



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 FORS2 Tutorial

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

Issue 1.8

Date 2015-06-24

Prepared: S. Moehler, C.E. García Dabo 2015-06-24
Name Date Signature

Approved: W.Freudling
Name Date Signature

Released: M. Sterzik
Name Date Signature

This page was intentionally left blank

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	3 of 45

Change record

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
1.0	02/05/2013	All	First public release 4.9.22
1.1	01/04/2013	All	Changes for release 4.9.23
1.2	01/10/2013	All	Changes for release 4.10.8
1.3	06/11/2013	All	Changes for release 4.11.11
1.4	10/04/2014	All	Changes for release 4.12.8
1.5	18/08/2014	All	Changes for release 4.12.21
1.6	02/10/2014	All	Changes for release 5.0.0
1.8	24/06/2015	All	Changes for release 5.0.11

This page was intentionally left blank

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	5 of 45

Contents

1	Introduction And Scope	7
2	Software Installation	8
3	Demo Data	9
4	Quick Start: Reducing The Demo Data	10
5	About The Reflex Canvas	15
5.1	Saving And Loading Workflows	15
5.2	Buttons	15
5.3	Workflow States	15
6	The FORS2 Workflow	16
6.1	Workflow Canvas Parameters	16
6.2	Workflow Actors	17
6.2.1	Simple Actors	17
6.2.2	Composite Actors	17
6.2.3	Recipe Execution within Composite Actors	18
6.2.4	Lazy Mode	21
6.3	Workflow Steps	21
6.3.1	Step 1: Data Organisation And Selection	21
6.3.2	Step 2: Creation Of Master Calibration Files	24
6.3.3	Step 3: Response Computation	24
6.3.4	Step 4: Science Reduction	24
6.3.5	Step 5: Output Organisation	24
6.4	Interactive Windows	26
6.4.1	fors_calib	26
6.4.2	fors_science (Response)	27
6.4.3	fors_science (Science)	28

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	6 of 45

7	Improving Your Results Through Workflow Interaction	30
7.1	Demo Data	30
7.2	Wavelength Calibration	30
7.3	Flat Field Normalization	32
7.4	Response Determination	32
8	Frequently Asked Questions	37
8.1	FORS specific questions	38
9	Troubleshooting	42

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	7 of 45

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 for further support.

This document is a tutorial designed to enable the user to employ the FORS2 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 delivered to PIs in the form of PI-Packs (until October 2011) or downloaded from the archive using the CalSelector tool¹ 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.

The FORS2 Reflex workflow described in this tutorial supports the reduction of FORS2 spectroscopic observations, only LSS/MOS/MXU, no PMOS. The user is referred to the FORS2 user manual (Saviane, 2012²) for more information on the instrument itself, and the FORS2 pipeline user manual (Izzo et al. 2012³) for the details of the spectroscopic FORS2 pipeline recipes.

The FORS2 pipeline can provide good quality results for most of the spectroscopic data (LSS/MOS/MXU), but may require fine-tuning of parameters to achieve the best results. Data with slits distributed across a large range along the dispersion axis can be difficult to process with one set of parameters for all slits. The workflow provides interactive windows for most recipes that allow the user to verify the quality of the reduction and tweak the parameters if necessary. The FORS2 pipeline underwent a thorough evaluation in 2013-2014, which greatly improved the quality of the results, esp. with respect to error propagation, distortion correction, and flux calibration.

The quick start section (see Section 4) describes the minimum effort to get started, and it makes up only two pages of text in this tutorial.

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

²available at

<http://www.eso.org/sci/facilities/paranal/instruments/fors2/doc>

³available at <ftp://ftp.eso.org/pub/dfs/pipelines/fors/fors-pipeline-manual-5.2.pdf>

