



# 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 II INSTRUMENT Tutorial

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

Issue 0.1

Date April 2016

Prepared: N. Surname April 2016  
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 April 2016
		Page:	3 of 19

#### 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 April 2016
		Page:	5 of 19

## Contents

<b>1</b>	<b>Introduction And Scope</b>	<b>7</b>
<b>2</b>	<b>Software Installation</b>	<b>8</b>
2.1	Installing Reflex workflows via <code>macports</code> . . . . .	8
2.2	Installing Reflex workflows via <code>rpm/yum</code> . . . . .	8
2.3	Installing Reflex workflows via <code>install_esoreflex</code> . . . . .	9
<b>3</b>	<b>Demo Data</b>	<b>11</b>
<b>4</b>	<b>Quick Start: Reducing The Demo Data</b>	<b>12</b>
<b>5</b>	<b>About The Reflex Canvas</b>	<b>13</b>
5.1	Saving And Loading Workflows . . . . .	13
5.2	Buttons . . . . .	13
5.3	Workflow States . . . . .	13
<b>6</b>	<b>The IIINSTRUMENT Workflow</b>	<b>14</b>
6.1	Workflow Canvas Parameters . . . . .	14
6.2	Workflow Actors . . . . .	15
6.2.1	Simple Actors . . . . .	15
6.2.2	Lazy Mode . . . . .	15
<b>7</b>	<b>Frequently Asked Questions</b>	<b>17</b>

<b>ESO</b>	<b>Reflex IIINSTRUMENT Tutorial</b>	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date April 2016
		Page:	6 of <a href="#">19</a>

<b>ESO</b>	<b>Reflex IIINSTRUMENT Tutorial</b>	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date April 2016
		Page:	7 of 19

## 1 Introduction And Scope

Reflex 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 contact [usd-help@eso.org](mailto:usd-help@eso.org) for further support.

This document is a tutorial designed to enable the user to employ the IIINSTRUMENT workflow to reduce his/her data in a user-friendly way, concentrating on high-level issues such as data reduction quality and signal-to-noise (S/N) optimisation.

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><http://www.eso.org/sci/archive/calselectorInfo.html>

ESO	Reflex IIINSTRUMENT Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date April 2016
		Page:	8 of 19

## 2 Software Installation

Reflex and the workflows can be installed in different ways: via package repositories, via `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 `macports` repositories support OS X, while the `rpm/yum` repositories support Fedora 20/21/22/23. For any other operating system it is recommended to use the `install_esoreflex` script.

### 2.1 Installing Reflex workflows via `macports`

This method is supported for OS X operating system. It is assumed that `macports` (<http://www.macports.org>) and `java` are installed. If you have any problem with this installation method, please read the full documentation at

<http://www.eso.org/sci/software/pipelines/installation/macports.html>.

For a quick installation, the following steps will install the ESO pipeline `macports` repository, the IIINSTRUMENT pipeline, including the Reflex workflow support and Reflex itself:

- Set up the repository:

```
# curl ftp://ftp.eso.org/pub/dfs/pipelines/repositories/macports/setup/Portfile -o Portfile
# sudo port install
# sudo port sync
```

- Install the IIINSTRUMENT pipeline:

```
# sudo port install esope-iiinstrument-all
```

### 2.2 Installing Reflex workflows via `rpm/yum`

This method is supported for Fedora 20/21/22/23 operating systems. If you have any problem with this installation method, please read the full documentation at

<http://www.eso.org/sci/software/pipelines/installation/rpm.html>.

For a quick installation, the following steps will install the ESO pipeline `rpm` repository, the IIINSTRUMENT pipeline, including the Reflex workflow support and Reflex itself:

- Set up the repository for Fedora 20/21:

```
# sudo yum install yum-utils
# sudo yum-config-manager \
  --add-repo=ftp://ftp.eso.org/pub/dfs/pipelines/repositories/fedora/esorepo.repo
```

- Set up the repository for Fedora 22/23:

```
# sudo dnf install dnf-plugins-core
# sudo dnf config-manager \
  --add-repo=ftp://ftp.eso.org/pub/dfs/pipelines/repositories/fedora/esorepo.repo
```



<b>ESO</b>	<b>Reflex IIINSTRUMENT Tutorial</b>	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date April 2016
		Page:	9 of 19

- Install the IIINSTRUMENT pipeline (Fedora 20/21):  
# `sudo yum install esopipe-iiinstrument-all`
- Install the IIINSTRUMENT pipeline (Fedora 22/23):  
# `sudo dnf install esopipe-iiinstrument-all`

### 2.3 Installing Reflex workflows via `install_esoreflex`

The software pre-requisites for Reflex 2.6 may be found at:  
[http://www.eso.org/sci/software/pipelines/reflex\\_workflows](http://www.eso.org/sci/software/pipelines/reflex_workflows)

