

European Organisation for Astronomical Research in the Southern Hemisphere

Programme: GEN

Project/WP: Science Operation Software Department

Reflex ERIS-NIX Tutorial

Document Number:

Document Version: 2.1.0

Document Type: Manual (MAN)

Released on: 2026-11-05

Document Classification: Public

Prepared by: ERIS Pipeline Team

Validated by:

Approved by:

Name

This page was intentionally left blank

Authors

Name	Affiliation
A. Modigliani	ESO

This page was intentionally left blank

Change Record from previous Version

Release	Affected Section(s)	Changes/Reason/Remarks
1.1.0	All	First public release
1.6.4		Some image
1.7.0		Updated
2.1.0		Updated

This page was intentionally left blank

Contents

1 Introduction to `Esoreflex` 9

2 Workflow Status 10

3 Software Installation 11

3.1 Installing `Esoreflex` workflows via `macports` . . . 11

3.2 Installing `Esoreflex` workflows via `rpm/yum/dnf` . . . 11

3.3 Installing `Esoreflex` workflows via `install_esoreflex` . . . 11

4 Demo Data 13

5 Quick Start: Reducing The Demo Data 14

6 About the main `esoreflex` canvas 17

6.1 Saving And Loading Workflows . . . 17

6.2 Buttons . . . 17

6.3 Workflow States . . . 17

7 The ERIS Workflow 18

7.1 Workflow Canvas Parameters . . . 18

7.2 Workflow Actors . . . 19

7.2.1 Simple Actors . . . 19

7.2.2 DataSetChooser . . . 19

7.2.3 The ProductExplorer . . . 20

7.2.4 Lazy Mode . . . 22

8 Reducing your own data 23

8.1 The `esoreflex` command . . . 23

8.2 Launching the workflow . . . 23

9 Reducing and Calibrating Your Own Science Data with Reflex 26

9.1 Available Reflex workflows . . . 26

9.2 Specifying data directories and selecting files . . . 26

9.3 Detector Linearity . . . 28
9.4 Master Dark Creation . . . 30
9.5 Master Flat Lamp . . . 31
9.6 Master Flat Twilight . . . 32
9.7 Removal of instrument signature . . . 33
9.8 Sky background correction . . . 34
9.9 World Coordinate System Calibration . . . 35
9.10 Photometric Calibration . . . 37
9.11 Frame Stacking . . . 39
9.12 Examining the workflow results . . . 41

10 Frequently Asked Questions **42**

1 Introduction to Esoreflex

This document is a tutorial designed to enable the user to to reduce his/her data with the ESO pipeline run under an user-friendly environment, called EsoReflex, concentrating on high-level issues such as data reduction quality and signal-to-noise (S/N) optimisation.

EsoReflex is the ESO Recipe Flexible Execution Workbench, an environment to run ESO VLT pipelines which employs a workflow engine to provide a real-time visual representation of a data reduction cascade, called a workflow, which can be easily understood by most astronomers. The basic philosophy and concepts of Reflex have been discussed by Freudling et al. (2013A&A...559A..96F). Please reference this article if you use Reflex in a scientific publication.

Reflex and the data reduction workflows have been developed by ESO and instrument consortia and they are fully supported. If you have any issue, please have a look to https://support.eso.org to see if this has been reported before or open a ticket for further support.

A workflow accepts science and calibration data, as downloaded from the archive using the CalSelector tool[1] (with associated raw calibrations) and organises them into DataSets, where each DataSet contains one science object observation (possibly consisting of several science files) and all associated raw and static calibrations required for a successful data reduction. The data organisation process is fully automatic, which is a major time-saving feature provided by the software. The DataSets selected by the user for reduction are fed to the workflow which executes the relevant pipeline recipes (or stages) in the correct order. Full control of the various recipe parameters is available within the workflow, and the workflow deals automatically with optional recipe inputs via built-in conditional branches. Additionally, the workflow stores the reduced final data products in a logically organised directory structure employing user-configurable file names.

This is the ESO-Reflex tutorial describing ESO-Rreflex reduction of ERIS-NIX IMG data using the eris_nix_imgworkflow. The user is referred to the ERIS web page (http://www.eso.org/sci/facilities/paranal/instruments/eris/) for more information on the instrument itself, and the ERIS-NIX pipeline user manual for the details of the pipeline recipes (http://www.eso.org/sci/software/pipelines/). The workflow uses association rules known to work with files downloaded from the ESO archive with the CalSelector tool.

[1] *http://www.eso.org/sci/archive/calselectorInfo.html*

2 Workflow Status

The ERIS-NIX Reflex workflow, in its current version, is capable, together with its underlying ERIS pipeline, of reducing ERIS-NIX data. The Reflex workflow was build by John Lighfoot from the ERIS consortium, with some contribute from the ESO SCPPS ERIS Team.

The first step of the ERIS-NIX Reflex workflow is to organise the data of this instrument into an associated, organised, and classified structure including for each science files, the required instrument calibration files with matching spectral-band. The user will be warned if any required calibration frames are missing.

The ERIS-NIX Reflex workflow, *eris_nix_img*, will correct the observed frames for their dark ans sky background level, flat-field the data, and performs WCS and photometric calibrations. A large number of data products are created and retained for the user to assess the quality of the pipeline processing.

During the processing within the Reflex workflow, the user has the ability to modify a number of pipeline parameters in order to optimise the data processing. This is done via interactive workflows that display the most important products and reduction parameters.

During the pipeline development, instrument commissioning, and the following experience of ERIS-NIX operations, the pipeline parameters have been set to default values that deliver the best results for the most cases. However, the user should make an effort to adjust and experiment with the parameters to optimize the results.

3 Software Installation

`Esoreflex` and the workflows can be installed in different ways: via package repositories, via the `install_esoreflex` script or manually installing the software tar files.

The recommended way is to use the package repositories if your operating system is supported. The pipelines and Reflex can be installed from the ESO `macports` repositories that support macOS platforms, the and the `rpm/yum` repositories that support Fedora and CentOS platforms. For any other operating system it is recommended to use the `install_esoreflex` script.

The installation from package repository requires administrative privileges (typically granted via sudo), as it installs files in system-wide directories under the control of the package manager. If you want a local installation, or you do not have sudo privileges, or if you want to manage different installations on different directories, then use the `install_esoreflex` script. Note that the script installation requires that your system fulfill several software prerequisites, which might also need sudo privileges.

Please note that in case of major or minor (affecting the first two digit numbers) Reflex upgrades, the user should erase the `$HOME/KeplerData`, `$HOME/.kepler` directories if present, to prevent possible aborts (i.e. a hard crash) of the `esoreflex` process.

3.1 Installing `Esoreflex` workflows via `macports`

This method is supported for the macOS operating system. It is assumed that macports (*http://www.macports.org*) is installed. Please read the full documentation at *http://www.eso.org/sci/software/pipelines/installation/macports.html*, which also describes the versions of macOS that are currently supported.

3.2 Installing `Esoreflex` workflows via `rpm/yum/dnf`

This method is supported for Fedora and CentOS platforms and requires sudo rights. Please read the full documentation at
http://www.eso.org/sci/software/pipelines/installation/rpm.html, which also describes the versions of Fedora and CentOS that are currently supported.

3.3 Installing `Esoreflex` workflows via `install_esoreflex`

This method is recommended for operating systems other than those indicated above, or if the user has no sudo rights. Software dependencies are not fulfilled by the installation script, therefore the user has to install all the prerequisites before running the installation script.

The software pre-requisites for `Reflex 2.11.5` may be found at:
http://www.eso.org/sci/software/pipelines/reflex_workflows

To install the `Reflex 2.11.5` software and demo data, please follow these instructions:

1. From any directory, download the installation script:

```
wget https://eso.org/sci/software/pipelines/install_esoreflex
```

2. Make the installation script executable:

```
chmod u+x install_esoreflex
```

3. Execute the installation script:

```
./install_esoreflex
```

   and the script will ask you to specify three directories: the download directory `<download_dir>`, the software installation directory `<install_dir>`, and the directory to be used to store the demo data `<data_dir>`. If you do not specify these directories, then the installation script will create them in the current directory with default names.

4. Follow all the script instructions; you will be asked whether to use your Internet connection (recommended: yes), the pipelines and demo-datasets to install (note that the installation will remove all previously installed pipelines that are found in the same installation directory).

5. To start `Reflex`, issue the command:

```
<install_dir>/bin/esoreflex
```

   It may also be desirable to set up an alias command for starting the `Reflex` software, using the shell command `alias`. Alternatively, the `PATH` variable can be updated to contain the `<install_dir>/bin` directory.

4 Demo Data

Together with the pipeline you will also receive a demo data set, that allows you to run the Reflex eris_nix_img workflow without any changes in parameters. This way you have a data set to verify the installation and to experiment with before you start to work on your own data. The demo data for ERIS-NIX includes example data for the workflow. Note that you will need a minimum of 3.5 GB, 1.5 GB and 4.4 GB of free disk space for the directories `<download_dir>`, `<install_dir>` and `<data_dir>`, respectively. The ERIS-NIX demo data have been retrieved with the CalSelector tool 2 .

5 Quick Start: Reducing The Demo Data

For the user who is keen on starting reductions without being distracted by detailed documentation, we describe the steps to be performed to reduce the science data provided in the ERIS demo data set supplied with the `esoreflex 2.11.5` release. By following these steps, the user should have enough information to perform a reduction of his/her own data without any further reading:

1. First, type:

   ```
   esoreflex -l
   ```

   If the `esoreflex` executable is not in your path, then you have to provide the command with the executable full path `<install_dir>/bin/esoreflex -l` . For convenience, we will drop the reference to `<install_dir>`. A list with the available `esoreflex` workflows will appear, showing the workflow names and their full path.

2. Open the ERIS NIX by typing:

   ```
   esoreflex eris_nix_img&
   ```

   Alternatively, you can type only the command `esoreflex` the empty canvas will appear (Figure 5.1) and you can select the workflow to open by clicking on `File -> Open File`. Note that the loaded workflow will appear in a new window. The ERIS NIX workflow is shown in Figure 5.2.

3. To aid in the visual tracking of the reduction cascade, it is advisable to use component (or actor) highlighting. Click on `Tools -> Animate at Runtime`, enter the number of milliseconds representing the animation interval (100 ms is recommended), and click `OK`.

4. Change directories set-up. Under “Setup Directories” in the workflow canvas there are seven parameters that specify important directories (green dots).

   By default, the `ROOT_DATA_DIR`, which specifies the working directory within which the other directories are organised. is set to your `$HOME/reflex_data` directory. All the temporary and final products of the reduction will be organized under sub-directories of `ROOT_DATA_DIR`, therefore make sure this parameter points to a location where there is enough disk space. To change `ROOT_DATA_DIR`, double click on it and a pop-up window will appear allowing you to modify the directory string, which you may either edit directly, or use the `Browse` button to select the directory from a file browser. When you have finished, click `OK` to save your changes.

   Changing the value of `RAW_DATA_DIR` is the only necessary modification if you want to process data other than the demo data

5. Click the  button to start the workflow

6. The workflow will highlight the `Data Organiser` actor which recursively scans the raw data directory (specified by the parameter `RAW_DATA_DIR` under “Setup Directories” in the workflow canvas) and constructs the datasets. Note that the raw and static calibration data must be present either in

RAW_DATA_DIR or in CALIB_DATA_DIR, otherwise datasets may be incomplete and cannot be processed. However, if the same reference file was downloaded twice to different places this creates a problem as esoreflex cannot decide which one to use.

