



# EUROPEAN SOUTHERN OBSERVATORY

Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral

Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

## VERY LARGE TELESCOPE

### Reflex IIINSTRUMENT Tutorial

VLT-MAN-ESO-19500-....

Issue 0.1

Date 2025-06-16

Prepared: N. Surname 2025-06-16  
.....  
Name Date Signature

Approved: N. Surname  
.....  
Name Date Signature

Released: N. Surname  
.....  
Name Date Signature

This page was intentionally left blank

<b>ESO</b>	<b>Reflex IIINSTRUMENT Tutorial</b>	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	3 of 26

**Change record**

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
1.0	25/09/2012	All	Beta release

This page was intentionally left blank

<b>ESO</b>	<b>Reflex IIINSTRUMENT Tutorial</b>	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	5 of 26

## Contents

<b>1</b>	<b>Introduction to Esoflex</b>	<b>7</b>
<b>2</b>	<b>Software Installation</b>	<b>8</b>
2.1	Installing Esoflex workflows via macports . . . . .	8
2.2	Installing Esoflex workflows via rpm/yum/dnf . . . . .	8
2.3	Installing Esoflex workflows via install_esoflex . . . . .	8
<b>3</b>	<b>Demo Data</b>	<b>10</b>
<b>4</b>	<b>Quick Start: Reducing The Demo Data</b>	<b>11</b>
<b>5</b>	<b>About the main esoflex canvas</b>	<b>15</b>
5.1	Saving And Loading Workflows . . . . .	15
5.2	Buttons . . . . .	15
5.3	Workflow States . . . . .	15
<b>6</b>	<b>The 4MOST Workflow</b>	<b>16</b>
6.1	Workflow Canvas Parameters . . . . .	16
6.2	Workflow Actors . . . . .	17
6.2.1	Simple Actors . . . . .	17
6.2.2	Lazy Mode . . . . .	17
6.3	Workflow Steps . . . . .	18
6.3.1	Data Organisation And Selection . . . . .	18
6.3.2	DataSetChooser . . . . .	19
6.3.3	The ProductExplorer . . . . .	20
<b>7</b>	<b>Frequently Asked Questions</b>	<b>23</b>

<b>ESO</b>	<b>Reflex IIINSTRUMENT Tutorial</b>	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	6 of 26

ESO	Reflex IIINSTRUMENT Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	7 of 26

## 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 environmet, 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<sup>1</sup> (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.

Add here some explanation about this specific workflow. Instrument modes supported, etc...

---

<sup>1</sup><https://www.eso.org/sci/archive/calselectorInfo.html>

ESO	Reflex IIINSTRUMENT Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	8 of 26

## 2 Software Installation

Esoflex and the workflows can be installed in different ways: via package repositories, via the `install_esoflex` 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_esoflex` 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_esoflex` script. Note that the script installation requires that your system fulfill several software prerequisites, which might also need `sudo` privileges.

Reflex 2.11.x needs java JDK 11 to be installed.

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 `esoflex` process.

### 2.1 Installing Esoflex workflows via `macports`

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

### 2.2 Installing Esoflex workflows via `rpm/yum/dnf`

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

### 2.3 Installing Esoflex workflows via `install_esoflex`

This method is recommended for operating systems other than what 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 2.6 may be found at: [https://www.eso.org/sci/software/pipelines/reflex\\_workflows](https://www.eso.org/sci/software/pipelines/reflex_workflows)

To install the Reflex 2.6 software and demo data, please follow these instructions:



<b>ESO</b>	<b>Reflex IIINSTRUMENT Tutorial</b>	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	9 of 26

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.

<b>ESO</b>	<b>Reflex IIINSTRUMENT Tutorial</b>	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	10 of <a href="#">26</a>

### 3 Demo Data

Add some explanation about the demo data used.

ESO	Reflex IIINSTRUMENT Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	11 of 26

## 4 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 4MOST demo data set supplied with the `esoreflex 2.6` 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 QMOST by typing:


```
esoreflex qmost&
```

Alternatively, you can type only the command `esoreflex` the empty canvas will appear (Figure 4.1) 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 QMOST workflow is shown in Figure 4.2.

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 .
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  button to select the directory from a file browser. When you have finished, click  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

<b>ESO</b>	<b>Reflex IIINSTRUMENT Tutorial</b>	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	12 of 26

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 4.3) that lists the datasets along with the values of a selection of useful header keywords<sup>2</sup>. 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 6.3.2.
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 4.4 shows the *Product Explorer* window. A full description of the *Product Explorer* will be presented in Section 6.3.3.
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 4MOST workflow that merit a look at the rest of this tutorial.

