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

EFOSC EDPS-GUI tutorial

1 rev. 20260516

Issue 1

Date May 16 2026

Prepared:	Write Y. Name	May 16 2026	.....
	Name	Date	Signature
Approved:			.....
	Name	Date	Signature
Released:			.....
	Name	Date	Signature

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 2.3.12

This page was intentionally left blank

Contents

1 Introduction

1.1 Scope

This document describes how to reduce EFOSC 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 EFOSC 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 EFOSC pipeline itself, consult the pipeline manual available at:
https://www.eso.org/sci/software/pipe_aem_table.html.

Note: this tutorial refers to:

- EFOSC instrument pipeline named `efosc`, version 2.3.12.
- EFOSC workflow: `efosc.efosc_wkf`
- `EDPS` version 1.5.7.
- `edps-gui` version 1.0.

Note:, the `efosc.efosc_wkf` deals both with imaging and spectroscopic data. The EFOSC pipeline distribution, however, includes two additional workflows `efosc.efosc_spectroscopy_wkf`, for spectroscopic data only, and `efosc.efosc_imaging_wkf`, for imaging data only. They can be used independetly.

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

In this Section we show how to reduce EFOSC demo data with the generic `efos.efosc_wkf` workflow. The same procedure applies for imaging data and spectroscopic datasets. More details on the individual instrument modes will be given in Sections **??** and **??**, respectively.

Follow this procedure to quickly reduce EFOSC demo data. We assume that the `EDPS`, `edps-gui`, the EFOSC 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).

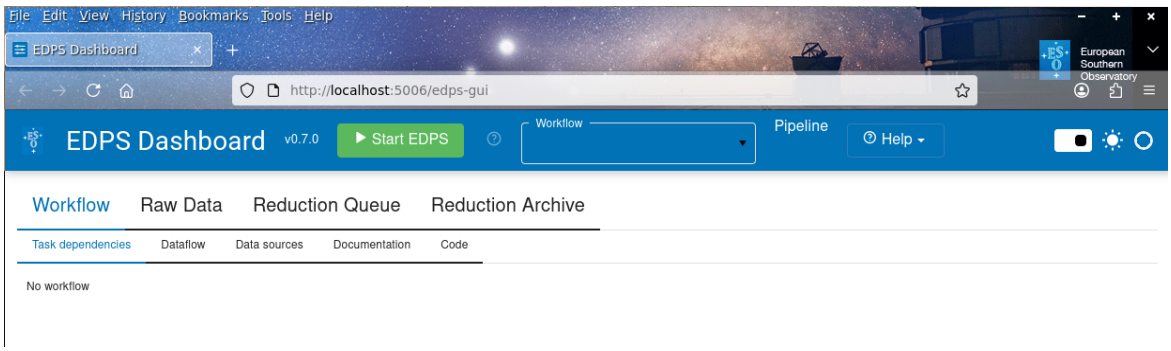

Figure 1: The empty `edps-gui` Dashboard; the underlying `EDPS` engine has not yet been started and no workflow has been loaded.

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

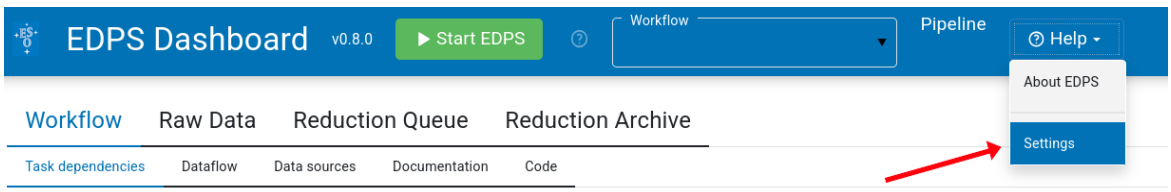

Figure 2: The “Help” –> “Settings” menu.

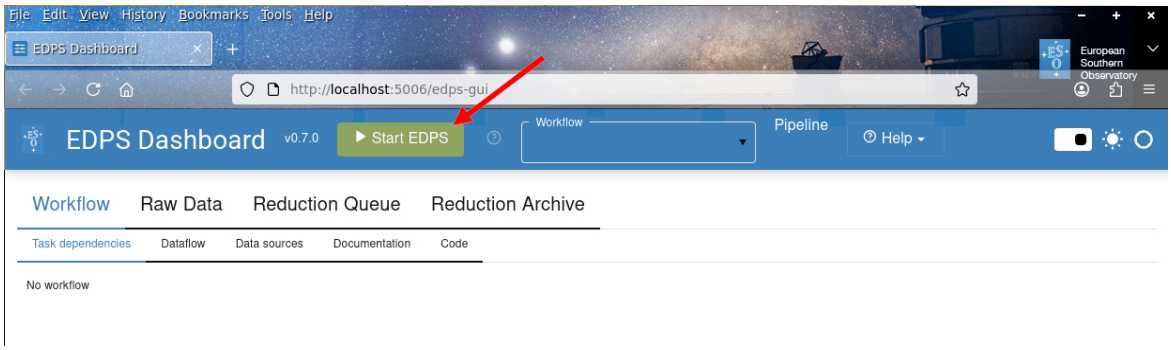

Figure 3: The “Start EDPS” button.

4. On the browser window with the dashboard, press the button ‘Start EDPS‘.

5. Choose the `efosc.efosc_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.

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 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).

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.

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.