7. The Data Set Chooser actor will be highlighted next and will display a "Select Datasets" window (see Figure 9.1) that lists the datasets along with the values of a selection of useful header keywords[2]. The first column consists of a set of tick boxes which allow the user to select the datasets to be processed. By default all complete datasets which have not yet been reduced will be selected. A full description of the options offered by the Data Set Chooser will be presented in Section 7.2.2.

8. Click the Continue button and watch the progress of the workflow by following the red highlighting of the actors. A window will show which dataset is currently being processed.

9. Once the reduction of all datasets has finished, a pop-up window called *Product Explorer* will appear, showing the datasets which have been reduced together with the list of final products. This actor allows the user to inspect the final data products, as well as to search and inspect the input data used to create any of the products of the workflow. Figure 9.16 shows the Product Explorer window. A full description of the *Product Explorer* will be presented in Section 7.2.3.

10. After the workflow has finished, all the products from all the datasets can be found in a directory under END_PRODUCTS_DIR named after the workflow start timestamp. Further subdirectories will be found with the name of each dataset.

Well done! You have successfully completed the quick start section and you should be able to use this knowledge to reduce your own data. However, there are many interesting features of Reflex and the ERIS workflow that merit a look at the rest of this tutorial.

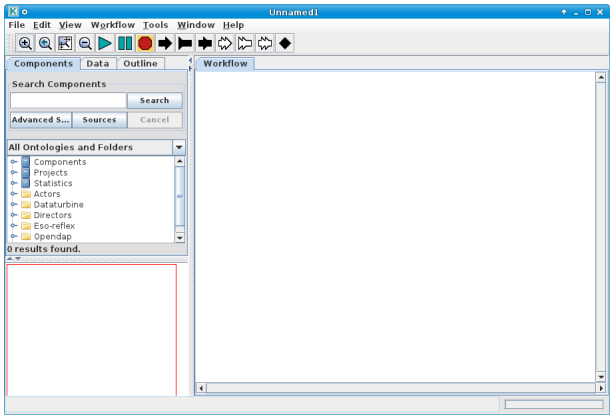


Figure 5.1: *The empty* Reflex *canvas.*

[2]The keywords listed can be changed by double clicking on the DataOrganiser Actor and editing the list of keywords in the second line of the pop-up window. Alternatively, instead of double-clicking, you can press the right mouse button on the DataOrganiser Actor and select Configure Actor to visualize the pop-up window.

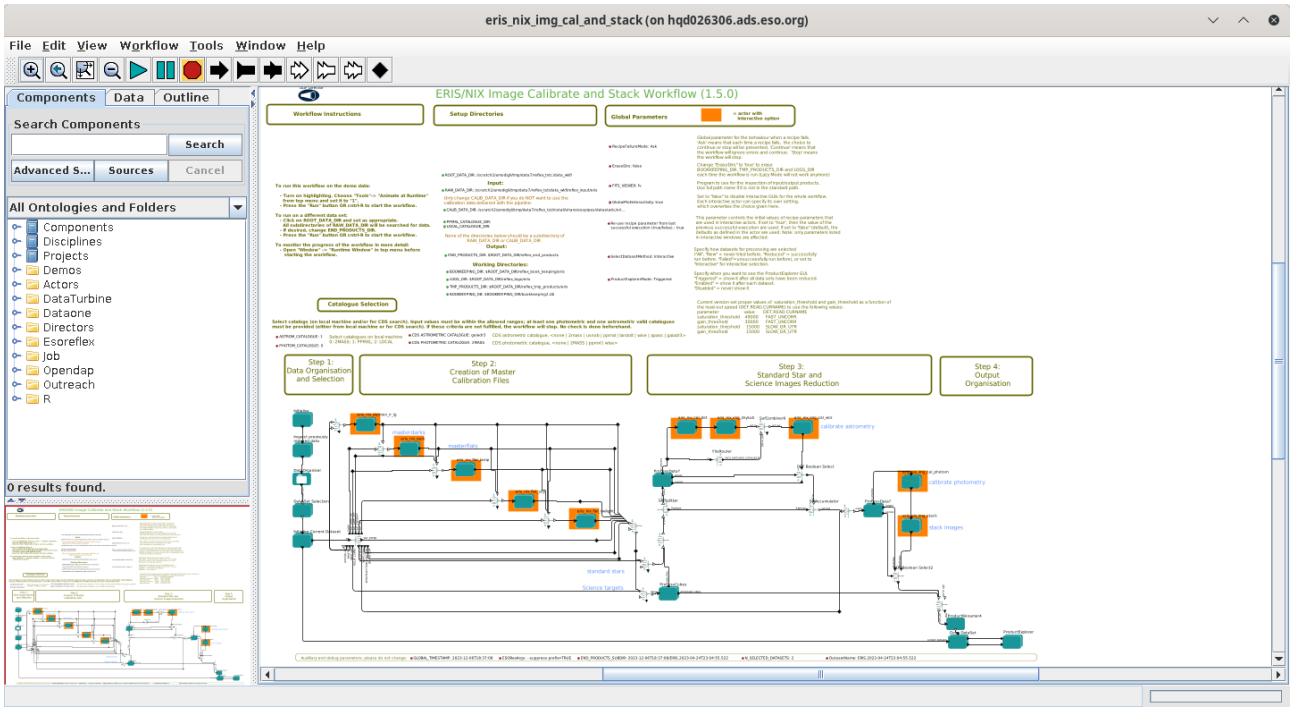


Figure 5.2: *ERIS NIX eris_nix_img workflow general layout.*

6 About the main `esoreflex` canvas

6.1 Saving And Loading Workflows

In the course of your data reductions, it is likely that you will customise the workflow for various data sets, even if this simply consists of editing the `ROOT_DATA_DIR` to a different value for each data set. Whenever you modify a workflow in any way, you have the option of saving the modified version to an `XML` file using `File -> Export As` (which will also open a new workflow canvas corresponding to the saved file). The saved workflow may be opened in subsequent `esoreflex` sessions using `File -> Open`. Saving the workflow in the default Kepler format (.kar) is only advised if you do not plan to use the workflow with another computer.

6.2 Buttons

At the top of the `esoreflex` canvas are a set of buttons which have the following functions:

-  - Zoom in.
-  - Reset the zoom to 100%.
-  - Zoom the workflow to fit the current window size (Recommended).
-  - Zoom out.
-  - Run (or resume) the workflow.
-  - Pause the workflow execution.
-  - Stop the workflow execution.

The remainder of the buttons (not shown here) are not relevant to the workflow execution.

6.3 Workflow States

A workflow may only be in one of three states: executing, paused, or stopped. These states are indicated by the yellow highlighting of the , , and  buttons, respectively. A workflow is executed by clicking the  button. Subsequently the workflow and any running pipeline recipe may be stopped immediately by clicking the  button, or the workflow may be paused by clicking the  button which will allow the current actor/recipe to finish execution before the workflow is actually paused. After pausing, the workflow may be resumed by clicking the  button again.

7 The ERIS Workflow

The ERIS workflow canvas is organised into a number of areas. From top-left to top-right you will find general workflow instructions, directory parameters, and global parameters. In the middle row you will find five boxes describing the workflow general processing steps in order from left to right, and below this the workflow actors themselves are organised following the workflow general steps.

7.1 Workflow Canvas Parameters

The workflow canvas displays a number of parameters that may be set by the user. Under "Setup Directories" the user is only required to set the `RAW_DATA_DIR` to the working directory for the dataset(s) to be reduced, which, by default, is set to the directory containing the demo data. The `RAW_DATA_DIR` is recursively scanned by the `Data Organiser` actor for input raw data. The directory `CALIB_DATA_DIR`, which is by default within the pipeline installation directory, is also scanned by the `Data Organiser` actor to find any static calibrations that may be missing in your dataset(s). If required, the user may edit the directories `BOOKKEEPING_DIR`, `LOGS_DIR`, `TMP_PRODUCTS_DIR`, and `END_PRODUCTS_DIR`, which correspond to the directories where book-keeping files, logs, temporary products and end products are stored, respectively (see the Reflex User Manual for further details; [**?**]).

There is a mode of the `Data Organiser` that skips the built-in data organisation and uses instead the data organisation provided by the CalSelector tool. To use this mode, click on `Use CalSelector associations` in the `Data Organiser` properties and make sure that the input data directory contains the XML file downloaded with the CalSelector archive request (note that this does not work for all instrument workflows).

Under the "Global Parameters" area of the workflow canvas, the user may set the `FITS_VIEWER` parameter to the command used for running his/her favourite application for inspecting FITS files. Currently this is set by default to `fv`, but other applications, such as `ds9`, `skycat` and `gaia` for example, may be useful for inspecting image data. Note that it is recommended to specify the full path to the visualization application (an alias will not work).

By default the `EraseDirs` parameter is set to `false`, which means that no directories are cleaned before executing the workflow, and the recipe actors will work in Lazy Mode (see Section 7.2.4), reusing the previous pipeline recipe outputs if input files and parameters are the same as for the previous execution, which saves considerable processing time. Sometimes it is desirable to set the `EraseDirs` parameter to `true`, which forces the workflow to recursively delete the contents of the directories specified by `BOOKKEEPING_DIR`, `LOGS_DIR`, and `TMP_PRODUCTS_DIR`. This is useful for keeping disk space usage to a minimum and will force the workflow to fully re-reduce the data each time the workflow is run.

The parameter `RecipeFailureMode` controls the behaviour in case that a recipe fails. If set to `Continue`, the workflow will trigger the next recipes as usual, but without the output of the failing recipe, which in most of the cases will lead to further failures of other recipes without the user actually being aware of it. This mode might be useful for unattended processing of large number of datasets. If set to `Ask`, a pop-up window will ask whether the workflow should stop or continue. This is the default. Alternatively, the `Stop` mode will stop the workflow execution immediately.

The parameter `ProductExplorerMode` controls whether the `ProductExplorer` actor will show its window or not. The possible values are `Enabled`, `Triggered`, and `Disabled`. `Enabled` opens the Produc-

tExplorer GUI at the end of the reduction of each individual dataset. `Triggered` (default and recommended) opens the ProductExplorer GUI when all the selected datasets have been reduced. `Disabled` does not display the ProductExplorer GUI.

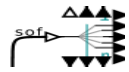
7.2 Workflow Actors

7.2.1 Simple Actors

Simple actors have workflow symbols that consist of a single (rather than multiple) green-blue rectangle. They may also have an icon within the rectangle to aid in their identification. The following actors are simple actors:

- 

- The `DataOrganiser` actor.
- 

- The `DataSetChooser` actor (inside a composite actor).
- 

- The `FitsRouter` actor Redirects files according to their categories.
- 

- The `ProductRenamer` actor.
- 

- The `ProductExplorer` actor (inside a composite actor).

Access to the parameters for a simple actor is achieved by right-clicking on the actor and selecting `Configure Actor`. This will open an “Edit parameters” window. Note that the `Product Renamer` actor is a jython script (Java implementation of the Python interpreter) meant to be customised by the user (by double-clicking on it).

