EDPS

Release 1.5.7

EDPS development team

Jun 22, 2026

CONTENTS

Graphic User Interface (EDPS-Dashboard)

CHAPTER

ONE

EDPS-GUI QUICK-START

1.1 Introduction

1.1.1 Scope

This document is meant for first time user of EDPS, and guides through the installation procedure and data reduction using the EDPS dashboard (Graphic User Interface). After working through this tutorial, the reader should be able to use other supported EDPS workflows for the reduction of user provided data.

1.1.2 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.

1.1.3 Additional documentation

The ESO instrument pipelines web pages are available here.

A more detailed description of the `EDPS` dashboard and its component is available here:

For specific instructions on how to reduce data from a particular instrument, consult the corresponding EDPS tutorials available in the main documentation page

---

Go to EDPS quick-start guide *index*

1.2 Installation

1.2.1 Prerequisites

1. Recent Firefox or Chrome browser, Python 3.11 or higher (but there are issues with Python 3.14).
2. 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.

3. Install graphviz, fv, and ds9, which have to be included in the system path (definying aliases not enough). Graphviz can be easily installed via:

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

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
```

1.2.2 Installation steps

To install EDPS follow these steps:

1. 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.

2. Install the required packages:

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

Go to EDPS quick-start guide *index*

1.3 Quick start

This Quick Start Guide walks through the steps to reduce data using the default setup.

1.3.1 1. Setting the workflow

a. After installation and while still in the installation virtual environment (if not, activate it *again*), start the EDPS dashboard by typing:

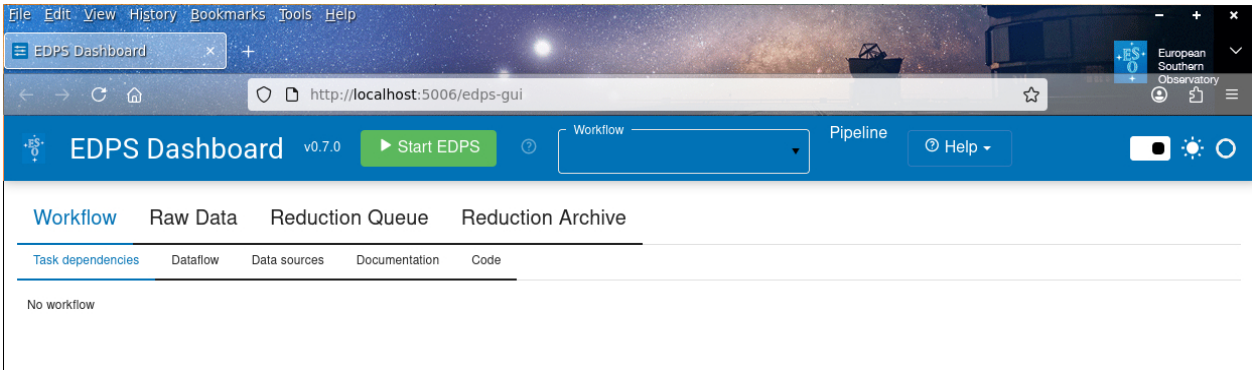
```
edps-gui
```

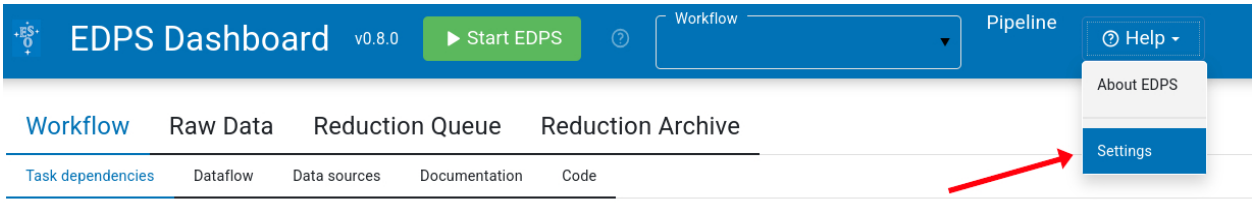
The EDPS dashboard will start in a browser window.

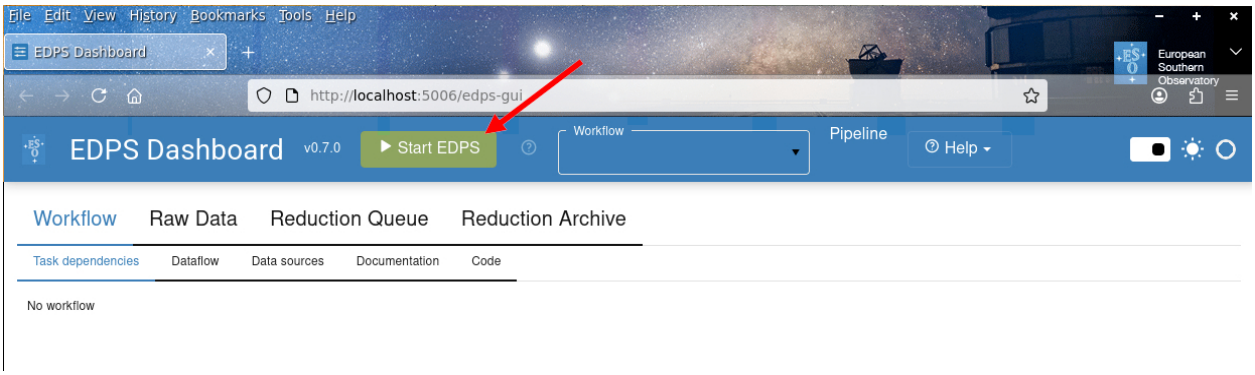
b. Optionally, before starting EDPS, one can specify new settings by pressing Help –> Settings (see here)

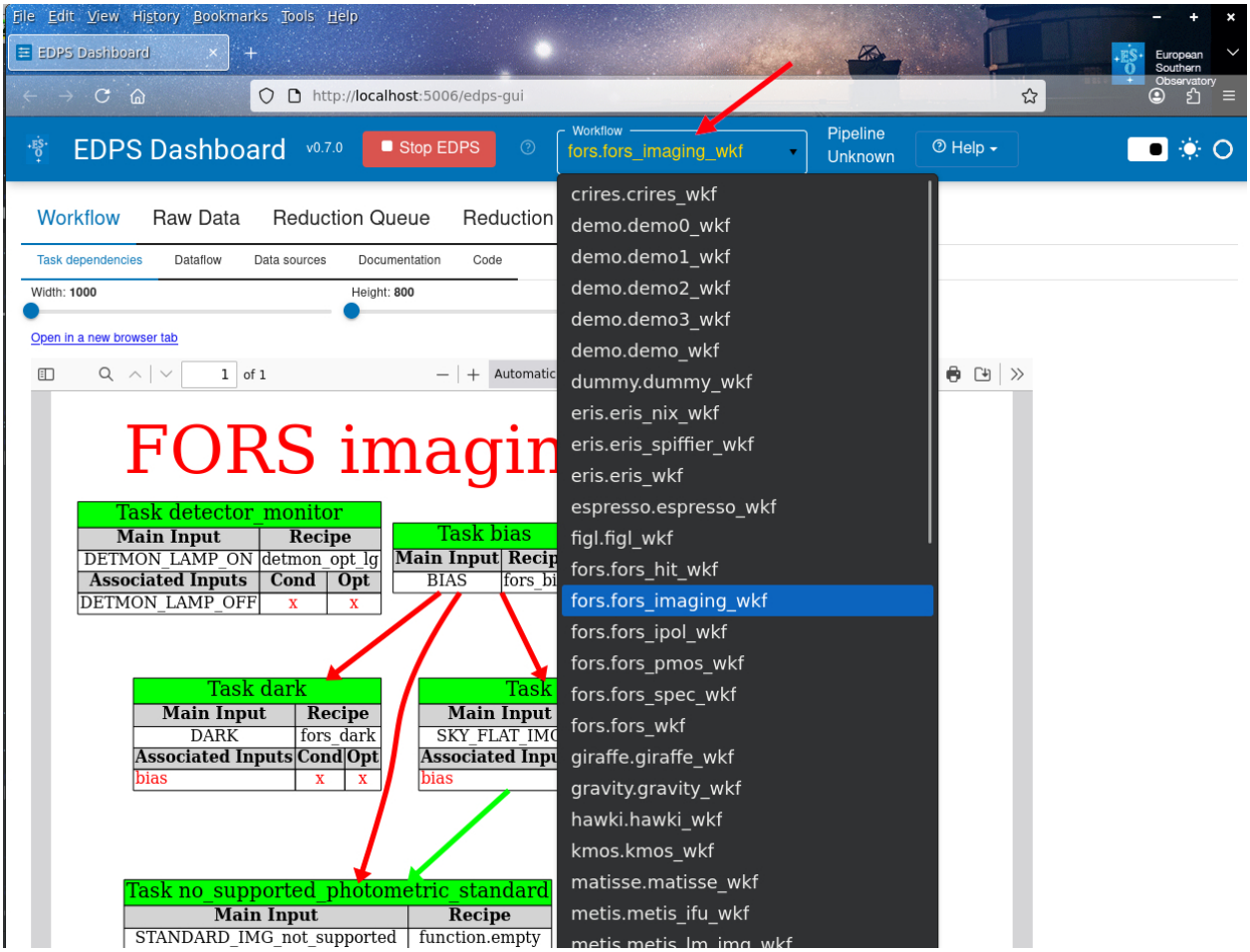
c. On the browser window with the EDPS dashboard, press the button `Start EDPS`.

d. Choose a pipeline workflow that matches the data to be reduced. The workflows offered in this selector depend on the installed pipelines.


File Edit View History Bookmarks Tools Help
EDPS Dashboard
http://localhost:5006/edps-gui
European Southern Observatory
EDPS Dashboard
v0.7.0
Start EDPS
Workflow
Pipeline
Help
Workflow
Raw Data
Reduction Queue
Reduction Archive
Task dependencies
Dataflow
Data sources
Documentation
Code
No workflow

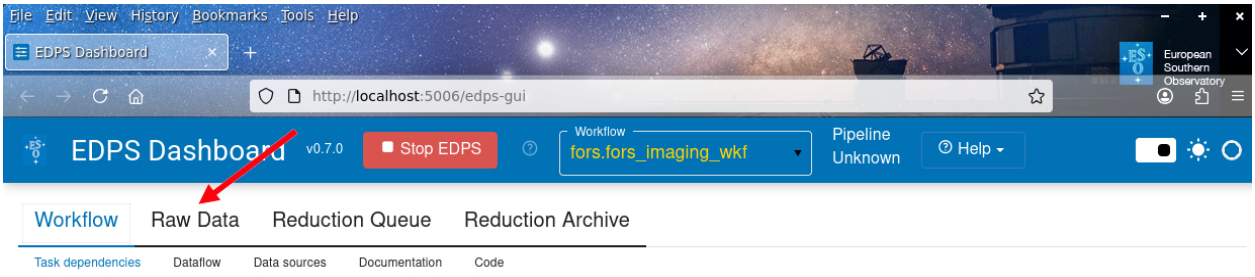

EDPS Dashboard
v0.8.0
Start EDPS
Workflow
Pipeline
Help
About EDPS
Settings
Workflow
Raw Data
Reduction Queue
Reduction Archive
Task dependencies
Dataflow
Data sources
Documentation
Code


File Edit View History Bookmarks Tools Help
EDPS Dashboard
http://localhost:5006/edps-gui
European Southern Observatory
EDPS Dashboard
v0.7.0
Start EDPS
Workflow
Pipeline
Help
Workflow
Raw Data
Reduction Queue
Reduction Archive
Task dependencies
Dataflow
Data sources
Documentation
Code
No workflow

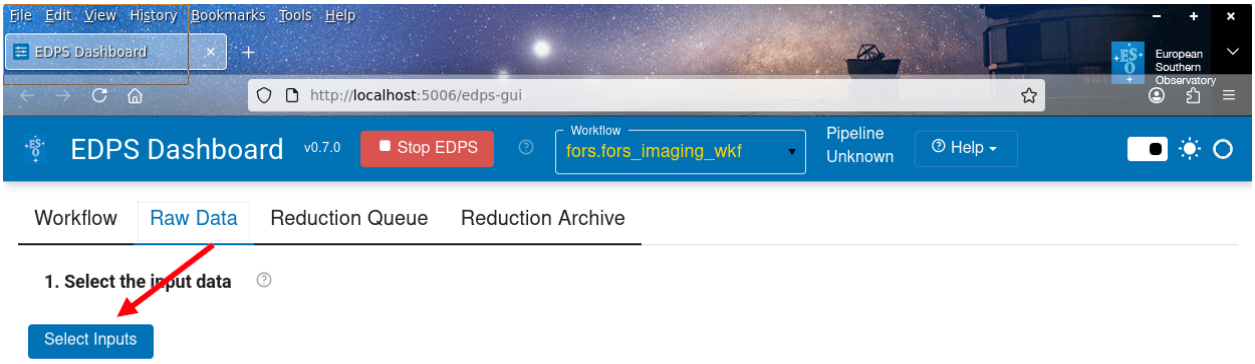

File Edit View History Bookmarks Tools Help
EDPS Dashboard
http://localhost:5006/edps-gui
European Southern Observatory
EDPS Dashboard v0.7.0
Stop EDPS
Workflow
fors.fors_imaging_wkf
Pipeline
Unknown
Help
Workflow
Raw Data
Reduction Queue
Reduction
Task dependencies
Dataflow
Data sources
Documentation
Code
Width: 1000
Height: 800
Open in a new browser tab
1 of 1
FORS imagin
Task detector_monitor
Main Input
Recipe
DETMON_LAMP_ON
detmon_opt_lg
Associated Inputs
Cond
Opt
DETMON_LAMP_OFF
x
x
Task bias
Main Input
Recip
BIAS
fors_bi
Task dark
Main Input
Recipe
DARK
fors_dark
Associated Inputs
Cond
Opt
bias
x
x
Task
Main Input
SKY_FLAT_IMG
Associated Inpu
bias
Task no_supported_photometric_standard
Main Input
Recipe
STANDARD_IMG_not_supported
function.empty
crires.crires_wkf
demo.demo0_wkf
demo.demo1_wkf
demo.demo2_wkf
demo.demo3_wkf
demo.demo_wkf
dummy.dummy_wkf
eris.eris_nix_wkf
eris.eris_spiffier_wkf
eris.eris_wkf
espresso.espresso_wkf
figl.figl_wkf
fors.fors_hit_wkf
fors.fors_imaging_wkf
fors.fors_ipol_wkf
fors.fors_pmos_wkf
fors.fors_spec_wkf
fors.fors_wkf
giraffe.giraffe_wkf
gravity.gravity_wkf
hawki.hawki_wkf
kmos.kmos_wkf
matisse.matisse_wkf
metis.metis_ifu_wkf

1.3.2 2. Selecting the input data

a. Press `Raw Data`.

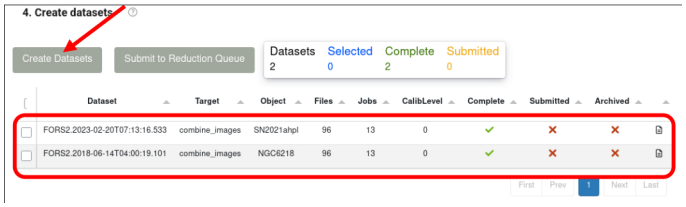


b. Press `Select Inputs`. A selection window will appear that allows to select data that are stored on a local disk.



c. (Optional). Select the reduction target, configure the workflow parameter and specify the association preferences. Thise steps are optional. For more information please consult the full edps-gui guide here.

d. Press `Create Datasets`. A list of datasets appears, one line for each set of science data.



e. Choose the datasets that should be processed

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

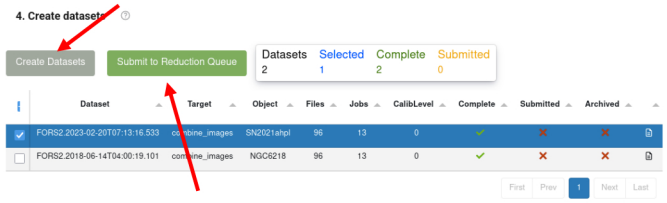
1.3.3 3. Start the reduction

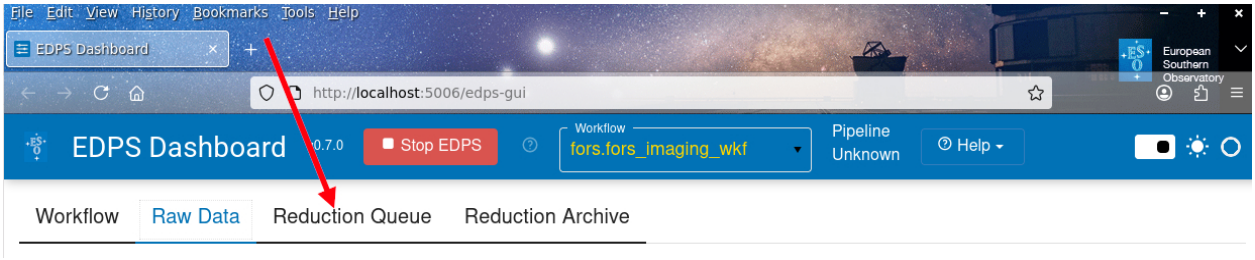
a. Press `Reduction Queue`.

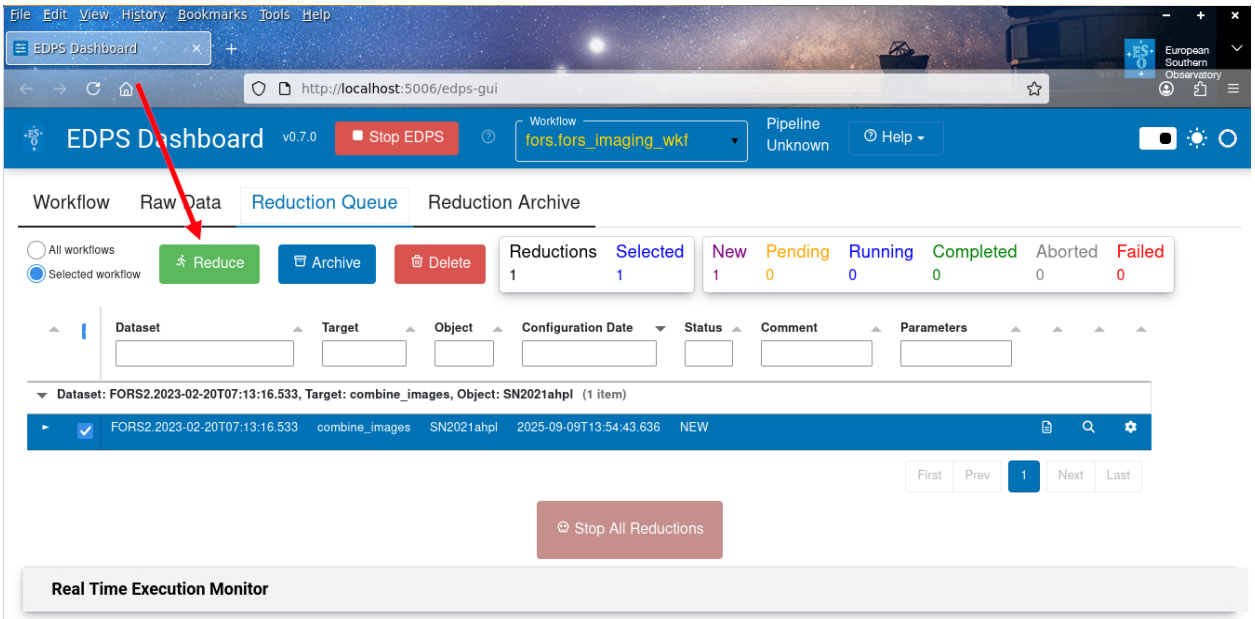
b. (Optional). Configure the workflow and recipe parameters by pressing the wheel button  to open the configuration editor.

c. Press the `Reduce` button. The selected data will now be processed with the default 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.


4. Create datasets
Create Datasets
Submit to Reduction Queue
Datasets Selected Complete Submitted
2 1 2 0
Dataset Target Object Files Jobs CalibLevel Complete Submitted Archived
FORS2.2023-02-20T07:13:16.533 combine_images SN2021ahpl 96 13 0
FORS2.2018-06-14T04:00:19.101 combine_images NGC6218 96 13 0
First Prev 1 Next Last


File Edit View History Bookmarks Tools Help
EDPS Dashboard
http://localhost:5006/edps-gui
European Southern Observatory
EDPS Dashboard
v0.7.0
Stop EDPS
Workflow
fors.fors_imaging_wkf
Pipeline
Unknown
Help
Workflow
Raw Data
Reduction Queue
Reduction Archive

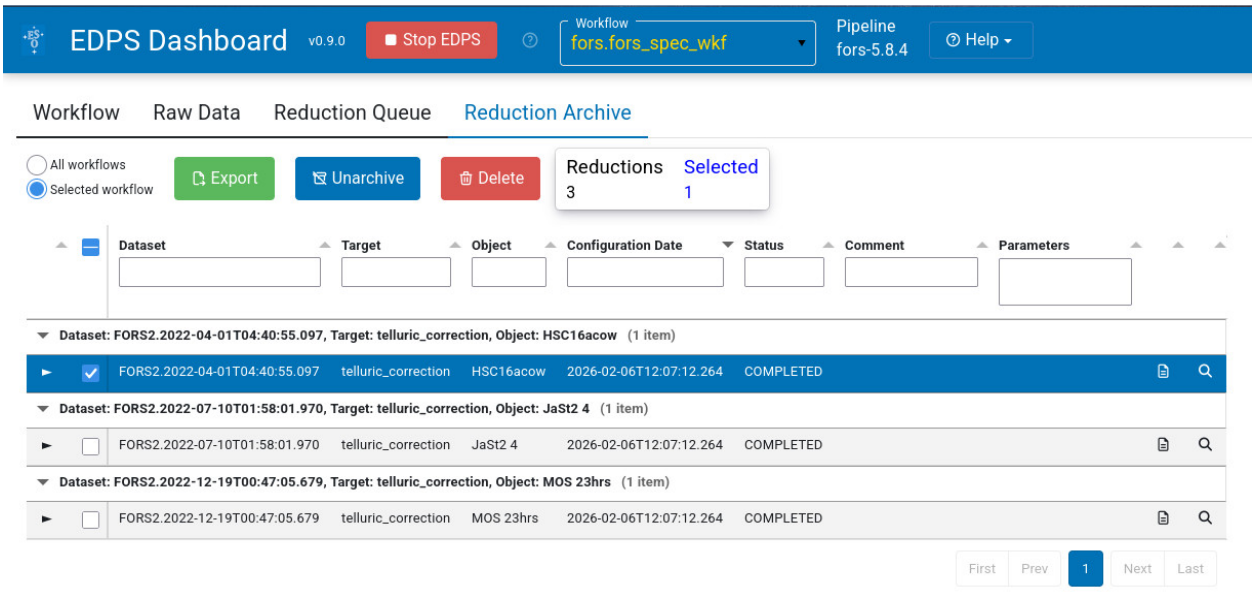

File Edit View History Bookmarks Tools Help
EDPS Dashboard
http://localhost:5006/edps-gui
European Southern Observatory
EDPS Dashboard
v0.7.0
Stop EDPS
Workflow
fors.fors_imaging_wkf
Pipeline
Unknown
Help
Workflow
Raw Data
Reduction Queue
Reduction Archive
All workflows
Selected workflow
Reduce
Archive
Delete
Reductions Selected
1 1
New Pending Running Completed Aborted Failed
1 0 0 0 0 0
Dataset Target Object Configuration Date Status Comment Parameters
Dataset: FORS2.2023-02-20T07:13:16.533, Target: combine_images, Object: SN2021ahpl (1 item)
FORS2.2023-02-20T07:13:16.533 combine_images SN2021ahpl 2025-09-09T13:54:43.636 NEW
First Prev 1 Next Last
Stop All Reductions
Real Time Execution Monitor

1.3.4 3. Save the final data (Optional).

All the products (final and intermediate) of all reduced configuration for all datasets are saved into the EDPS_data directory. In addition, EDPS offers the possibility to "archive" a desired reduction (e.g., the one that gives the best results) by "exporting" only the final reduction products into another location.

Select the desired reduced configurations and press the **Archive button**. This configuration disappears from the Reduction Queue and appear in the tab **Reduction Archive**.

The content of the "Reduction Archive", e.g. the list of archived reductions, can be inspected by pressing the **Reduction Archive** tab.



Each configuration and its jobs can be inspected, unarchived and eventually deleted. In order to copy the final products into a desired location, press the Export button. A window will appear allowing

- to select all or just the last configuration for the selected datasets
- to specify a directory where to save the files. A check is done

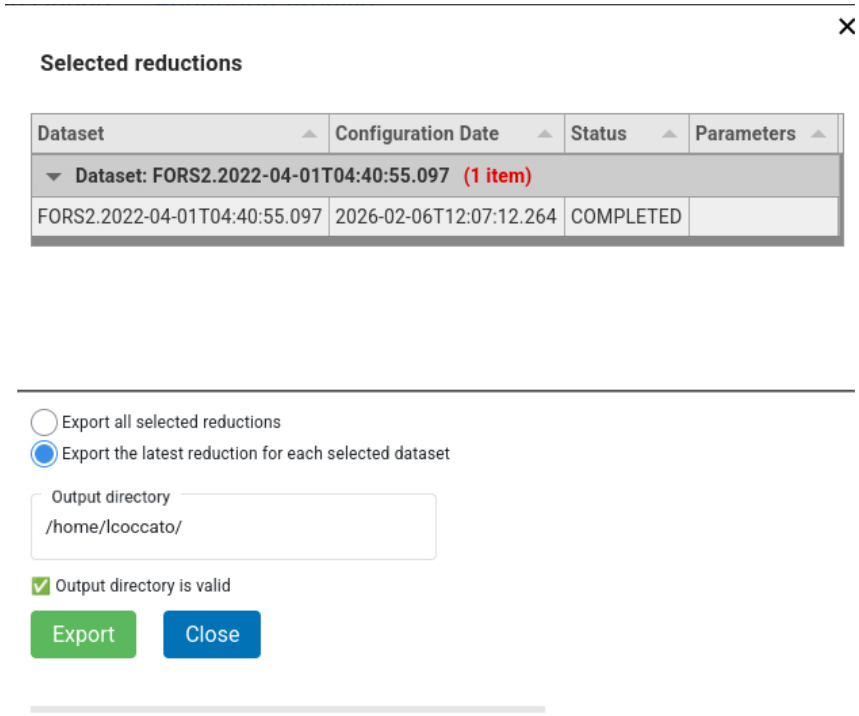
Press **Export** to save the files. The types of files to copy depend on each workflow. Typically, only most important files of the scientific tasks are saved. Other files can be found in the EDPS_data directory. The operation can take several minutes, depending on the sizes of the files. The default structure is `<dataset_name>/<reduction_time_stamp>/<category>.fits`.

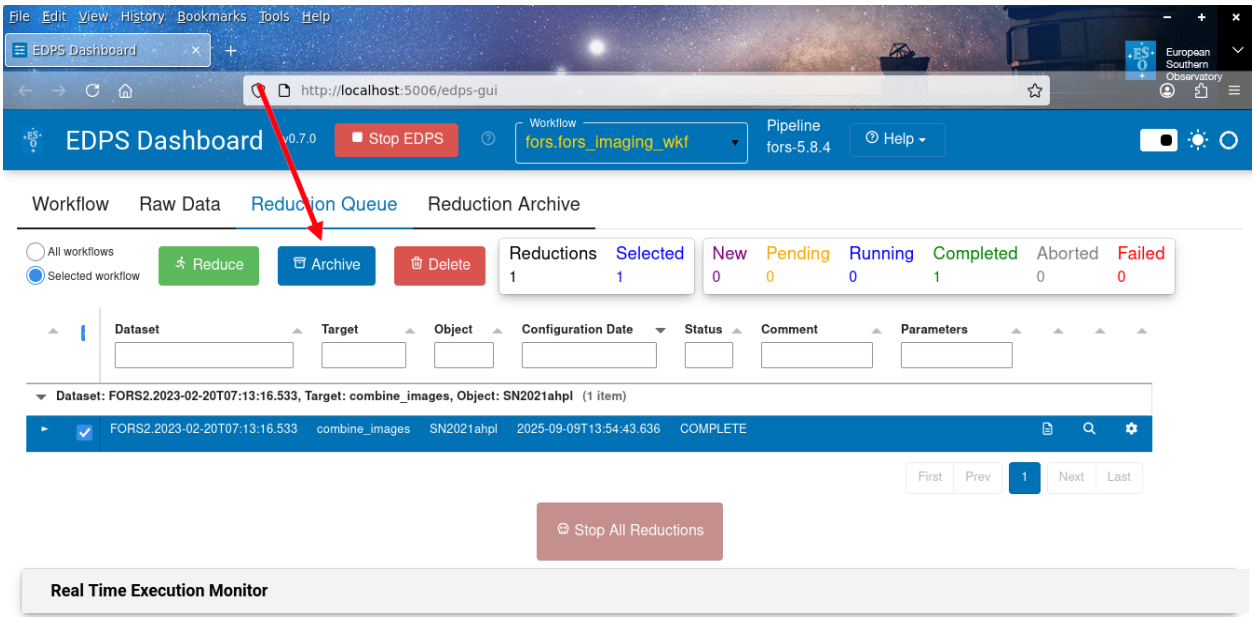
Go to EDPS quick-start guide *index*

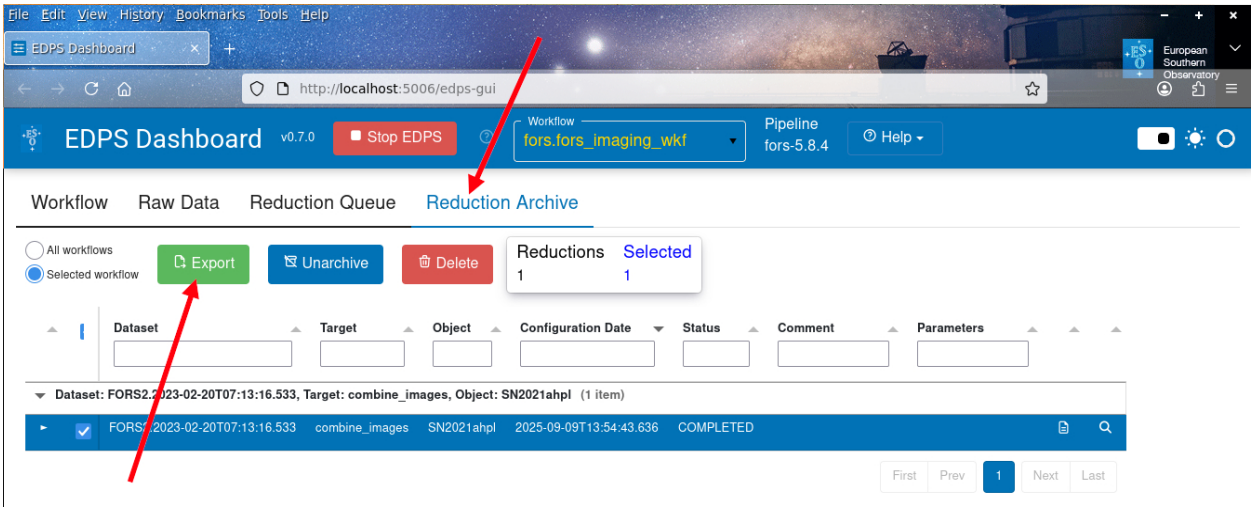
1.4 Frequently Asked Questions

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. To do so, proceed as follows:

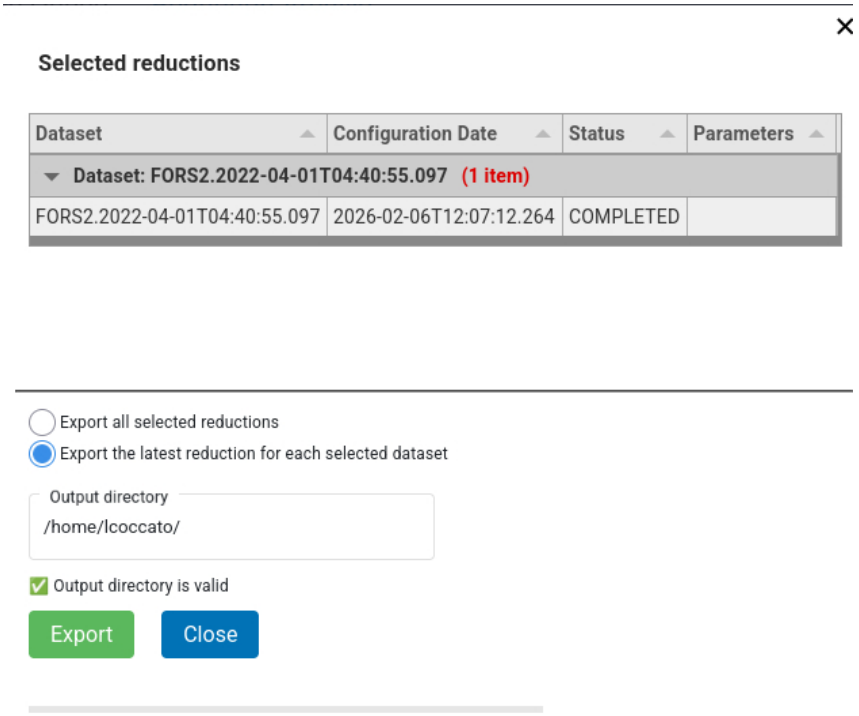
1. In the `Reduction Queue` tab, select the dataset and the dataset for which you want to export the final products. Click on the `Archive` button.
2. Go in the `Reduction Archive` tab,


Selected reductions
Dataset
Configuration Date
Status
Parameters
Dataset: FORS2.2022-04-01T04:40:55.097 (1 item)
FORS2.2022-04-01T04:40:55.097
2026-02-06T12:07:12.264
COMPLETED
Export all selected reductions
Export the latest reduction for each selected dataset
Output directory
/home/lcoccato/
Output directory is valid
Export
Close


File Edit View History Bookmarks Tools Help
EDPS Dashboard
http://localhost:5006/edps-gui
EDPS Dashboard
Stop EDPS
Workflow
fors.fors_imaging_wkf
Pipeline
fors-5.8.4
Help
Workflow
Raw Data
Reduction Queue
Reduction Archive
All workflows
Selected workflow
Reduce
Archive
Delete
Reductions
Selected
New
Pending
Running
Completed
Aborted
Failed
Dataset
Target
Object
Configuration Date
Status
Comment
Parameters
Dataset: FORS2.2023-02-20T07:13:16.533, Target: combine_images, Object: SN2021ahpl (1 item)
FORS2.2023-02-20T07:13:16.533
combine_images
SN2021ahpl
2025-09-09T13:54:43.636
COMPLETE
First
Prev
Next
Last
Stop All Reductions
Real Time Execution Monitor



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.



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>
```