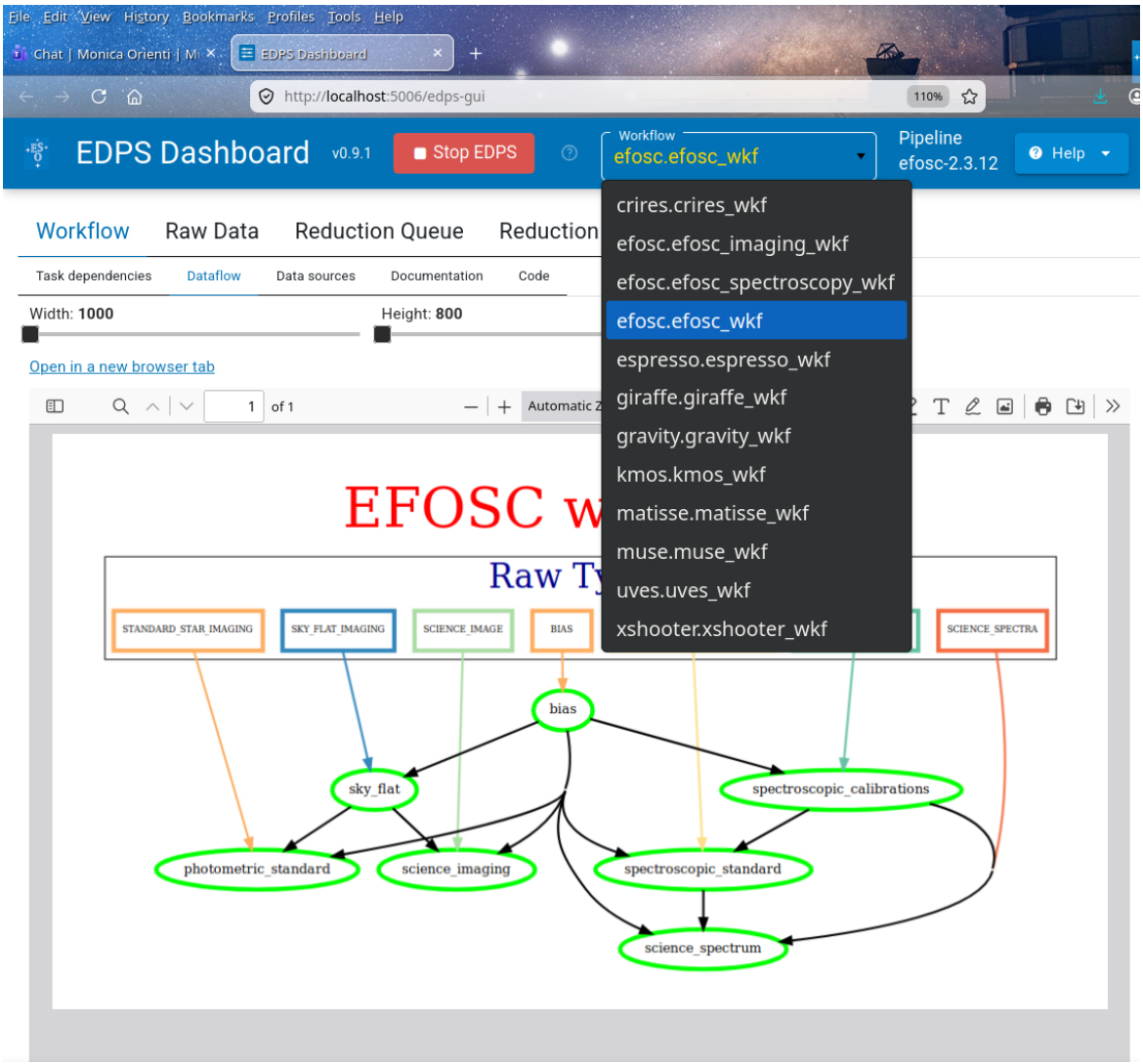

Figure 4: The `edps-gui` with the EFOSC workflow loaded.

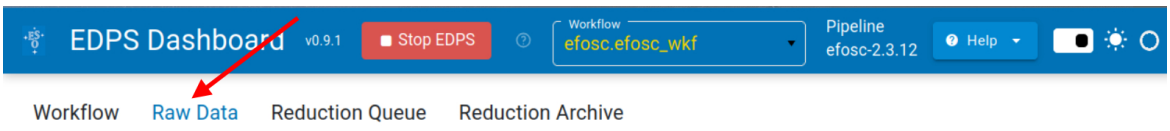

Figure 5: How to select RAW data Tab.

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.

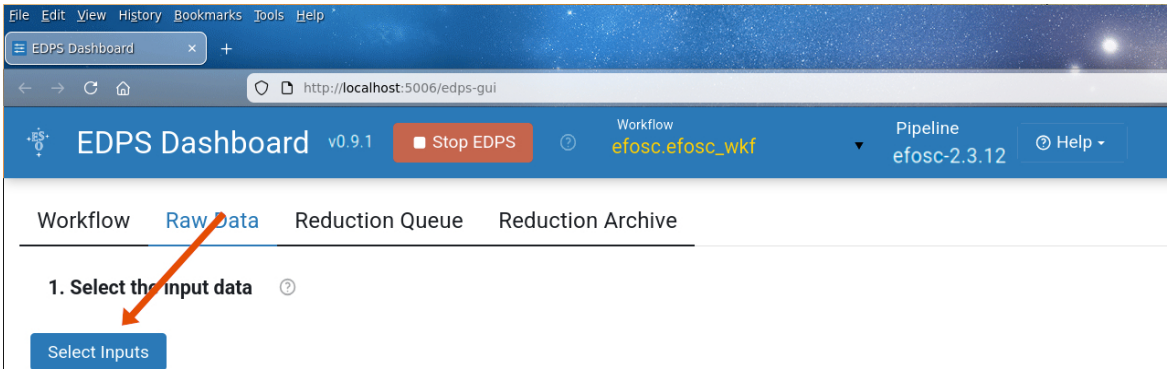

Figure 6: How to select input data.