7.2.2 DataSetChooser

The `DataSetChooser` displays the DataSets available in the “Select Data Sets” window, activating vertical and horizontal scroll bars if necessary (Fig. 9.1).

Some properties of the DataSets are displayed: the name, the number of files, a flag indicating if it has been successfully reduced (a green OK), if the reduction attempts have failed or were aborted (a red FAILED), or if it is a new dataset (a black "-"). The column "Descriptions" lists user-provided descriptions (see below), other columns indicate the instrument set-up and a link to the night log.

Sometimes you will want to reduce a subset of these DataSets rather than all DataSets, and for this you may individually select (or de-select) DataSets for processing using the tick boxes in the first column, and the buttons `Deselect All` and `Select Complete` at the bottom, or configure the “Filter” field at the bottom

left. Available filter options are: "New" (datasets not previously reduced will be selected), "Reduced" (datasets previously reduced will be selected), "All" (all datasets will be selected), and "Failed" (dataset with a failed or aborted reduction will be selected).

You may also highlight a single DataSet in blue by clicking on the relevant line. If you subsequently click on `Inspect Highlighted`, then a “Select Frames” window will appear that lists the set of files that make up the highlighted DataSet including the full filename[3], the file category (derived from the FITS header), and a selection tick box in the right column. The tick boxes allow you to edit the set of files in the DataSet which is useful if it is known that a certain calibration frame is of poor quality (e.g: a poor raw flat-field frame). The list of files in the DataSet may also be saved to disk as an `ASCII` file by clicking on `Save As` and using the file browser that appears.

By clicking on the line corresponding to a particular file in the “Select Frames” window, the file will be highlighted in blue, and the file FITS header will be displayed in the text box on the right, allowing a quick inspection of useful header keywords. If you then click on `Inspect`, the workflow will open the file in the selected FITS viewer application defined by the workflow parameter `FITS_VIEWER`.

To exit from the “Select Frames” window, click `Continue`.

To add a description of the reduction, press the button `...` associated with the field "Add description to the current execution of the workflow" at the bottom right of the Select Dataset Window; a pop-up window will appear. Enter the desired description (e.g. "My first reduction attempt") and then press `OK`. In this way, all the datasets reduced in this execution, will be flagged with the input description. Description flags can be visualized in the `SelectFrames` window and in the `ProductExplorer`, and they can be used to identify different reduction strategies.

To exit from the “Select DataSets” window, click either `Continue` in order to continue with the workflow reduction, or `Stop` in order to stop the workflow.

7.2.3 The ProductExplorer

The ProductExplorer is an interactive component in the `esoreflex` workflow whose main purpose is to list the final products with the associated reduction tree for each dataset and for each reduction attempt (see Fig. 9.16).

Configuring the ProductExplorer

You can configure the ProductExplorer GUI to appear after or before the data reduction. In the latter case you can inspect products as reduction goes on.

1. To display the ProductExplorer GUI at the end of the datareduction:

- Click on the global parameter “ProductExplorerMode” before starting the data reduction. A configuration window will appear allowing you to set the execution mode of the Product Explorer. Valid options are:
  - "Triggered" (default). This option opens the ProductExplorer GUI when all the selected datasets have been reduced.

[3] keep the mouse pointer on the file name to visualize the full path name.

  - "Enabled". This option opens the ProductExplorer GUI at the end of the reduction of each individual dataset.
  - “Disable”. This option does not display the ProductExplorer GUI.

- Press the  button to start the workflow.

2. To display the ProductExplorer GUI “before” starting the data reduction:

- double click on the composite Actor "Inspect previously reduced data". A configuration window will appear. Set to "Yes" the field "Inspect previously reduced data (Yes/No)". Modify the field "Continue reduction after having inspected the previously reduced data? (Continue/Stop/Ask)". "Continue" will continue the workflow and trigger the DataOrganizer. "Stop" will stop the workflow; "Ask" will prompt another window deferring the decision whether continuing or not the reduction after having closed the Product Explorer.
- Press the  button to start the workflow. Now the ProductExplorer GUI will appear before starting the data organization and reduction.

Exploring the data reduction products

The left window of the ProductExplorer GUI shows the executions for all the datasets (see Fig. 9.16). Once you click on a dataset, you get the list of reduction attemps. Green and red flags identify successfull or unsucessfull reductions. Each reduction is linked to the “Description” tag assigned in the “Select Dataset” window.

1. To identify the desired reduction run via the “Description” tag, proceed as follows:

- Click on the symbol at the left of the dataset name. The full list of reduction attempts for that dataset will be listed. The column Exec indicates if the reduction was succesful (green flag: "OK") or not (red flag: "Failed").
- Click on the entries in the field "Description" to visualize the description you have entered associated to that dataset on the Select Dataset window when reducing the data.
- Identify the desired reduction run. All the products are listed in the central window, and they are organized following the data reduction cascade.

You can narrow down the range of datasets to search by configuring the field "Show" at the top-left side of the ProductExplorer (options are: "All", "Successful", "Unsuccessful"), and specifying the time range (Last, all, From-to).

2. To inspect the desired file, proceed as follows:

- Navigate through the data reduction cascade in the ProductExplorer by clicking on the files.
- Select the file to be inspected and click with the mouse right-hand button. The available options are:

- Options available always:
  - Copy full path. It copies the full name of the file onto the clipboard. Shift+Ctr+v to past it into a terminal.
  - Inspect Generic. It opens the file with the FITS viewer selected in the main workflow canvas.
  - Inspect with. It opens the file with an executable that can be specified (you have to provide the full path to the executable).
- Options available for files in the `TMP_PRODUCTS_DIR` directory only:
  - command line. Copy of the environment configuration and recipe call used to generate that file.
  - Xterm. It opens an Xterm at the directory containing the file.
- Options available for products associated to interactive windows only:
  - Display pipeline results. It opens the interactive windows associated to the recipe call that generated the file. Note that this is for visualization purposes only; the recipe parameters cannot be changed and the recipe cannot be re-run from this window.

7.2.4 Lazy Mode

By default, all `RecipeExecuter` actors in a pipeline workflow are "Lazy Mode" enabled. This means that when the workflow attempts to execute such an actor, the actor will check whether the relevant pipeline recipe has already been executed with the same input files and with the same recipe parameters. If this is the case, then the actor will not execute the pipeline recipe, and instead it will simply broadcast the previously generated products to the output port. The purpose of the Lazy Mode is therefore to minimise any reprocessing of data by avoiding data re-reduction where it is not necessary.

One should note that the actor's Lazy Mode depends on the contents of the directory specified by the parameter `BOOKKEEPING_DIR` and the relevant FITS file checksums. Any modification to the directory contents and/or the file checksums will cause the corresponding actor to run the pipeline recipe again when executed, thereby re-reducing the input data.

The re-reduction of data at each execution may sometimes be desirable. To force a re-reduction of data for any single `RecipeExecuter` actor in the workflow, right-click the actor, select `Configure Actor`, and uncheck the Lazy mode parameter tick-box in the "Edit parameters" window that is displayed. For many workflows the `RecipeExecuter` actors are actually found inside the composite actors in the top level workflow. To access such embedded `RecipeExecuter` actors you will first need to open the sub-workflow by right-clicking on the composite actor and then selecting `Open Actor`.

To force the re-reduction of all data in a workflow (i.e. to disable Lazy mode for the whole workflow), you must uncheck the Lazy mode for every single `RecipeExecuter` actor in the entire workflow. It is also possible to change the name of the bookkeeping directory, instead of modifying any of the Lazy mode parameters. This will also force a re-reduction of the given dataset(s). A new reduction will start (with the lazy mode still enabled), but the results of previous reduction will not be reused. Alternatively, if there is no need to keep any of the previously reduced data, one can simply set the `EraseDirs` parameter under the "Global Parameters" area of the workflow canvas to `true`. This will then remove all previous results that are stored in the bookkeeping, temporary, and log directories before processing the input data, in effect, starting a new clean data reduction and re-processing every input dataset. *Note: The option* `EraseDirs` = `true` *does not work in* `esoreflex` *version 2.9.x and causes the workflow to crash.*

8 Reducing your own data

In this section we describe how to reduce your own data set.

First, we suggest the reader to familiarize with the workflow by reducing the demo dataset first (Section 5), but it is not a requirement.

8.1 The `esoreflex` command

We list here some options associated to the `esoreflex` command. We recommend to try them to familiarize with the system. In the following, we assume the `esoreflex` executable is in your path; if not you have to provide the full path `<install_dir>/bin/esoreflex`

To see the available options of the `esoreflex` command type:

```
esoreflex -h
```

The output is the following.

```
-h | -help              print this help message and exit.
-v | -version           show installed Reflex version and pipelines and exit.
-l | -list-workflows    list available installed workflows and from
                        ~/KeplerData/workflows.
-n | -non-interactive   enable non-interactive features.
-e | -explore           run only the Product Explorer in this workflow
-p <workflow> | -list-parameters <workflow>
                        lists the available parameters for the given
                        workflow.
-config <file>          allows to specify a custom esoreflex.rc configuration
                        file.
-create-config <file>   if <file> is TRUE then a new configuration file is
                        created in ~/.esoreflex/esoreflex.rc. Alternatively
                        a configuration file name can be given to write to.
                        Any existing file is backed up to a file with a '.bak'
                        extension, or '.bakN' where N is an integer.
-debug                  prints the environment and actual Reflex launch
                        command used.
```

8.2 Launching the workflow

We list here the recommended way to reduce your own datasets. Steps 1 and 2 are optional and one can start from step 3.

1. Type: `esoreflex -n <parameters>` ERIS NIX to launch the workflow non interactively and reduce all the datasets with default parameters.

<parameters> allows you to specify the workflow parameters, such as the location of your raw data and the final destination of the products.

For example, type (in a single command line):

```
esoreflex -n
  -RAW_DATA_DIR /home/user/my_raw_data
  -ROOT_DATA_DIR /home/user/my_reduction
  -END_PRODUCTS_DIR $ROOT_DATA_DIR/reflex_end_products
  eris_nix_img
```

to reduce the complete datasets that are present in the directory /home/user/my_raw_data and that were not reduced before. Final products will be saved in /home/user/my_reduction/reflex_end_products, while book keeping, temporary products, and logs will be saved in sub-directories of /home/user/my_reduction/. If the reduction of a dataset fails, the reduction continues to the next dataset. It can take some time, depending on the number of datasets present in the input directory. For a full list of workflow parameters type esoreflex -p ERIS NIX. Note that this command lists only the parameters, but does not launch the workflow.

Once the reduction is completed, one can proceed with optimizing the results with the next steps.

2. Type:

   esoreflex -e eris_nix_img

   to launch the Product Explorer. The Product Explorer allows you to inspect the data products already reduced by the ERIS NIX esoreflex workflow. Only products associated with the workflow default bookkeeping database are shown. To visualize products associated to given bookkeeping database, pass the full path via the BOOKKEEPING_DB parameter:

   esoreflex -e BOOKKEEPING_DB <database_path> eris_nix_img

   to point the product explorer to a given <database_path>, e.g., /home/username/reflex/reflex_bookkeeping/test.db

   The Product Explorer allows you to inspect the products while the reduction is running. Press the button Refresh to update the content of the Product Explorer. This step can be launched in parallel to step 1.