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

Answer: Install the `datademo` package provided with the pipeline installation or download the "Demo Data" package from href=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 -0 https://eso.org/sci/software/apptainer/eso_download_demodata.sh
bash ./eso_download_demodata.sh
```

```
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>
```

Answer: For suggestions, questions, or feedback in general, please open a ticket with the EDPS Support team. This link should take you directly to a webpage for creating and EDPS feedback ticket, but incase 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]".

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 migth break some internal links. Recipes requiring external packages such as telluriccorr might not work (impacts on kmos, xshooter, fors, and molecfit pipelines)

Answer: This is a know bug that will be fixed in the next release. The cause is that the input data directory contains 2 files with the same properties and the same mjd-obs, but with different names (e.g. the same static calibration that

is present in 2 places). Since their mjd-obs is the same, EDPS sometimes adds one file, sometimes the other. When a different file is associated, the dataset is labelled as new.

Go to EDPS quick-start guide *index*

EDPS-GUI Version 0.9 - 20260212

CHAPTER

TWO

EDPS-GUI FULL TUTORIAL

A PDF tutorial is available here.

EDPS command line execution

CHAPTER

THREE

EDPS COMMAND LINE EXECUTION

It is possible to run `EDPS` also via command line. This option is useful, for example, when reducing data via scripts.

A tutorial is available here.

Instrument specific tutorials

CHAPTER

FOUR

EFOSC TUTORIAL

A PDF tutorial is available here.

CHAPTER

FIVE

ESPRESSO TUTORIAL

A PDF tutorial is available here.

CHAPTER

SIX

FORS2 SPECTROPOLARIMETRY TUTORIAL

A PDF tutorial is available here.

CHAPTER

SEVEN

KMOS TUTORIAL

A PDF tutorial is available here.

CHAPTER

EIGHT

MUSE TUTORIAL

A PDF tutorial is available here.

CHAPTER

NINE

SPHERE IFS TUTORIAL

9.1 Introduction

9.1.1 Scope

This document describes how to reduce SPHERE-IFS data with the EDPS dashboard (Graphic User Interface), the recommended infrastructure to reduce data from the ESO telescope, using the new ESO Data Processing System (EDPS). Details on the SPHERE-IFS 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 brief description of EDPS and its main concepts, see here.

For a detailed documentation on the SPHERE-IFS pipeline itself, consult the pipeline manual available here.

9.2 Reducing demo data

Follow this procedure to quickly reduce SPHERE-IFS demo data. We assume that the `EDPS` Graphic User Interface, the SPHERE-IFS pipeline and the demo data are installed in your system. For general instructions on how to install `EDPS` and the pipeline, please visit https://www.eso.org/sci/software/pipe_aem_main.html.

9.2.1 1. Setting the workflow

a. After installation and while still in the installation virtual environment (if not, activate it *again*), start the `EDPS` dashboard by typing:

```
edps-gui
```

The `EDPS` dashboard will start in a browser window.

b. Optionally, before starting EDPS, one can specify new settings by pressing Help –> Settings (see here)

c. On the browser window with the `EDPS` dashboard, press the button `Start EDPS`.

d. Choose the `SPHERE-IFS` pipeline from the list in the `workflows` field. The workflows offered in this selector depend on the installed pipelines.

The graphic workflow representation will appear as in Fig. 9.1.

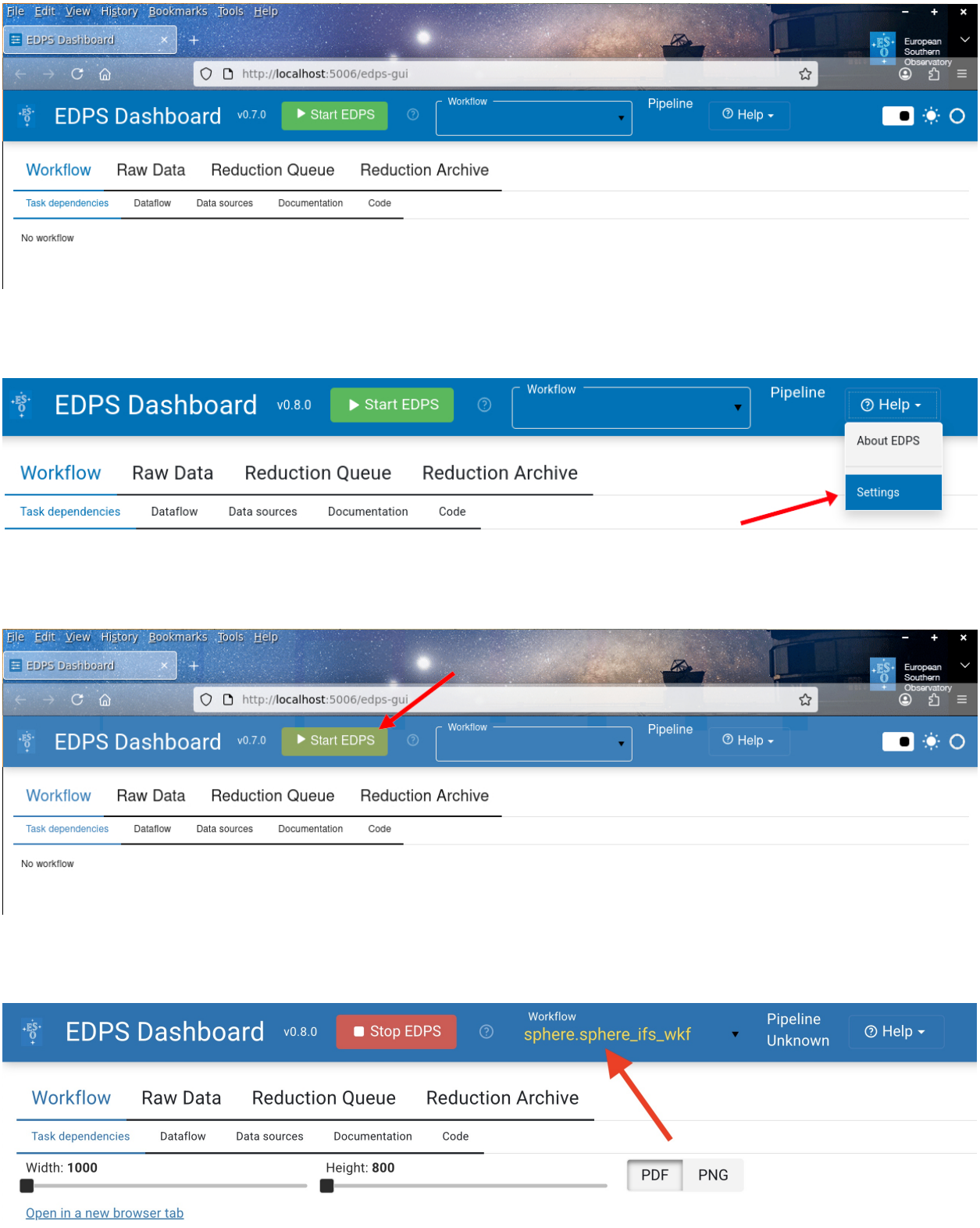
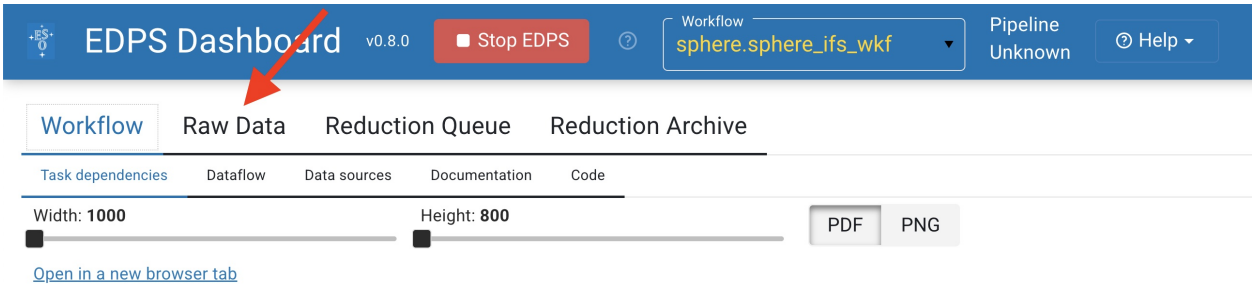


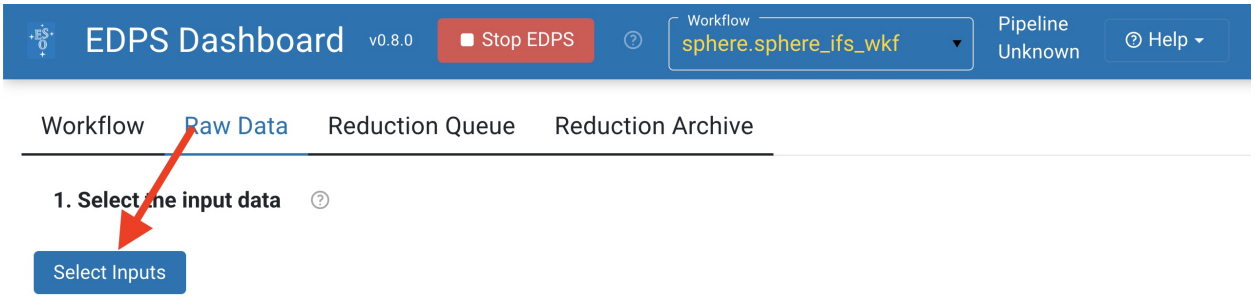
Fig. 9.1: The EDPS Dashboard (Graphic User Interface) with the SPHERE-IFS workflow loaded.

9.2.2 2. Selecting the input data

a. Press `Raw Data`.

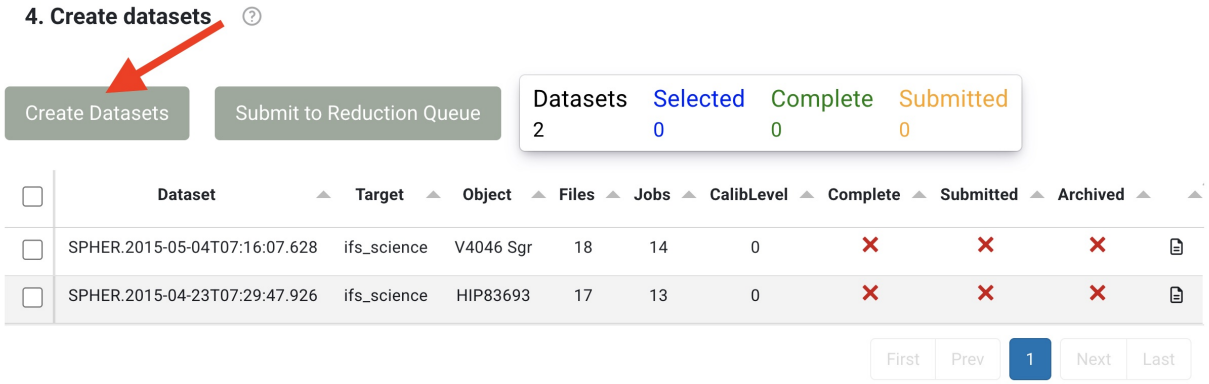


b. Press `Select Inputs`. A selection window will appear that allows to select data that are stored on a local disk.



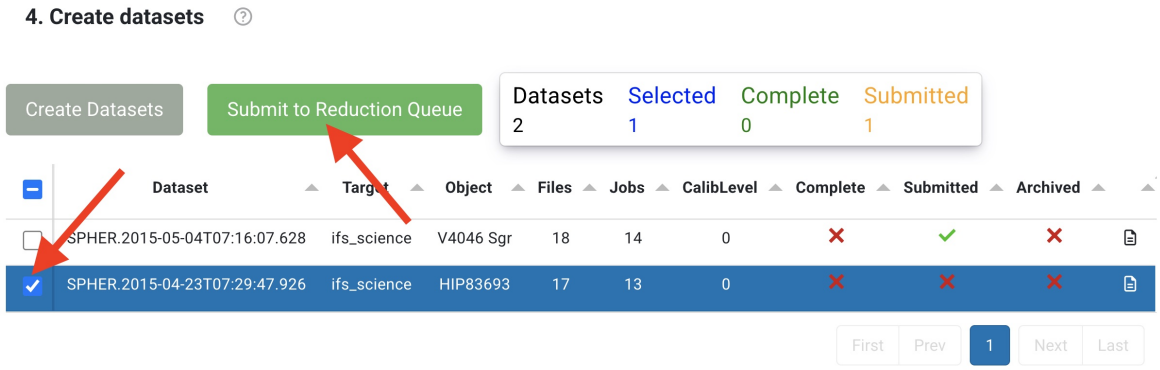
c. (Optional). Select the reduction target, configure the workflow parameter and specify the association preferences. Thise steps are optional. For more information see here.

d. Press `Create Datasets`. A list of datasets appears, one line for each set of science data.



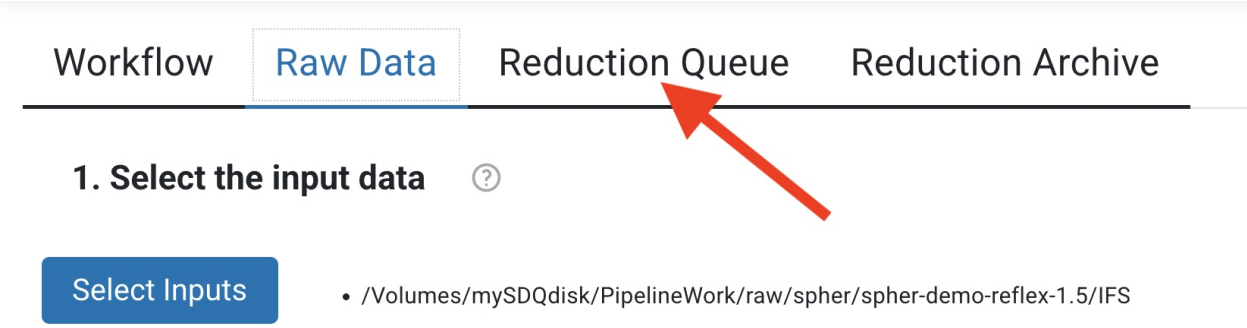
e. Choose the datasets that should be processed

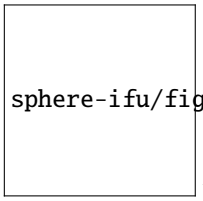
and send them to the data reduction queue by pressing `Submit to Reduction Queue`. Note that this action does not start the reduction automatically.



9.2.3 3. Start the reduction

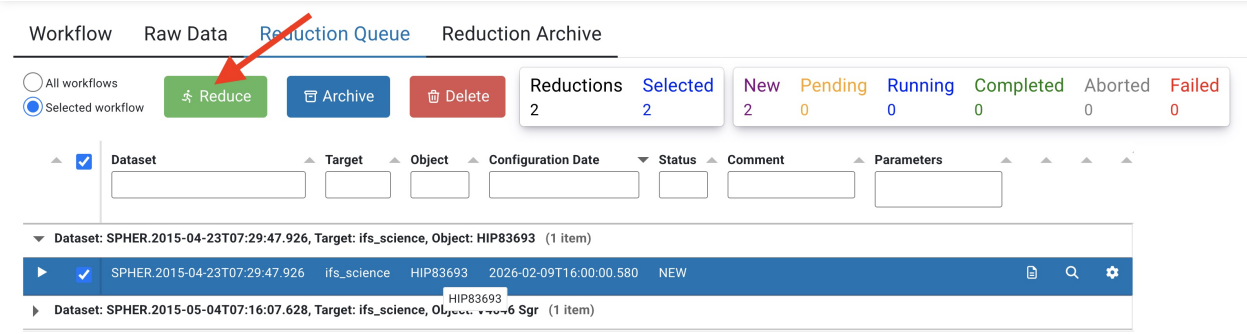
a. Press `Reduction Queue`.



b. (Optional). Configure the workflow and recipe parameters by pressing the wheel button 

to open the configuration editor.

c. Press the `Reduce` button. The selected data will now be processed with the default 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.

9.2.4 4. Final products

By default, EDPS saves all the recipe products for all the executions in the directory specified at the first execution (default: `$HOME/EDPS_data`). However, it is possible to save only the final products into a desired location. This can be achieved by exporting archived data reduction: press the `Export` button in the `Archived Data` tab (see here).

To archive a data reduction, press the button `Archive` in the `Reduction Queue` tab (see here).

The exported products are organized by DATASET (named as the first scientific exposure of the dataset), and `TIMESTAMP` (time of start of reduction).

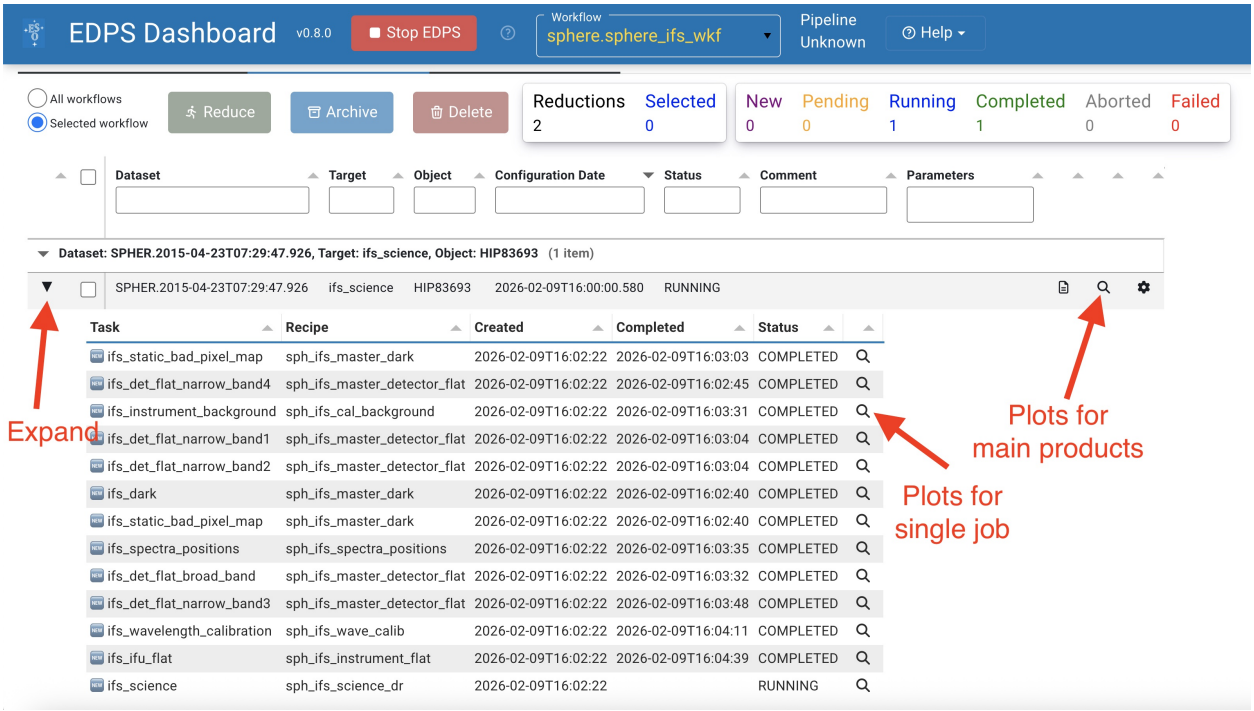
The final product saved in the specified directory is the wavelength-calibrated datacube, with name format `DATACUBE_` followed by the archive file name (corresponding to header keyword `ARCFILE`).

Other useful products can be found in the EDPS_data directory, where all intermediate products are saved.

9.2.5 5. 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),

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



Go to *top*

Go to SPHERE-IFS EDPS tutorial *index*

9.3 The SPHERE-IFS data reduction flow

The overall data flow of the SPHERE-IFS pipeline is displayed here.

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, inputs, 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 `EDPS` workflow is designed to execute the tasks that deliver the final reduced data cube for each dataset. Currently, the pipeline does not stack science data.

It is possible to set `EDPS` to perform the data reduction until a certain step of the reduction chain (e.g. to reduce only standard 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 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 *here* for more information.

The reduction steps are:

9.3.1 1. Subworkflow: Dark background

Recipes: `sph_ifs_cal_background`, `sph_ifs_master_dark`

This subworkflow generates the instrumental background and dark-current calibration products.

It consists of the tasks:

- **ifs_instrument_background**
- **ifs_sky_background**
- **ifs_dark**
- **ifs_static_bad_pixel_map**

which run the pipeline recipes `sph_ifs_cal_background` and `sph_ifs_master_dark`.

The first recipe measures the instrumental background and/or a sky background in counts per second per pixel.
The output is a 4-extension FITS file containing the background image, bad-pixel map, RMS map, and weight map.

The second recipe derives the detector dark current (`IFS_MASTER_DARK`) and a bad pixel map (`IFS_STATIC_BADPIXELMAP`; also contained in the 2nd extension of the master dark).

The master dark is obtained by combining the input raw frames using the selected collapse algorithm. Bad pixels are identified through the following steps:

1. Initial thresholding using `min_acceptable` and `max_acceptable`.
2. Smoothing of the master dark with a Gaussian kernel and subtraction of the smoothed frame to remove large-scale variations.
3. Sigma clipping of the residuals using the `sigma_clip` parameter.

The final (unsmoothed) master dark is written with extensions for the bad-pixel map, RMS, and the number of contributing raw pixels per output pixel.

Customization

Recipe parameters:

- `ifs.master_dark.coll_alg`: collapse algorithm (0 = Mean, 1 = Median, 2 = Clean Mean; default).
- `ifs.master_dark.min_acceptable`, `ifs.master_dark.max_acceptable`: pixel-value limits for thresholding (default 0 and 1000; reasonably adjustable within [-100:0] and [800:1000], respectively).
- `ifs.master_dark.sigma_clip`: sigma-clipping threshold (default 3; increase to 5 if you believe too many pixels are flagged).

2. Spectra positions

Recipe: `sph_ifs_spectra_positions`

The task **ifs_spectra_positions** runs the pipeline recipe to associate detector pixels with IFS lenslets and assigns an initial wavelength solution to each pixel.

Spectral traces are detected via thresholding and used to determine their centroid positions. These measured positions are compared to those predicted by a scaled and shifted lenslet model, allowing refinement of the model's scale and position. Optionally, a 2D distortion model is fitted and incorporated into the lenslet model to improve wavelength calibration.

The output is a pixel description table (PDT) written out as a FITS image with 6 extensions, corresponding to: wavelength, spectra region id, lenslet id, wavelength width, second derivative of wavelength and illumination fraction.

Customization

Recipe parameters:

- `ifs.master_dark.coll_alg`: collapse algorithm (0 = Mean, 1 = Median, 2 = Clean Mean; default).

9.3.2 3. Wavelength calibration

Recipe: `sph_ifs_wave_calib`

The task **ifs_wavelength_calibration** runs the pipeline recipe to perform the wavelength calibration by refining the pixel-to-wavelength associations (starting from the model produced by **ifs_spectra_positions**; see above), using observed wavelength calibration frames and an existing lenslet model.

1D spectra are then extracted for each spectral region, and the positions of known calibration lines are measured using a flux-weighted centroid within a configurable fitting window. These measured line positions are fitted with a polynomial that defines the wavelength solution for each spectrum, provided the solution is consistent with the expected dispersion; otherwise, the original model wavelengths are retained or the region is flagged as bad.

From the final wavelength solutions, the recipe computes per-spectrum quality-control metrics, including minimum, maximum, and central wavelengths, as well as the resolving power derived from the local wavelength gradient. The corrected PDT is written as the main product, together with updated QC keywords.

Customization

Recipe parameters:

- `ifs.wave_calib.coll_alg`: collapse algorithm (0 = Mean, 1 = Median, 2 = Clean Mean; default).

9.3.3 4. Subworkflow Detector Flat

Recipe: `sph_ifs_master_detector_flat`

This subworkflow executes the following series of tasks to generate detector flat-field calibrations using exposures obtained with narrow or broadband calibration lamps:

- **ifs_det_flat_narrow_band1**

- **ifs_det_flat_narrow_band3**
- **ifs_det_flat_narrow_band4**
- **ifs_det_flat_narrow_band2**
- **ifs_det_flat_broad_band**

It can produce several flat-field components, including a standard detector flat, a large-scale (smoothed) flat, and a preamplifier correction flat. In practice, the standard `IFS_MASTER_DFF_LONG` products are sufficient to correct pixel-to-pixel detector response variations.

Raw frames are dark-subtracted and corrected for preamplifier variations, after which the detector flat is derived by fitting, for each pixel, a linear relation between its signal and the mean illumination level of each exposure. The slope of this fit represents the relative pixel response. The main output is the detector flat-field image, with optional products including a non-linearity map and a smoothed large-scale flat.

Customization

Recipe parameters:

- `ifs.master_detector_flat.coll_alg`: collapse algorithm (0 = Mean, 1 = Median, 2 = Clean Mean; default).
- The fit between signal and illumination level can be performed using either a maximum-likelihood (`ifs.master_detector_flat.robust_fit` = TRUE; default) or a robust linear method (change parameter to FALSE), the latter being more resistant to outliers such as cosmic rays.

9.3.4 5. IFU Flat

Recipe: `sph_ifs_instrument_flat`

The task **ifs_ifu_flat** uses the pipeline recipe to generate flat-field calibrations. It can operate in two modes, depending on the inputs and selected parameters.

In detector flat (total flat) mode, raw calibration frames are combined to produce a flat field that includes the detector response. This product can be used by the `spectra-positions` and `wavelength-calibration` recipes (see above).

In IFU flat mode, the recipe uses the PDT and wavelength calibration to remove the detector response and isolate the IFU (lenslet) contribution. A wavelength-dependent "super-flat" is first constructed from master detector flats and used to flat-field the combined raw frames. Using the lenslet model from the wavelength calibration, spectra are extracted for all lenslets and collapsed along the wavelength direction to derive one flat-field value per lenslet. The main output is a table of IFU flat-field values used by subsequent recipes, along with a diagnostic interpolated image.

The outputs are:

- `IFS_INSTRUMENT_FLAT_FIELD` - The total instrument flat field, with 4 extensions: the flat values, the bad pixels, the rms error on the flat and a weightmap.
- `IFS_IFU_FLAT_FIELD` - The IFU flat field, with 4 extensions: the flat values, the bad pixels, the rms error on the flat, a weightmap, and a 1 table extension containing the lenslet flat values.
- `IFS_STATIC_BADPIXELMAP` - Optional output of all non-linear pixels.

Customization

Recipe parameters:

- `ifs.instrument_flat.coll_alg`: collapse algorithm (0 = Mean, 1 = Median, 2 = Clean Mean; default).

9.3.5 6. Distortion map

Recipe: `sph_ifs_distortion_map`

The task **ifs_distortion_map** runs the pipeline recipe to measure geometric distortion in the SPHERE IFS lenslet grid. Raw calibration frames are reduced similarly to science data, with optional background or dark subtraction and flat-fielding. The resulting monochromatic images are collapsed along the wavelength axis and point sources are detected using a user-defined threshold.

If no reference point pattern is provided, the recipe derives one from the detected sources and saves it as a product; otherwise, the supplied pattern is used directly. Distortion is computed by comparing detected and expected source positions, rejecting vectors larger than a user-defined limit, and fitting the remaining offsets with a 2D polynomial model. An optional self-consistency check applies the derived distortion to the data and outputs a residual distortion map, which can be inspected to assess the quality and accuracy of the calibration.

The outputs are:

- `IFS_POINT_PATTERN` - The point pattern, in case that an input point pattern was provided. This product may be used as reference input for future runs of this recipe.
- `IFS_DISTORTION_MAP` - The distortion map, with 8 extensions. The first 4 contain the distortion in the x direction, the badpixels, the rms on the distortion and a weightmap. The second set of 4 extensions contain the same information but for the y direction.

Customization

Recipe parameters:

- `ird.distortion_map.coll_alg`: collapse algorithm (0 = Mean, 1 = Median, 2 = Clean Mean; default).
- `ird.distortion_map.threshold`: threshold for source detection (default is 3.0).

9.3.6 7. Subworkflow Science Data Reduction

Recipe: `sph_ifs_science_dr`

This subworkflow executes the tasks:

- **ifs_science_flux**
- **ifs_coronagraph_center**
- **ifs_science**

to reduce IFS science observations, with optional background or dark subtraction and pre-amplifier stripe correction.

It also processes FLUX frames acquired during coronagraphic observations, where the telescope is offset to move the target away from the coronagraph, allowing to measure the stellar flux.

In addition, it processes CENTER frames to determine the stellar position behind the coronagraph. This is required to accurately define the center of rotation for pupil-stabilized observations.

Large-scale flat-field effects are removed using a wavelength-dependent "super flat" built from multi-colour lamp flats and the wavelength calibration, while small-scale, time-dependent pixel-to-pixel variations are corrected using a broad-band master flat.

The recipe supports automatic frame combination using "spectral deconvolution" (a method used to reduce speckle noise; see Mesa et al. 2015) and angular differential imaging (ADI), which can be enabled or disabled via flags.

Customization

Recipe parameters:

- `ifs.science_dr.coll_alg`: collapse algorithm (0 = Mean, 1 = Median, 2 = Clean Mean; default).

- `ifs.science_dr.use_adi`: enable use of ADI (0 = not applied, 1 = always applied, 2 = applied only if the total rotation is larger than `ifs.science_dr.min_adi_angle`).
- `ifs.science_dr.min_adi_angle`: minimum angle for automatic ADI switch (when previous parameter equals 2).
- `ifs.science_dr.spec_deconv`: enable use of spectral deconvolution (0 = not applied, 1 = always applied).

9.3.7 8. Subworkflow Standard Data Reduction

Recipe: `sph_ifs_science_dr`

This subworkflow executes the tasks:

- **ifs_standard_astrometry**
- **ifs_standard_flux**

At the moment, astrometry and flux standard observations are processed like science data.

Go to SPHERE-IFS EDPS tutorial *index*

9.4 Customizing the data reduction

9.4.1 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 (see here).

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.

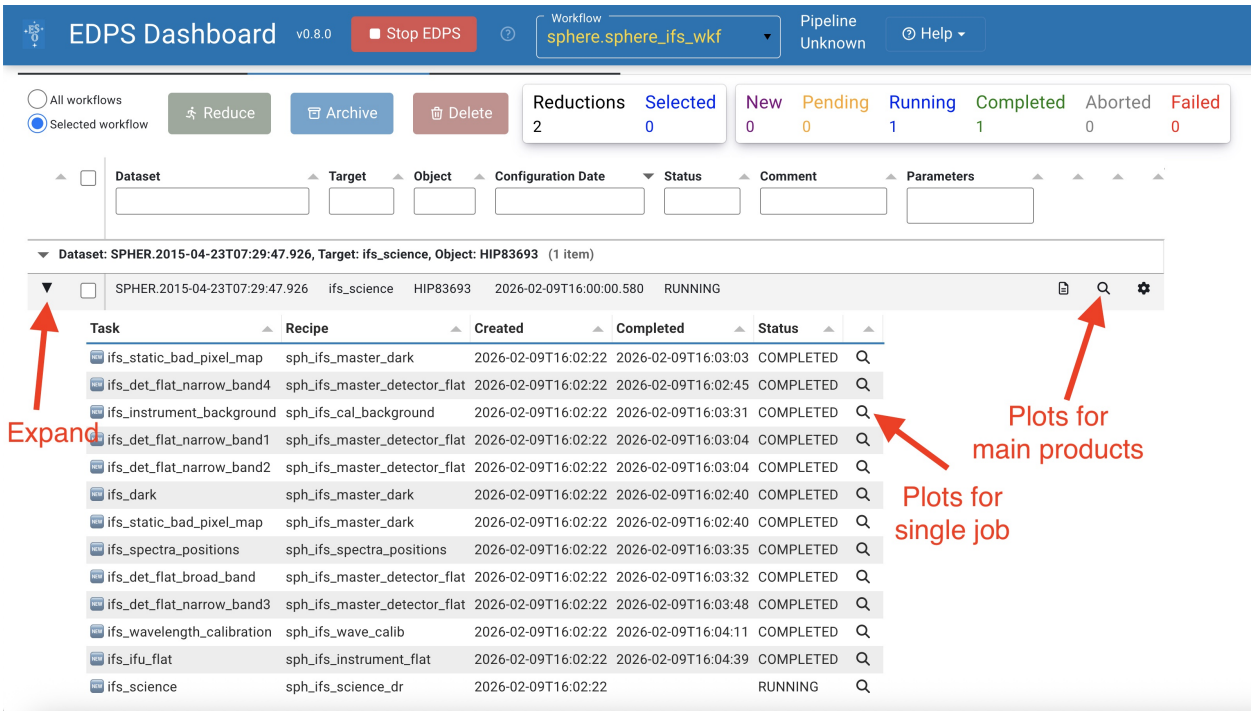
More information on the application properties file can be found here.

More explanations on the concept of "association levels" can be found here.

9.4.2 Quality reports

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),

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



9.4.3 Configuration of parameters

The data reduction of each dataset can be configured according to the scientific needs using an appropriate configuration editor. This editor allows to configure the data reduction for a given dataset by specifying workflow and recipe parameters.

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** 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.

sphere-ifu/figures/configure_dataset.jpg

To open the `Reduction configuration` 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 the figure below.

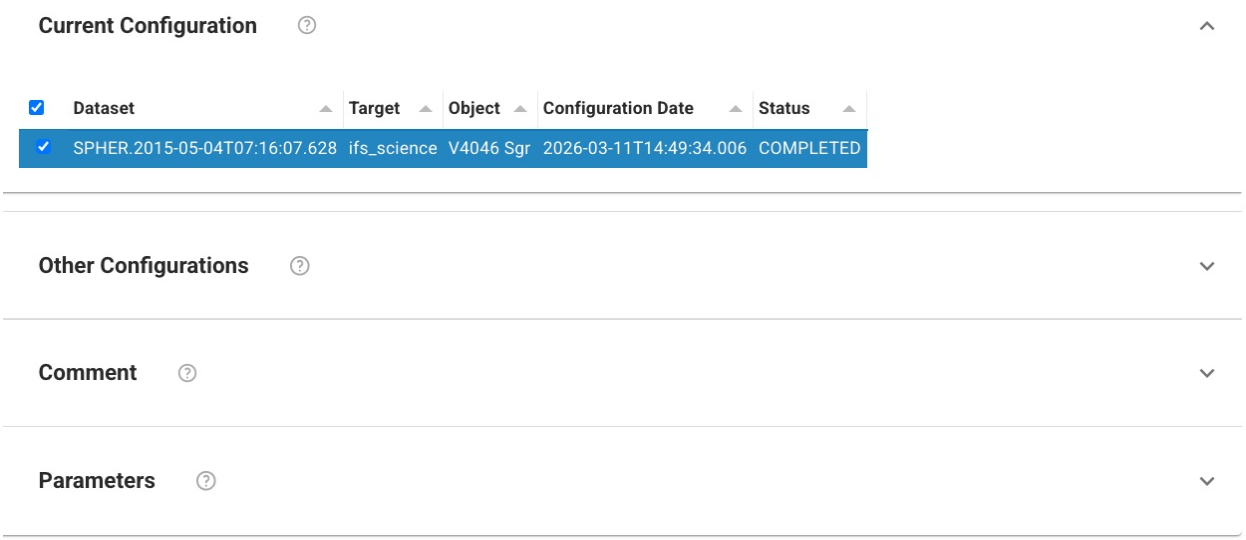
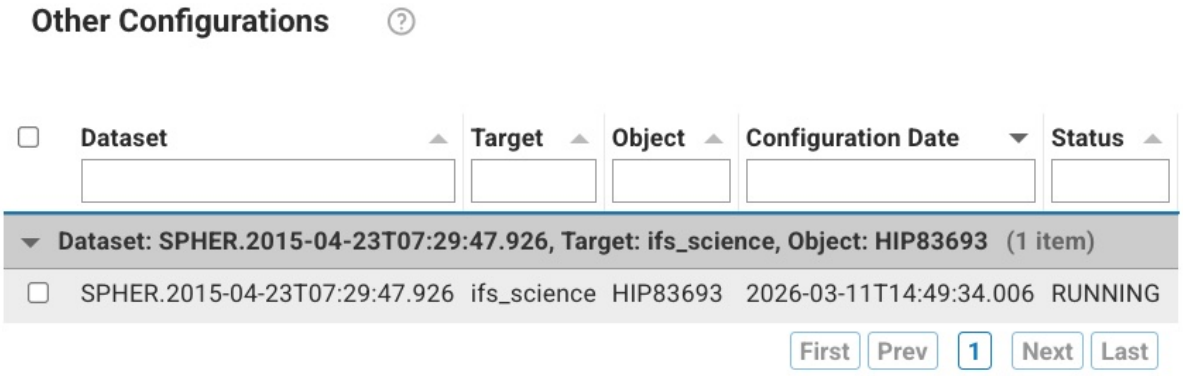


Fig. 9.2: The Reduction Configuration editor.

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

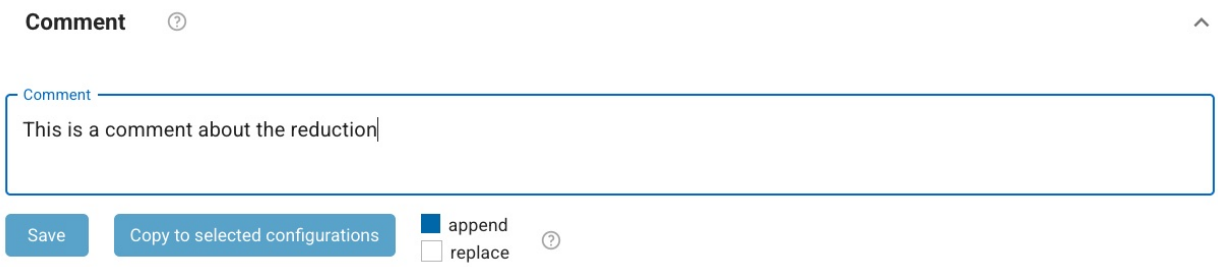
Current configuration It indicates the name of the selected configuration for a given dataset.

Other configurations It allows to specify other configurations, to which the changes shall be copied to.



Comment It allows to specify a comment to describe the configuration. It is possible to append or replace a comment.

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

This window is visible 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.

9.4.4 Configuration and troubleshooting

This section provides guidance on configuring the reduction, as well as diagnosing and resolving common issues encountered during the SPHERE-IFS data-reduction cascade.

Master darks

`sph_ifs_master_dark` produces the detector dark-current calibration (`IFS_MASTER_DARK`) and generates a static bad-pixel map (`IFS_STATIC_BADPIXELMAP`).

If you believe too many bad pixels are detected, increase `ifs.master_dark.sigma_clip` from the default value of 3 to 5 to make the clipping less aggressive.

Distortion map

The pipeline recipe `sph_ifs_distortion_map`, intended to measure the geometric distortion of IFS frames, has been found to be insufficiently robust. Calibration data acquired close in time can yield inconsistent distortion solutions, including variations in the derived central axes. When applied to science data, these discrepancies can introduce artificial distortions rather than correct them. For this reason, the recipe is not included in the workflow.

The dominant distortion component is a stable anamorphism of approximately 0.6%. This can be corrected in a simple and reproducible way by rescaling the detector coordinates, multiplying the X and Y axes by factors of 1.0059 and 1.0011, respectively.

Parameters

Parameter set

science_parameters

Workflow parameters

Parameter	Default value	Custom value
force_distortion_correction	FALSE	

Click on a parameter to view its description

Recipe parameters

Task

ifs_dark

Parameter	Default value	Custom value
ifs.master_dark.outfilename	master_dark.fits	
ifs.master_dark.coll_alg	2	
ifs.master_dark.badpixfilename	static_badpixels_unused.fits	
ifs.master_dark.clean_mean.reject_high	0	
ifs.master_dark.clean_mean.reject_low	0	
ifs.master_dark.sigma_clip	3.0	
ifs.master_dark.smoothing	5.0	

Wavelength calibration

The consortium identified significant systematic wavelength shifts when using the pipeline product `IFS_SPECPOS` generated by recipe `sph_ifs_spectra_positions`. To mitigate this issue, an ad hoc correction was developed and implemented in IDL. The corresponding implementation within the pipeline, however, does not yet reproduce the same level of performance and remains under evaluation.

To facilitate independent verification of the wavelength solution, the recipe `sph_ifs_wave_calib` produces a wavelength-calibrated data cube of the arc lamp exposure. This allows users to directly assess the quality and stability of the wavelength calibration.

Star center

The recipe `sph_ifs_star_center`, which processes OBJECT,CENTER observations obtained with the IFS, has been found to introduce significant systematic errors in the derived stellar position. As a consequence, its output is not used by the science reduction recipe.

Spectral deconvolution

When `sph_ifs_science_dr` reduces science data with `ifs.science_dr.spec_deconv` = TRUE, the following recipe parameters are ignored and internally set to fixed values:

- `ifs.science_dr.order` = 5
- `ifs.science_dr.user_cent` = FALSE
- `ifs.science_dr.cx` = 0 (unused since `user_cent` = FALSE)
- `ifs.science_dr.cy` = 0 (unused since `user_cent` = FALSE)
- `ifs.science_dr.reflambda` = 1.0
- `ifs.science_dr.fwhm` = 2.0

These choices reflect the experience of the High Contrast Data Center.

Science data reduction

Following the consortium recommendation, `ifs.science_dr.spec_deconv` is set to FALSE in recipe `sph_ifs_science_dr`, as the current pipeline implementation relies on an outdated algorithm. Because the science case is not known a priori, `ifs.science_dr.use_adi` is set to 0.

At the Data Center, crosstalk and bad-pixel corrections were historically performed using IDL scripts. These algorithms are now implemented in the pipeline and can be enabled via `ifs.science_dr.badpixco.apply` and `ifs.science_dr.xtalkco.apply` (both FALSE by default).

The large-scale crosstalk correction may improve point-source detection but can negatively impact extended sources (e.g., discs). It must therefore be enabled separately by setting `ifs.science_dr.xtalkco.largescale.apply` to TRUE.

Go to *top*

Go to SPHERE-IFS EDPS tutorial *index*

9.5 List of workflow tasks

This is the list of all the tasks and associated recipes in the SPHERE-IFS 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
ifs_coronagraph	sph_ifs_science_	optional	Reduces center frames to determine stellar position behind coronagraph.
ifs_dark	sph_ifs_master_(	yes	Creates master dark (`IFS_MASTER_DARK`) and bad-pixel map.
ifs_det_flat_bro	sph_ifs_master_(	yes	Generates broadband detector flat used for small-scale pixel-to-pixel correction.
ifs_det_flat_nar	sph_ifs_master_(	yes	Produces narrow-band detector flat component.
ifs_det_flat_nar	sph_ifs_master_(	yes	Produces narrow-band detector flat component.
ifs_det_flat_nar	sph_ifs_master_(	yes	Produces narrow-band detector flat component.
ifs_det_flat_nar	sph_ifs_master_(	yes	Produces narrow-band detector flat component.
ifs_distortion_m	sph_ifs_distortio	no	Measures geometric distortion via point-pattern comparison; outputs distortion map and optional residual map.
ifs_ifu_flat	sph_ifs_instrume	yes	Produces total instrument flat and/or IFU (lenslet) flat.
ifs_instrument_l	sph_ifs_cal_back	yes	Measures instrumental background in counts/s per pixel.
ifs_science	sph_ifs_science_	yes	Main science reduction task; applies flats, background/dark subtraction, optional ADI and spectral deconvolution.
ifs_science_flux	sph_ifs_science_	optional	Processes FLUX frames (star offset from coronagraph) to measure stellar flux.
ifs_sky_backgrc	sph_ifs_cal_back	yes	Processes sky background frames for background calibration.
ifs_spectra_posi	sph_ifs_spectra_	yes	Associates pixels with lenslets and assigns initial wavelength solution; produces Pixel Description Table (PDT).
ifs_standard_ast	sph_ifs_science_	optional	Reduces astrometric standard observations (currently treated as science data).
ifs_standard_flu	sph_ifs_science_	optional	Reduces flux standard observations (currently treated as science data).
ifs_static_bad_p	sph_ifs_master_(	optional	Produces static bad-pixel map (`IFS_STATIC_BADPIXELMAP`) from master dark.
ifs_wavelength_	sph_ifs_wave_ca	yes	Refines pixel-to-wavelength solution using calibration lines; updates PDT and QC metrics.

Notes

- For IFS science and standard tasks, backgrounds are associated in the following priority order: **sky background**, then **internal background**, then **master dark**.
- The detector-flat tasks are split by calibration lamp / wavelength. `ifs_det_flat_narrow_band4` is only relevant for the **YJH** setting.
- Distortion is associated as **mandatory or optional depending on configuration**, through the `distortion_mandatory` / `distortion_optional` conditions in the workflow.
- `ifs_science_flux` and `ifs_coronagraph_center` are also associated to `ifs_science` mainly for **calselector support**, not because their products are always required by the science recipe.

CHAPTER

TEN

SPHERE IRDIS TUTORIAL

10.1 Introduction

10.1.1 Scope

This document describes how to reduce SPHERE-IRDIS data with the EDPS dashboard (Graphic User Interface), the recommended infrastructure to reduce data from the ESO telescope, using the new ESO Data Processing System (EDPS). Details on the SPHERE-IRDIS 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 brief description of EDPS and its main concepts, see here.

For a detailed documentation on the SPHERE-IRDIS pipeline itself, consult the pipeline manual available here.

10.2 Reducing demo data

Follow this procedure to quickly reduce SPHERE-IRDIS demo data. We assume that the `EDPS` Graphic User Interface, the SPHERE-IRDIS pipeline and the demo data are installed in your system. For general instructions on how to install `EDPS` and the pipeline, please visit https://www.eso.org/sci/software/pipe_aem_main.html.

10.2.1 1. Setting the workflow

a. After installation and while still in the installation virtual environment (if not, activate it *again*), start the `EDPS` dashboard by typing:

```
edps-gui
```

The `EDPS` dashboard will start in a browser window.

b. Optionally, before starting EDPS, one can specify new settings by pressing Help –> Settings (see here)

c. On the browser window with the `EDPS` dashboard, press the button `Start EDPS`.

d. Choose the `SPHERE-IRDIS` pipeline from the list in the `workflows` field. The workflows offered in this selector depend on the installed pipelines.

The graphic workflow representation will appear as in Fig. 10.1.

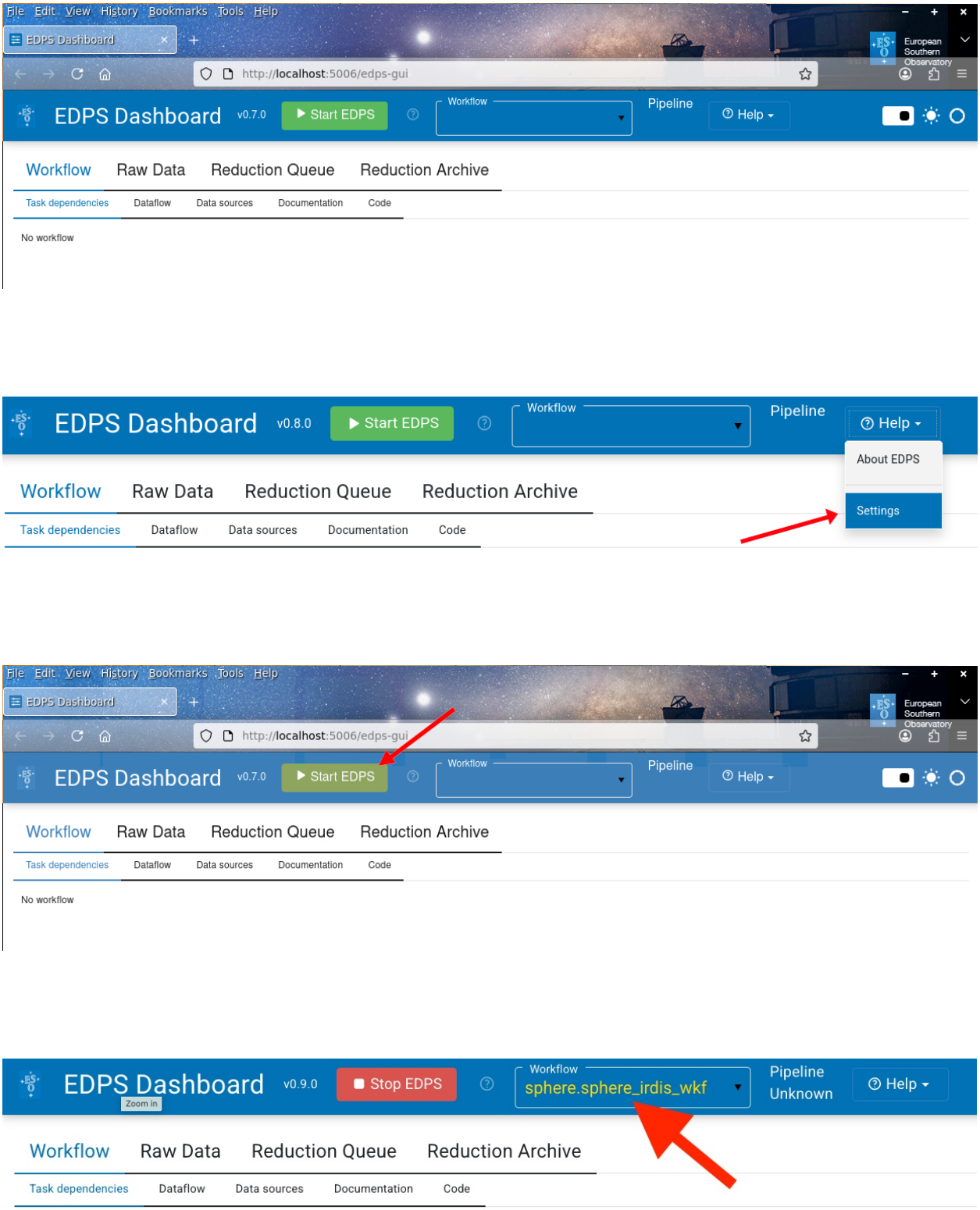
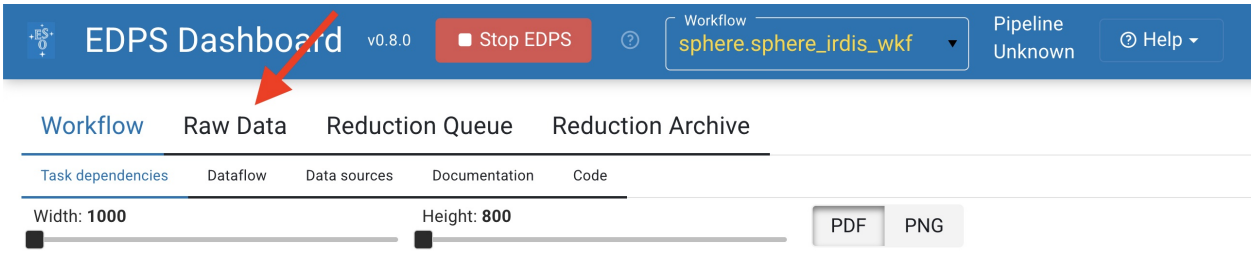


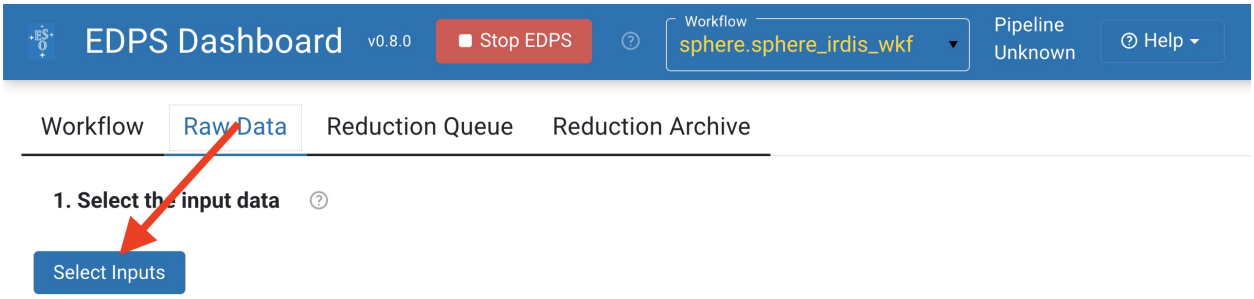
Fig. 10.1: The EDPS Dashboard (Graphic User Interface) with the SPHERE-IRDIS workflow loaded.

10.2.2 2. Selecting the input data

a. Press `Raw Data`.

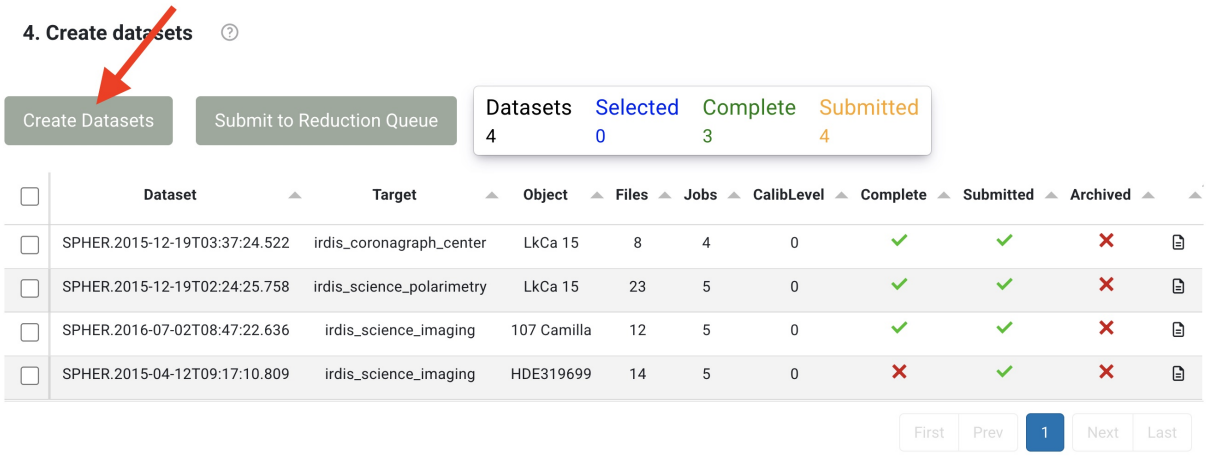


b. Press `Select Inputs`. A selection window will appear that allows to select data that are stored on a local disk.



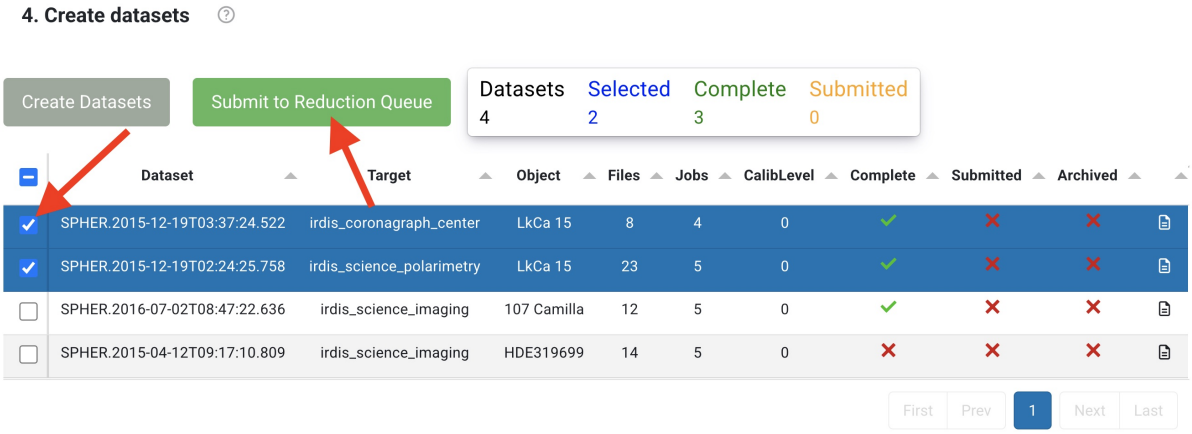
c. (Optional). Select the reduction target, configure the workflow parameter and specify the association preferences. These steps are optional. For more information see here.

d. Press `Create Datasets`. A list of datasets appears, one line for each set of science data.



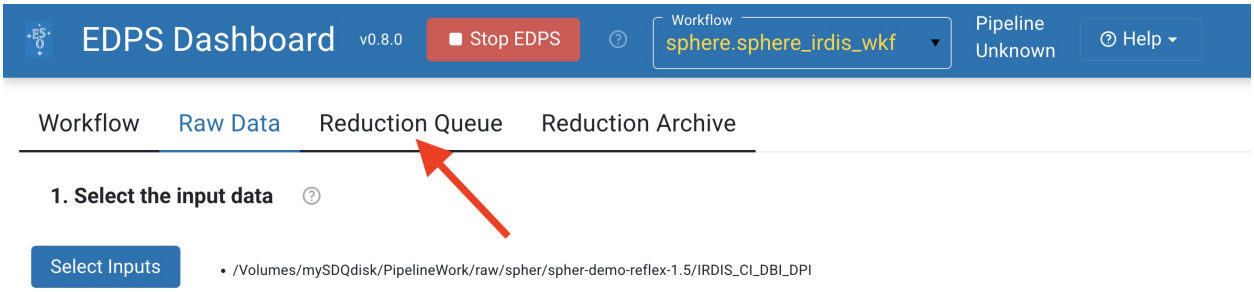
e. Choose the datasets that should be processed

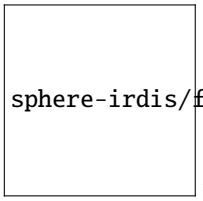
and send them to the data reduction queue by pressing `Submit to Reduction Queue`. Note that this action does not start the reduction automatically.



10.2.3 3. Start the reduction

a. Press `Reduction Queue`.



b. (Optional). Configure the workflow and recipe parameters by pressing the wheel button 

to open the configuration editor.

c. Press the `Reduce` button. The selected data will now be processed with the default parameters.

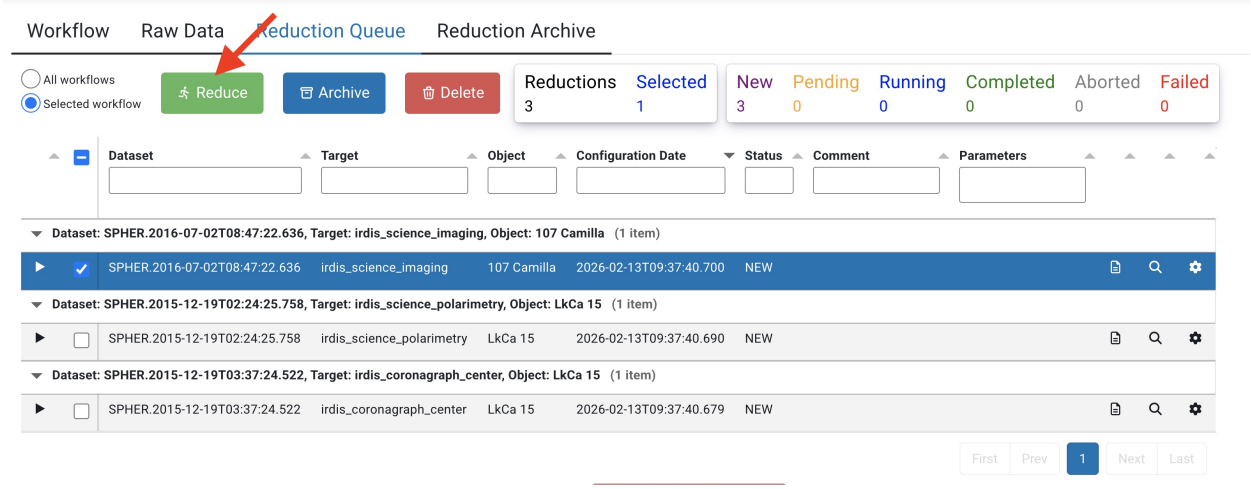
Congratulations! You reduced your first data with the `EDPS` dashboard! All the reduced data are saved in the EDPS_data directory specified when executing `edps-gui` for the first time.

10.2.4 4. Final products

By default, EDPS saves all the recipe products for all the executions in the directory specified at the first execution (default: `$HOME/EDPS_data`). However, it is possible to save only the final products into a desired location. This can be achieved by exporting archived data reduction: press the `Export` button in the `Archived Data` tab (see here).

To archive a data reduction, press the button `Archive` in the `Reduction Queue` tab (see here).

The exported products are organized by DATASET (named as the first scientific exposure of the dataset), and `TIMESTAMP` (time of start of reduction).



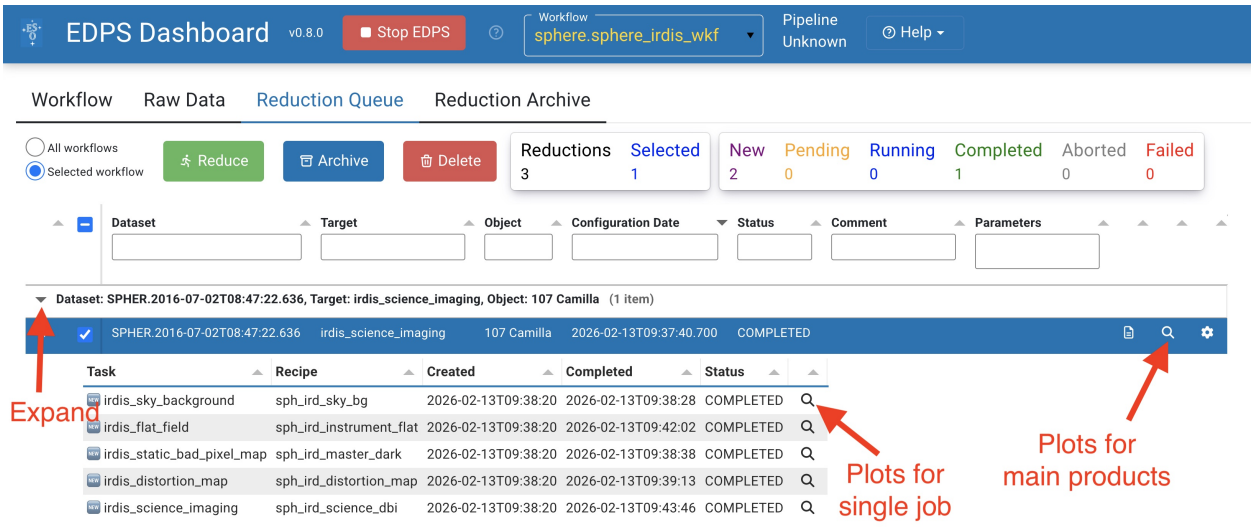
The final product saved in the specified directory is the wavelength-calibrated datacube, with name format `IRDIS_` followed by the observing mode (e.g., `IMAGE_`, `POLARIMETRY_`, `STAR_CENTER_`), and archive file name (corresponding to header keyword `ARCFILE`).

Other useful products can be found in the EDPS_data directory, where all intermediate products are saved.

10.2.5 5. 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),

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



Go to *top*

Go to SPHERE-IRDIS EDPS tutorial *index*

10.3 The SPHERE-IRDIS data reduction flow

The overall data flow of the SPHERE-IRDIS pipeline is displayed here.

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, inputs, 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 `EDPS` workflow is designed to execute the tasks that deliver the final reduced data product for each dataset.

It is possible to set `EDPS` to perform the data reduction until a certain step of the reduction chain (e.g. to reduce only standard 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 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 *here* for more information.

The reduction steps are:

10.3.1 1. Subworkflow: Dark background

Recipes: `sph_ird_master_dark`, `sph_ird_ins_bg`, `sph_ird_sky_bg`

This subworkflow generates the dark-current frame, instrumental background and sky background calibration products.

It consists of the tasks:

- **irdis_dark_imaging**
- **irdis_static_bad_pixel_map**
- **irdis_instrument_background**
- **irdis_sky_background**

First, `sph_ird_master_dark` creates the master dark calibration frame by combining raw dark exposures using the selected collapse algorithm (typically clean mean). Bad pixels are identified after combination. The master dark is written as the main product, and a separate hot-pixel map is also generated.

Next, `sph_ird_ins_bg` generates the instrument background calibration frame using only raw background exposures. The input frames are combined with the selected collapse algorithm to produce the final background product. Unlike the master dark recipe, no bad-pixel map is created.

Finally, `sph_ird_sky_bg` generates the sky background calibration frame by combining raw background exposures. Unlike the master dark recipe, it does not produce a bad-pixel map.

The resulting `MASTER_DARK`, `INS_BG`, and `SKY_BG` products are intended solely for subtraction from raw data and are not flat-fielded or corrected for other dark or background frames. They contain 4 extensions: the image, badpixels, the weightmap (how many frames contribute to each pixel), and the rms map.

Customization

Recipe parameters:

- `ird.sky_bg.coll_alg`: collapse algorithm (0 = Mean, 1 = Median, 2 = Clean Mean; default).

10.3.2 2. Master Flat

Recipe: `sph_ird_master_detector_flat`

The task `irdis_flat_fiel` generates the IRDIS instrument flat field using narrow- or broadband lamp exposures and varying flux levels. The resulting flat must be applied only to matching data. For optimal dark subtraction, raw dark frames with matching DITs should be provided; each flat is corrected using the closest matching dark. If raw darks are unavailable, background products are used in the following priority order: `INS_BG_FIT` → `INS_BG` → `MASTER_DARK`. The recipe can run without darks, but this is discouraged.

After dark correction, illuminated regions are identified using thresholding and detector windows are treated separately. For each pixel, a linear fit between its signal and the exposure mean level is performed; the slope defines the flat-field value (typically close to unity). The fit can be maximum-likelihood or robust (better against outliers). The main output is the flat-field image. Optional products include a non-linearity map and bad-pixel information. The master flat's bad-pixel extension contains all bad pixels at this stage (including input static bad pixels), while the optional static bad-pixel output includes only those newly identified by this recipe.

Customization

Recipe parameters:

- `ird.instrument_flat.coll_alg`: collapse algorithm (0 = Mean, 1 = Median, 2 = Clean Mean; default).
- `ird.instrument_flat.robust_fits` is FALSE by default. Changing to TRUE can provide better results against outliers (e.g., cosmic rays).

10.3.3 3. Distortion map

Recipe: `sph_ird_distortion_map`

The task **irdis_distortion_map** runs the pipeline recipe to derive the IRDIS geometric distortion map from calibration frames reduced as standard science data (field-stabilized, no dithering), optionally including dark subtraction and flat-fielding. After combining the frames, point sources are detected using a user-defined threshold. If no reference point pattern is provided, the recipe creates one; otherwise, it compares detected positions with the expected pattern to measure distortion. The optical axis is estimated from the most central point, the patterns are aligned, outliers beyond a user-defined maximum distortion are rejected, and 2D polynomial fits are computed to model the X and Y distortion across the detector.

The distortion map is saved in a FITS file with a total of 16 extensions. The first 4 extensions contain values, badpixels, rms and weightmap for the distortion in the x direction and the next 4 extensions the same information for the distortion in the y direction. The first 8 extension contain the information for the left FOV the next 8 extension the information for the right FOV. Optional QC products include images of the input point patterns, corrected detector and FoV images, and residual distortion maps (typical residuals < 0.1 pixels for good solutions). The estimated optical axis position and full polynomial coefficients are recorded in header keywords for monitoring and verification.

Customization

Recipe parameters:

- `ird.distortion_map.coll_alg`: collapse algorithm (0 = Mean, 1 = Median, 2 = Clean Mean; default).
- `ird.distortion_map.threshold`: threshold for source detection (default is 3.0).

10.3.4 4. Subworkflow Standard Imaging

In this subworkflow, flux and astrometric standard frames are processed using the tasks **irdis_standard_flux_imaging** and **irdis_standard_astrometry_imaging**.

These tasks execute the recipes `sph_ird_science_imaging` (for classical imaging, CI) and `sph_ird_science_dbi` (for dual-band imaging, DBI), respectively.

4a. Standard Flux Imaging

Recipe: `sph_ird_science_imaging`

The task **irdis_standard_flux_imaging** uses the pipeline recipe to produce reduced flux standard frames for zero point calibration. Raw frames are dark-subtracted (mandatory), optionally flat-fielded, and a combined bad-pixel map is generated. Left and right subframes are extracted using the IRDIS instrument model. Frames are combined using weighted mean, mean (with bad-pixel rejection), or median. The final product is a multi-extension FITS file (8 extensions: image, bad-pixel map, N map, RMS for left and right FoVs).

Customization

Recipe parameters:

- `ird.science_imaging.coll_alg`: collapse algorithm (0 = Mean, 1 = Median, 2 = Clean Mean; default).

4b. Standard Astrometry Imaging

Recipe: `sph_ird_science_dbi`

The task **irdis_standard_astrometry_imaging** uses the pipeline recipe to produce reduced astrometry standard frames. Raw frames are dark-subtracted (mandatory), optionally flat-fielded, and a combined bad-pixel map is generated. Left and right subframes are extracted using the IRDIS instrument model. Frames are combined using weighted mean, mean (with bad-pixel rejection), or median. The final product is a multi-extension FITS file (8 extensions: image, bad-pixel map, N map, RMS for left and right FoVs).

Customization

Recipe parameters:

- `ird.science_dbi.coll_alg`: collapse algorithm (0 = Mean, 1 = Median, 2 = Clean Mean; default).

10.3.5 5. Subworkflow On-Sky Science Calibrations

Recipes: `sph_ird_science_dpi`, `sph_ird_science_dbi`, `sph_ird_star_center`

This subworkflow processes on-sky calibrations for imaging data (IRDIS-IMG).

5a. Imaging Flux Standard

The task **irdis_science_flux_imaging** uses `sph_ird_science_dbi` to reduce flux frames acquired for coronagraphic observations, where the telescope is offset to observe the star outside the coronagraph.

These reductions are not used in the science processing (no flux calibration is applied to the science data).

5b. Polarimetry Flux Standard

The task **irdis_science_flux_polarimetry** uses `sph_ird_science_dpi` to reduce flux frames acquired for coronagraphic observations, where the telescope is offset to observe the star outside the coronagraph.

These reductions are not used in the science processing (no flux calibration is applied to the science data).

5c. Coronagraph Center

The task **irdis_coronagraph_center** executes the pipeline recipe `sph_ird_star_center` to determine the stellar center position behind the coronagraph and produce a table of frame centers.

The illuminated left and right detector regions are analysed separately using a source-detection algorithm with a user-defined sigma threshold. For each waffle image, the output table records the exposure start time, the derived center position, and the IRDIS DMS position converted from microns to pixels.

Customization

Recipe parameters:

- `ird.star_center.coll_alg`: collapse algorithm (0 = Mean, 1 = Median, 2 = Clean Mean; default).
- `ird.star_center.sigma`: the sigma threshold for source detection (default = 10.0).

10.3.6 6. Subworkflow Wavelength Calibration

Recipe: `sph_ird_wave_calib`

In spectroscopy mode (IRDIS-LSS) the task **irdis_wavecal_spectroscopy** performs the wavelength calibration with the pipeline recipe `sph_ird_wave_calib`. Raw frames are combined, dark-subtracted, and flat-fielded, with bad pixels flagged. The combined image is sliced along the wavelength direction, and calibration lines are detected within a window of $\pm$`ird.wave_calib.line_tolerance` around their expected positions.

Measured line positions are matched to known wavelengths and fitted with a polynomial, which defines the wavelength solution and interpolates values between calibration lines. The updated solution is written to the pixel description table (PDT) as the final product.