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	8 of 45

2 Software Installation

The software pre-requisites for Reflex 2.8 may be found at:

http://www.eso.org/sci/software/pipelines/reflex_workflows

To install the Reflex 2.8 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:

```
<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.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	9 of 45

3 Demo Data

Together with the pipeline you will also receive a demo data set, that allows you to run the `Reflex FORS2` workflow without any changes in parameters. This way you have a data set to experiment with before you start to work on your own data.

Note that you will need a minimum of ~ 0.5 GB, ~ 0.6 GB and ~ 1 GB of free disk space for the directories `<download_dir>`, `<install_dir>` and `<data_dir>`, respectively. The FORS2 demo data have been retrieved with the CalSelector tool⁴.

One of the demo data sets (`FORS2.2010-03-17T09:17:19.236`, 2nd) does not contain flux standard observations and the data are therefore not flux-calibrated. This happens currently if one downloads data from the lower chips only. Standard stars are observed only on the upper CCD. If instead full SCIENCE observations (upper and lower chip) are downloaded with `calSelector` any flux standard star observed for the upper chip will also be associated to the lower chip. For the second data set from the lower chip (`FORS2.2010-03-22T06:55:17.701`, 3rd) we identified and downloaded the standard star observation and its calibrations manually. The data sets `FORS2.2010-05-03T03:36:35.486` (4th) and `FORS2.2012-06-22T04:00:32.921` (5th) were carefully selected to illustrate potential problems and Sect. 7.1 describes how to improve the results obtained for these data sets.

⁴<http://www.eso.org/sci/archive/calselectorInfo.html>

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	10 of 45

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 FORS2 demo data set supplied with the Reflex 2.8 release. By following these steps, the user should have enough information to attempt a reduction of his/her own data without any further reading:



Figure 4.1: *The empty Reflex canvas.*

1. Start the Reflex application:

```
reflex &
```

The empty Reflex canvas as shown in Figure 4.1 will appear.

2. Now open the FORS2 workflow by clicking on File -> Open File, selecting first fors-5.0.14 and then the file ForsSpec.xml in the file browser. You will be presented with the workflow canvas shown in Figure 4.2. Note that the workflow will appear as a canvas in a new window.
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. Under “Setup Directories” in the workflow canvas there are seven parameters that specify important directories (green dots). Setting the value of ROOT_DATA_DIR is the only necessary modification if you want to process data other than the demo data⁵, since the value of this parameter specifies the working directory within which the other directories are organised. Double-click on the parameter ROOT_DATA_DIR 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.

⁵If you used the install script install_reflex, then the value of the parameter ROOT_DATA_DIR will already be set correctly to the directory where the demo data was downloaded.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	11 of 45

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 `RAWDATA_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 `RAWDATA_DIR` or in `CALIB_DATA_DIR`, otherwise `DataSets` may be incomplete and cannot be processed. However, if the same reference file was downloaded twice in different places this creates a problem as `Reflex` 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 keywords⁶. The first column consists of a set of tick boxes which allow the user to select the `DataSets` to be processed, and by default all complete `DataSets` are selected.

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. For each execution of the `ForsCalib` actor, an interactive window will appear which shows some plots to evaluate the quality of the slit tracing, the wavelength calibration and the master flat creation. Figure 4.4 shows the look of that window. The first and fifth data set will create an error message during that step. See p. 30 how to proceed.

This window will appear as many times as inputs of the calibration recipe exist. Note that some datasets contain calibrations for the standard star as well. For the time being just click on `Continue` to proceed to the next steps. Also the *Response Curve* (Fig. 6.4, p. 27) and *ForsScience* (Fig. 6.5, p. 28) actors have associated interactive windows, where you should proceed in the same way.

10. When the reduction of the current `DataSet` finishes, a pop-up window called *Provenance Explorer* will appear showing the datasets which have been so far 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 4.5 shows the *Provenance Explorer* window.

11. The workflow will continue with the remaining `DataSets` following the same steps described above.

12. After the workflow has finished, all the products from all the `DataSets` can be found in a directory under `END_PRODUCTS_DIR` with the named with 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 FORS2 workflow that merit a look at the rest of this tutorial.

⁶The keywords listed can be changed by right-clicking on the `DataOrganiser` Actor, selecting `Configure Actor`, and then changing the list of keywords in the second line of the pop-up window.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	12 of 45

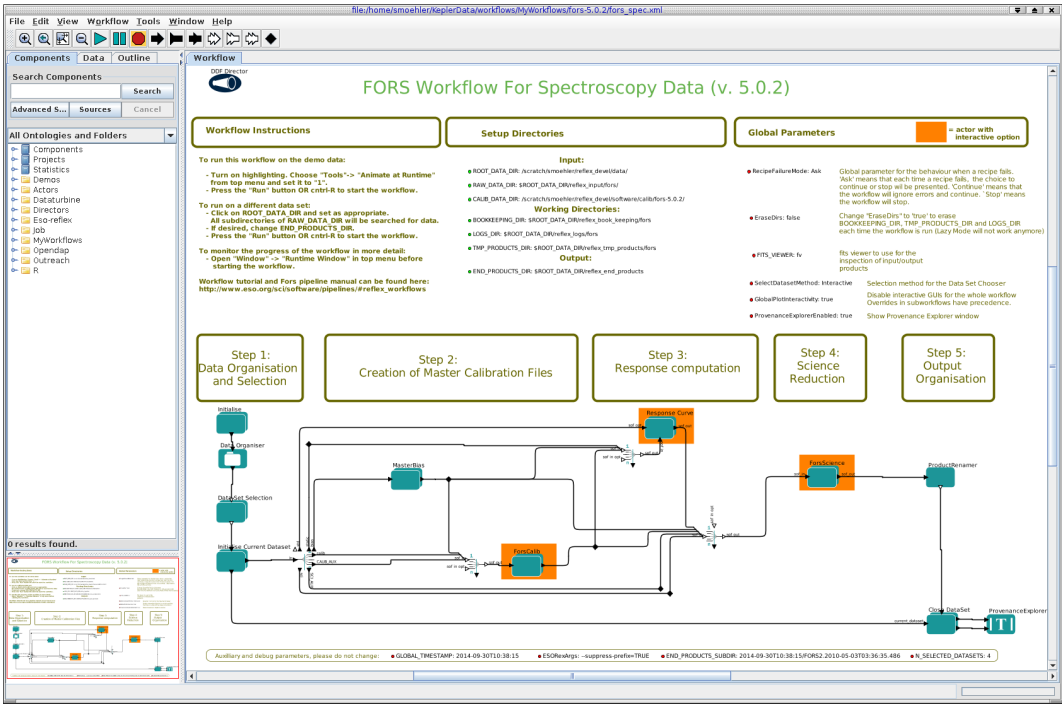


Figure 4.2: FORS2 workflow general layout.

Select Datasets				
Selected	Data Set	#Files	OBS.TARG.NAME	INS. GRIS1.NAM
<input checked="" type="checkbox"/>	FORS2.2008-07-20T00:14:58.014	25	Rhooph2x08-28f	GRIS_600R
<input checked="" type="checkbox"/>	FORS2.2010-09-17T09:17:19.236	15	RC52.1514.15.284181_10.05788	GRIS_150I
<input checked="" type="checkbox"/>	FORS2.2010-03-22T06:59:17.701	30	BULB	GRIS_500R