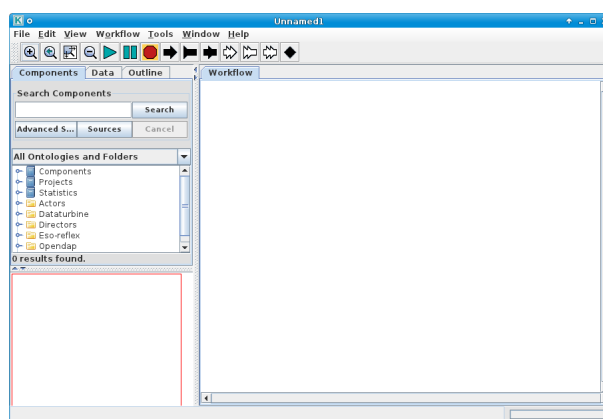


Figure 4.1: *The empty Reflex canvas.*

<sup>2</sup>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.

ESO	Reflex IIINSTRUMENT Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	13 of 26

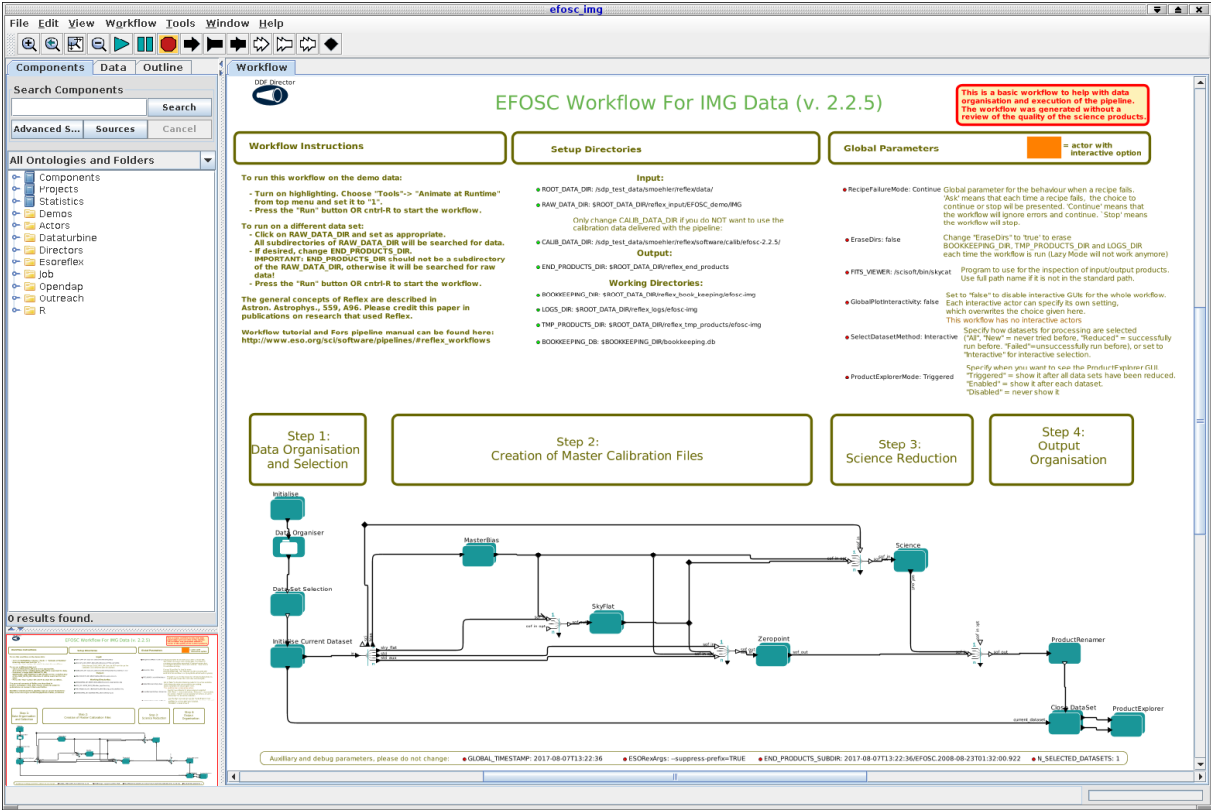


Figure 4.2: IIInstrument workflow general layout.