The output is a FITS file containing six image extensions, each corresponding to a column of the PDT: wavelength, spectrum ID, slit ID, wavelength width (or uncertainty), second derivative, and illumination fraction. Additional extensions include the combined image, a bad-pixel map, and an RMS map.

Customization

Recipe parameters:

- `ird.wave_calib.coll_alg`: collapse algorithm (0 = Mean, 1 = Median, 2 = Clean Mean; default).

7. Science Imaging

Recipe: `sph_ird_science_dbi`

The task **irdis_science_imaging** uses the recipe to produce reduced science frames for IRDIS observations in dual-band imaging (DBI) mode, supporting dithering as well as optional angular differential imaging (ADI), and spectral differential imaging (SDI) processing. Raw frames are dark-subtracted (mandatory), optionally flat-fielded, and combined with static bad pixels from dark and flat frames. Left and right subframes are extracted using the IRDIS instrument model.

The output is an 8-extension FITS file containing image, bad-pixel map, N map, and RMS for both left and right FoVs. A dark/background frame is required, with priority: `SKY_BG_FIT` $\rightarrow$ `SKY_BG` $\rightarrow$ `INS_BG_FIT` $\rightarrow$ `INS_BG` $\rightarrow$ `MASTER_DARK`; matching DIT/readout (and ideally filter) is the user's responsibility. Optional QC includes Strehl ratio estimation when calibration tables are available.

Customization

Recipe parameters:

- `ird.science_dbi.coll_alg`: collapse algorithm (0 = Mean, 1 = Median, 2 = Clean Mean; default).
- `ifs.science_dbi.use_adi`: enable use of ADI (0 = not applied; Default , 1 = applied).
- `ifs.science_dbi.use_sdi`: enable use of SDI (0 = not applied; Default , 1 = applied).

8. Science Polarimetry

Recipe: `sph_ird_science_dpi`

The task **irdis_science_polarimetry** uses the recipe to produce reduced science frames for IRDIS observations in DPI mode. The processing is essentially identical to the DBI science recipe, but the final products are saved as polarization (P) and intensity (I) images instead of left and right FoVs (see the IRDIS DBI recipe for processing details).

The main output is a FITS file with eight extensions: the first four contain the polarization (P) image, bad-pixel map, RMS map, and weight map, while the last four contain the same products for the intensity (I) image. When ADI is enabled, only four extensions are written, containing the P image products.

Recipe parameters:

- `ird.science_dpi.coll_alg`: collapse algorithm (0 = Mean, 1 = Median, 2 = Clean Mean; default).
- `ifs.science_dpi.use_adi`: enable use of ADI (0 = not applied; Default , 1 = applied).
- `ifs.science_dpi.use_sdi`: enable use of SDI (0 = not applied; Default , 1 = applied).

9. Science Spectroscopy

Recipe: `sph_ird_science_spectroscopy`

The task **irdis_science_spectroscopy** uses the recipe to produce reduced science frames for spectroscopy mode. The processing consists of dark subtraction and flat-fielding followed by frame combination using a user-selected method. If an atmospheric calibration is provided, it is subtracted from the combined result.

The output is a FITS file containing the reduced 2D spectrum for the left FoV as a full-detector image. The file includes four extensions: the science image, bad-pixel map, N-combination map (number of contributing frames per pixel), and an RMS map.

Recipe parameters:

- `ird.science_spectroscopy.coll_alg`: collapse algorithm (0 = Mean, 1 = Median, 2 = Clean Mean; default).

Go to SPHERE-IRDIS EDPS tutorial *index*

10.4 Customizing the data reduction

10.4.1 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 (see here).

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.

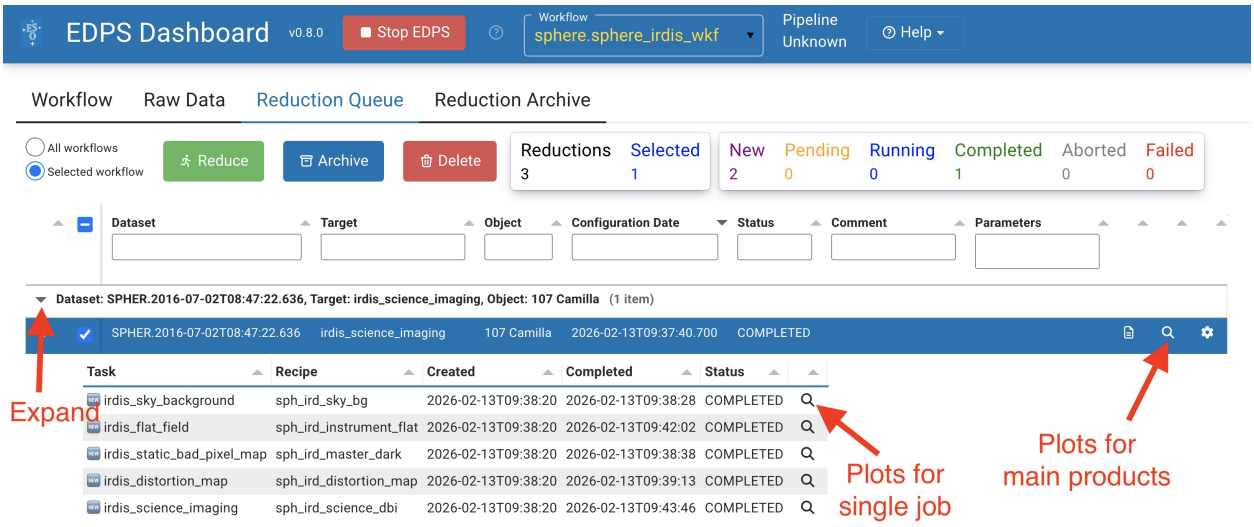
More information on the application properties file can be found here.

More explanations on the concept of "association levels" can be found here.

10.4.2 Quality reports

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),

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



10.4.3 Configuration of parameters

The data reduction of each dataset can be configured according to the scientific needs using an appropriate configuration editor. This editor allows to configure the data reduction for a given dataset by specifying workflow and recipe parameters.

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** 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.

sphere-irdis/figures/configure_dataset.jpg

To open the `Reduction configuration` 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 the figure below.

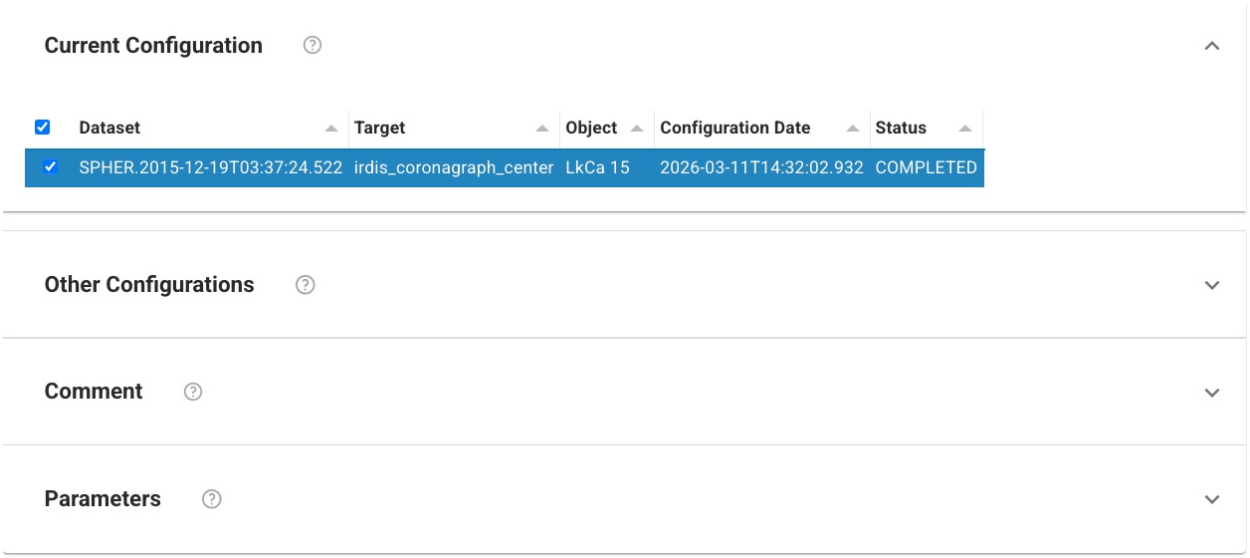
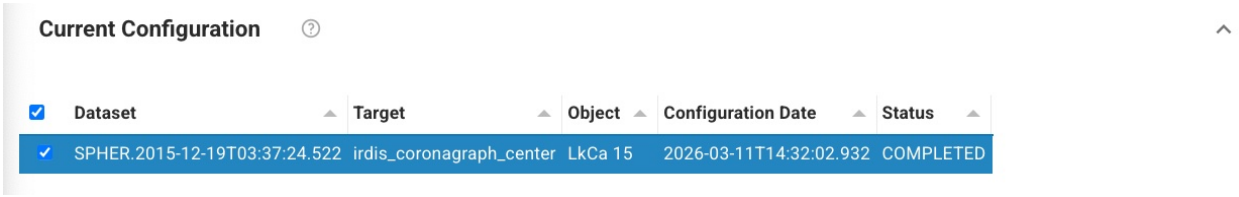


Fig. 10.2: The Reduction Configuration editor.

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.



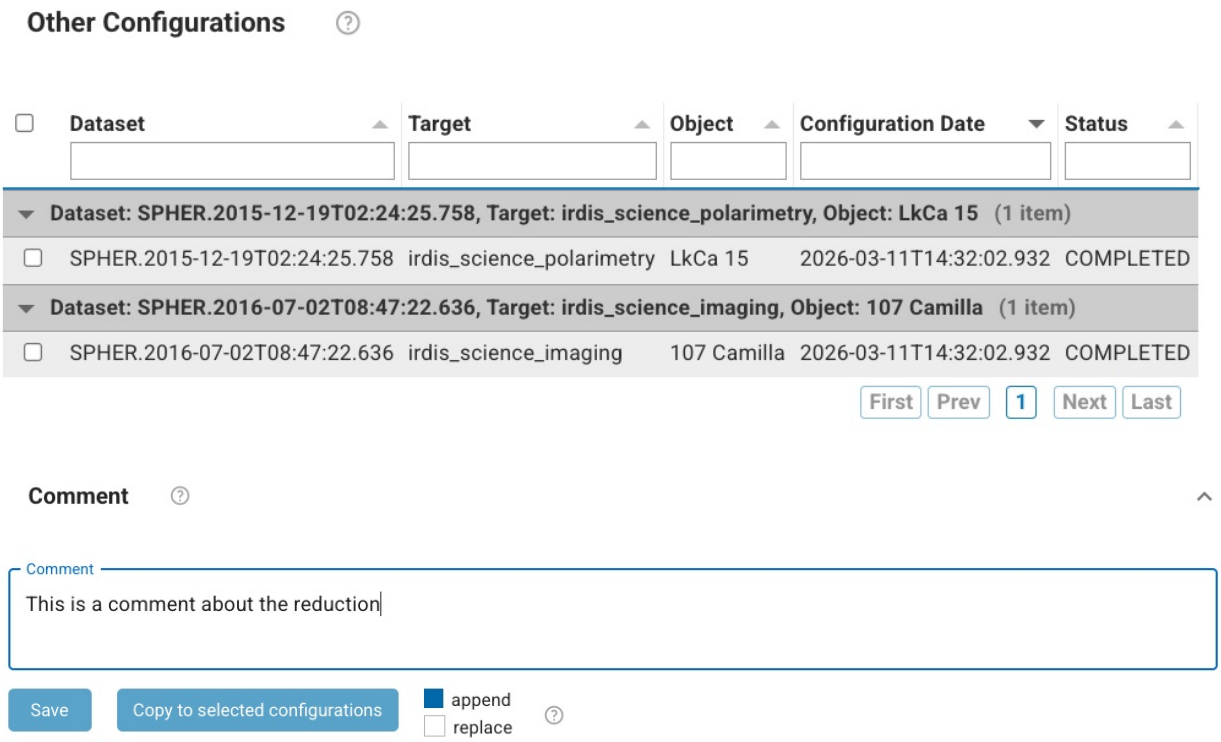
Other configurations It allows to specify other configurations, to which the changes shall be copied to.

Comment It allows to specify a comment to describe the configuration. It is possible to append or replace a comment. 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

This window is visible 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.

10.4.4 Configuration and troubleshooting

This section provides guidance on configuring the reduction, as well as diagnosing and resolving common issues encountered during the SPHERE-IRDIS data-reduction cascade.

Dark regions in the background

Fig. 10.3 shows an example of dark regions appearing in the products of the imaging workflow. These features are caused by astrophysical sources present in the SKY frames used for background subtraction.

If SKY frames are not provided to the pipeline, dark frames are used instead for the background correction, which removes this pattern.

Fine tuning the instrument flat

The recipe `sph_ird_instrument_flat` is executed using modified bad-pixel rejection thresholds. Instead of the default values of 0.1 and 10 for `ird.instrument_flat.badpix_lowtolerance` and `ird.instrument_flat.badpix_uptolerance`, respectively, values of 0.75 and 1.25 are adopted.

For K-band data, the High Contrast Data Center recommends increasing `ird.instrument_flat.badpix_uptolerance` to 1.5.

Parameters

Parameter set

science_parameters

Workflow parameters

Parameter	Default value	Custom value
force_distortion_correction	FALSE	

Click on a parameter to view its description

Recipe parameters

Task

irdis_coronagraph_center

Parameter	Default value	Custom value
ird.star_center.outfilename	star_center.fits	
ird.star_center.coll_alg	2	
ird.star_center.clean_mean.reject_high	1	
ird.star_center.clean_mean.reject_low	1	
ird.star_center.sigma	10.0	
ird.star_center.use_waffle	TRUE	
ird.star_center.qc	FALSE	

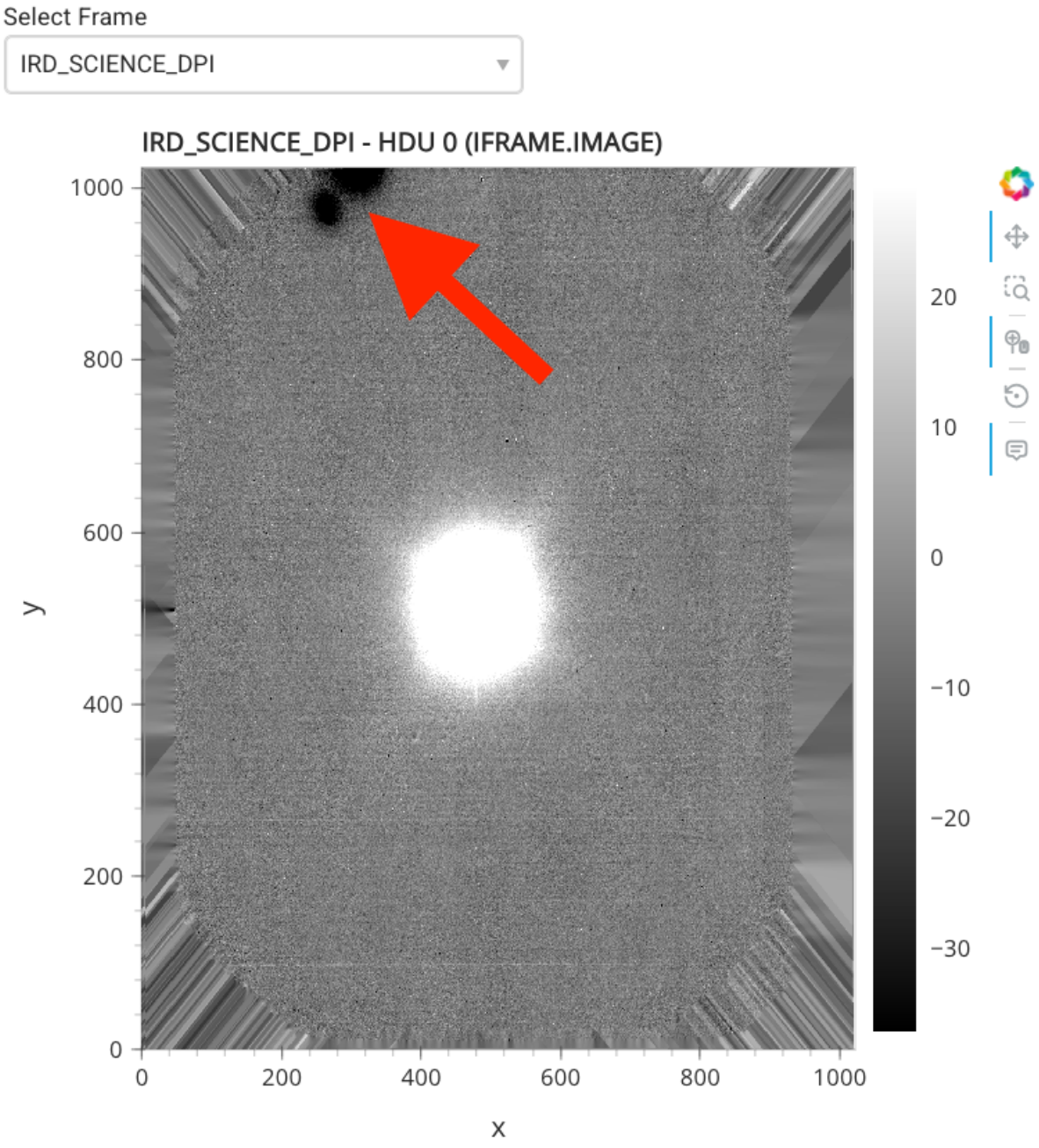


Fig. 10.3: Dark regions in the final image product for `SPHER.2015-12-19T02:24:25.758_tpl` (DPI).

Distortion map are not used

The pipeline recipe `sph_ird_distortion_map` determines the geometric distortion of IRDIS data using internal calibration observations obtained with a pinhole mask. The recipe can also measure the relative offset between the LEFT and RIGHT detector images when the workflow parameter `force_distortion_correction` is enabled.

In practice, distortion solutions derived from internal calibration data have been found to be insufficiently robust for routine processing. Calibration sequences acquired close in time may produce noticeably different distortion solutions, including variations in the derived geometric mapping. When applied to science or astrometric observations, these solutions generally do not improve the calibration beyond what is achieved using a simple anamorphism correction alone, and in some cases may even degrade the data.

For this reason, processing of internal distortion calibrations is disabled. A more robust approach is to correct only the dominant distortion component, which is a stable anamorphism of approximately 0.6%. This can be applied in a simple and reproducible manner by rescaling the detector coordinates, multiplying the X and Y axes by factors of 1.0059 and 1.0011, respectively.

Waffle spots cannot be detected

In task **irdis_coronagraph_center**, the recipe `sph_ird_star_center` may require parameter adjustments depending on the observing setup.

For K-band data, the detection of waffle spots can be challenging due to the high thermal background.
Increasing `ird.star_center.sigma` to 100 may improve the detection. Alternatively, the background can be removed manually prior to running the recipe.

The default parameters are optimized for detecting waffle spots in coronagraphic observations.
For data acquired without a coronagraph, the following parameters should be modified:

- `ird.star_center.use_waffle` = `FALSE`,
- `ird.star_center.sigma` = `500` (the stellar PSF is typically very bright in this configuration).

Angular Differential Imaging and Spectral Differential Imaging

The recipe `sph_ird_science_dbi` runs with both ADI and SDI disabled by default.
If SDI processing is enabled, the Data Centre (DC) recommends setting `ird.science_dbi.minr` and `ird.science_dbi.maxr` to 20 and 70, respectively, instead of the default values of 4 and 40.

These values are calibrated for a wavelength of = 1.6 μm and should be scaled linearly with wavelength for other observing bands.

Getting additional data products

The recipe `sph_ird_science_dbi` is used to process both Classical Imaging (CI) nd Dual-Band Imaging (DBI) data. The workflow is configured to save stacked products for each detector arm (`ird.science_dbi.save_addprod` = `TRUE`) while intermediate products are not retained by default.

To generate and preserve all intermediate products that may be useful for further analysis, set the recipe parameter `ird.science_dbi.save_interprod` = `TRUE`.

Go to *top*

Go to SPHERE-IRDIS EDPS tutorial *index*

10.5 List of workflow tasks

This is the list of all the tasks and associated recipes in the SPHERE-IRDIS 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
irdis_coronagraph_c	sph_ird_star_cent	yes (coronagraphic data)	Determines stellar position behind coronagraph using waffle images; outputs center table for registration.
irdis_dark_imaging	sph_ird_master_d	yes	Creates `MASTER_DARK` by combining dark frames and produces hot/static bad-pixel map.
irdis_distortion_map	sph_ird_distortion	no	Derives geometric distortion solution via point-pattern matching; outputs polynomial distortion model.
irdis_flat_field	sph_ird_master_d	yes	Generates detector master flat; includes linearity fit and updated bad-pixel map.
irdis_instrument_ba	sph_ird_ins_bg	yes	Produces instrumental background (`INS_BG`) used for subtraction when darks are unavailable.
irdis_science_flux_i	sph_ird_science_	optional	Reduces off-axis flux frames for coronagraphic observations; not applied to final science calibration.
irdis_science_flux_p	sph_ird_science_	optional	Flux reduction equivalent for DPI mode; output not used in science calibration.
irdis_science_imagi	sph_ird_science_	yes (DBI/CI modes)	Main DBI science reduction with dark/background subtraction, flat-fielding, optional ADI/SDI.
irdis_science_polari	sph_ird_science_	yes (DPI mode)	Produces polarization (P) and intensity (I) products.
irdis_science_spectr	sph_ird_science_	yes (LSS mode)	Reduces spectroscopy data via dark subtraction, flat-fielding, and frame combination.
irdis_sky_backgrour	sph_ird_sky_bg	optional	Creates sky background (`SKY_BG`) calibration used for science background subtraction.
irdis_standard_astro	sph_ird_science_	optional	Processes astrometric standard observations for calibration monitoring.
irdis_standard_flux_	sph_ird_science_	optional	Reduces photometric standards for zero-point calibration.
irdis_static_bad_pix	sph_ird_master_d	yes	Static bad pixels identified during master dark creation and propagated to later reductions.
irdis_wavecal_spect	sph_ird_wave_ca	yes (LSS only)	Computes wavelength solution and updates Pixel Description Table (PDT).

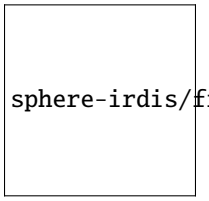
Notes

- For science tasks, backgrounds are associated in the following priority order: **sky background**, then **internal background**, then **master dark**.
- In DPI mode, the coronagraph-center association is constrained so that only suitable center frames taken **before** the science can be used.

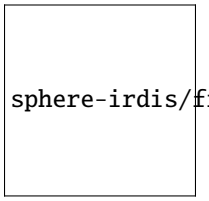
10.6 Frequently Asked Questions

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. To do so, proceed as follows:

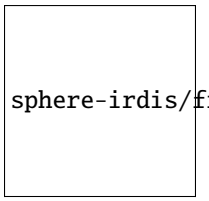
1. In the `Reduction Queue` tab, select the dataset and the dataset for which you want to export the final products. Click on the `Archive` button.



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.



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>
```