<input checked="" type="checkbox"/>	FORS2.2010-05-03T03:38:35.486	30	VCC1694	GRIS_1400V
<input checked="" type="checkbox"/>	FORS2.2012-06-22T04:00:32.921	29	V707-Scp	GRIS_300V

Select complete

Select all

Deselect all

Save all

Inspect highlighted

Continue

Stop

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

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	13 of 45

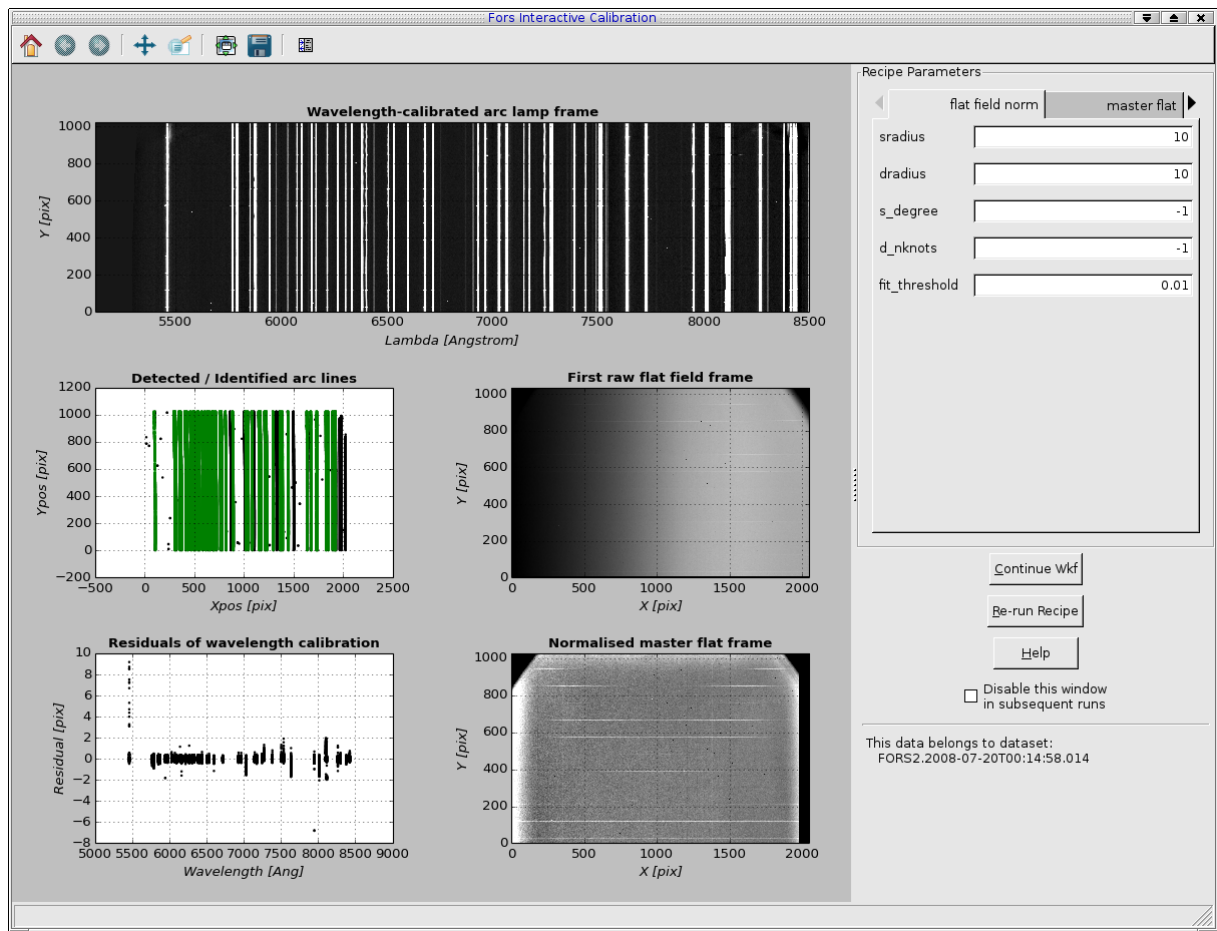


Figure 4.4: The interactive window of the `fors_calib` actor for the first demo DataSet. First the calibration data of the flux standard star are processed.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	14 of 45

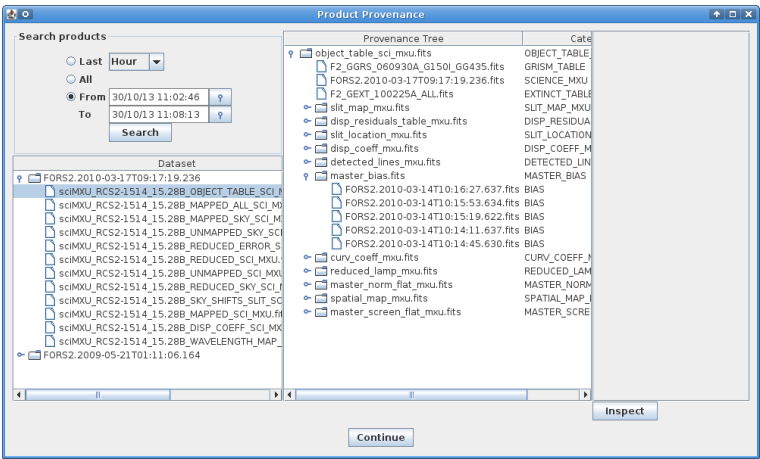


Figure 4.5: The Provenance Explorer shows all datasets reduced in previous executions together with the full reduction chain for all the pipeline products.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	15 of 45








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 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 FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	16 of 45

6 The FORS2 Workflow

The FORS2 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 `RAWDATA_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 `RAWDATA_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; [Forchi \(2012\)](#)).

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.

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 6.2.4), 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 fails 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. Additionally, the `Stop` mode will stop the workflow execution immediately.

The parameter `GlobalInteractivity` controls whether the interactive windows will appear for those windows which are *enabled* by default. The possible values are `true`, `false`. Take into account that some windows are disabled in the default configuration and therefore are not affected by this parameter.



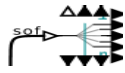


The parameter `ProvenanceExplorerEnabled` controls whether the `ProvenanceExplorer` actor will show its window or not. The possible values are `true`, `false`.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	17 of 45

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.
- 
 - The Fits Router actor
- 
 - The Product Renamer actor.
- 
 - The Provenance Explorer actor.




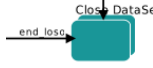
Access to the parameters for a simple actor is achieved by right-clicking on the actor and selecting `Configure Actor`. This will open an “Edit parameters” window. Note that the `Product Renamer` actor is a jython script (Java implementation of the Python interpreter) meant to be customised by the user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

6.2.2 Composite Actors

Composite Actors have workflow symbols that consist of multiple-layered green-blue rectangles. They generally do not have a logo within the rectangle. A Composite Actor represents a combination of more Simple or Composite Actors which hides over-complexity from the user in the top-level workflow. In the FORS2 workflow, the following actors are composite actors:

- 
 - The Initialise actor.
- 
 - The Initialise Current DataSet actor.
- 
 - The MasterBias actor.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	18 of 45

-  - The `ForsCalib` actor.
-  - The `ResponseCurve` actor.
-  - The `ForsScience` actor.
-  - The `Close DataSet` actor.

Composite Actors may also be expanded for inspection. To do this, right-click on the actor and select `Open Actor`, which will expand the Composite Actor components in a new `Reflex` canvas window. If the Composite Actor corresponds to a pipeline recipe, then the corresponding `RecipeExecutor` actor will be present as a Simple Actor, and its parameters are accessible as for any other Simple Actor. Alternatively you may still find Composite Actors, on which you need to repeat the first step to access the `Recipe Executor`.