Select Datasets						
Selected	Data Set	#Files	Reduced	OBS.TARG.NAME	DPR.TYPE	INS.FILT
<input checked="" type="checkbox"/>	EFOSC.2008-05-29T08:44:34.706	27	-	Q2044-369	OBJECT	OIII#687
<input checked="" type="checkbox"/>	EFOSC.2010-03-19T03:28:32.393	29	-	4Cm02p55	OBJECT	g#782
<input checked="" type="checkbox"/>	EFOSC.2010-11-05T00:26:55.510	22	-	GRB100621A	OBJECT	U#640
<input checked="" type="checkbox"/>	EFOSC.2010-11-05T00:40:14.626	24	-	SA113-493	STD	U#640
<input checked="" type="checkbox"/>	EFOSC.2010-12-27T04:13:18.271	18	-	Ru152	STD	V#641

Save all

Inspect highlighted

Select complete

Deselect all

Filter:

Continue

Stop

Figure 4.3: The “Select Datasets” pop-up window.

ESO	Reflex IIINSTRUMENT Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	14 of 26

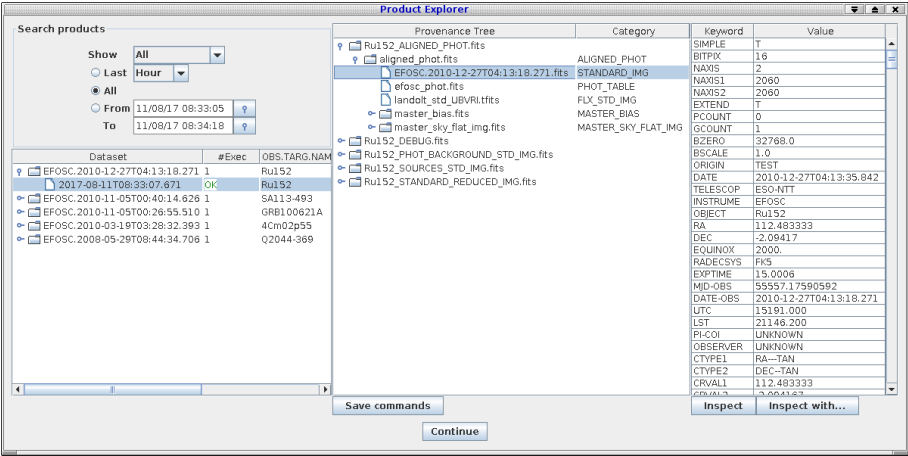


Figure 4.4: The Product Explorer shows all datasets reduced in previous executions together with the full reduction chain for all the pipeline products.

ESO	Reflex IIINSTRUMENT Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	15 of 26








## 5 About the main `esoreflex` canvas

### 5.1 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.

### 5.2 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.

### 5.3 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.

ESO	Reflex IIINSTRUMENT Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	16 of 26

## 6 The 4MOST Workflow

The 4MOST 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.

### 6.1 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; [1]).

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 6.2.2), 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 `Product-`





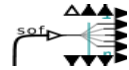


ESO	Reflex IIINSTRUMENT Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	17 of 26

Explorer 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.

## 6.2 Workflow Actors

### 6.2.1 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).

Add here a description of this workflow specific actors (probably including some composite actors).

### 6.2.2 Lazy Mode

By default, all `RecipeExecutor` 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

ESO	Reflex IIINSTRUMENT Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	18 of 26

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 `RecipeExecutor` 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 `RecipeExecutor` actors are actually found inside the composite actors in the top level workflow. To access such embedded `RecipeExecutor` 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 `RecipeExecutor` 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 makes the workflow to crash.*

Add here a description of the workflow steps like Data organisation, routing, creation of calibration files and science reduction.

## 6.3 Workflow Steps

### 6.3.1 Data Organisation And Selection

The `DataOrganiser` (DO) is the first crucial component of a Reflex workflow. The DO takes as input `RAW_DATA_DIR` and `CALIB_DATA_DIR` and it detects, classifies, and organises the files in these directories and any subdirectories. The output of the DO is a list of “DataSets”. A `DataSet` is a special Set of Files (SoF). A `DataSet` contains one or several science (or calibration) files that should be processed together, and all files needed to process these data. This includes any calibration files, and in turn files that are needed to process these calibrations. Note that different `DataSets` might overlap, i.e. some files might be included in more than one `DataSet` (e.g., common calibration files).

A `DataSet` lists three different pieces of information for each of its files, namely 1) the file name (including the path), 2) the file category, and 3) a string that is called the “purpose” of the file. The DO uses the OCA<sup>3</sup> rules to find the files to include in a `DataSet`, as well as their categories and purposes. The file category identifies different types of files, and it is derived by information in the header of the file itself. A category

<sup>3</sup>OCA stands for OrganisationClassificationAssociation and refers to rules, which allow to classify the raw data according to the contents of the header keywords, organise them in appropriate groups for processing, and associate the required calibration data for processing. They can be found in the directory `<install_dir>/share/esopipes/<pipeline-version>/reflex/`, carrying the extension `.oca`. The variable `<install_dir>` depends on the operative system and installation procedure. For installation through rpm: `<install_dir>=/usr`; for installation through macport `<install_dir>=/opt/local`; for installation through the installation script `install_esoreflex` it depends on the path specified during installation, e.g. `<install_dir>=<specified_path>/install`