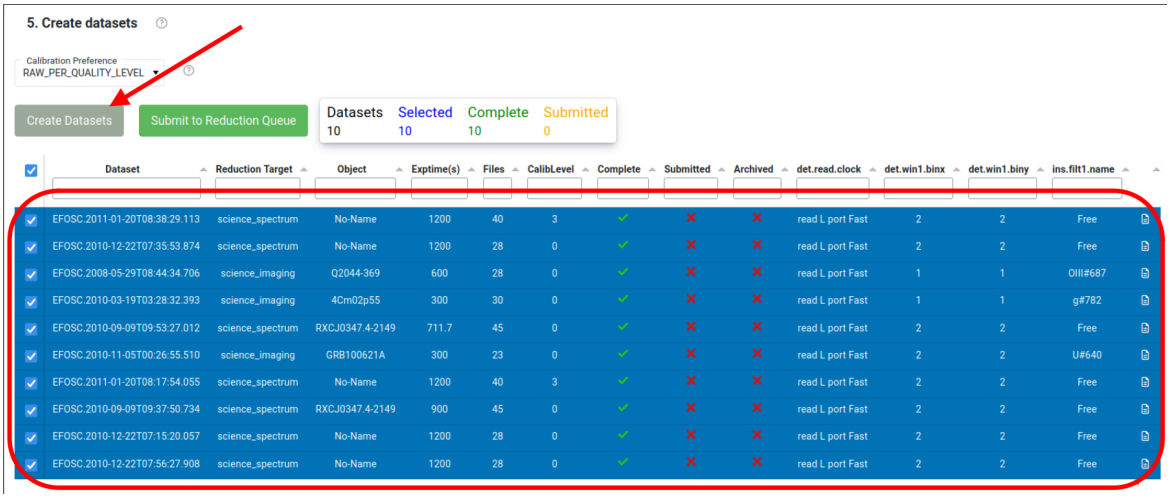

Figure 7: How to inspect the input data directory to create datasets.

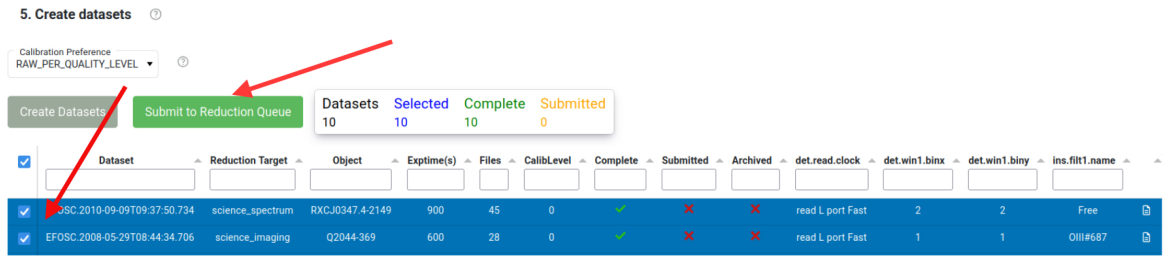

Figure 8: How to send the selected datasets to the Reduction Queue for processing.

2.3.1 Quality plots

Almost all processing tasks can display the input raw frames and the products in the so called "quality plots", which can be inspected from the ‘Reduction Queue‘ window. Those associated for the main product can be inspected by pressing the magnifying glass symbol at the right side of each dataset. To inspect those associated to each individual job (if created),

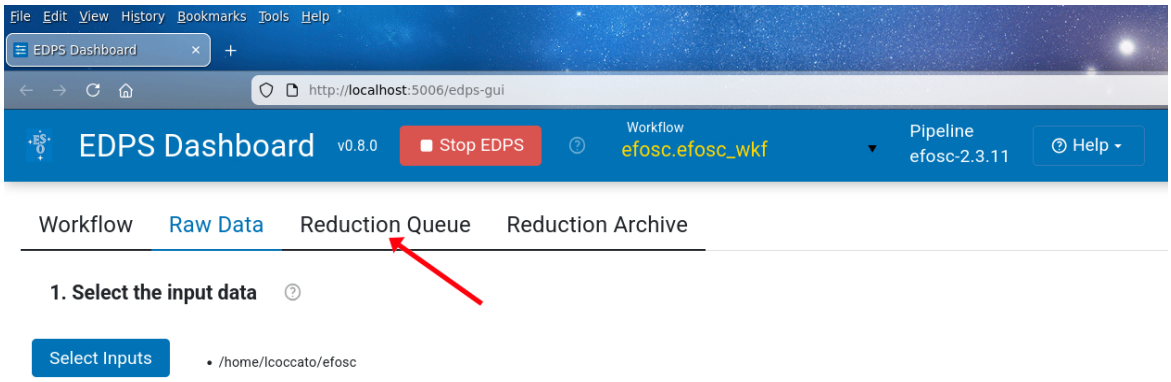

Figure 9: How to select Reduction Queue tab.