6.2.3 Recipe Execution within Composite Actors

The FORS2 workflow contains Composite Actors to run pipeline recipes. This is in the most simple case due to the `SoF Splitter/SoF Accumulator`⁷, which allow to process calibration data from different settings within one given `DataSet` (e.g. lamp frames taken with different slits/masks). More complex Composite Actors contain several actors (e.g. `Recipe Executor`).

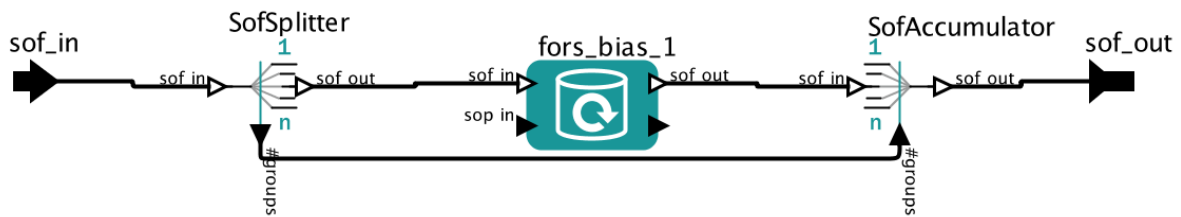


Figure 6.1: This is the window you get when you choose `Open Actor` for the Composite Actor `MasterBias`. This is the most simple case for a Composite Actor. Using `Configure Actor` on `fors_bias_1` gives you Fig. 6.2.

The central elements of any `Reflex` workflow are the `RecipeExecutor` actors that actually run the recipes. One basic way to embed a `RecipeExecutor` in a workflow is shown in Fig 6.1, which is the most simple version of a Composite Actor. The `RecipeExecutor` is preceded by an `SofSplitter`, and followed by an `SofAccumulator`. The function of the `SofSplitter` is to investigate the incoming `SoFs`, sort them by “purpose”, and create separate `SoFs` for each purpose. The `RecipeExecutor` then processes each of

⁷ `SoF` stands for `Set of Files`, which is an ASCII file containing the name (and path) of each input file and its category (e.g. `BIAS`).

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	19 of 45

Table 6.1: The FORS2 pipeline actors and their contents

actor	recipes	description
MasterBias	fors_bias	create master bias
ForsPmosCalib	fors_calib	create master flat, determine coefficients for wave-length calibration and correction of spatial distortion
ForsPmosScience	fors_science	reduce science and standard star data

the SoFs independently (unless they are actually the same files). Finally, the `SofAccumulator` packs all the results into a single output SoF. The direct relation between the `SofSplitter` and `SofAccumulator` is used to communicate the number of different SoFs created by the `SofSplitter`. A workflow will only work as intended if the purpose of all the files a recipe needs as input is identical. The only exception to this rule is that a purpose can also be “default”. In this case, the file is included in any output SoF created by the `SofSplitter` and `SofAccumulator`.

The reason for this scheme is best explained by an example. For a complex `DataSet`, the `Data Organiser` might have selected a large number of individual raw lamp frames (arc and flat field). The different lamp frames are to be used to calibrate different frames, e.g. the science frames and the standard star frames. The `Data Organiser` determines and records this “purpose” of each lamp frame, and this information is included in the `DataSet` and each SoF created from this `DataSet`. The `FitsRouter` directs all raw lamp frames to the `ForsCalib Composite Actor`. The `SofSplitter` then creates SoFs, one for the lamp frames to be used for the science frames, and (probably) separate ones for the lamp frames to be used for the standard star observations. The `fors_calib` recipe creates one master flat field (and other products) for each SoF, and the `SofAccumulator` then creates a SoF that contains all the products.

A `RecipeExecutor` actor is used in the workflow to run a single FORS2 pipeline recipe (e.g. in the `MasterBias` actor the recipe `fors_bias` is executed). In order to configure the `RecipeExecutors`, one has to first use `Open Actor` to get to the level of the recipe executors (see Fig. 6.1).