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

1. From any directory, download the installation script:

```
wget ftp://ftp.eso.org/pub/dfs/reflex/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. You will be asked whether you want to use your Internet connection. Unless you want to reuse already downloaded packages (only advanced users), use the default Yes.
5. You will be given a choice of pipelines (with the corresponding workflows) to install. Please specify the numbers for the pipelines you require, separated by a space, or type “A” for all pipelines.
6. For the pipelines to be installed you will be prompted for the demo data sets to be installed. Type “A” for all demo datasets. Take into account that if you are installing in a directory that already contains data, it won’t be removed.
7. The script will also detect whether previous versions of the workflows or Reflex were installed and in this case you have the option to update links or remove obsolete cache directories. It is advised to use the defaults.
8. To start Reflex, issue the command:

<b>ESO</b>	<b>Reflex IIIINSTRUMENT Tutorial</b>	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date April 2016
		Page:	10 of 19

`<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 April 2016
		Page:	11 of 19

### 3 Demo Data

Add some explanation about the demo data used.

<b>ESO</b>	<b>Reflex IIINSTRUMENT Tutorial</b>	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date April 2016
		Page:	12 of <a href="#">19</a>

## 4 Quick Start: Reducing The Demo Data

Add here a two pages maximum explanation on how to quickly reduce the demo data set.

ESO	Reflex IIINSTRUMENT Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date April 2016
		Page:	13 of 19








## 5 About The Reflex 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 Reflex 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 in another computer.








### 5.2 Buttons

At the top of the Reflex canvas are a set of buttons which have the following useful 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 April 2016
		Page:	14 of 19

## 6 The IIINSTRUMENT Workflow

The IIINSTRUMENT 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 CalSelection associations` in the `Data Organiser` properties and make sure that the input data directory contains the XML file downloaded with the `CalSelector` archive request.

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 in general it is recommended to specify the full path to the visualization application.

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 where 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 rereduce 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.



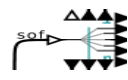
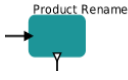
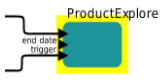
The parameter `ProductExplorerMode` controls whether the `ProductExplorer` actor will show its window or not. The possible values are `Enabled`, `Disabled` and `Triggered`. The later, recommended, means that the `ProductExplorer` actor will be shown only at the end of the workflow execution.

ESO	Reflex IIINSTRUMENT Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date April 2016
		Page:	15 of 19

## 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 a logo within the rectangle to aid in their identification. The following actors are simple actors:

- 
 - The Data Organiser actor.
- 
 - The Data Set Chooser actor (inside a composite actor).
- 
 - The Fits Router actor
- 
 - The Product Renamer actor.
- 
 - The Product Explorer 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 recipe executer 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 rereduction where it is not necessary.

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

The forced rereduction of data at each execution may of course be desirable. To force a rereduction of all data for all `RecipeExecuter` actors in the workflow (i.e. to disable Lazy mode for the whole workflow), set the

<b>ESO</b>	<b>Reflex IIINSTRUMENT Tutorial</b>	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date April 2016
		Page:	16 of <a href="#">19</a>

EraseDirs parameter under the “Global Parameters” area of the workflow canvas to `true`. This will then remove all previous results as well. To force a rereduction of data for any single `RecipeExecutor` actor in the workflow (which will be inside the relevant composite actor), right-click the `RecipeExecutor` actor, select `Configure Actor`, and uncheck the Lazy mode parameter tick-box in the “Edit parameters” window that is displayed. If the Lazy mode is switched off for an actor, all subsequent actors that use products from that one will also reprocess the data, as they see new products.

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


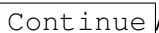

Add here a description on how to optimise the results of the workflow.



ESO	Reflex IIINSTRUMENT Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date April 2016
		Page:	17 of 19

## 7 Frequently Asked Questions

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

Press the  key together with your left mouse button to move the window upwards and to the left. At the bottom the   buttons will be visible. This bug is known but a fix has not yet been found.

- **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 to Kepler and the workflows that avoid any graphical display (including pipeline interactive windows).

It is possible to specify workflow variables (those that appear in the workflow canvas) in the command line. For instance, to set the raw data directory can be done 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 executor. This actor must be manually instantiated using `Tools -> Instantiate Component`. Fill in the “Class name” field with `org.eso.RecipeExecutor` 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.

<b>ESO</b>	<b>Reflex IIINSTRUMENT Tutorial</b>	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 0.1
		Date:	Date April 2016
		Page:	18 of 19

- **How can I run manually 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 some 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 in the command line a recipe that used a given 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 before, 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 doesn't 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 are in the pipeline. However there are situations in which a previously created bookkeeping directory will cause problems due to pipeline versions incompatibility. This is specially true if the parameters of the pipeline recipes have changed. In that case, please remove completely the bookkeeping directory.

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 April 2016
		Page:	19 of 19

- [1] Forchì V. *Reflex User's Manual*. ESO/SDD/DFS, <http://www.eso.org/gasgano/>, 0.7 edition, 2012. VLT-MAN-ESO-19000-5037. 14, 17