ESO	Reflex IIIINSTRUMENT Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	19 of 26

could for example be `RAW_CALIBRATION_1`, `RAW_CALIBRATION_2` or `RAW_SCIENCE`, depending on the instrument. The purpose of a file identifies the reason why a file is included in a `DataSet`. The syntax is `action_1/action_2/action_3/ ... /action_n`, where each `action_i` describes an intended processing step for this file (for example, creation of a `MASTER_CALIBRATION_1` or a `MASTER_CALIBRATION_2`). The actions are defined in the OCA rules and contain the recipe together with all file categories required to execute it (and predicted products in case of calibration data). For example, a workflow might include two actions `action_1` and `action_2`. The former creates `MASTER_CALIBRATION_1` from `RAW_CALIBRATION_1`, and the later creates a `MASTER_CALIBRATION_2` from `RAW_CALIBRATION_2`. The `action_2` action needs `RAW_CALIBRATION_2` frames and the `MASTER_CALIBRATION_1` as input. In this case, these `RAW_CALIBRATION_1` files will have the purpose `action_1/action_2`. The same `DataSet` might also include `RAW_CALIBRATION_1` with a different purpose; irrespective of their purpose the file category for all these biases will be `RAW_CALIBRATION_1`.

The `Datasets` created via the `DataOrganiser` will be displayed in the `DataSet Chooser`. Here the users have the possibility to inspect the various datasets and decide which one to reduce. By default, `Datasets` that have not been reduced before are highlighted for reduction. Click either `Continue` in order to continue with the workflow reduction, or `Stop` in order to stop the workflow. A full description of the `DataSet Chooser` is presented in Section 6.3.2.

Once the `Continue` is pressed, the workflow starts to reduce the first selected `DataSet`. Files are broadcasted according to their purpose to the relevant actors for processing.

The categories and purposes of raw files are set by the DO, whereas the categories and purpose of products generated by recipes are set by the `RecipeExecutor`. The file categories are used by the `FitsRouter` to send files to particular processing steps or branches of the workflow (see below). The purpose is used by the `SofSplitter` and `SofAccumulator` to generate input `SoFs` for the `RecipeExecutor`. The `SofSplitter` and `SofAccumulator` accept several `SoFs` as simultaneous input. The `SofAccumulator` creates a single output `SoF` from the inputs, whereas the `SofSplitter` creates a separate output `SoF` for each purpose.

### 6.3.2 DataSetChooser

The `DataSetChooser` displays the `Datasets` available in the “Select Data Sets” window, activating vertical and horizontal scroll bars if necessary (Fig. 4.3).

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

<b>ESO</b>	<b>Reflex IIINSTRUMENT Tutorial</b>	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	20 of 26

, then a “Select Frames” window will appear that lists the set of files that make up the highlighted DataSet including the full filename<sup>4</sup>, 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  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 , 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 .

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 . 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  in order to continue with the workflow reduction, or  in order to stop the workflow.


### 6.3.3 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. 4.4).

#### *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 data reduction:

- 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.
  - "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.

<sup>4</sup>keep the mouse pointer on the file name to visualize the full path name.

<b>ESO</b>	<b>Reflex IIINSTRUMENT Tutorial</b>	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	21 of 26

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. 4.4). Once you click on a dataset, you get the list of reduction attempts. Green and red flags identify successful or unsuccessful 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 successful (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+Ctrl+v to paste it into a terminal.
    - \* Inspect Generic. It opens the file with the fits viewer selected in the main workflow canvas.

<b>ESO</b>	<b>Reflex IIINSTRUMENT Tutorial</b>	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	22 of 26

- \* 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.

ESO	Reflex IIINSTRUMENT Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	23 of 26

## 7 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 ([1]). 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:



<b>ESO</b>	<b>Reflex IIINSTRUMENT Tutorial</b>	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	24 of 26

```
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 ([1]) 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 ([1]) 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.



<b>ESO</b>	<b>Reflex IIINSTRUMENT Tutorial</b>	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	25 of 26

- **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 2, 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](http://www.eso.org/sci/data-processing/Python_and_esoreflex.pdf)

Add here a section on troubleshooting problems.

<b>ESO</b>	<b>Reflex IIINSTRUMENT Tutorial</b>	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date 2025-06-16
		Page:	26 of 26

- [1] Forchì V. *Reflex User's Manual*. ESO/SDD/DFS, <http://www.eso.org/gasgano/>, 0.7 edition, 2012. VLT-MAN-ESO-19000-5037. 16, 23, 24