
response	muse_standard	yes	Computes the response function and atmospheric transmission.
sky	muse_create_sky	yes	Computes the sky background from dedicated sky exposures.
sky_combination	muse_exp_combine	no	Combines sky frames as they were science exposures.
sky_flat	muse_twilight	optional	Processes sky flats for illumination correction.
source_detection	muse_detection	yes	Detects sources in the datacube and creates a source catalog.
source_detection_sky	muse_detection	no	Detects sources in the datacube of sky exposure and creates a source catalog.
throughput	muse_ampl	no	Computes the throughput.
wavelength	muse_wavecal	yes	Computes wavelength calibration.

Table 1: MUSE pipeline tasks overview

6 Frequently Asked Questions

- **Q1) Where can I find the final reduced data?**

  Answer: all the products of all the datasets and the reductions are saved into the EDPS_data directory, specified when executing the `edps-gui` for the first time. One can decide to export only the final products for selected datasets and only for the desired reduction attempts into another location for further analysis. See Section 2.4 for further instructions.

- **Q2) How do I stop the application?**

  Answer: Proceed as follows:

  1. Press "Stop EDPS" in the Dashboard.
  2. Type Ctrl-C in the terminal where the application is running. If the application doesn't terminate, type Ctrl-C again.
  3. Alternatively, kill the 'panel serve' process on your system, for example:

     ```
     ps -e | grep panel # get the process ID of the gui (<pid>).
     kill -9 <pid>
     ```

- **Q3) I have closed the browser window where the application is running. How can I reopen the application?**

  Answer: Point your browser to: `http://localhost:5006/edps-gui`

- **Q4) Where can I find some data that I can use to test the application?**

  Answer: Install the 'datademo' package provided with the pipeline installation or download the "Demo Data" package from https://www.eso.org/sci/software/pipe_aem_table.html.

  Please note that the demo data can be large (tens of Gigabytes).

  A convenient script to download demo data for any pipeline is also available and can be used from the command line:

  ```
  curl -O https://eso.org/sci/software/apptainer/eso_download_demodata.sh
  bash ./eso_download_demodata.sh
  ```

- **Q5) How can I start the edps-gui if the following message appears?**

  ```
  Cannot start Bokeh server, port 5006 is already in use
  ```

  Answer: The panel server was not closed properly. Kill it by typing:

  ```
  ps -e | grep panel # get the process ID of the gui (<pid>).
  kill -9 <pid>
  ```

- **Q6) How do I get additional support on EDPS or data reduction in general?**

  Answer: For suggestions, questions, or feedback in general, please open a ticket with the EDPS Support team. This https://support.eso.org/new-ticket?ticket%5Bticket_field_13%5D%5Bdata%5D=227 should take you directly to a webpage for creating and `EDPS` feedback ticket, but in case you want to navigate there 'manually', go to https://support.eso.org, login, click on "Submit Helpdesk Ticket", and specify the Help topic: "Post Observations", "ESO Data Processing System [EDPS]".

- **Q7) I have a lot of disk space, but when I install `EDPS` with `pip` or an ESO pipeline with `Homebrew` I get the error message: `Cannot mkdir:  No space left on device`. How do I fix it?**

  Answer: This depends on how much disk space is allocated to the `/home`, `/var`, and `/tmp` directories. The final solution would be to resize the space allocated to the in the organization of the filesystem. However, we list here few tricks that might do the job.

  – Clearing the pip .cache to make space for new packages. Type the command:

  ```
  pip cache purge
  ```

  before installing `EDPS`.

  – Redirect the cache, `Homebrew` temporary build directories into a partition with enough space. Set some of the following environmental variables in your `.bashrc` file:

  ```
  export HOMEBREW_CACHE=<path_to_new_cache_directory>
  export XDG_CACHE_HOME=<path_to_new_cache_directory>
  export HOMEBREW_TEMP=<path_to_new_temporary_directory>
  export TMPDIR=<path_to_new_temporary_directory>
  ```

  The first moves only the location of `Homebrew` cache, the second the cache of most applications (instead of the default `/home/username/.cache`), the third moves the directory where `Homebrew` builds, extracts, and saves temporary files (instead of the defaults `/tmp` and `/var/tmp`). The last changes the global system temporary directory and affects most of the linux commands.

  – As extreme measure, one can move the `/home/linuxbrew/.linuxbrew` directory somewhere else, and create a symbolic link in `/home/linuxbrew`. For example:

  ```
  cd /home/linuxbrew
  mv -f .linuxbrew <path_to_new_directory>
  ln -s <path_to_new_directory> .linuxbrew
  ```

  Important note: this operation might break some internal links. Recipes requiring external packages such as `telluriccorr` might not work (impacts on KMOS, XSHOOTER, FORS2, and MOLECFIT pipelines).