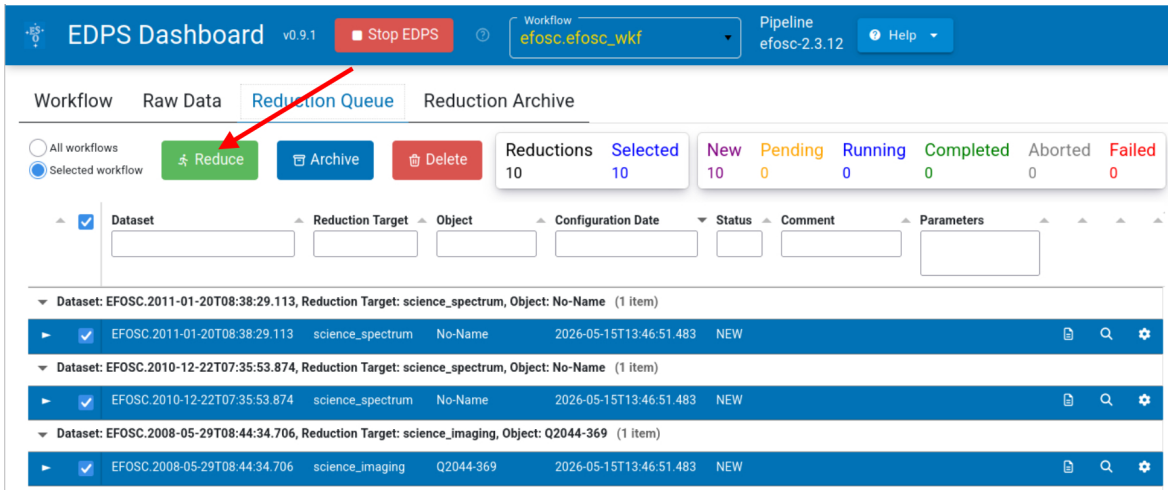

Figure 10: Reduce.

- 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)

- Press the magnifying glass symbol at the right side of the job you want to inspect. Only plots for completed jobs can be inspected.

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:

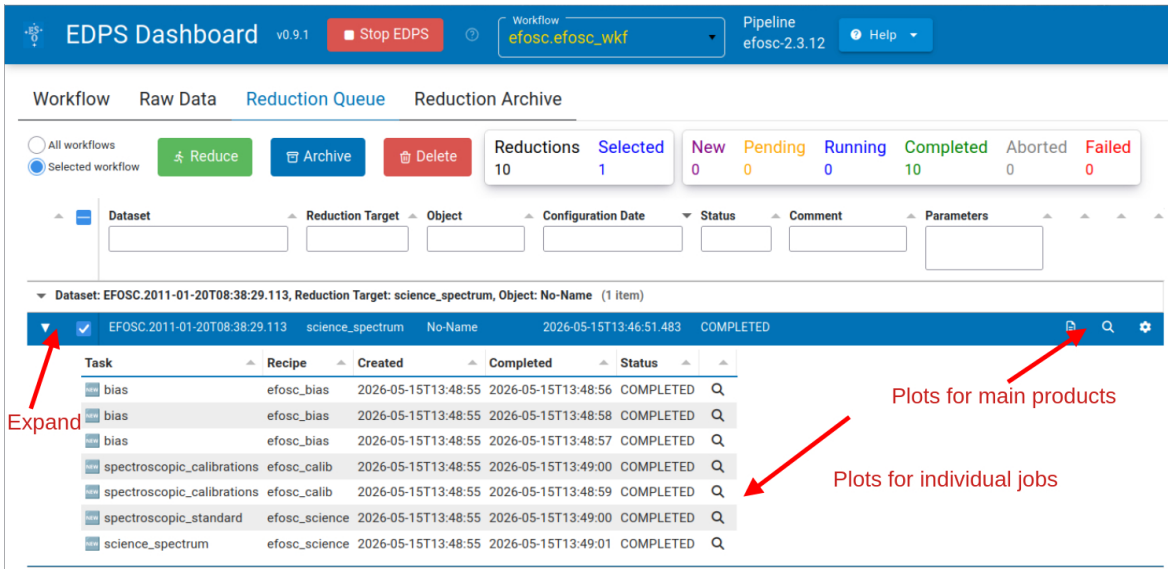

Figure 11: How to look for quality plots from the Reduction Queue tab.

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:

- IMAGE_`arcfile`. Reduce image corresponding to the exposure specified by the `arcfile` keyword, imaging workflow only.
- CATALOG_`arcfile`. Catalogue of detected sources in IMAGE_`arcfile`, imaging worlflow only.
- SPECTRUM_MOS_`arcfile`. Reduced spectra extracted from the exposure specified by the `arcfile` keyword, spectroscopic workflow only.