In Figure 6.2 we show the “Edit parameters” window for a typical `RecipeExecutor` actor, which can be displayed by right-clicking on the actor and selecting `Configure Actor`. In the following we describe in more detail the function of some of the parameters for a `RecipeExecutor` actor:

- The “recipe” parameter states the FORS2 pipeline recipe which will be executed.
- The “mode” parameter has a pull-down menu allowing the user to specify the execution mode of the actor. The available options are:
 - Run: The pipeline recipe will be executed, possibly in Lazy mode (see Section 6.2.4). This option is the default option.
 - Skip: The pipeline recipe is not executed, and the actor inputs are passed to the actor outputs.
 - Disabled: The pipeline recipe is not executed, and the actor inputs are not passed to the actor outputs.
- The “Lazy Mode” parameter has a tick-box (selected by default) which indicates whether the `RecipeExecutor` actor will run in Lazy mode or not. A full description of Lazy mode is provided in Sect. 6.2.4.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	20 of 45



Figure 6.2: The “Edit parameters” window for a typical `RecipeExecutor` actor, the `fors_bias_1` actor which runs the `fors_bias` pipeline recipe.

- The “Recipe Failure Mode” parameter has a pull-down menu allowing the user to specify the behaviour of the actor if the pipeline recipe fails. The available options are:
 - Stop: The actor issues an error message and the workflow stops.
 - Continue: The actor creates an empty output and the workflow continues.
 - Ask: The actor displays a pop-up window and asks the user whether he/she wants to continue or stop the workflow. This option is the default option.
- The set of parameters which start with “recipe param” and end with a number or a string correspond to the parameters of the relevant FORS2 pipeline recipe. By default in the `RecipeExecutor` actor, the pipeline recipe parameters are set to their pipeline default values. If you need to change the default parameter value for any pipeline recipe, then this is where you should edit the value⁸. A special case are parameters that can be changed in the interactive windows (see Sect. 6.4). For more information on the FORS2 pipeline recipe parameters, the user should refer to the FORS2 pipeline user manual (Izzo et al. 2012⁹).

The description of the remainder of the `RecipeExecutor` actor parameters are outside the scope of this tutorial, and the interested user is referred to the Reflex User Manual for further details (Forchì 2012). Any

⁸Some of the pipeline parameters are read from the `GRISM_TABLES`, which contain grism-specific parameters. These cannot be changed here.

⁹Available at <ftp://ftp.eso.org/pub/dfs/pipelines/fors/fors-pipeline-manual-5.2.pdf>

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	21 of 45

changes that you make in the “Edit parameters” window may be saved in the workflow by clicking the Commit button when you have finished.

6.2.4 Lazy Mode

By default, all recipe executor 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 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 `RecipeExecutor` actors in the workflow (i.e. to disable Lazy mode for the whole workflow), set the `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.

6.3 Workflow Steps

6.3.1 Step 1: Data Organisation And Selection

On clicking the  button on the `Reflex` canvas, 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 6.1).

1. The `DataOrganiser` (DO) is the first crucial component of a `Reflex` workflow. The DO takes as input `RAWDATA_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. `BIAS` 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

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	22 of 45