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

Answer: Install the `datademo` package provided with the pipeline installation or download the "Demo Data" package from href=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
```

```
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>
```

Answer: For suggestions, questions, or feedback in general, please open a ticket with the EDPS Support team. This link should take you directly to a webpage for creating and EDPS feedback ticket, but incase 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]".

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 migth break some internal links. Recipes requiring external packages such as telluriccorr might not work (impacts on kmos, xshooter, fors, and molecfit pipelines)

Answer: This is a know bug that will be fixed in the next release. The cause is that the input data directory contains 2 files with the same properties and the same mjd-obs, but with different names (e.g. the same static calibration that is present in 2 places). Since their mjd-obs is the same, EDPS sometimes adds one file, sometimes the other. When a different file is associated, the dataset is labelled as new.

Go to SPHERE-IRDIS tutorial *index*

CHAPTER

ELEVEN

SPHERE ZIMPOL TUTORIAL

A PDF tutorial is available here.

CHAPTER

TWELVE

UVES TUTORIAL

A PDF tutorial is available here.

CHAPTER

THIRTEEN

XSHOOTER TUTORIAL

13.1 Introduction

13.1.1 Scope

This document describes how to reduce XSHOOTER data with the EDPS dashboard (Graphic User Interface), the recommended infrastructure to reduce data from the ESO telescope, using the new ESO Data Processing System (EDPS). Details on the XSHOOTER 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 brief description of EDPS and its main concepts, see here.

For a detailed documentation on the XSHOOTER pipeline itself, consult the pipeline manual available here.

13.2 Reducing demo data

Follow this procedure to quickly reduce XSHOOTER demo data. We assume that the `EDPS` Graphic User Interface, the XSHOOTER pipeline and the demo data are installed in your system. For general instructions on how to install `EDPS` and the pipeline, please visit https://www.eso.org/sci/software/pipe_aem_main.html.

13.2.1 1. Setting the workflow

a. After installation and while still in the installation virtual environment (if not, activate it *again*), start the `EDPS` dashboard by typing:

```
edps-gui
```

The `EDPS` dashboard will start in a browser window.

b. Optionally, before starting EDPS, one can specify new settings by pressing Help –> Settings (see here)

c. On the browser window with the `EDPS` dashboard, press the button `Start EDPS`.

c. Choose the `XSHOOTER` pipeline from the list in the `workflows` field. The workflows offered in this selector depend on the installed pipelines.

The graphic workflow representation will appear as in `fig-wkf`.

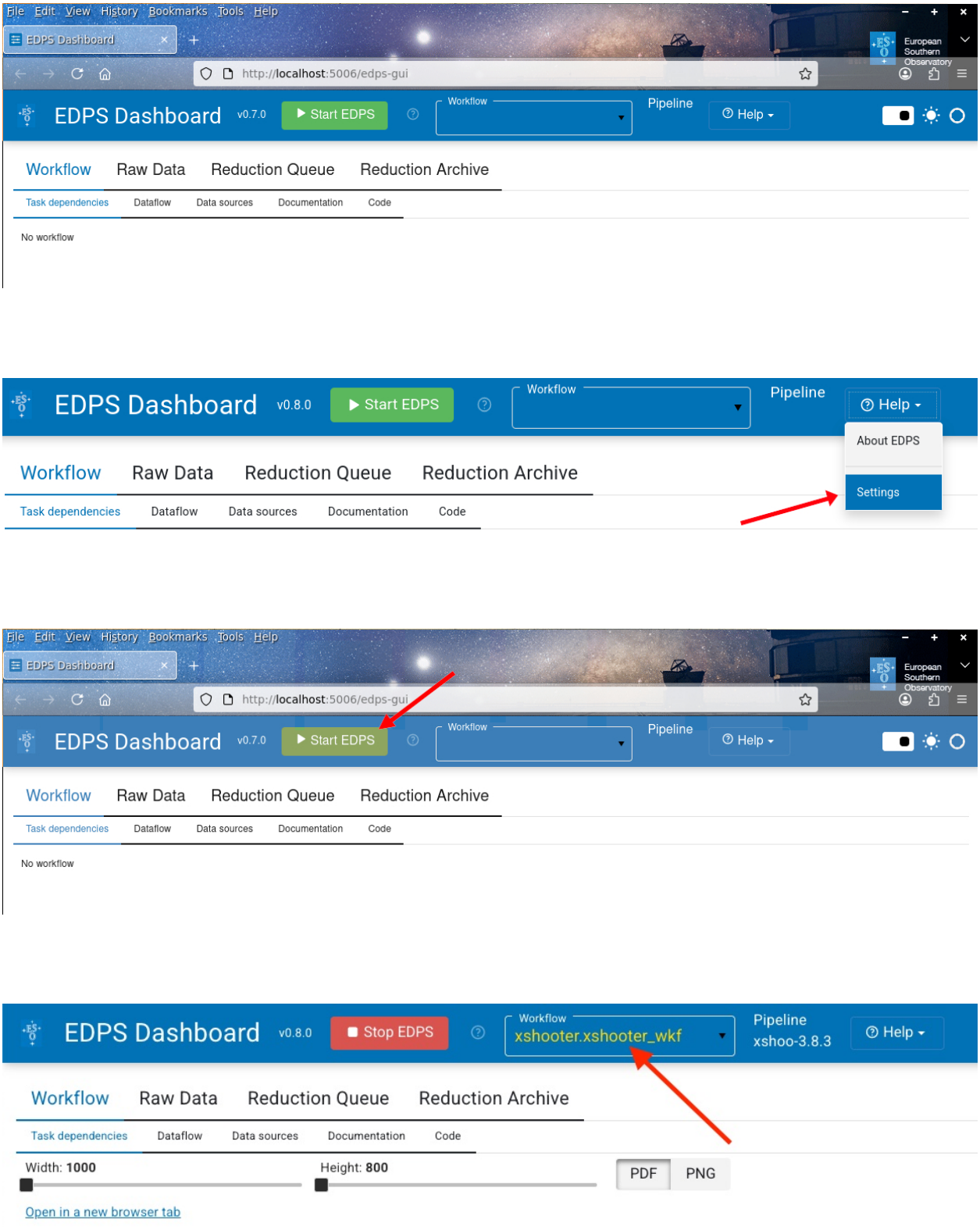
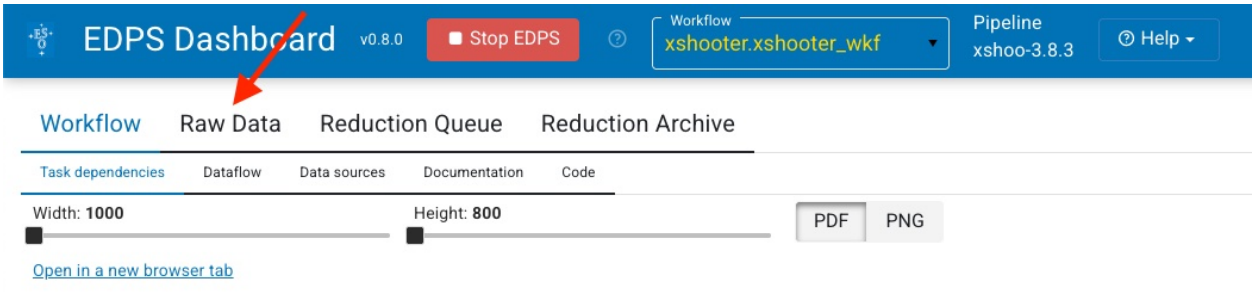


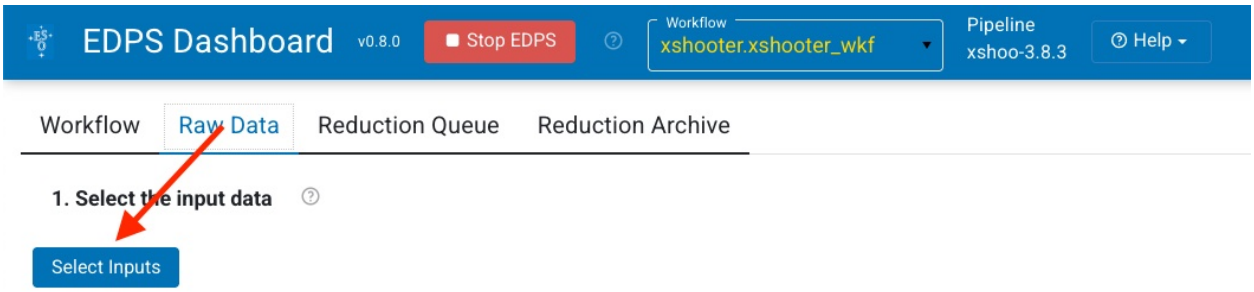
Fig. 13.1: The EDPS Dashboard (Graphic User Interface) with the XSHOOTER workflow loaded.

13.2.2 2. Selecting the input data

a. Press `Raw Data`.

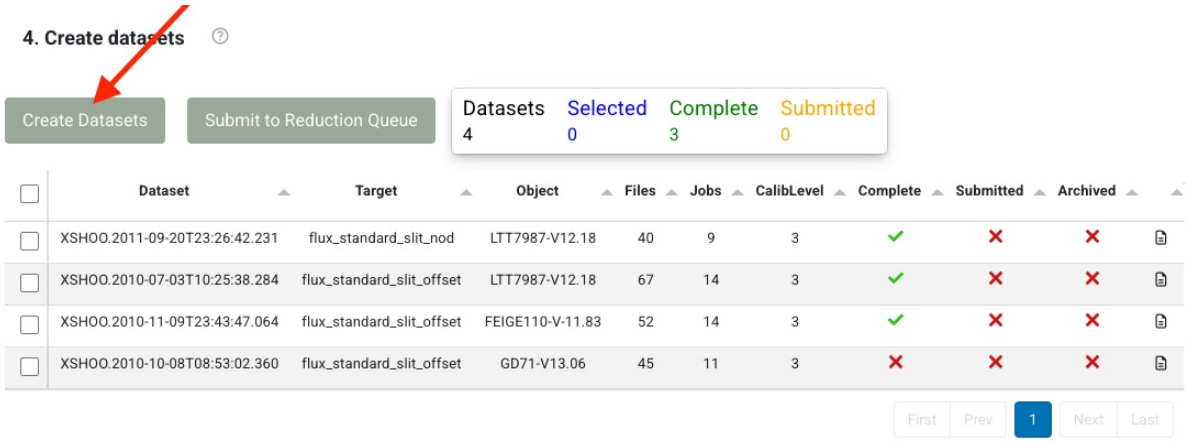


b. Press `Select Inputs`. A selection window will appear that allows to select data that are stored on a local disk.



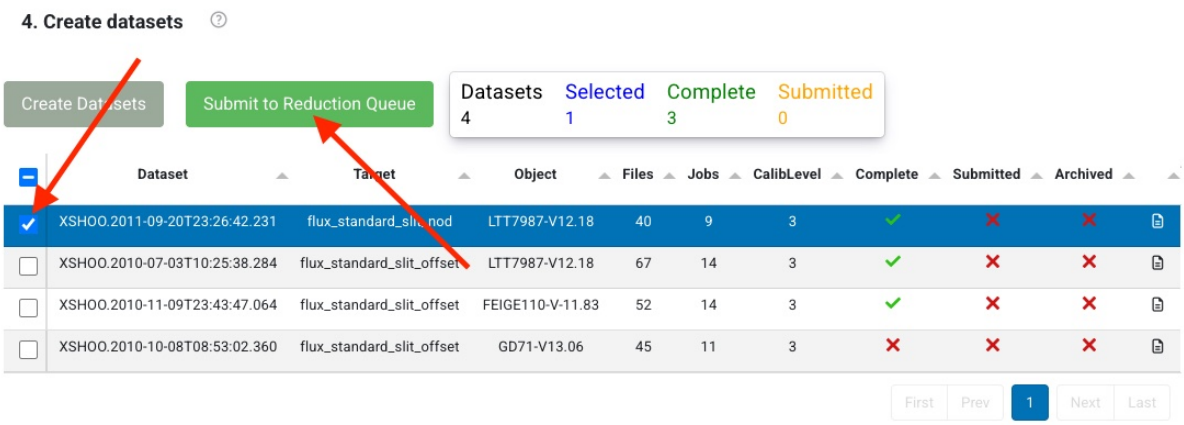
c. (Optional). Select the reduction target, configure the workflow parameter and specify the association preferences. Thise steps are optional. For more information see here.

d. Press `Create Datasets`. A list of datasets appears, one line for each set of science data.



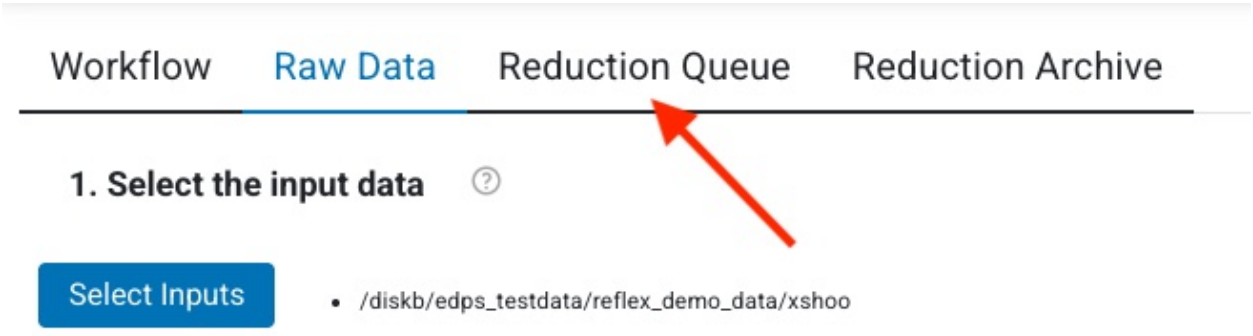
e. Choose the datasets that should be processed

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



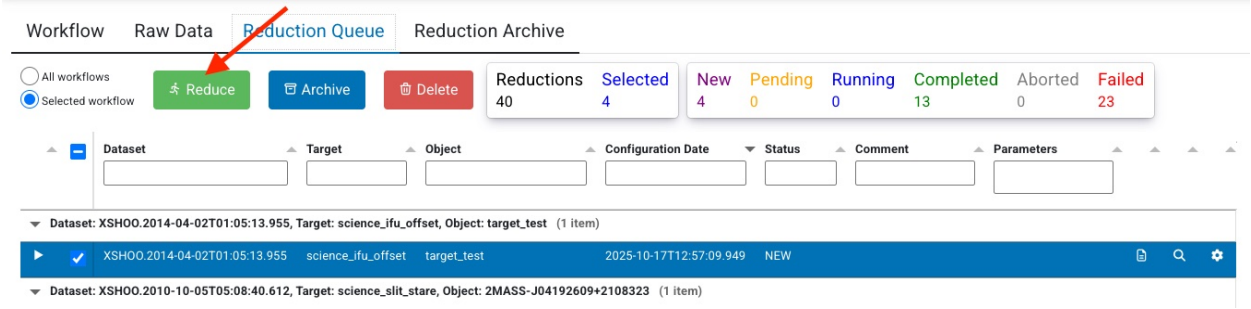
13.2.3 3. Start the reduction

a. Press `Reduction Queue`.



b. (Optional). Configure the workflow and recipe parameters by pressing the wheel button  to open the configuration editor.

c. Press the `Reduce` button. The selected data will now be processed with the default 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.

Go to *top*

13.2.4 4. Final products

By default, EDPS saves all the recipe products for all the executions in the directory specified at the first execution (default: `$HOME/EDPS_data`). However, it is possible to save only the final products into a desired location. This can be achieved by exporting archived data reduction: press the `Export` button in the `Archived Data` tab (see here).

To archive a data reduction, press the button `Archive` in the `Reduction Queue` tab (see here)

The exported products are organized by TARGET_NAME (as read from the header), 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:

- The 1D spectrum, telluric corrected, obtained from the combination of individual exposures. Its name format is: SPECTRUM_COMBINED_<target_name>, where target name is read from the header.
- The individual 1D spectra prior to combination (either telluric corrected or not). They are in a subdirectory of the combined spectrum they refer to their name formats are SPECTRUM_<arm>_<arcfile> or SPECTRUM_TELLCORR_<arm>_<arcfile> if telluric corrected. The values of arm (UVB, VIS, NIR) and arcfile (the file name of the raw science exposure) are read from the file header.
- The datacubes obtained from individual exposures for IFU observations. Its name format is: DATACUBE_<arm>_<arcfile>.

This applies to all slit modes (stare, offset, and nod) and ifu mode.

Other useful products, such as 2D spectra prior to the extraction, can be found in the EDPS_data directory, where all intermediate products are saved.

13.2.5 5. 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),

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

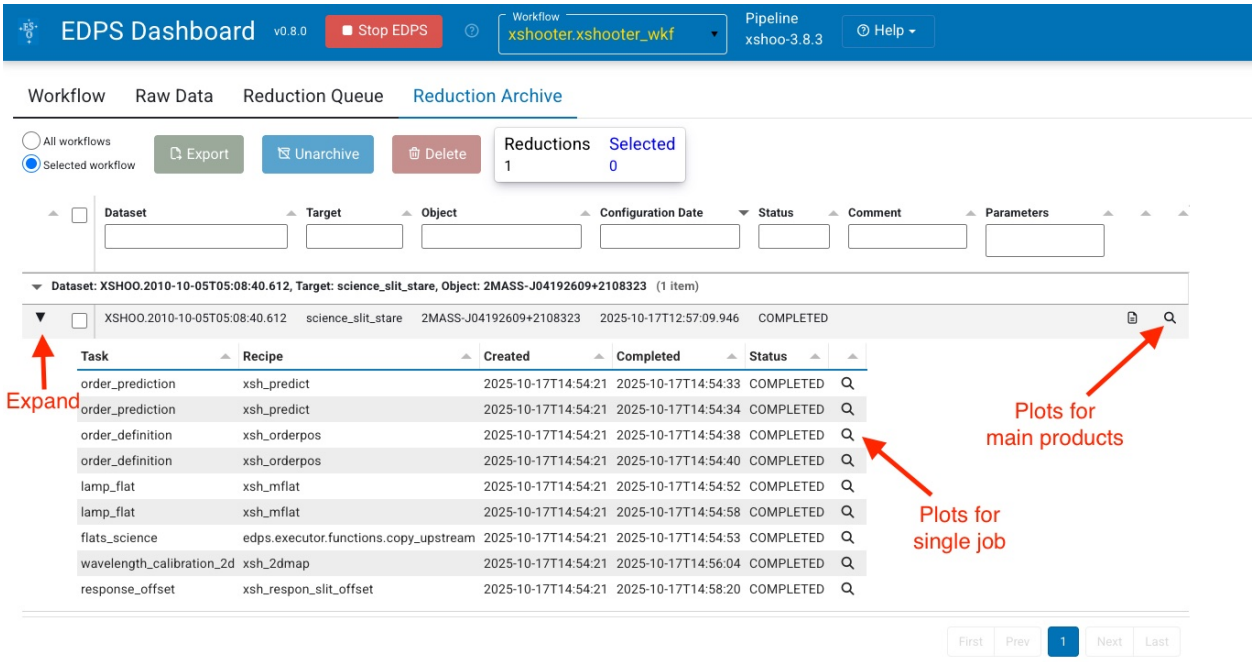
Go to *top*

Go to XSHOOTER EDPS tutorial *index*

13.3 The XSHOOTER data reduction flow

The overall data flow of the XSHOOTER pipeline is displayed here.

The reduction cascade is organized in tasks, which represent well-defined steps in the process. Tasks are 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 the most important features.



The EDPS workflow is designed to execute the tasks that deliver the final reduced spectrum (per order and merged) for each dataset, optionally flux- and telluric-calibrated. Only calibrations needed by the selected 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 standard stars, or only flat fields). This is controlled by the `Select the reduction target` option in the `Raw Data` tab.

The reduction steps 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 *here* for more information.

13.3.1 1. Generate Master Bias

This step is carried in the task **bias**, which runs the recipe **xsh_mbias**.

Produces a master bias for UVB/VIS arms. NIR frames do not use bias frames.

Customization

Recipe parameters:

- Choose stacking method (average, median) with `stack-method`.
- Adjust sigma-clipping with `klow` and `khigh`.

13.3.2 2. Generate Master Dark

This step is carried in the task **dark**, which runs the recipe **xsh_mdark**.

Produces a master dark. Optional for UVB/VIS (dark current negligible). For NIR it is only needed for stare observations; generally not used in nodding/offset where ON–OFF subtraction removes the dark.

Customization

Recipe parameters:

- Enable the use of darks for UVB/VIS data reduction with `use_optical_dark`.
- Change `stack-method` and `klow` / `khigh` thresholds to adjust stacking behaviour, e.g. for better cosmic-ray rejection.

13.3.3 3. Fit Orders

Computes initial guesses for the wavelength solution and order positions. Performed in two steps, described hereafter.

3a. Instrument Model Prediction

Recipe: `xsh_predict`

Uses the instrument physical model and information about ambient conditions during the observations (e.g. atmospheric pressure, temperature, instrument setting) to predict line positions and a first dispersion solution.

Customization

Workflow parameters:

- Select between physical-model (recommended) and polynomial mode. See *here* for more information.

3b. Order Tracing (Determining Order Geometry)

Recipe: `xsh_orderpos`

Uses pinhole order-definition frames (ORDERDEF_*) to trace the location and curvature of each echelle order on the detector. Produces order tables that later help rectification and wavelength calibration.

Customization

Recipe parameters:

- Tuning of detection thresholds can help when continuum levels are low or orders are partially vignetted.

13.3.4 4. Generate Master Flat

Recipe: `xsh_mflat`

Produces a master flat, refined version of the table from `xsh_orderpos` and a bad pixel map for each arm. The flat defines the detector response and updates the true geometry of the orders from slit illumination. For IFU flat fields the edges of the slices are traced.

Customization

Recipe parameters:

- Adjust bad-pixel handling (via `decode-bp`) if master flats saturate or raise quality control errors.
- Change `stack-method` and `klow` / `khigh` thresholds to adjust stacking behaviour, e.g. for better cosmic-ray rejection.

13.3.5 5. Wavelength Calibration

The wavelength calibration creates the wavelength and spatial resampling solutions and computes the arc-line tilts and instrumental resolution. It is done in two steps:

5a. 2D Mapping

Recipe: `xsh_2dmap`

Determines the two-dimensional wavelength solution needed to resample the orders. Creates `WAVE_TAB_2D_ARM`, which provides a full mapping from detector coordinates to wavelength+slit coordinates, and `SPECTRAL_FORMAT_TAB_ARM`, which defines, for each order, the wavelength range, pixel boundaries, predicted order edge traces, and spatial-to-spectral coordinate conversion.

Customization

Workflow parameters:

- Select between physical-model (recommended) and polynomial mode.

Recipe parameters:

- Adjust line-detection thresholds for weak arcs via `detectarclines-fit-win-hsize` and `detectarclines-search-win-hsize`. These parameters have to be small enough not to include a doublet but large enough to be able to detect and fit the line.

5b. Wavelength Calibration

Recipe: `xsh_wavecal`

Computes arc lines tilt and resolving power.

Customization

Workflow parameters:

- Select between physical-model (recommended) and polynomial mode.

13.3.6 6. Flexure Compensation

Recipe: `xsh_flexcomp`

Refines the wavelength solution to account for flexure, especially when arcs are not taken at the same rotator angle as science.

Customization

Workflow parameters:

- Select between physical-model (recommended) and polynomial mode.

13.3.7 7. Flat Strategy

This allows the user to choose between two strategies to select a flat field for the flux calibrator. The default strategy is to use the same flat as for the science observation. The alternative is to use the flat field selected by the rules, i.e., those taken closest in time to the flux calibrator.

Customization

Workflow parameters:

- By default, the `use_flat` parameter is set to *science*, meaning the flats used for science frames are also applied to standard stars. Set it to *standard* to use the flats taken closest in time to the standard-star observations.

13.3.8 8. Instrument Response and Efficiency

Recipes: `xsh_respon_slit_stare`, `xsh_respon_slit_offset` and `xsh_respon_slit_nod`

Use standard stars to derive the per-order and merged instrument response (mapping detector counts to physical flux), the blaze correction and the telescope + instrument + detector efficiency. Used for flux calibration of science exposures. The pipeline does not create response curves for IFU data, and they are therefore not flux-calibrated.

13.3.9 9. Science Reduction SLIT

Recipes: `xsh_scired_slit_stare`, `xsh_scired_slit_offset` and `xsh_scired_slit_nod`

Perform the data reduction:

- Prepare the science frame (bias/dark/inter-order background correction, flat-fielding).
- Rectify orders using the model and wavelength solution.
- Perform sky subtraction (mode-dependent).
- Localize the object and extract `ORDER1D` and `MERGE1D` products.

When several exposures are fed together, the recipes stack them (mean/median) before extraction, and you get one combined 2D and 1D spectrum per run, not one per exposure. To obtain one spectrum per exposure (or per AB pair in nodding), you must run `xsh_scired_slit_*` separately on each exposure or nod pair.

Customization

Workflow parameters:

- By default, `telluric_correction_mode`=*standard* derives the atmospheric parameters from the telluric standard. Set it to *science* to derive them directly from the science frame, or to *none* to disable telluric correction.
- Select between physical-model (recommended) and polynomial mode.

Recipe parameters:

- Change `stack-method` and `klow` / `khigh` thresholds to adjust stacking behaviour.
- Sky modelling: Use `sky-method` to select the method. In the NIR, *BSPLINE2* provides the best residuals, *BSPLINE1* is faster and acceptable for UVB/VIS, and *MEDIAN* is the fatest but may leave residuals.
- Sky region selection: use parameters `sky-position1`, `sky-hheight1`, `sky-position2`, and `sky-hheight2` to manually select sky zones. Required when object not centered, multiple objects on slit, or strong NIR gradient.
- Spectroscopic extraction: Standard extraction (`localize-method`=*MANUAL*) recommended for faint sources. Automatic detection (`localize-method`=*AUTO*) is usually fine for bright sources.

13.3.10 10. Science Reduction IFU (instrument mode decommissioned)

Recipes: `xsh_scired_ifu_stare` and `xsh_scired_ifu_offset`

Reduces a science IFU stare or on-off exposure and builds a 3D cube. Contrary to the slit mode, the pipeline does not create response curves for IFU data and they are therefore not flux-calibrated.

13.3.11 11. Atmospheric Modelling with Calibration Star

Recipe: `xsh_molecfit_model`

Models the atmosphere by fitting an atmospheric model to the input telluric-standard spectrum to derive the column densities of several molecular species.

The fitting process accounts for instrument-dependent parameters, including telescope background, spectral resolution, and wavelength calibration accuracy.

The products of this recipe are:

- ATMOS_PARM: table containing atmospheric parameters (e.g. pressure, temperature, humidity),
- BEST_FIT_PARM: table with the best-fit atmospheric and instrumental parameters,
- BEST_FIT_MODEL: the best-fit atmospheric model to the data.

13.3.12 12. Atmospheric Modelling with Science

Recipe: xsh_molecfit_model

Same as step 11, but using the science spectrum to be corrected as input instead of a telluric-standard spectrum.

13.3.13 13. Telluric Correction

The telluric absorption correction is performed using the Molecfit software in two steps, as described below:

13.3.14 13a. Determine Correction

Recipe: xsh_molecfit_calctrans

Computes the telluric correction by combining the atmospheric model produced by xsh_molecfit_model with a single reduced observation (science, flux standard, or telluric standard). This step derives the atmospheric transmission over the full wavelength range of the data.

If xsh_molecfit_model was computed using a different target than the one processed by xsh_molecfit_calctrans (e.g. a standard star instead of the science target), the difference in airmass between the two observations is taken into account when computing the transmission.

13.3.15 13b. Apply Correction

Recipe: xsh_molecfit_correct

Applies the telluric correction derived from the atmospheric transmission to the input data.

13.3.16 14. Spectra Combination

Recipe: esotk_spectrum1d_combine

This sub-workflow combines spectra selected according to the grouping rule defined in the raw_spectra_to_combine datasource. The process is performed in two steps: first, the products of the relevant science reduction tasks are collected based on the raw frames; second, the spectra are combined in a task named according to input_type. The recipe used for the combination is esotk_spectrum1d_combine. It resamples all input 1D spectra onto a common wavelength grid and stacks them. It is well suited for combining a small number of spectra affected by cosmic rays, using a median-based preprocessing step to identify and reject outliers; for larger stacks, sigma clipping alone is usually sufficient.

13.3.17 15. Additional Tasks not used in the Science Reduction Cascade

15a. Detector Linearity

Recipes: detmon_opt_lg, detmon_ir_lg

These recipes identify pixels whose response to different flux levels deviates significantly from that of the majority of the detector (optical and infrared). Such pixels are flagged in the output mask with the value 32768 and are considered potential bad pixels. This step can be useful for diagnostic purposes, but is not strictly required and, by default, it is not included in the Reduction Cascade. The UVB/VIS detectors exhibit very few pixels of this type.

Go to XSHOOTER EDPS tutorial *index*

13.4 Overview of all the data reduction configuration options

13.4.1 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 (see here).

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.

More information on the application properties file can be found here.

More explanations on the concept of "association levels" can be found here.

13.4.2 Quality reports

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),

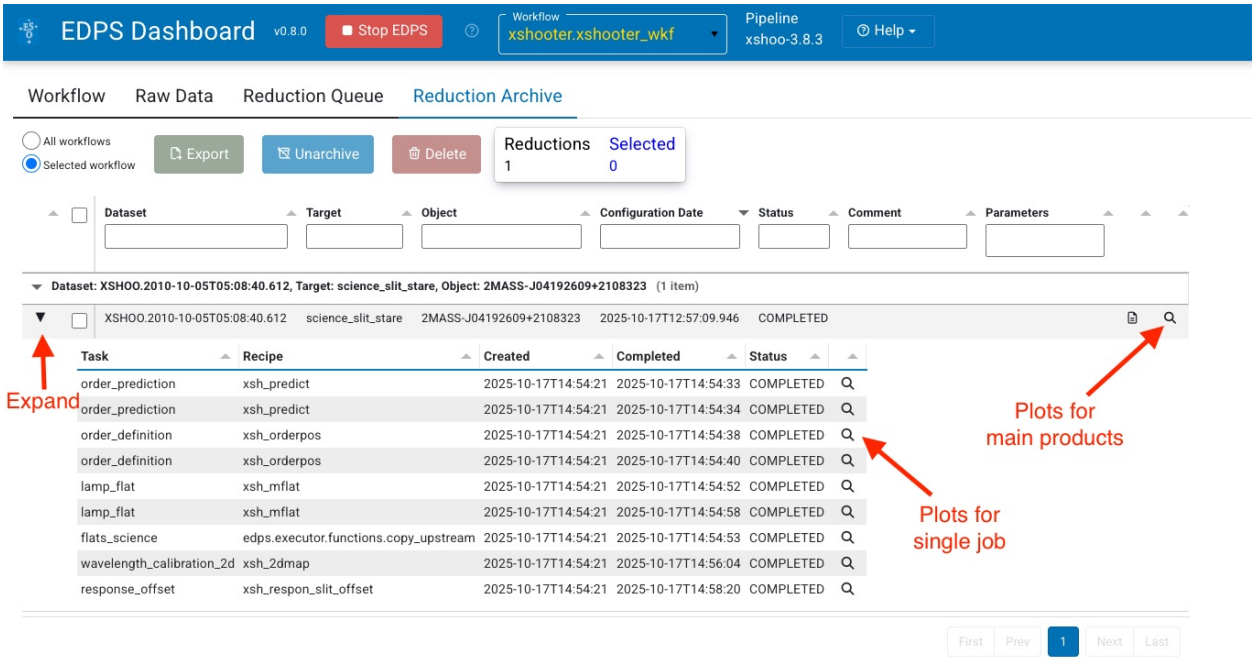
- Expand the desired dataset by pressing the black arrow on its left. The list of jobs will appear with the associated status (COMPLETED, RUNNING, PENDING)
- Press the magnifying glass symbol at the right side of the job you want to inspect. Only plots for completed jobs can be inspected.

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

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** 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.

To open the `Reduction configuration` 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 the figure below.

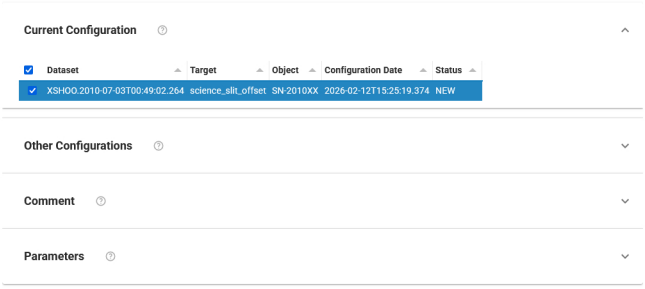
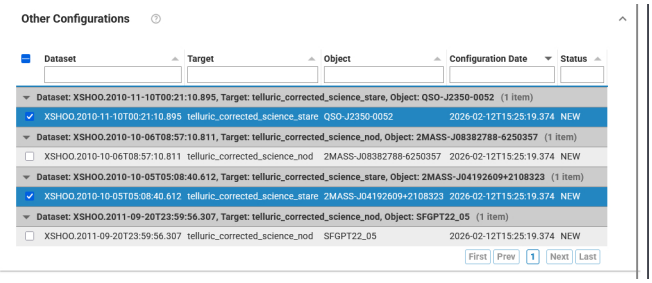


Fig. 13.2: The Reduction Configuration editor.

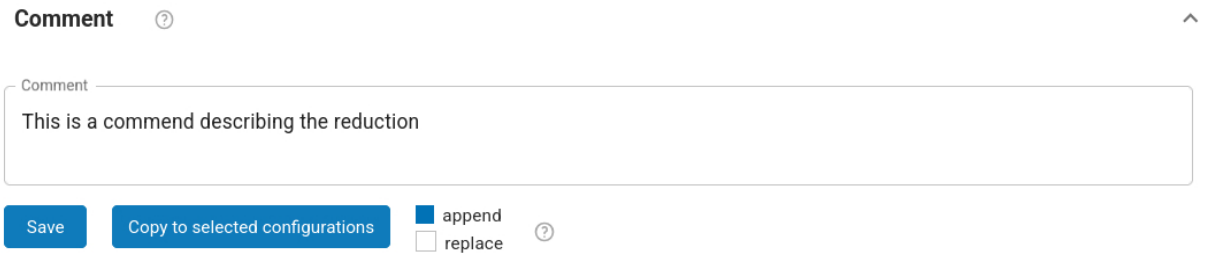
The editor is divided into 4 parts, which can be accessed pressing the corresponding expansion arrow.

Current configuration It indicates the name of the selected configuration for a given dataset.

Other configurations It allows to specify other configurations, to which the changes shall be copied to.



Comment It allows to specify a comment to describe the configuration. It is possible to append or replace a comment. 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

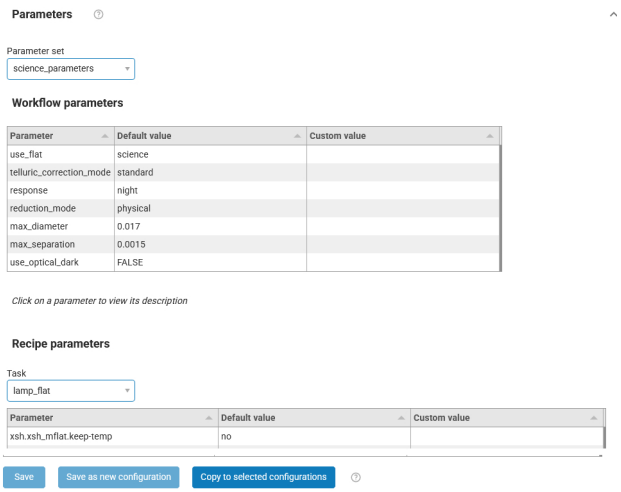
This window is visible allows to:

- 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.

For XSHOOTER, the following workflow parameters can be adjusted when the `Target Category` is set to *science*:

- `use_flat`: By default, this parameter is set to *science*, meaning the flats used for science frames are also applied to standard stars. Set it to *standard* to use the flats taken closest in time to the standard-star observations instead.
- `telluric_correction_mode`: Atmospheric parameters can be derived either from a telluric standard or from the science frame itself. Set this parameter to *standard* (default) to use a telluric star observed the same night with the same instrument setup. Set it to *science* to derive the parameters directly from the science. Use *none* to disable telluric correction.
- `response`: This parameter selects the response curve used for flux calibration. *night* (default) uses the response from the standard star observed that night; if unavailable, fall back to the master response. *master* uses the master response from CalSelector, built by combining standards from multiple nights.
- `reduction-mode`: Select between the physical-model mode (recommended) and the polynomial mode to map each wavelength onto the CCDs. The physical model derives the solution from the actual optical path, while the



polynomial method uses empirical multi-coefficient fits per order. If the reduction fails or the physical model appears inconsistent, the polynomial mode can be used for troubleshooting or as a fallback.

- `max_diameter`: TODO
- `max_separation`: TODO
- `use_optical_dark`: This parameter controls the use of a dark frame in the UVB/VIS data reduction. Because the dark current is negligible at these wavelengths, it is set to *FALSE* by default. Set it to *TRUE* if you wish to enable dark correction.

13.4.4 Troubleshooting

This section provides guidance for diagnosing and resolving common issues encountered during the XSHOOTER data-reduction cascade.

Not enough arc lines detected

The recipe `xsh_2dmap` may fail with an error indicating that not enough arc lines were detected. Typical expected values for `QC.NLINE.FOUND.CLEAN` are:

- UVB: ~2550 lines
- VIS: ~3950 lines