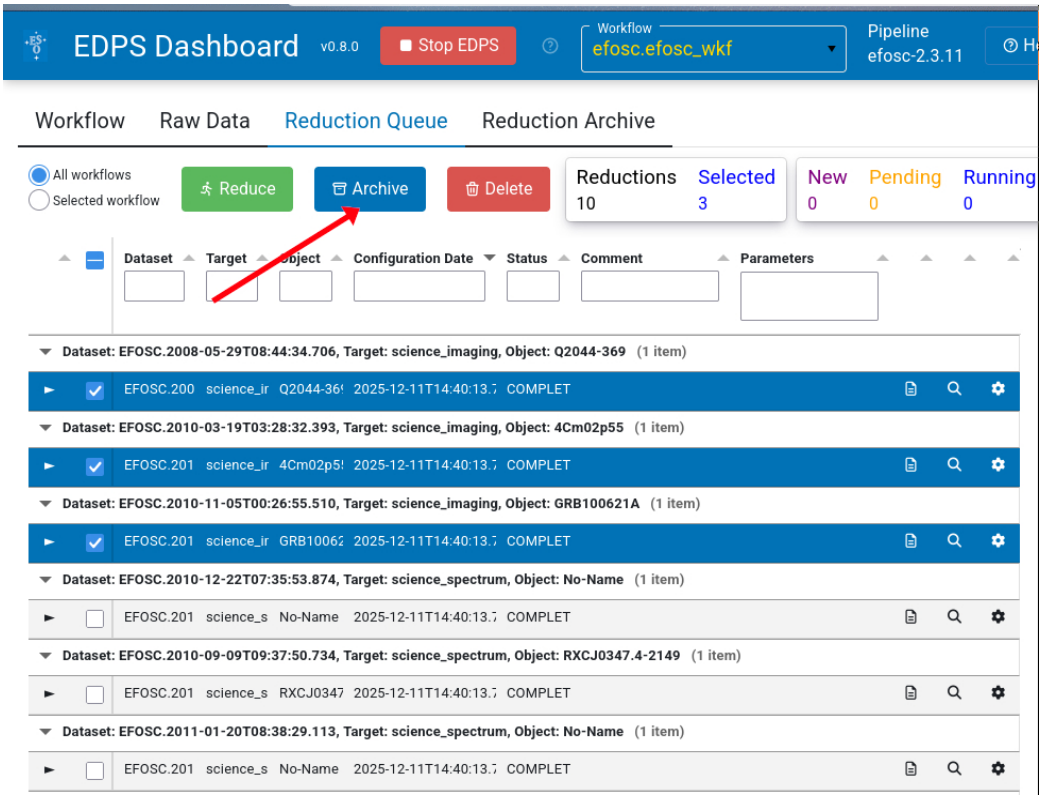

Figure 12: How to archive a completed reduction from the Reduction Queue tab.

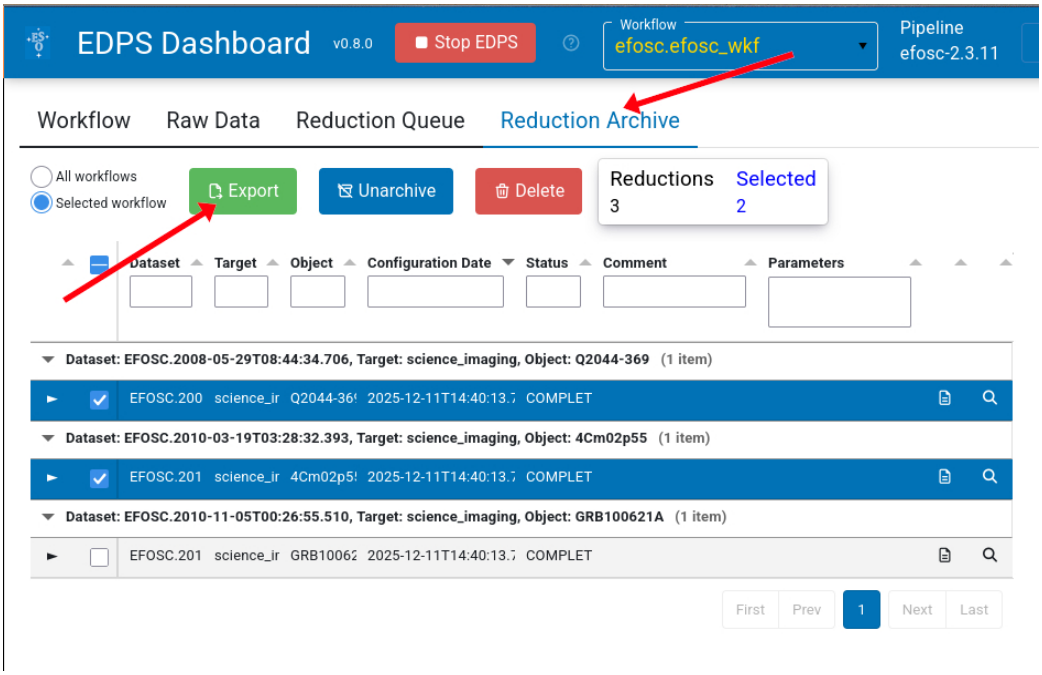

Figure 13: The reduction archive tab. This table contains all the different configurations of datasets that are declared "finished" and removed from the Reduction Queue. From this page, the user can export the most important files into a desired local directory.

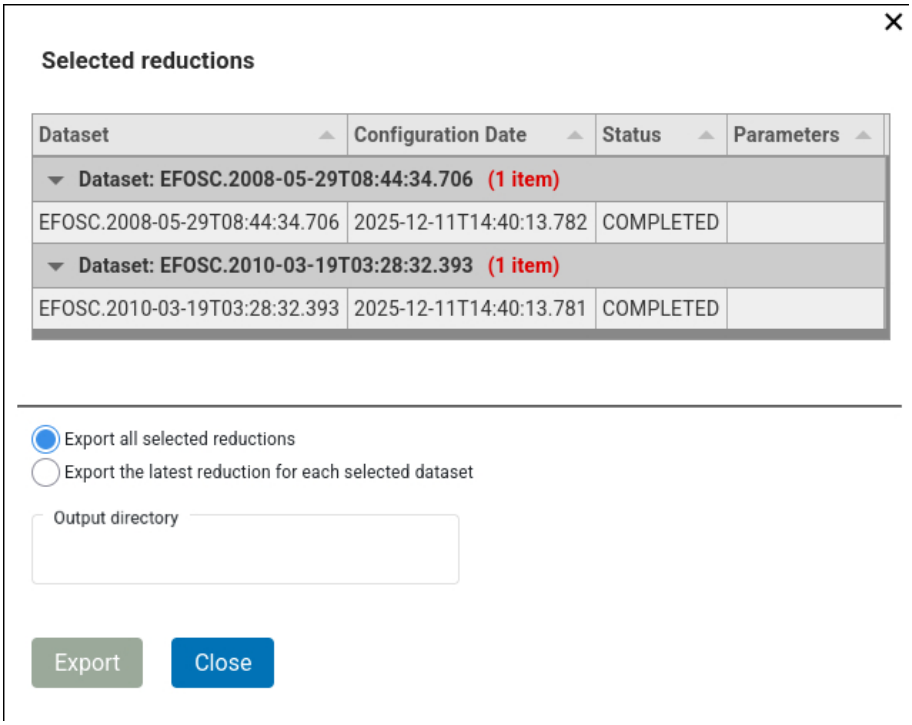

Figure 14: The EXPORT dialogue window, where the user can decide which reduced configuration to save and where.