   A full description of the Product Explorer will be given in Section 7.2.3

3. Type:

   esoreflex eris_nix_img &

   to launch the ERIS NIX esoreflex workflow. The ERIS NIX workflow window will appear (Fig. 5.2). Please configure the set-up directories ROOT_DATA_DIR, RAW_DATA_DIR, and other workflow parameters as needed. Just double-click on them, edit the content, and press OK. Remember to specify the same <database_path> as for the Product Explorer, if it has been opened at step #2, to synchronize the two processes.

4. (Recommended, but not mandatory) On the main esoreflex menu set Tools –> Animate at Runtime to 1 in order to highlight in red active actors during execution.

5. Press the button  to start the workflow. First, the workflow will highlight and execute the `Initialise` actor, which among other things will clear any previous reductions if required by the user (see Section 7.1). Secondly, if set, the workflow will open the Product Explorer, allowing the user to inspect previously reduced datasets (see Section 7.2.3 for how to configure this option).

9 Reducing and Calibrating Your Own Science Data with Reflex

9.1 Available Reflex workflows

The workflow eris_nix_img (shown in Fig. 5.2) allows to reduce NIX data. In case DET.FRAM.FORMAT=="cube" only the actor eris_nix_cal_det is executed. Note that on the Global Parameter section at the left of the top-level workflow canvas there is a gobal parameter “Re-use recipe parameters”, by default set to false, that controls the initial values of the parameters used in the interactive workflows. If set to true the values of the previous execution are used. This setting allows a consistent behaviour of the ESOReflex based data reduction. If the user would like to change parameters to optimise their value to a given data set has to set this parameter to false (default), and then play with the parameter values controlling the steps relevant for the given use case.

9.2 Specifying data directories and selecting files

To reduce your own science data, simply change the paths to the root (optional) and data directories. Under the root directory, Reflex will create sub-directories which will contain temporary and end products, as well as book keeping and log files. The data directory, normally also under the root directory, contains directories with your raw files downloaded from the ESO archive. The paths are defined at the top of the workflow window in the area labeled `Setup Directories`. Simply double click on the `RAW_DATA_DIR`, enter the path to your raw science directory and then start the workflow in the same way as you did for the tutorial demo data. In case the data sets listed in the first window created by the work flow (Fig. 9.1) are greyed out, calibration files are missing (hovering with the mouse over the grey file entry will give more details). You can click the entry and a GUI opens up showing the dependency tree of the science (or calibrator) file on calibrations (Fig. 9.1).

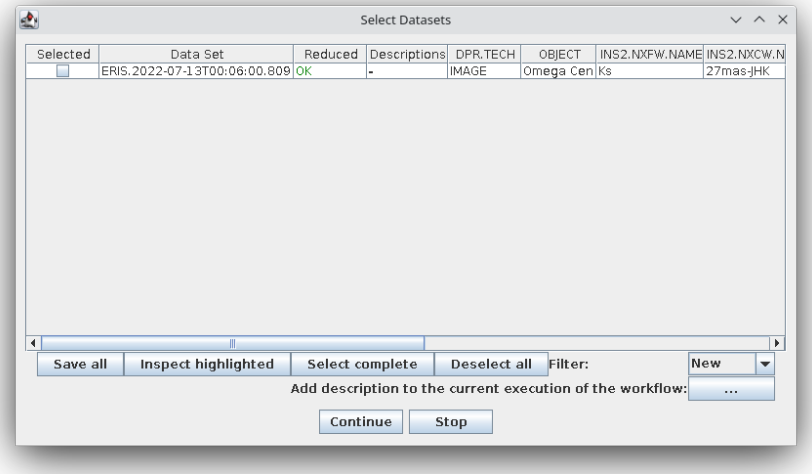


Figure 9.1: The “Select Datasets” pop-up window.

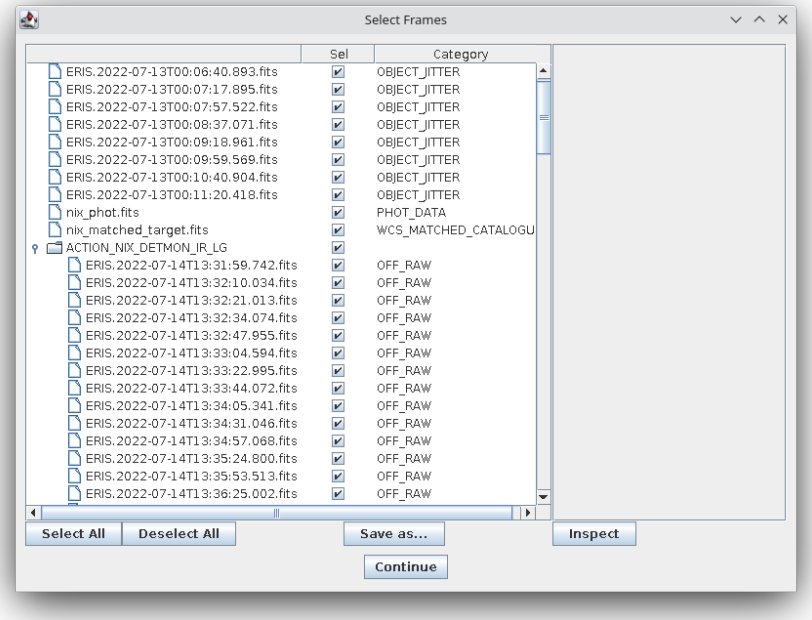


Figure 9.2: The “Select Frame” pop-up window, obtained after pressing the “Inspect highlighted” button in the “Select Datasets” window.

Reflex message	missing raw data type
Missing GAIN_TABLE	raw linearity frames
Missing COEFFS_CUBE	raw linearity frames
Missing BP_MAP_NL	raw linearity frames
Missing MASTER_DARK_IMG	raw dark frames
Missing MASTER_FLAT_LAMP_HIFREQ	raw flat lamp frames
Missing MASTER_FLAT_LAMP_LOFREQ	raw flat lamp frames
Missing MASTER_BPM_LAMP	raw flat lamp frames
Missing MASTER_FLAT_TWILIGHT_LOWFREQ	raw flat twilight frames
Missing MASTER_FLAT_SKY_HIFREQ	raw flat sky frames
Missing MASTER_FLAT_SKY_LOFREQ	raw flat sky frames
Missing MASTER_BPM_SKY	raw flat sky frames
Missing IMG_STD_COMBINED	raw std star frames (astrometry)

Table 9.1: This table reports the possible message indicated by the cursor when moved on a grey out dataset, and the corresponding missing raw data type, the user has to retrieve from the ESO archive to obtain a complete data set.

It may occur that the user misses some input data required to reduce a given dataset. In this case the affected dataset(s) are grey out. Moving the mouse over the grey datasets provides information about the missing data product to perform the chain (see Fig. 9.3). In such case the user should consult the pipeline user manual to identify which raw data allow to generate that data set and add to the input raw data set (of the appropriate instrument setting) to the missing data.

9.3 Detector Linearity

The pixel to pixel detector sensitivity maybe not linear over the usual illumination range. For this reason the pipeline implements a recipe, `eris_nix_detmon_ir_lg`, that analyses a set of flat frames acquired with different exposure times (and thus illumination levels), to measure the response of each pixels to different levels of input signal and fit a low order polynomial to identify in what intensity rande the detector starts to have a not linear response. A map of not linear pixels is created and displayed by the corresponding interactie reflex workflow. The reflex workflow set proper defaults of the two parameters `saturation_limit` and `gain_threshold` based on the DET.READ.CURNAME FITS keyword value, as they are very different in the two cases. The user may eventually refine those two good default faluues to improve further the quality of the resultin non linear bad pixel map.

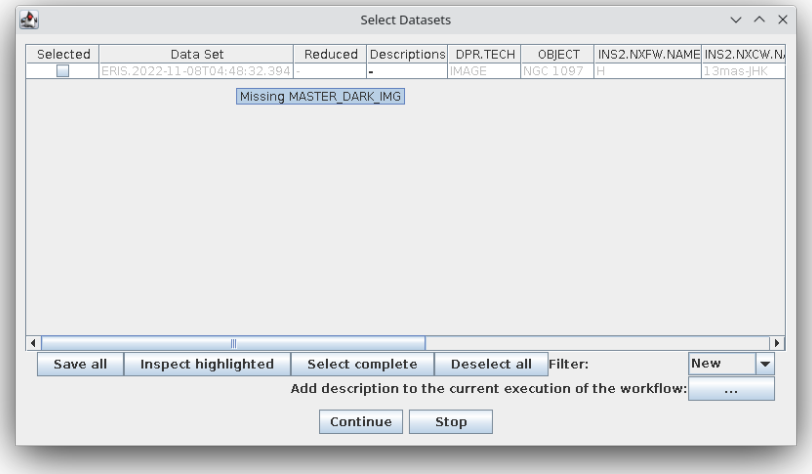


Figure 9.3: The "Select Datasets" pop-up window. If required input data are missing, the corresponding dataset is grey out.

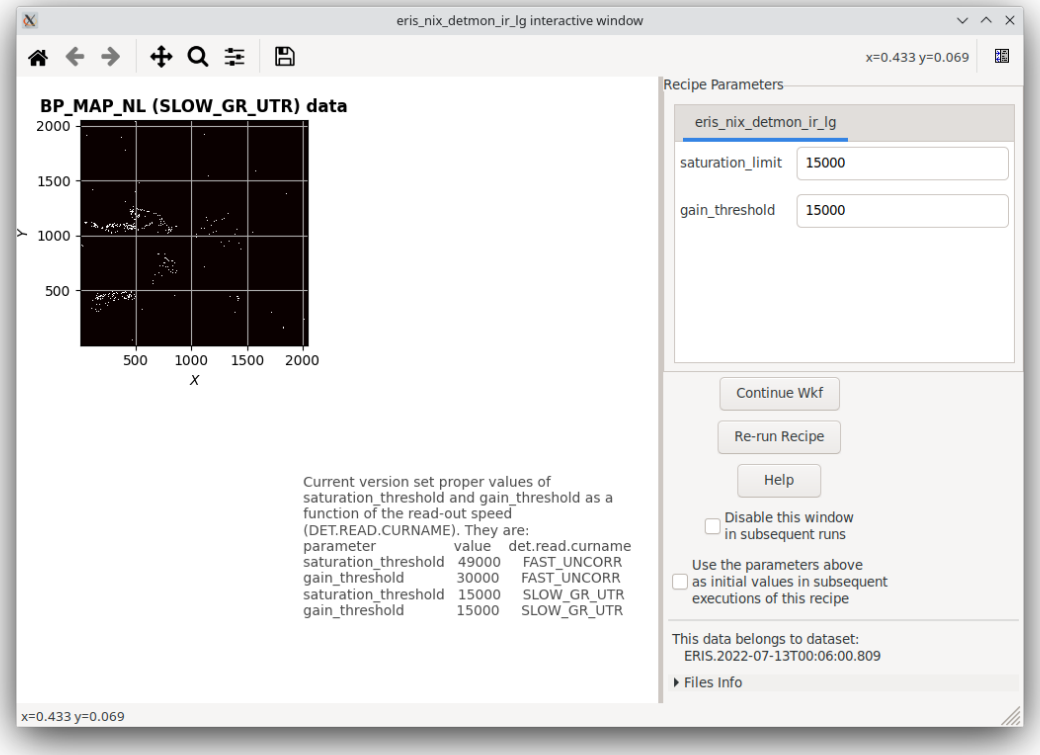


Figure 9.4: *The interactive window to edit parameters of recipe eris_nix_detmon_ir_lg.*

9.4 Master Dark Creation

The signal on the detector when the shutter is closed is called dark (current) level. To measure the dark current a set of usually five dark frames is acquired and stacked into a master frame to reduce the noise present in each frame. The eris_nix_img offers an interactive workflow to inspect and eventually optimise the main results of the master dark recipe.