- NIR: ~1300 lines

Verify the quality and consistency of the input calibration products, in particular:

- `ARC_LINE_LIST_ARM`
- `THEO_TAB_MULT_ARM` (poly mode)
- `ORDER_TAB_EDGES_SLIT_ARM`
- `WAVE_TAB_GUESS_ARM`

In addition, adjust the line-detection parameters:

- increase `detectarclines-search-winhsize`
- decrease `detectarclines-min-sn` to ensure more arc lines are detected.

Go to *top*

Sky subtraction residuals (especially NIR)

Residual tilts between the sky model and the observed 2D frame can introduce sky–subtraction artefacts, particularly in the NIR. In practice, this manifests as alternating under- and over-subtracted regions around OH emission lines. See Figure Fig. 13.3 as an example.

Possible workarounds:

– #1 — Verify the accuracy of the `xsh_2dmap` products. Run the reduction chain in physical-model mode, which provides a more robust 2D geometry than the polynomial mode. Ensure that the residuals after model optimisation are small; if needed, re-run `xsh_2dmap` with additional iterations.

– #2 — Try setting `sky-method` = MEDIAN. This can improve stability when the BSPLINE models fail due to slight geometric mismatches.

– #3 — Apply flexure corrections computed by `xsh_flexcomp`. Flexure compensation can realign the sky model and reduce systematic sky–subtraction residuals.

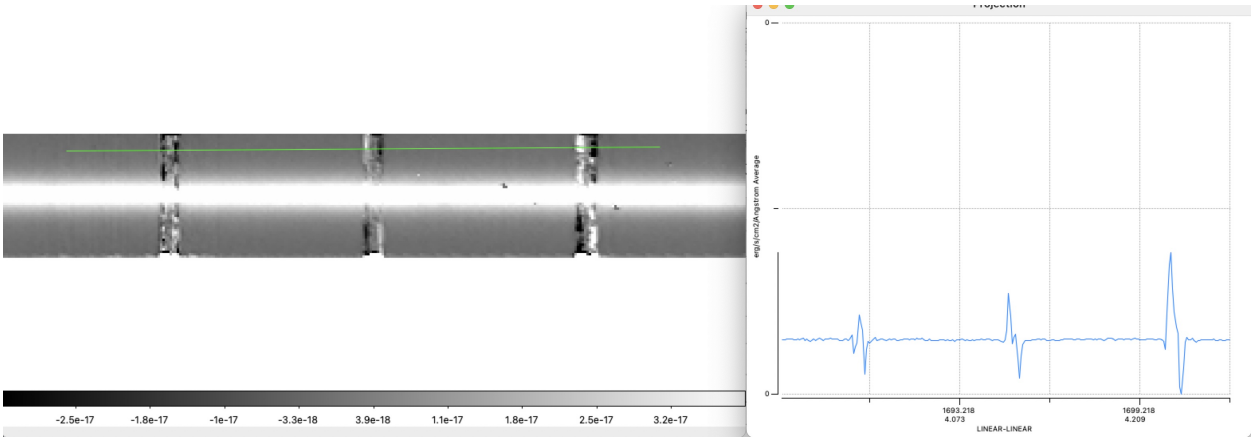


Fig. 13.3: Example of alternating under- and over-subtracted regions around OH emission lines, caused by a slight tilt of the sky lines relative to the object trace. The plot on the right shows the flux profile extracted along the green line in the 2D image.

Go to *top*

Trace-localization failures

`xsh_scired_slit_XXX` may fail when `extraction-method` = LOCALIZATION and `localize-method` is set to either GAUSSIAN or MAXIMUM.

This issue typically occurs with low S/N science exposures, where the object trace cannot be reliably detected using either the Gaussian cross-order profile or the maximum-detection algorithm.

The user should try adjusting `localize-slit-position` and `localize-slit-hheight` to guide the trace-localization process, or alternatively set `extraction-method` = MANUAL, which avoids automatic trace detection altogether.

Go to *top*

Two object traces in the slit

Reduce each object trace separately by providing appropriate values in the `xsh_scired_slit_XXX` recipe for the parameters `sky-position1`, `sky-hheight1`, `sky-position2`, and `sky-hheight2`. These parameters define the specific regions of the slit used to estimate the sky during single-frame sky subtraction.

By default, all four parameters are set to zero, meaning that the sky is taken from all pixels outside the object localization region (and outside the masked slit edges). However, when multiple objects are present along the slit, or when the default choice is not appropriate, the user should manually adjust these positions.

Both the central positions and the half-heights of the sky regions are expressed in arcseconds.

Go to *top*

Order-edge artefacts in extracted spectra

Artefacts may affect the edges of the 2D and 1D orders, degrading the quality of the merged extracted 2D and 1D spectra.

In such case, use the appropriate reference format-check frames and verify that the values of `WLMIN` and `WLMAX` in `xsh_scired_slit_XXX` are correct. They should correspond to the last wavelength imaged on the illuminated part of the order, divided by 1.007.

This can be checked by projecting the region files produced by the script `test_xsh_data_wave_tab_2d` onto the 2D sky-subtracted science image.

Alternatively, in physical-model mode, the user may provide a reference sky-line table (`SKY_LINE_LIST_ARM`) and project the predicted sky-line positions onto the 2D sky-subtracted frame to assess the consistency of order edges.

Go to *top*

Small jumps between orders

Small discontinuities may appear between orders during merging, typically caused by slit-edge artefacts introduced by the flat-field. To mitigate this, we recommend setting `extract-method` = LOCALIZATION in `xsh_scired_slit_XXX`.

For low S/N objects, use `localize-method` = MANUAL and provide appropriate values for `localize-slit-position` and `localize-slit-hheight`. For high S/N objects, `localize-method` = AUTO is generally sufficient.

Go to *top*

Overestimated uncertainties in the extracted spectrum

This issue is typically caused by slit-edge artefacts introduced by the flat-field. To mitigate it, set `extract-method` = LOCALIZATION in `xsh_scired_slit_XXX` and choose an appropriate localization strategy:

For low S/N objects, use `localize-method` = MANUAL and specify suitable values for `localize-slit-position` and `localize-slit-hheight`.

For high S/N objects, localize-method = AUTO is usually sufficient.

Go to *top*

The extracted merged 1D spectrum looks poor

The `xsh_respon_slit_xxx` and/or `xsh_scired_slit_xxx` recipes may have failed to localize the trace correctly during extraction.

Try setting `extract-method` = LOCALIZATION and provide appropriate values for `localize-slit-position` and `localize-slit-hheight`, based on the appearance of the merged 2D spectral image (e.g. `SCI_SLIT_FLUX_MERGE2D_NIR.fits`).

These parameters should be tuned so that:

- the object lies near the centre of the extraction aperture, and
- the extraction window fully includes the entire positive flux profile.

Go to *top*

Spectral regions of zero flux values

Large regions of zero-flux values in the final extracted 1D spectrum may be caused by spurious Cosmic-Ray (CR) detections. See Fig. 13.4 as an example of affected spectrum.

CR-contaminated pixels are identified using the van Dokkum algorithm (2001, PASP, 113, 1420) applied to the input science frames. The detection threshold is controlled by the parameter `removecrhsingle-sigmalim`, whose default value is 5.0 for the UVB and VIS arms. For the NIR arm, CR detection is disabled by default by setting `removecrhsingle-niter` to 0.

In observations obtained under excellent seeing conditions (0.4 or better), the stellar PSF may appear sharp enough to be mistakenly flagged as CR-affected. To reduce these false detections, increase the value of `removecrhsingle-sigmalim`. Figure Fig. 13.5 illustrates the improvement obtained on the same observation.

IFU cube not well aligned

Go to *top*

Go to XSHOOTER EDPS tutorial *index*

13.5 List of workflow tasks

This is the list of all the tasks and associated recipes in the XSHOOTER 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
acquisition_camera_flat	xsh_mflat	no	Create a master flat for the acquisition camera.
arc	xsh_wavecal	yes	Compute arc-line tilts and instrument resolution.

continues on next page

Table 13.1 – continued from previous page

TASK	RECIPE	Used in science reduction	Notes
atmospheric_model_on_science_	xsh_molecfit_model	optional	Run molecfit_model on flux-standard spectrum observed in nod mode.
atmospheric_model_on_science_	xsh_molecfit_model	optional	Run molecfit_model on flux-standard spectrum observed in offset mode.
atmospheric_model_on_science_	xsh_molecfit_model	optional	Run molecfit_model on flux-standard spectrum observed in stare mode.
atmospheric_model_on_science_	xsh_molecfit_model	optional	Run molecfit_model on science spectrum observed in nod mode.
atmospheric_model_on_science_	xsh_molecfit_model	optional	Run molecfit_model on science spectrum observed in offset mode.
atmospheric_model_on_science_	xsh_molecfit_model	optional	Run molecfit_model on science spectrum observed in stare mode.
atmospheric_model_on_science_	xsh_molecfit_model	optional	Run molecfit_model on telluric-standard spectrum observed in nod mode.
atmospheric_model_on_science_	xsh_molecfit_model	optional	Run molecfit_model on telluric-standard spectrum observed in offset mode.
atmospheric_model_on_science_	xsh_molecfit_model	optional	Run molecfit_model on telluric-standard spectrum observed in stare mode.
atmospheric_model_on_standard	xsh_molecfit_model	optional	Run molecfit_model on telluric-standard spectrum observed in nod mode.
atmospheric_model_on_standard	xsh_molecfit_model	optional	Run molecfit_model on telluric-standard spectrum observed in offset mode.
atmospheric_model_on_standard	xsh_molecfit_model	optional	Run molecfit_model on telluric-standard spectrum observed in stare mode.
atmospheric_transmission_for_fl	xsh_molecfit_calctrans	optional	Apply telluric correction on flux-standard spectrum observed in nod mode.
atmospheric_transmission_for_fl	xsh_molecfit_calctrans	optional	Apply telluric correction on flux-standard spectrum observed in offset mode.
atmospheric_transmission_for_fl	xsh_molecfit_calctrans	optional	Apply telluric correction on flux-standard spectrum observed in stare mode.
atmospheric_transmission_for_s	xsh_molecfit_calctrans	optional	Apply telluric correction on science spectrum observed in nod mode.

continues on next page

Table 13.1 – continued from previous page

TASK	RECIPE	Used in science reduction	Notes
atmospheric_transmission_for_s	xsh_molecfit_calctrans	optional	Apply telluric correction on science spectrum observed in nod mode.
atmospheric_transmission_for_s	xsh_molecfit_calctrans	optional	Apply telluric correction on science spectrum observed in nod mode.
atmospheric_transmission_for_te	xsh_molecfit_calctrans	optional	Apply telluric correction on telluric-standard spectrum observed in nod mode.
atmospheric_transmission_for_te	xsh_molecfit_calctrans	optional	Apply telluric correction on telluric-standard spectrum observed in offset mode.
atmospheric_transmission_for_te	xsh_molecfit_calctrans	optional	Apply telluric correction on telluric-standard spectrum observed in stare mode.
bias	xsh_mbias	yes	Create a master bias.
dark	xsh_mdark	optional	Create a master dark.
detmon_nir	detmon_ir_lg	no	Compute NIR detector gain and linearity-map, flag abnormal pixels.
detmon_optical	detmon_opt_lg	no	Compute VIS detector gain and linearity-map, flag abnormal pixels.
flats_science	-	yes	Select flats associated with science observations for flat-fielding the flux calibrator.
flats_standard	-	optional	Select flats closer in time to the flux calibrator for flat-fielding the flux calibrator.
flexures	xsh_flexcomp	yes	Refine the wavelength solution to account for flexure.
flux_standard_combination	-	optional	Resample flux-standard spectra to a common grid and stack them.
flux_standard_slit_nod	xsh_scired_slit_nod	optional	Reduces flux-standard exposure in slit nod mode.
flux_standard_slit_offset	xsh_scired_slit_offset	optional	Reduces flux-standard exposure in slit offet mode.
flux_standard_slit_stare	xsh_scired_slit_stare	optional	Reduces flux-standard exposure in slit stare mode.
lamp_flat	xsh_mflat	yes	Create master flat and order edge traces table frame.

continues on next page

Table 13.1 – continued from previous page

TASK	RECIPE	Used in science reduction	Notes
lamp_flat_hc	xsh_mflat	yes	Create master flat (high counts).
order_definition	xsh_orderpos	yes	Trace orders location and curvature.
order_prediction	xsh_predict	yes	Compute a first guess dispersion solution and order trace.
response_offset	xsh_respon_slit_offset	optional	Compute the instrument response in slit offset mode.
response_slit_nod	xsh_respon_slit_nod	optional	Compute the instrument response in slit nod mode.
response_slit_stare	xsh_respon_slit_stare	optional	Compute the instrument response in slit stare mode.
science_combination	-	yes	Resample science spectra to a common grid and stack them.
science_ifu_offset	science_ifu_offset	yes	Reduce science exposure in IFU offset mode.
science_ifu_stare	science_ifu_stare	yes	Reduce science exposure in IFU stare mode.
science_slit_nod	xsh_scired_slit_nod	yes	Reduce science exposure in IFU nod mode.
science_slit_offset	xsh_scired_slit_offset	yes	Reduce science exposure in slit offset mode.
science_slit_stare	xsh_scired_slit_stare	yes	Reduce science exposure in slit stare mode.
select_flux_standard_to_comb	esotk_spectrum1d_combine	optional	Collect flux-standard reduction products and select tasks to process them.
select_science_to_combine	esotk_spectrum1d_combine	optional	Collect science reduction products and select tasks to process them.
select_telluric_standard_to_c	esotk_spectrum1d_combine	optional	Collect telluric-standard reduction products and select tasks to process them.
telluric_corrected_flux_standa	xsh_molecfit_correct	optional	Apply telluric correction to flux-standard spectrum observed in nod mode.
telluric_corrected_flux_standa	xsh_molecfit_correct	optional	Apply telluric correction to flux-standard spectrum observed in offset mode.
telluric_corrected_flux_standa	xsh_molecfit_correct	optional	Apply telluric correction to flux-standard spectrum observed in stare mode.
telluric_corrected_science_no	xsh_molecfit_correct	optional	Apply telluric correction to science spectrum in nod mode.

continues on next page

Table 13.1 – continued from previous page

TASK	RECIPE	Used in science reduction	Notes
telluric_corrected_science_off	xsh_molecfit_correct	optional	Apply telluric correction to science spectrum in offset mode.
telluric_corrected_science_sta	xsh_molecfit_correct	optional	Apply telluric correction to science spectrum in stare mode.
telluric_corrected_telluric_sta	xsh_molecfit_correct	optional	Apply telluric correction to telluric-standard spectrum observed in nod mode.
telluric_corrected_telluric_sta	xsh_molecfit_correct	optional	Apply telluric correction to telluric-standard spectrum observed in offset mode.
telluric_corrected_telluric_sta	xsh_molecfit_correct	optional	Apply telluric correction to telluric-standard spectrum observed in stare mode.
telluric_standard_combination	-	optional	Resample telluric-standard spectra to a common grid and stack them.
telluric_standard_slit_nod	xsh_scired_slit_nod	optional	Reduce telluric exposure in slit nod mode.
telluric_standard_slit_offset	xsh_scired_slit_offset	optional	Reduce telluric exposure in slit offset mode.
telluric_standard_slit_stare	xsh_scired_slit_stare	optional	Reduce telluric exposure in slit stare mode.
wavelength_calibration_2d	xsh_2dmap	yes	Compute wavelength and spatial resampling solution.

13.6 Frequently Asked Questions

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. 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. Click on 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.

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.

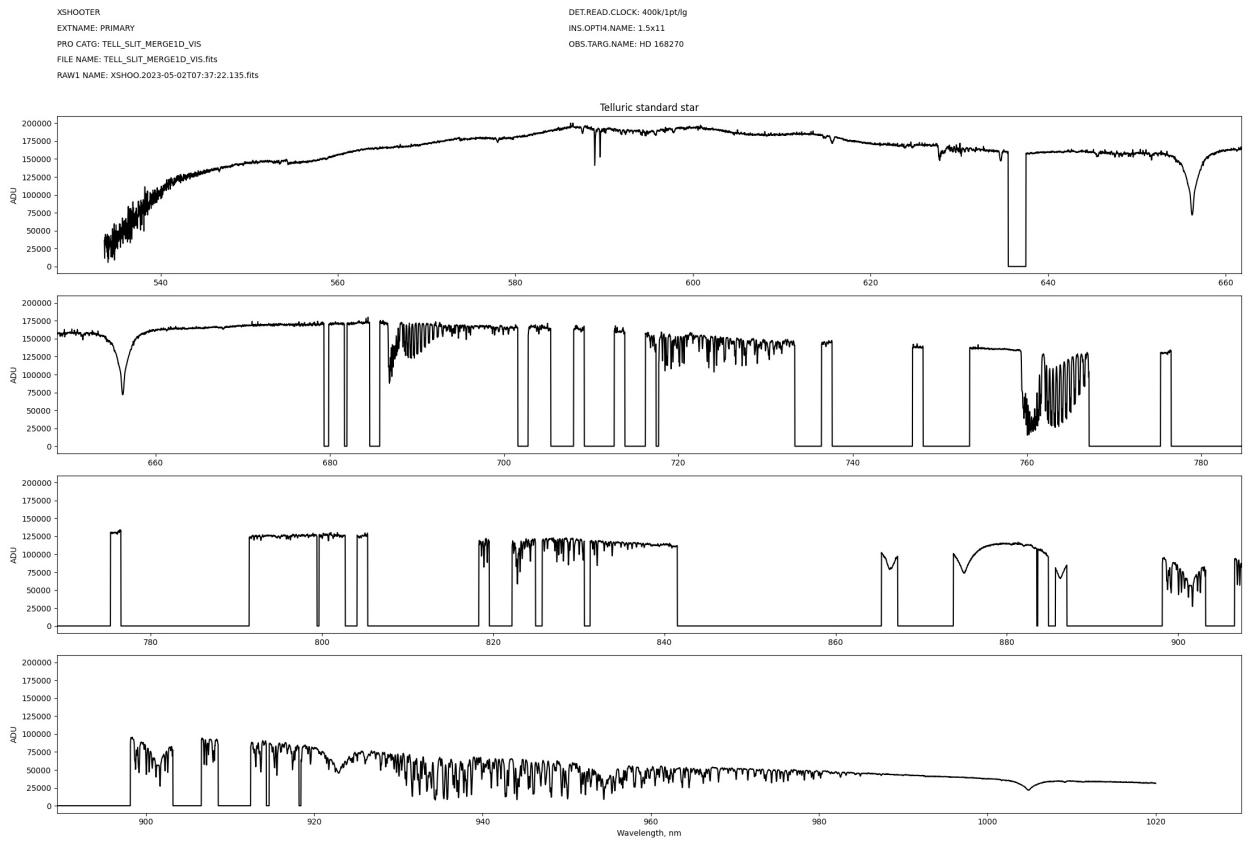


Fig. 13.4: Here we show the VIS spectrum of the telluric standard star Hip089684, as displayed in the `Graphical reports`. The observation was obtained on May 1, 2023 (OB identifier: `XSHOO.2023-05-02T07:37:22.135`) under excellent seeing conditions (0.4). With the default value `removecrhsingle-sigmalim`=5.0, the algorithm produces spurious cosmic-ray detections, which in turn lead to extended wavelength intervals with zero flux in the extracted 1D spectrum.

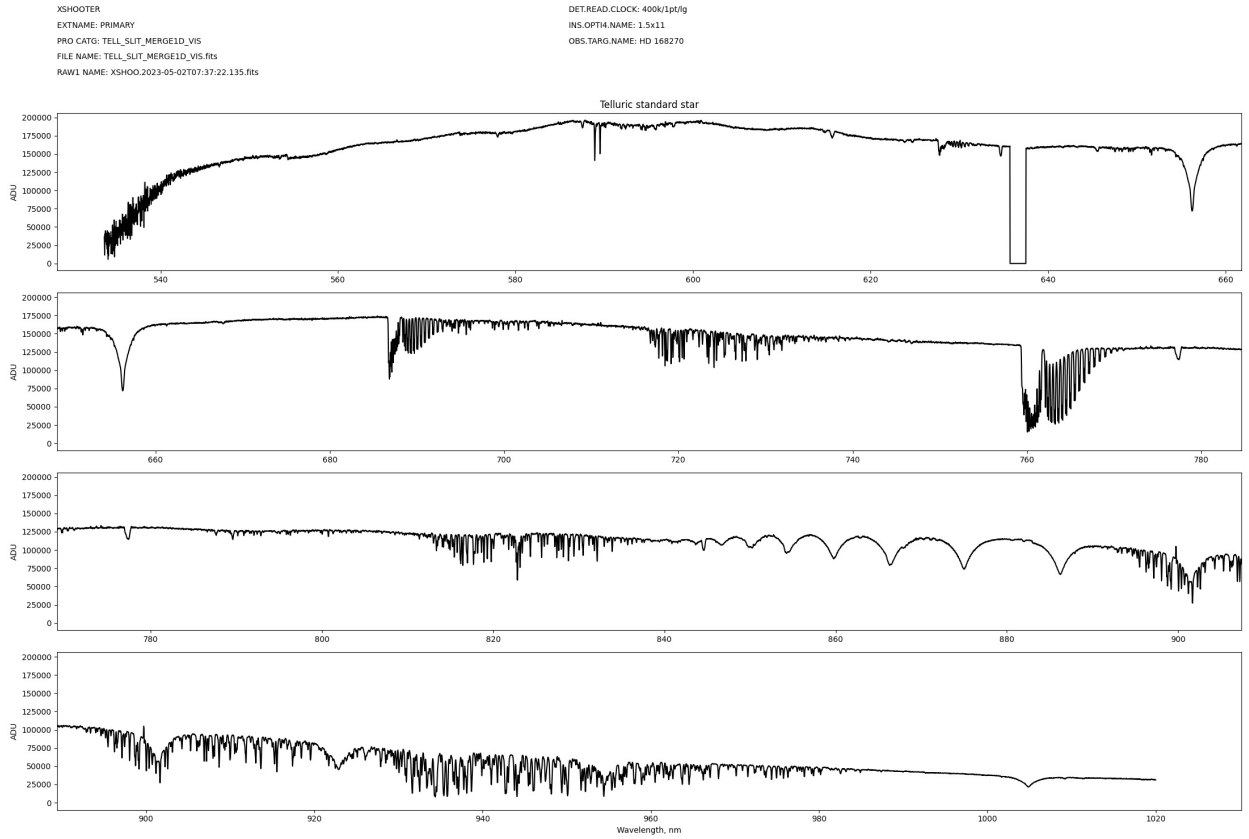
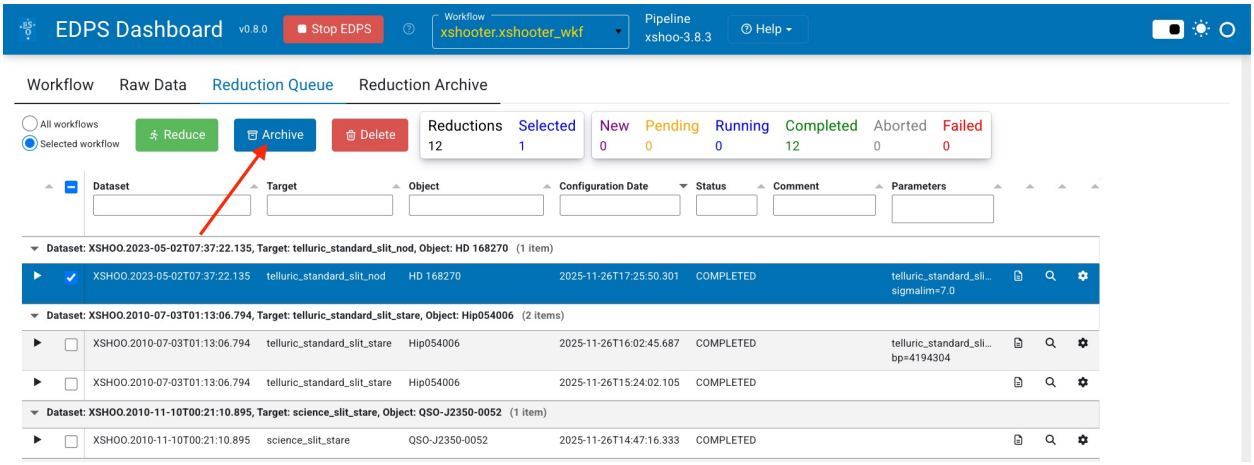
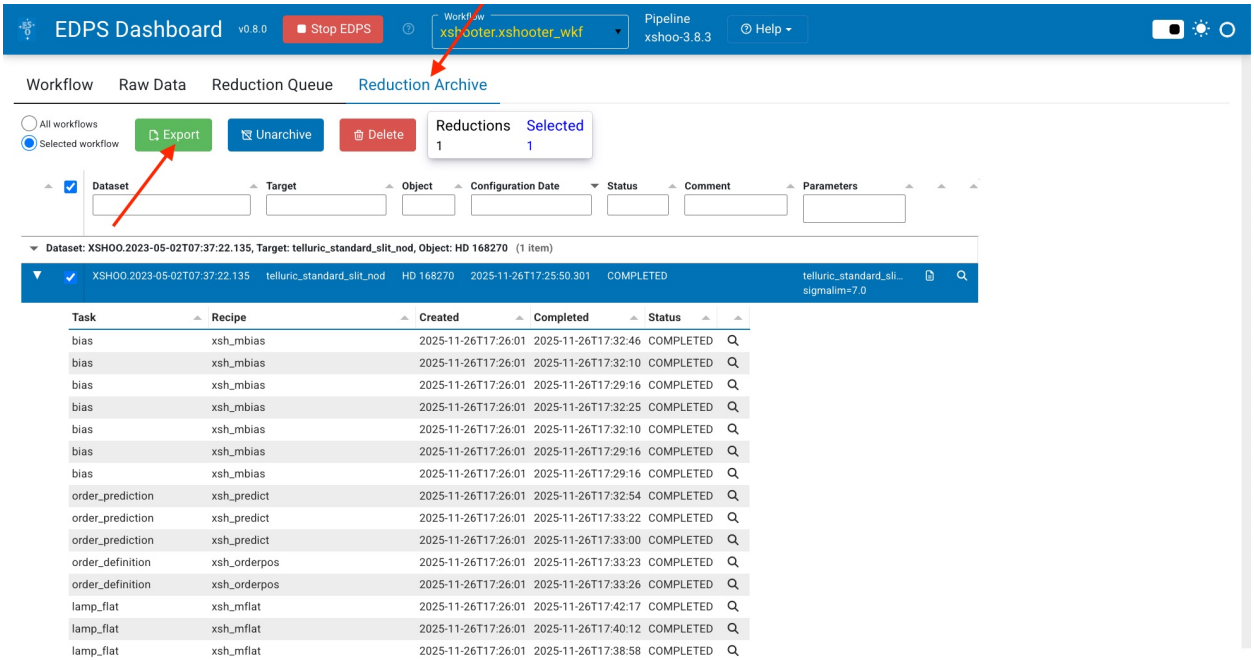
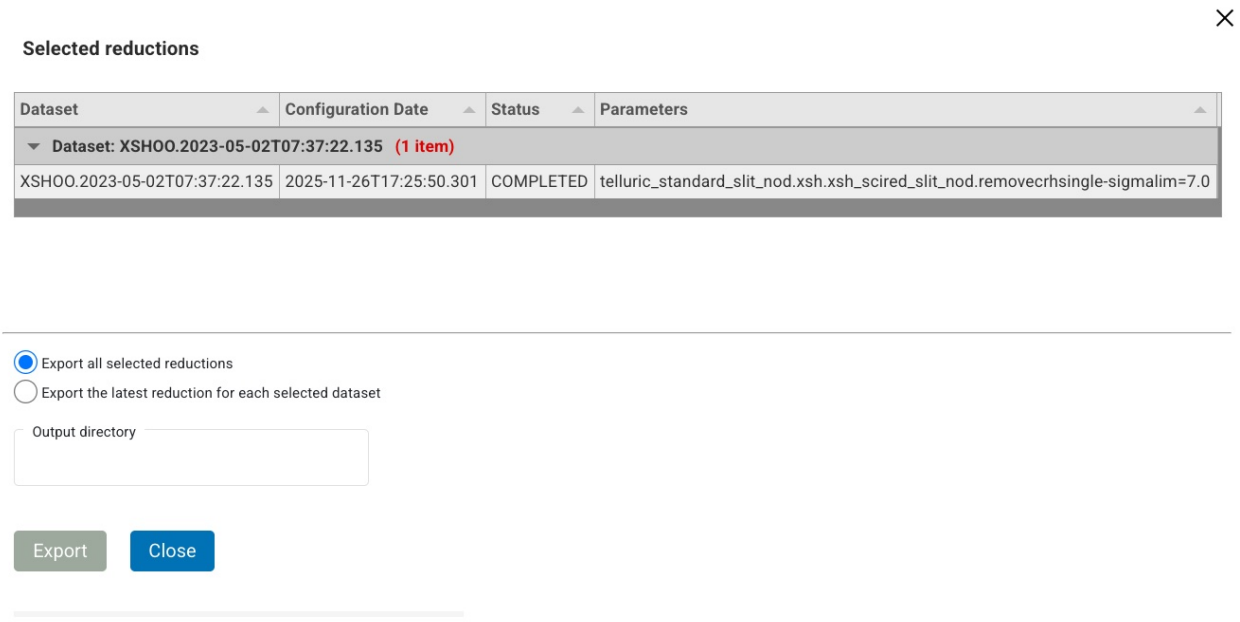


Fig. 13.5: Same as Fig. 13.4, but with `removecrhsingle-sigmalim`=7.0. In this case, this avoids the flagging of good pixels as CR-affected.




EDPS Dashboard
v0.8.0
Stop EDPS
Workflow
xshooter.xshooter_wkf
Pipeline
xshoo-3.8.3
Help
Workflow
Raw Data
Reduction Queue
Reduction Archive
All workflows
Selected workflow
Export
Unarchive
Delete
Reductions
1
Selected
1
Dataset
Target
Object
Configuration Date
Status
Comment
Parameters
Dataset: XSHOO.2023-05-02T07:37:22.135, Target: telluric_standard_slit_nod, Object: HD 168270 (1 item)
XSHOO.2023-05-02T07:37:22.135 telluric_standard_slit_nod HD 168270 2025-11-26T17:25:50.301 COMPLETED telluric_standard_sli... sigmalim=7.0
Task Recipe Created Completed Status
bias xsh_mbias 2025-11-26T17:26:01 2025-11-26T17:32:46 COMPLETED
bias xsh_mbias 2025-11-26T17:26:01 2025-11-26T17:32:10 COMPLETED
bias xsh_mbias 2025-11-26T17:26:01 2025-11-26T17:29:16 COMPLETED
bias xsh_mbias 2025-11-26T17:26:01 2025-11-26T17:32:25 COMPLETED
bias xsh_mbias 2025-11-26T17:26:01 2025-11-26T17:32:10 COMPLETED
bias xsh_mbias 2025-11-26T17:26:01 2025-11-26T17:29:16 COMPLETED
bias xsh_mbias 2025-11-26T17:26:01 2025-11-26T17:29:16 COMPLETED
order_prediction xsh_predict 2025-11-26T17:26:01 2025-11-26T17:32:54 COMPLETED
order_prediction xsh_predict 2025-11-26T17:26:01 2025-11-26T17:33:22 COMPLETED
order_prediction xsh_predict 2025-11-26T17:26:01 2025-11-26T17:33:00 COMPLETED
order_definition xsh_orderpos 2025-11-26T17:26:01 2025-11-26T17:33:23 COMPLETED
order_definition xsh_orderpos 2025-11-26T17:26:01 2025-11-26T17:33:26 COMPLETED
lamp_flat xsh_mflat 2025-11-26T17:26:01 2025-11-26T17:42:17 COMPLETED
lamp_flat xsh_mflat 2025-11-26T17:26:01 2025-11-26T17:40:12 COMPLETED
lamp_flat xsh_mflat 2025-11-26T17:26:01 2025-11-26T17:38:58 COMPLETED


Selected reductions
Dataset
Configuration Date
Status
Parameters
Dataset: XSHOO.2023-05-02T07:37:22.135 (1 item)
XSHOO.2023-05-02T07:37:22.135 2025-11-26T17:25:50.301 COMPLETED telluric_standard_slit_nod.xsh.xsh_scired_slit_nod.removecrhsingle-sigmalim=7.0
Export all selected reductions
Export the latest reduction for each selected dataset
Output directory
Export
Close

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>
```

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

Answer: Install the `datademo` package provided with the pipeline installation or download the "Demo Data" package from href=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
```

```
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>
```

Answer: For suggestions, questions, or feedback in general, please open a ticket with the EDPS Support team. This link should take you directly to a webpage for creating and EDPS feedback ticket, but incase 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]".

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 migth break some internal links. Recipes requiring external packages such as telluriccorr might not work (impacts on kmos, xshooter, fors, and molecfit pipelines)

Answer: This is a know bug that will be fixed in the next release. The cause is that the input data directory contains 2 files with the same properties and the same mjd-obs, but with different names (e.g. the same static calibration that is present in 2 places). Since their mjd-obs is the same, EDPS sometimes adds one file, sometimes the other. When a different file is associated, the dataset is labelled as new.

Go to XSHOOTER tutorial *index*

XSHOOTER Version 3.8.3 - 20260212