3 The EFOSC data reduction flow.

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

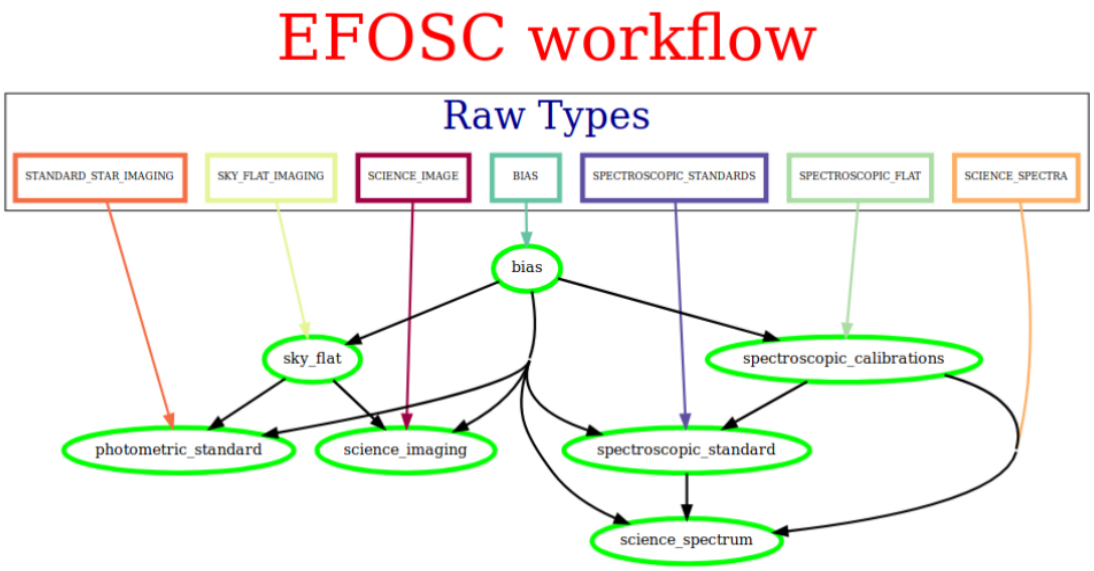

Figure 15: The data reduction cascade of the EFOSC workflow.

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 `efosc.efosc_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 `efosc.efosc_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 Imaging workflow

3.1.1 Bias

The task **bias** reduces bias frames and generates a master bias calibration. Typically, default parameters give satisfactory results.

3.1.2 Sky flat

The task **bias** reduces sky flats and generates a master flat field calibration. Typically, default parameters give satisfactory results. If the field does not contain bright stars, it might be advisable to set to the 'stack_method' to 'mean' fron the configuration editor (Section 4.2).

3.1.3 Photometric Standard

The task **photometric_standard** runs the recipe `efosc_zeropoint` to process observations of standard stars and compute the photometric zeropoint and exinction coefficient. The recipe runs sextractor to identify objects and perform aperture photometry. Typically, default recipe parameters deliver good results. Standard stars are used to determine the zeropoint only for filters "U#640" (using catalogue from Landolt 1992, 2007), and filters "B#639", "V#641", "R#642", "i#705" (using catalogs from Stetson 2000).

This extinction coefficient may not be the correct one for a given night. In such a case the zeropoint will reflect the mismatch of the extinction. If the airmass differs between the observations of the standard star and that of the science the mismatch will not be fully corrected by the zeropoint. Please refer to the EFOSC2 pipeline user manual (Izzo et al. 2016: Sections 9 and 10) for the details of this recipe and the extraction algorithms employed.

Note that the results of this task are not used to calibrate the scientific exposure.

3.1.4 Science image

The task **science_imaging** runs the recipe **efosc_img_science** to reduce the raw image, and compiles a catalogue of extracted sources using sextractor.

If zeropoint and exinction coefficient determined from the standard star have to be applied, they have to be applied outside the workflow, as the feature is not yet supported by the pipeline.

3.2 Spectroscopic workflow

3.2.1 Bias

The task **bias** reduces bias frames and generate a master bias calibration. Typically, default parameters give satisfactory results.

3.2.2 Spectroscopic calibrations

The task **Spectroscopic calibrations** runs the recipe `efosc_calib` to process arc lamps and flat fiels to determine the wavelength calibration, spectral distortion, and the flat fielding of spectroscopic calibrations. All spectroscopic modes are supported. Typically, default parameters give satisfactory results.

3.2.3 Spectroscopic standard

The task **spectroscopic_standard** runs the recipe `efosc_science` on the standar stars and produces a response curve, which will subsequently be used to flux-calibrate the science observation. Typically, default recipe parameters deliver good results.

3.2.4 Science spectrum

The task **Science spectrum** reduces scinetific spectroscopic observations for all spectroscopic modes. The EFOSC workflow will flux-calibrate the science observation using the instrument response curve derived from the standard star observation if it exists in the current DataSet. If no standard star observation exists in the current DataSet, then the science observation will not be flux-calibrated.

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. The current version of the EFOSC workflow contains only **recipe parameters**, therefore the field associated to the **Workflow parameters** is currently empty.

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 the recipe parameters.

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 16.