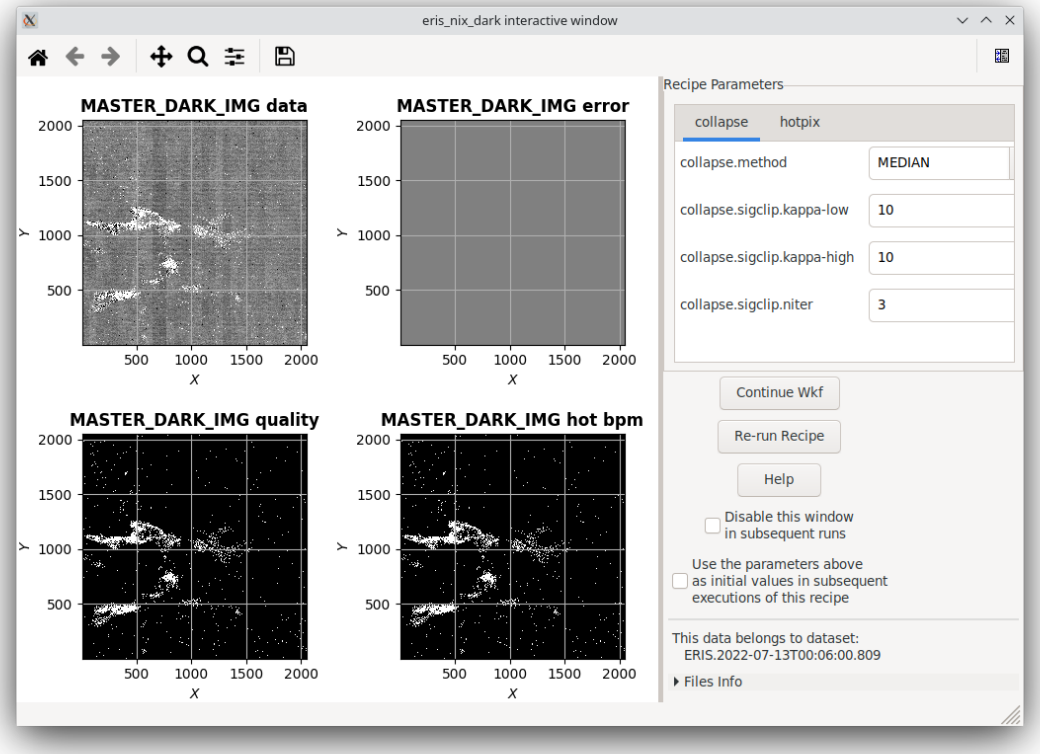


Figure 9.5: *The interactive window to edit parameters of recipe eris_nix_dark.*

For images are shown showing images of the data, the corresponding error, the pixel quality and the bad pixel map.

The user may modify a number of data reduction parameters, like parameters controlling the collapsing of the dark frames and the creation of the hot pixel map.

9.5 Master Flat Lamp

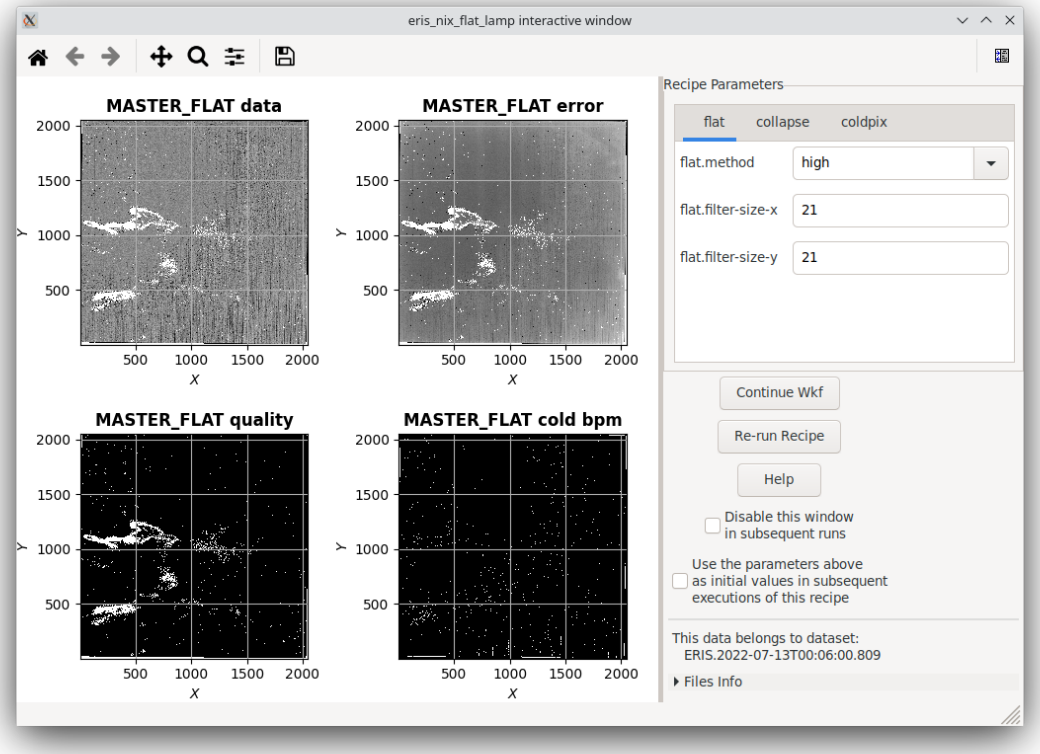


Figure 9.6: *The interactive window to edit parameters of recipe eris_nix_flat_lamp.*

Four images are shown showing images of the data, the corresponding error, the pixel quality and the bad pixel map.

The user may modify a number of data reduction parameters, like parameters controlling the flat creation, the collapsing of the flat frames and the creation of the cold pixel map.

9.6 Master Flat Twilight

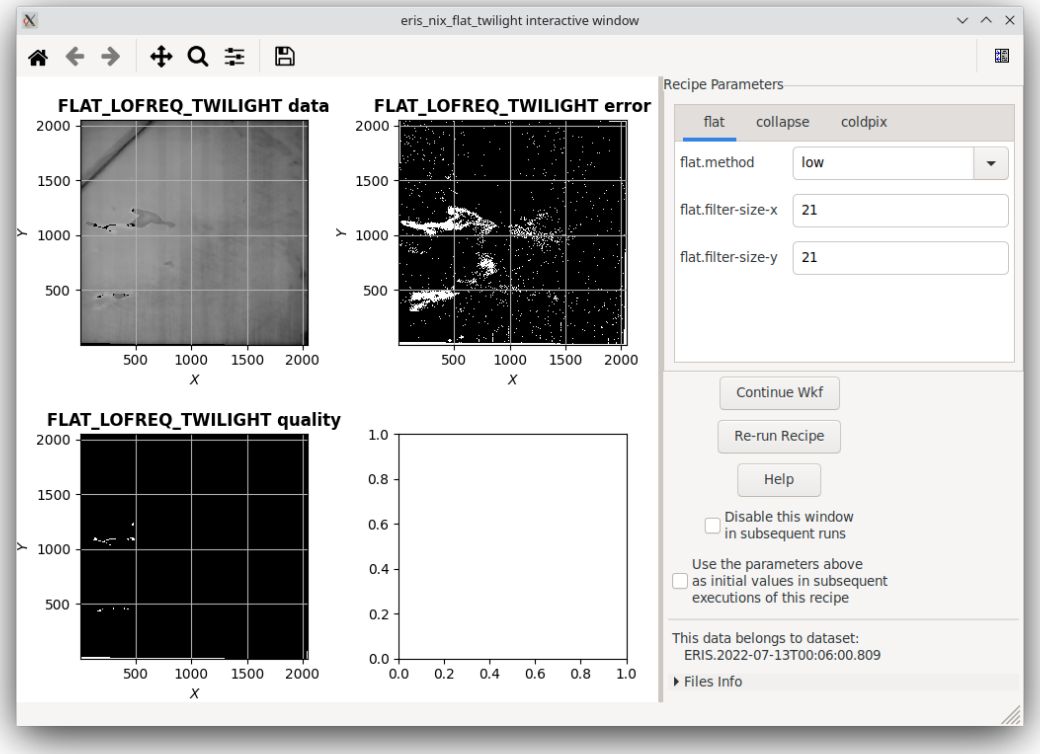


Figure 9.7: *The interactive window to edit parameters of recipe eris_nix_flat_twilight.*

Four images are shown showing images of the data, the corresponding error, the pixel quality and the bad pixel map.

The user may modify a number of data reduction parameters, like parameters controlling the flat creation, the collapsing of the flat frames and the creation of the cold pixel map.

9.7 Removal of instrument signature

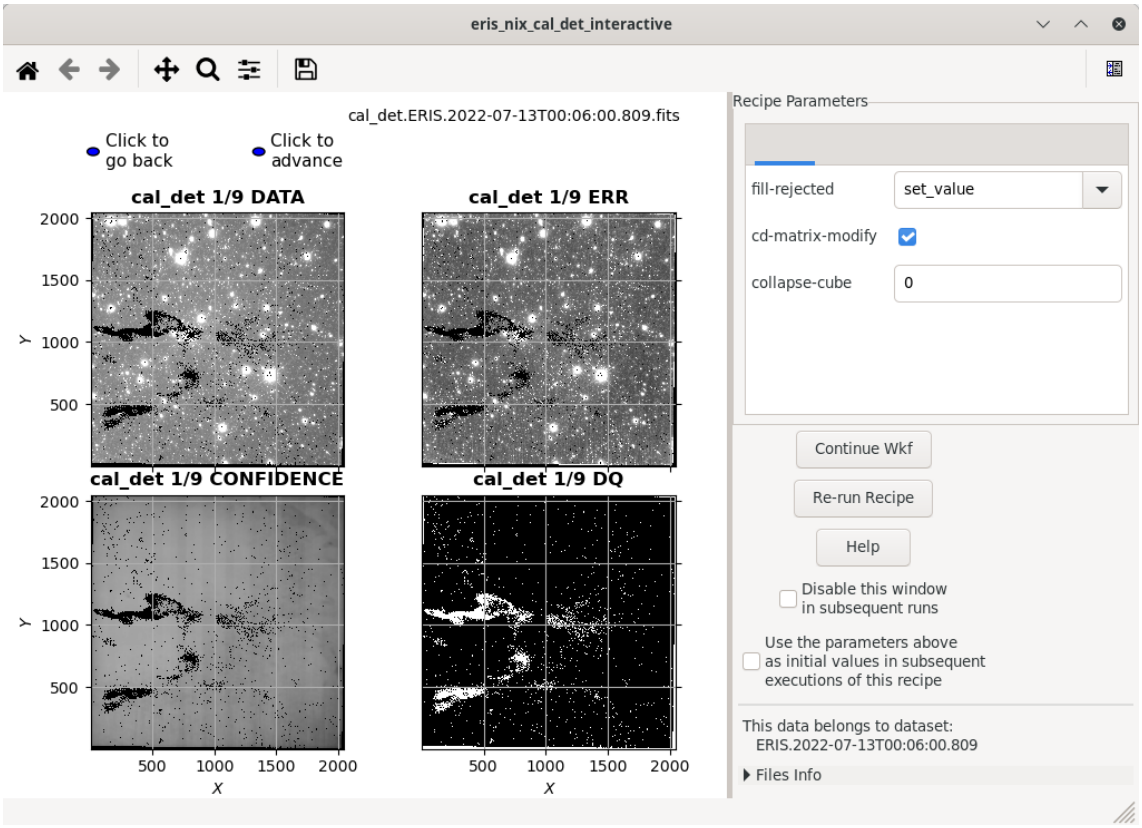


Figure 9.8: *The interactive window to edit parameters of recipe eris_nix_cal_det.*

The first step to reduce science frames is the removal of the instrument detector signature. This implies correction of the dark level, and division by the master flat to correct the small and long scale detector responsiveness variations.