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

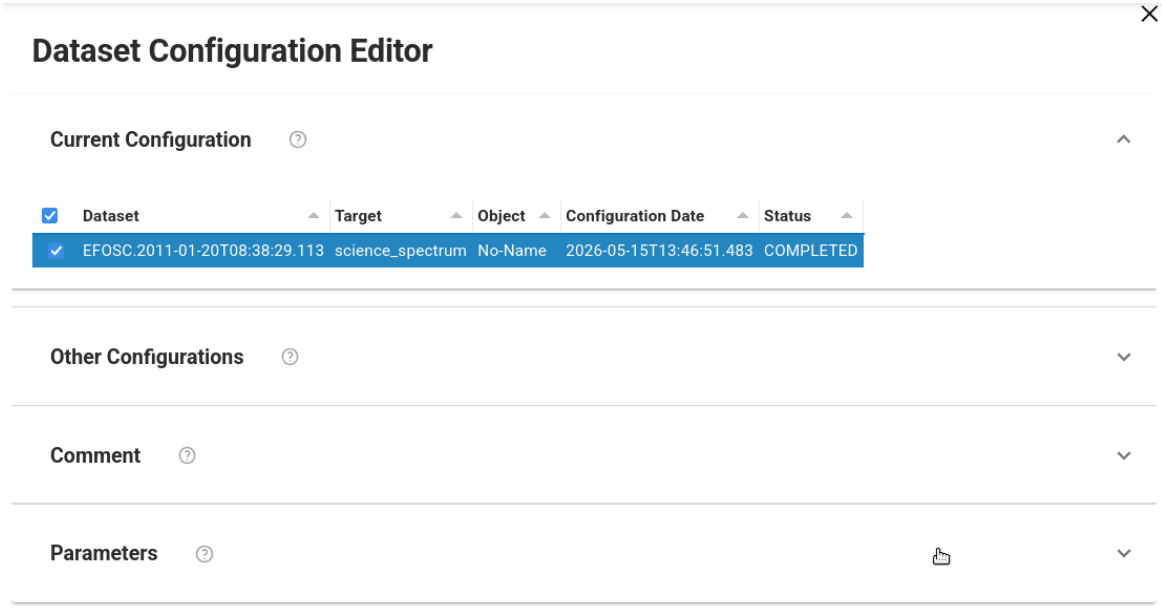

Figure 16: The Reduction Configuration editor. It contains 4 sections, that indicate the current configuration, list of other configurations to set, comments to insert, and the parameters to modify.

- **Current configuration.** It indicates the name of the selected configuration for a given dataset (Figure 17).
- **Other configurations.** It allows to specify other configurations, to which the changes shall be copied to (Figure 18.
- **Comment** It allows to specify a comment to describe the configuration. It is possible to append or replace a comment (Figure 19). 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 20 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.

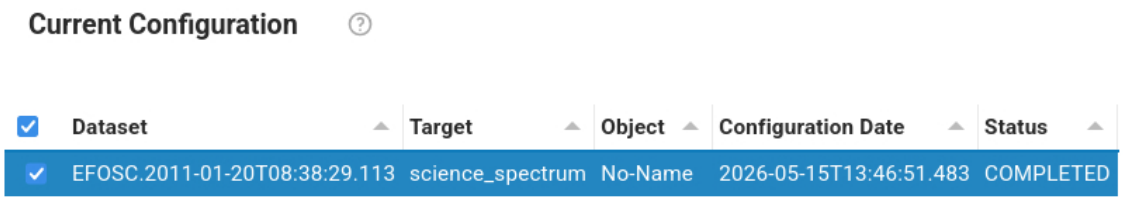

Figure 17: The first part of the Reduction Configuration Editor, that indicates the selected configuration.

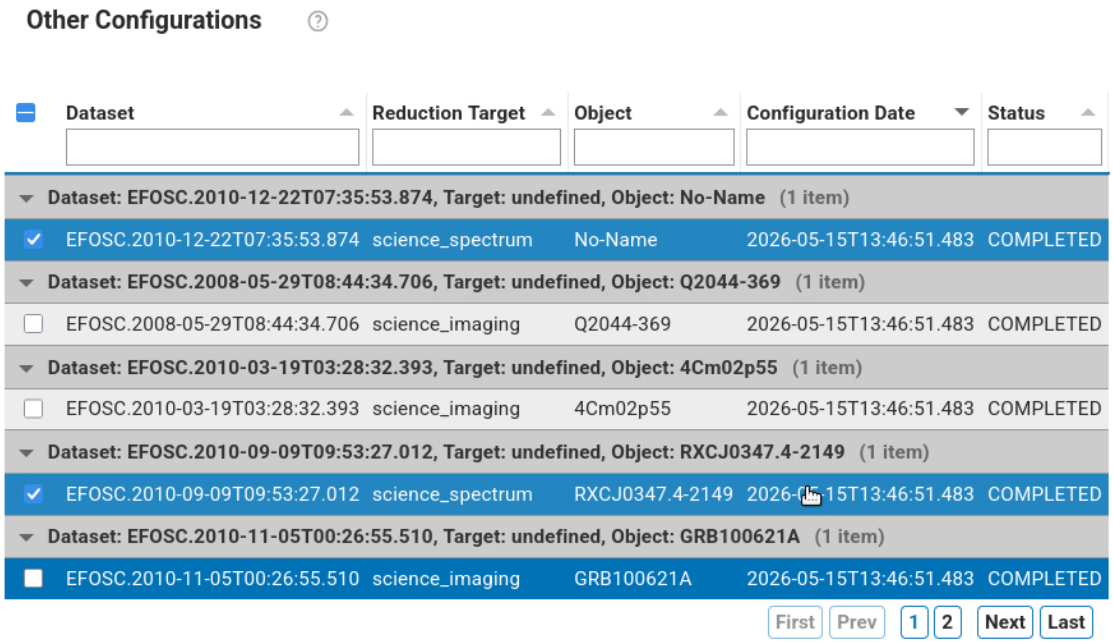

Figure 18: The second part of the Reduction Configuration Editor, that indicates other configurations for which we'd like to apply the changes.

OTHER CONFIGURATIONS

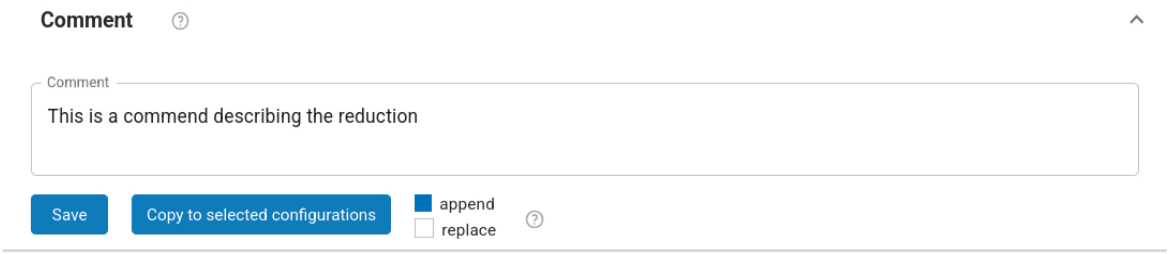

Figure 19: The third part of the Reduction Configuration Editor, that allows to specify a comment to the selected configurations.

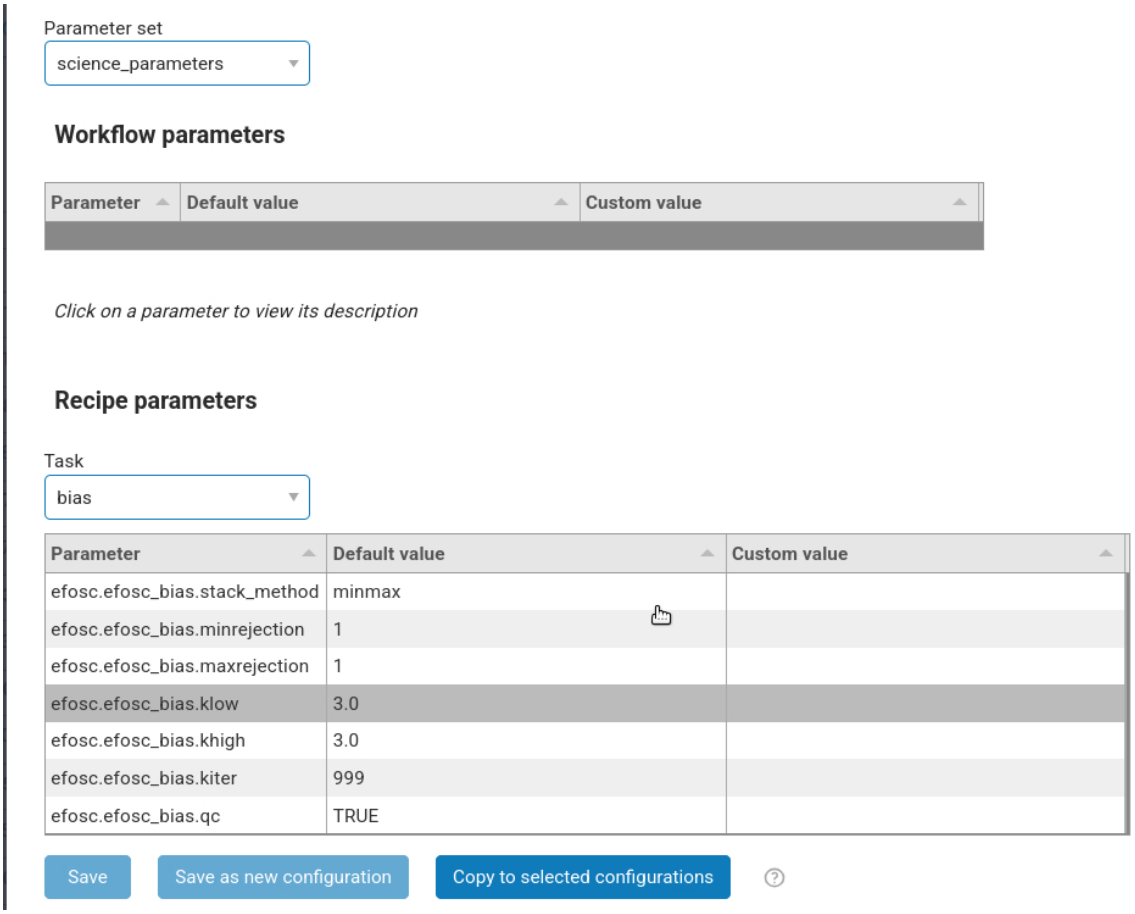

Figure 20: The fourth part of the Reduction Configuration Editor, that allows to specify the parameters sets and the recipe parameter per task. These settings can be applied to the "Selected Configuration" (Fig. 17) or to the "Other Configurations (Fig. 18).

5 List of workflow tasks

This is the list of all the tasks and associated recipes in the EFOSC workflow. Note that the task **photometric_standard** is not directly associated to the process of science observations.

TASK	RECIPE	Used in workflow	Notes
bias	efosc_bias	both	Creates MASTER_BIAS
photometric_standard	efosc_zeropoint	imaging	Computes photometric zeropoint and processes standard star
science_imaging	efosc_img_science	imaging	Reduces raw images
science_spectrum	efosc_science	spectroscopy	Reduces raw spectra (long-slit and MOS)
sky_flat	efosc_img_sky_flat	imaging	Reduces sky flats
spectroscopic_calibrations	efosc_calib	spectroscopy	Process spectroscopic calibrations (flats and arcs)
spectroscopic_standard	efosc_science	spectroscopy	Reduces raw standard stars

Table 5.0.0: EFOSC tasks and associated recipes.

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).