The ESOReflex interactive GUI present to the users information on the data, the corresponding error, confidence map and data quality for each observation. The user may display the results for different data sets by clinking on the “Click to advance” or “Click to go back” buttons.

The user may also de/select the parameter “fill-rejected”. The **cd-matrix-modify** parameter default is true. In that case the pipeline corrects for small distortions introduced by the optics. The CD matrix geometry corrections coefficients have been calibrated with data obtained with Paosition Angle (PA) set to 0. If the user has observations obtained with PA different from 0, the astrometry calibration may be more accurate by setting **cd-matrix-modify** to false.

9.8 Sky background correction

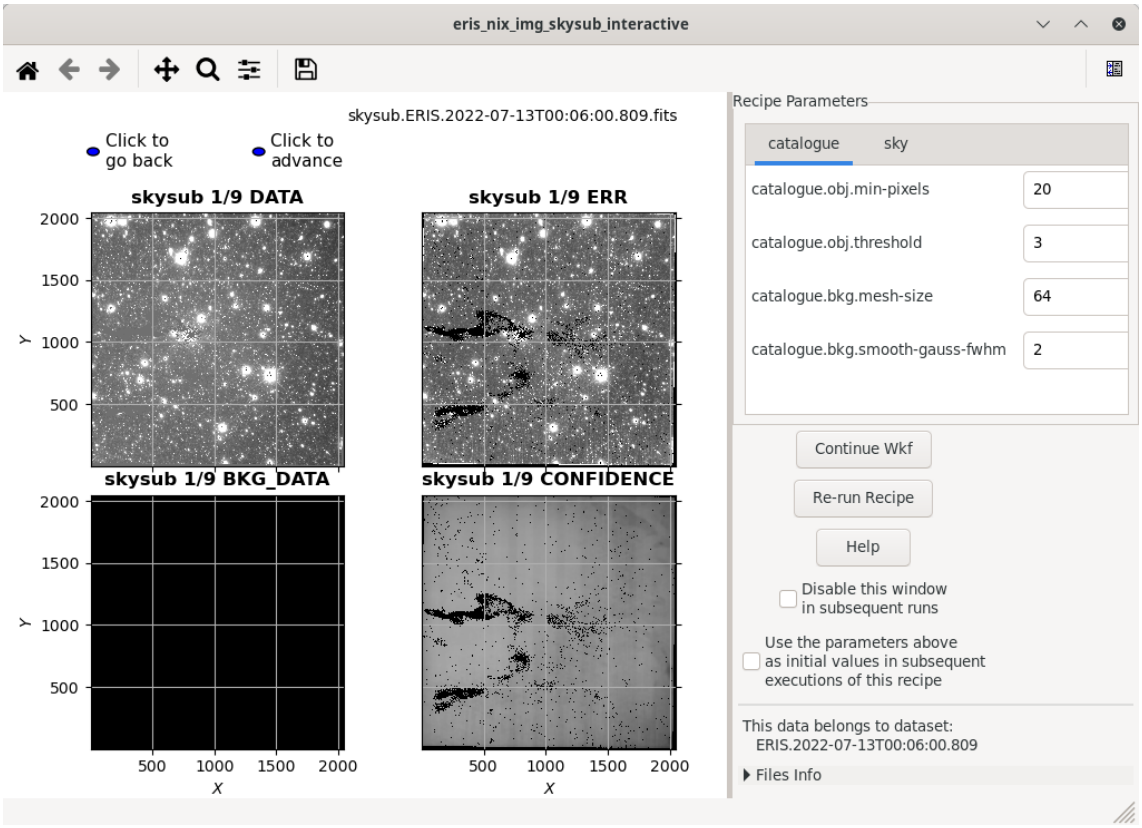


Figure 9.9: *The interactive window to edit parameters of recipe eris_nix_img_skysub.*

The second step to reduce science frames is the subtraction of the Sky background.

The ESOReflex interactive GUI present to the users information on the data, the corresponding error, the sky background image and the confidence map for each observation. The user may display the results for different data sets by clinking on the “Click to advance” or “Click to go back” buttons.

The user may also optimise parameters controlling the detection of the objects present in the instrument field of view and parameters that control the determination of the sky background image.

9.9 World Coordinate System Calibration

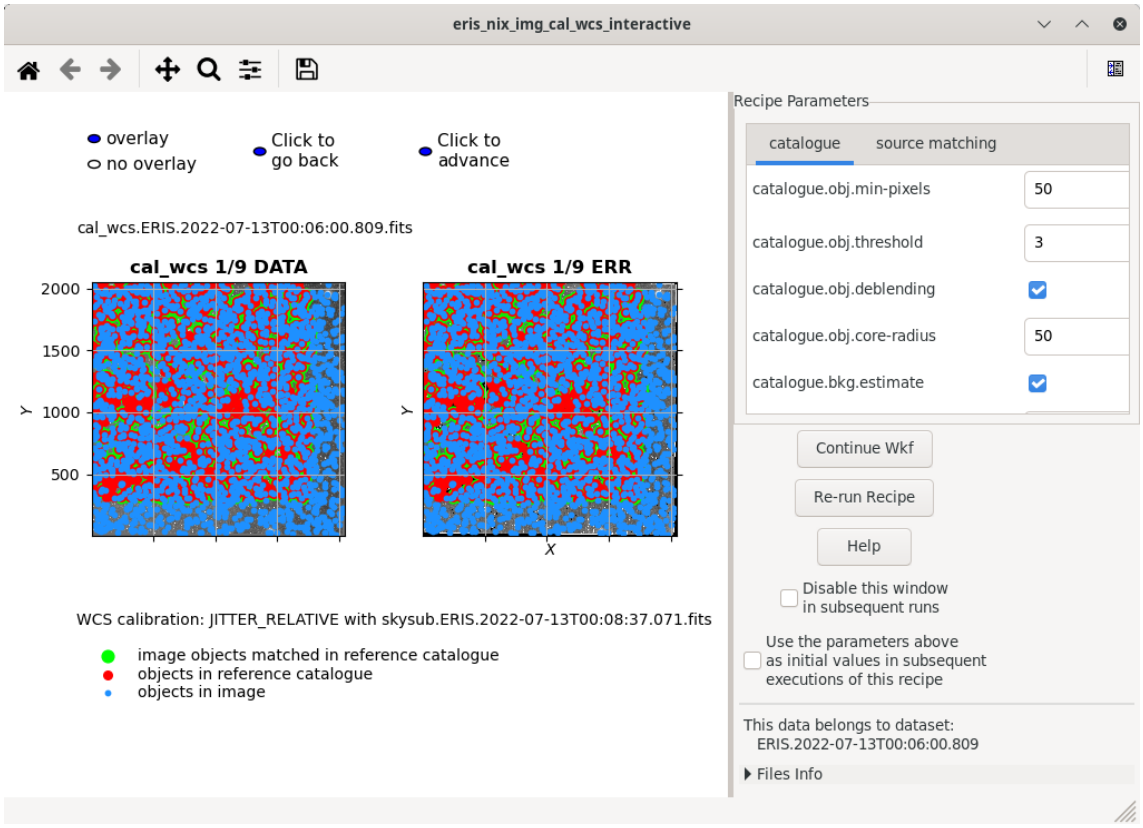


Figure 9.10: *The interactive window to edit parameters of recipe eris_nix_img_cal_wcs. Detected objects locations appear in overlay.*

The third step to reduce science frames is the determination of the World Coordinate System (WCS).

The ESOReflex interactive GUI present to the users information on the data and the corresponding error. The user may display the results for different data sets by clinking on the “Click to advance” or “Click to go back” buttons. The user may also display (default) or not the overlay of the detected objects on the images.

The user may also optimise parameters controlling the detection of the objects present in the instrument field of view and parameters that control the determination of the objects on the image. For example if the Field Of View contains a few sources with coordinates matching with the ones of the selected catalog and some are adjacent one to the other, we recommend the user to a smaller value of the parameter **catalogue.bkg.mesh-size**.

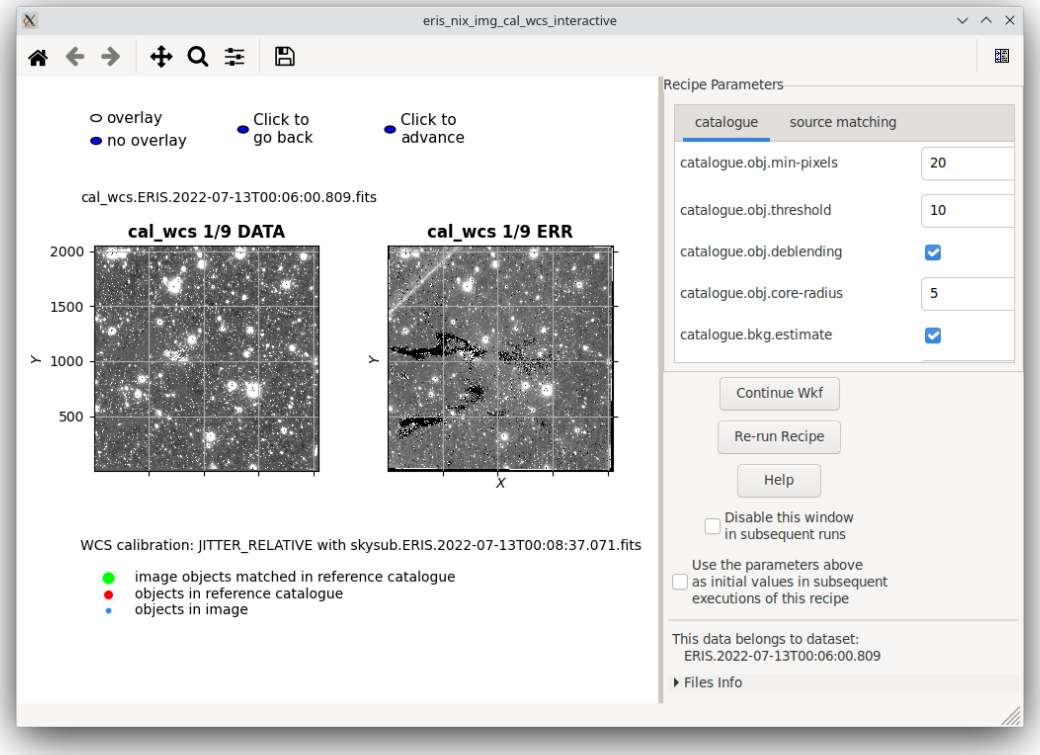


Figure 9.11: *The interactive window to edit parameters of recipe eris_nix_img_cal_wcs.*

9.10 Photometric Calibration

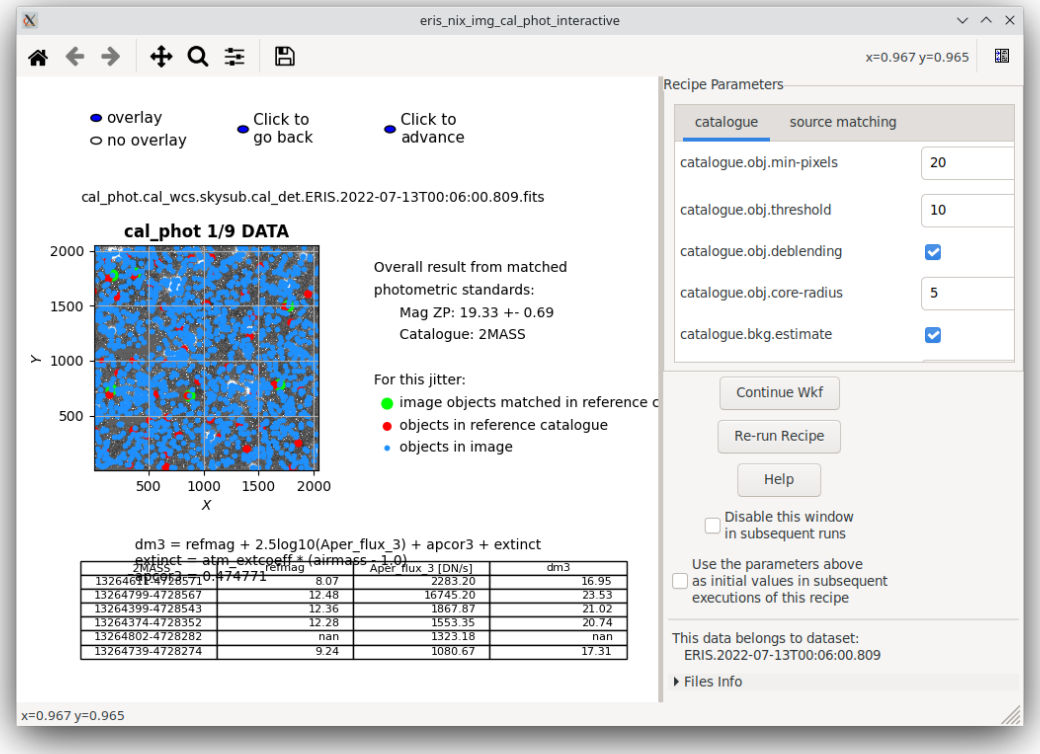


Figure 9.12: *The interactive window to edit parameters of recipe eris_nix_img_cal_phot. Detected objects locations appear in overlay.*

The fourth step to reduce science frames is the determination of the photometric calibration of the field of view.

The ESOReflex interactive GUI present to the users information on the data. The user may display the results for different data sets by clinking on the “Click to advance” or “Click to go back” buttons. The user may also display (default) or not the overlay of the detected objects on the images.

The GUI display the observed image Zero Point Magnitude with associated error and the name of the catalogue used to perform this calibration step.

The formulae used to compute dm3, the extinction are also indicated. Then a table with information on the objerved objects, its reference magnitute, corresponding aperture and value of dm3 is shown.

The user may also optimise parameters controlling the detection of the objects present in the instrument field of view and parameters that control the determination of the objects on the image.

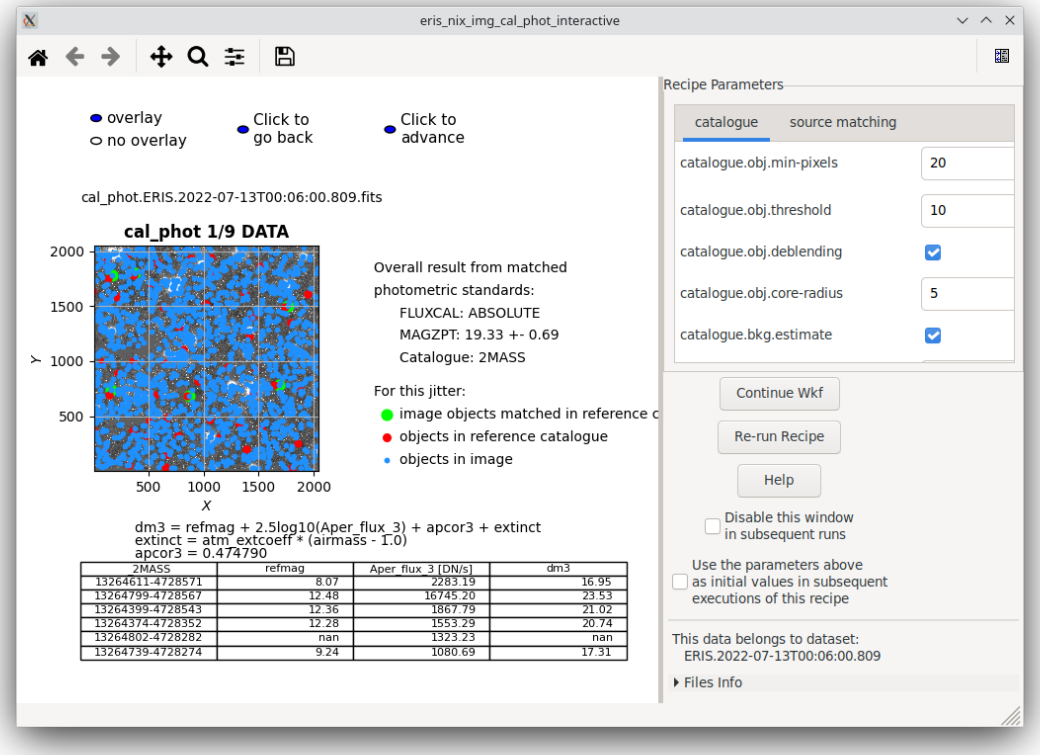


Figure 9.13: *The interactive window to edit parameters of recipe eris_nix_img_cal_phot.*

9.11 Frame Stacking

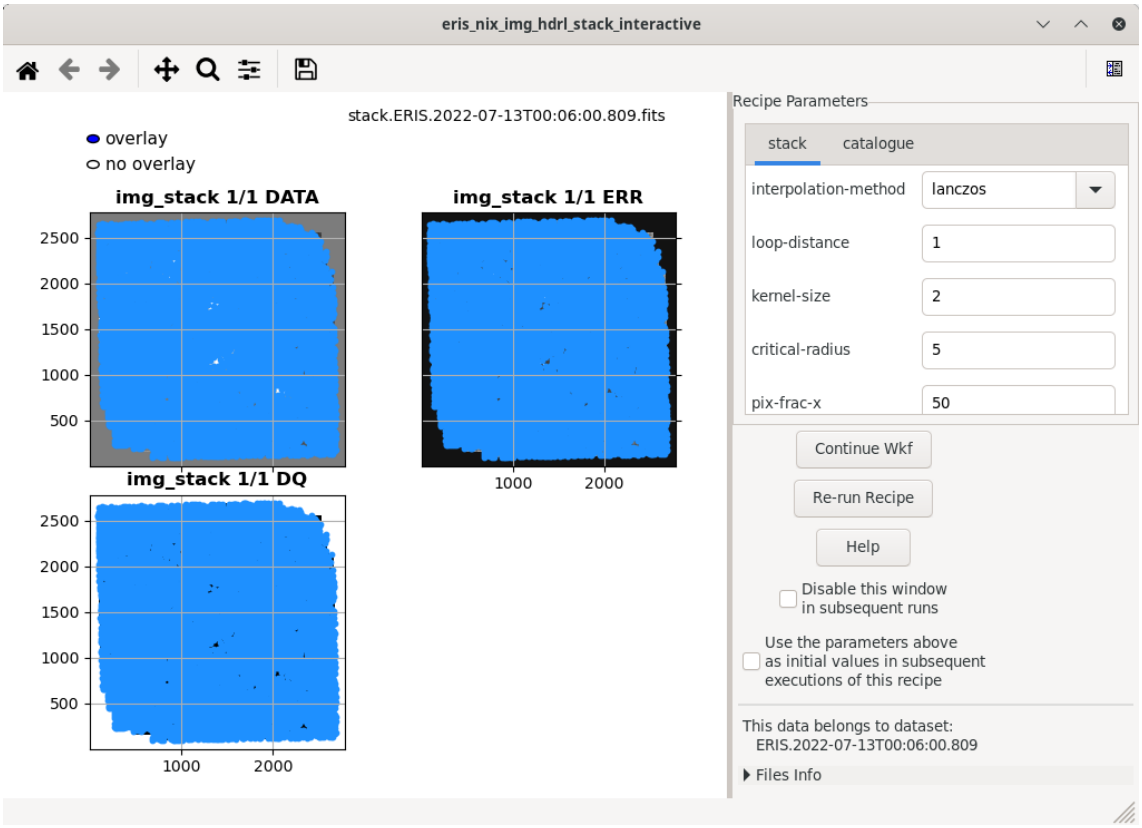


Figure 9.14: *The interactive window to edit parameters of recipe eris_nix_img_hdrl_stack. Detected objects locations appear in overlay.*

The last step to reduce science frames is the stacking of the previously calibrated frames to maximise the overall SNR.

The ESOReflex interactive GUI present to the users information on the data. The user may display the results for different data sets by clinking on the “Click to advance” or “Click to go back” buttons. The user may also display (default) or not the overlay of the detected objects on the images.

The user may also optimise parameters controlling the frame stacking and the detection of the objects present in the instrument field of view.

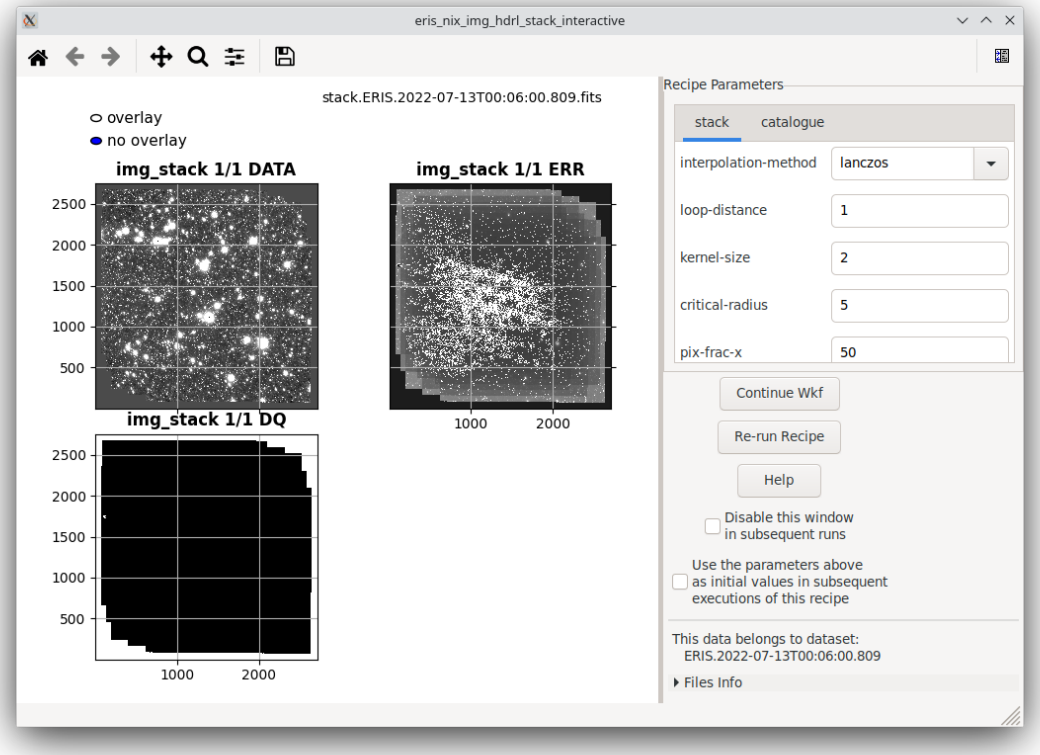


Figure 9.15: *The interactive window to edit parameters of recipe eris_nix_img_hdrl_stack.*

9.12 Examining the workflow results

When the workflow has finished, the Product Explorer window opens (Fig. 9.16). Select a data file and unfold the file tree in the “Provenance Tree” window. This provides information on the dependency of product files on the calibration files and other files from recipes executed before. You can inspect a data file by clicking the “Inspect with...” button, and entering the path to your favourite FITS file viewer (e.g., *fv*).

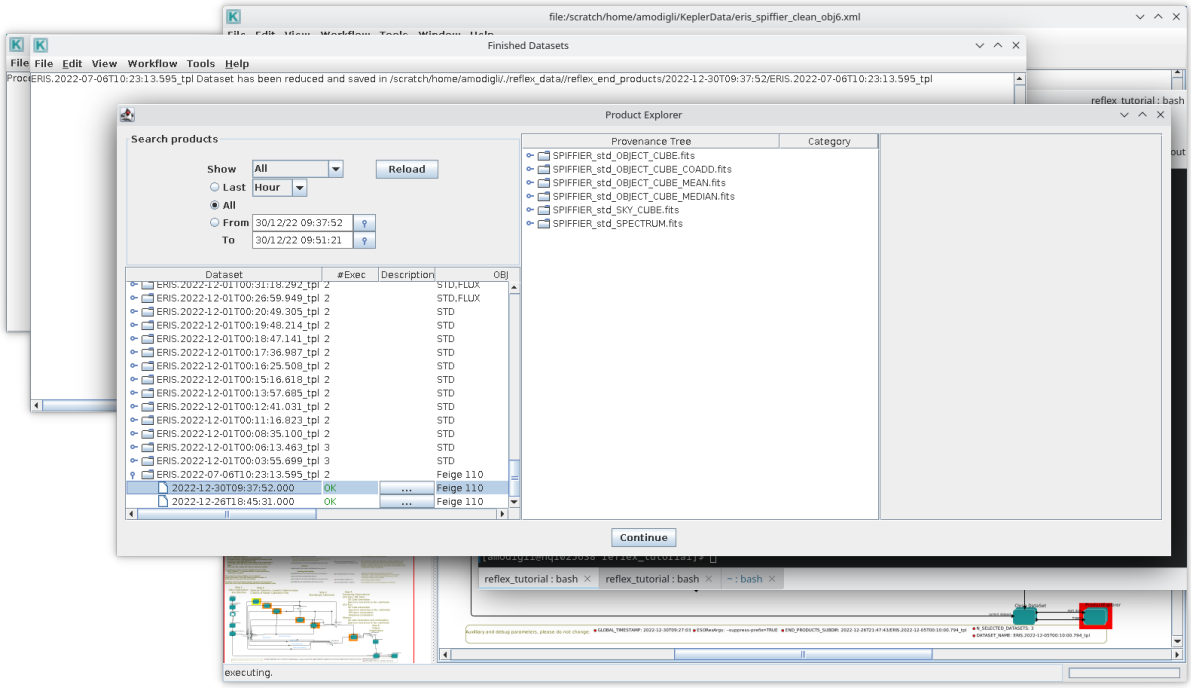


Figure 9.16: *The ERIS-SPIFFIER product explorer for the eris_nix_img workflow.*

10 Frequently Asked Questions

- **The error window fills the whole screen - how can I get to the `Continue`/`Stop` buttons?**

  Press the Alt key together with your left mouse button to move the window upwards and to the left. At the bottom the `Continue`/`Stop` buttons will be visible. This bug is known but could not yet be fixed.

- **I tried to `Open` (or `Configure`) an `Actor` while the workflow is running and now it does not react any more. What should I do?**

  This is a limitation of the underlying Kepler engine. The only way out is to kill the workflow externally. If you want to change anything while a workflow is running you first need to pause it.

- **After a successful reduction of a data set, I changed this data set in some way (e.g. modified or removed some files, or changed the rules of the Data Organizer). When I restart Reflex, the Data Set Chooser correctly displays my new data set, but marks it as "reduced ok", even though it was never reduced before. What does this mean?**

  The labels in the column "Reduced" of the Data Set Chooser mark each dataset with "OK", "Failed" or "-". These labels indicate whether a data set has previously successfully been reduced at least once, all previous reductions failed, or a reduction has never been tried respectively. Data sets are identified by their name, which is derived from the first science file within the data set. As long as the data set name is preserved (i.e. the first science file in a data set has not changed), the Data Organizer will consider it to be the same data set. The Data Organizer recognizes any previous reductions of data sets it considers to be the same as the current one, and labels the current data set with "OK" if any of them was successful, even if the previously reduced data set differs from the current one.

  Note that the Product Explorer will list all the previous reductions of a particular data set only at the end of the reduction. This list might include successful and/or unsuccessful reduction runs with different parameters, or in your case with different input files. The important fact is that these are all reductions of data sets with the same first raw science file. By browsing through all reductions of a particular raw science file, the users can choose the one they want to use.

- **Where are my intermediate pipeline products?** Intermediate pipeline products are stored in the directory `<TMP_PRODUCTS_DIR>` (defined on the workflow canvas, under Setup Directories) and organised further in directories by pipeline recipe.

- **Can I use different sets of bias frames to calibrate my flat frames and science data?** Yes. In fact this is what is currently implemented in the workflow(s). Each file in a DataSet has a purpose attached to it ([**?**]). It is this purpose that is used by the workflow to send the correct set of bias frames to the recipes for flat frame combination and science frame reduction, which may or may not be the same set of bias frames in each case.

- **Can I run `Reflex` from the command line?** Yes, use the command:

  ```
  esoreflex -n <workflow_path>/<workflow>.xml
  ```

  The -n option will set all the different options for Kepler and the workflows to avoid opening any GUI elements (including pipeline interactive windows).

It is possible to specify workflow variables (those that appear in the workflow canvas) in the command line. For instance, the raw data directory can be set with this command:

```
esoreflex -n -RAW_DATA_DIR <raw_data_path> \
             <workflow_path>/<workflow>.xml
```

You can see all the command line options with the command `esoreflex -h`.

Note that this mode is not fully supported, and the user should be aware that the path to the workflow must be absolute and even if no GUI elements are shown, it still requires a connection to the window manager.

- **How can I add new actors to an existing workflow?** You can drag and drop the actors in the menu on the left of the `Reflex` canvas. Under `Eso-reflex -> Workflow` you may find all the actors relevant for pipeline workflows, with the exception of the recipe executer. This actor must be manually instantiated using `Tools -> Instantiate Component`. Fill in the "Class name" field with `org.eso.RecipeExecuter` and in the pop-up window choose the required recipe from the pull-down menu. To connect the ports of the actor, click on the source port, holding down the left mouse button, and release the mouse button over the destination port. Please consult the Reflex User Manual ([**?**]) for more information.

- **How can I broadcast a result to different subsequent actors?** If the output port is a multi-port (filled in white), then you may have several relations from the port. However, if the port is a single port (filled in black), then you may use the black diamond from the toolbar. Make a relation from the output port to the diamond. Then make relations from the input ports to the diamond. Please note that you cannot click to start a relation from the diamond itself. Please consult the Reflex User Manual ([**?**]) for more information.

- **How can I manually run the recipes executed by Reflex?** If a user wants to re-run a recipe on the command line he/she has to go to the appropriate reflex_book_keeping directory, which is generally reflex_book_keeping/<workflow>/<recipe_name>_<number> There, subdirectories exist with the time stamp of the recipe execution (e.g. 2013-01-25T12:33:53.926/). If the user wants to re-execute the most recent processing he/she should go to the `latest` directory and then execute the script `cmdline.sh`. Alternatively, to use a customized `esorex` command the user can execute

  ```
  ESOREX_CONFIG="INSTALL_DIR/etc/esorex.rc"
  PATH_TO/esorex --recipe-config=<recipe>.rc <recipe> data.sof
  ```

  where INSTALL_DIR is the directory where Reflex and the pipelines were installed.

  If a user wants to re-execute on the command line a recipe that used a specific raw frame, the way to find the proper data.sof in the bookkeeping directory is via `grep <raw_file> */data.sof`. Afterwards the procedure is the same as before.

  If a recipe is re-executed with the command explained above, the products will appear in the directory from which the recipe is called, and not in the reflex_tmp_products or reflex_end_products directory, and they will not be renamed. This does not happen if you use the `cmdline.sh` script.

- **Can I reuse the bookkeeping directory created by previous versions of the pipeline?**

  In general no. In principle, it could be reused if no major changes were made to the pipeline. However there are situations in which a previously created bookkeeping directory will cause problems due to pipeline versions incompatibility. This is especially true if the parameters of the pipeline recipes have changed. In that case, please remove the bookkeeping directory completely.

- **How to insert negative values into a textbox?**

  Due to a bug in wxPython, the GUI might appear to freeze when attempting to enter a negative number in a parameter's value textbox. This can be worked around by navigating away to a different control in the GUI with a mouse click, and then navigating back to the original textbox. Once focus is back on the original textbox the contents should be selected and it should be possible to replace it with a valid value, by typing it in and pressing the enter key.

- **I've updated my Reflex installation and when I run esoreflex the process aborts. How can I fix this problem?**

  As indicated in Section 3, in case of major or minor (affecting the first two digit numbers) Reflex upgrades, the user should erase the `$HOME/KeplerData`, `$HOME/.kepler` directories if present, to prevent possible aborts (i.e. a hard crash) of the esoreflex process.

- **How can include my analysis scripts and algorithms into the workflow?**

  EsoReflex is capable of executing any user-provided script, if properly interfaced. The most convenient way to do it is through the Python actor. Please consult the tutorial on how to insert Python scripts into a workflow available here: www.eso.org/sci/data-processing/Python_and_esoreflex.pdf