OCA¹⁰ 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. A category could for example be `FLAT_LSS`, `LAMP_MOS` or `SCIENCE_MOS`. 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. 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 `BIAS` and `CAL_LSS`. The former creates a master bias from raw biases, and the later creates (among other products) a master flat from raw flats. The `CAL_LSS` action needs raw lamp frames (arc and flat field) and the master bias as input. In this case, these biases will have the purpose `BIAS/CAL_LSS`. The same DataSet might also include biases with a different purpose, e.g. `BIAS/CAL_MOS`. Irrespective of their purpose the file category for all these biases will be `BIAS`.

2. Next the `DataSet Chooser` displays the DataSets available in the “Select Data Sets” window, activating a vertical scroll bar on the right if necessary (see Figure 4.3). 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 `Select All` and `Deselect All` at the bottom left.

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 and path for each file, the file category (from the FITS header), and a selection tick box in the right column (see Figure 6.3). 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 (see Figure 6.3), 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`, and 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.

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 `RecipeExecuter` (see Sect. 6.2.3). 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 `RecipeExecuter`. Note that while the DO includes files into a DataSet for a reason, and records this reason as the “purpose” of the file, the workflow itself can use these files in a different manner. 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.

¹⁰OCA 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`

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	23 of 45



Figure 6.3: The “Select Frames” window with a single file from the current Data Set highlighted in blue, and the corresponding FITS header displayed in the text box on the right. Hidden partially behind the “Select Frames” window is the “Select DataSets” window with the currently selected DataSet highlighted in blue.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	24 of 45

6.3.2 Step 2: Creation Of Master Calibration Files

In this step of the workflow, the following FORS2 recipes are executed in the order listed below. Please refer to the FORS2 pipeline user manual (Izzo et al. 2012: Sections 9 and 10) for the details of each recipe and the algorithms employed:

1. The `MasterBias` actor will execute the FORS2 pipeline recipe `fors_bias` in order to create a combined master bias frame from the set of raw bias frames
2. The `ForsCalib` actor will execute the FORS2 pipeline recipe `fors_calib` in order to create from the set of raw flat and arc frames a combined master flat frame as well as coefficients for wavelength calibration and correction of spatial distortions.

6.3.3 Step 3: Response Computation

The `ResponseCurve` actor will execute the FORS2 pipeline recipe `fors_science` in order to create an instrument response curve from the observation of a standard star (if it is found in the standard star tables), which will subsequently be used to flux-calibrate the science observation. Please refer to the FORS2 pipeline user manual (Izzo et al. 2012: Sections 9 and 10) for the details of this recipe.

Note that this actor will be skipped if there are no observations of a standard star in the current `DataSet`. `calSelector` usually includes a standard star observation only if it was taken within ± 7 nights of the science observation. For some combinations of grism and filter, FORS2 PI-Packs as well as `CalSelector` data sets are supplied with master instrument response curves. As they were produced with older pipeline versions that had some flaws in the response creation the workflow ignores these files.

6.3.4 Step 4: Science Reduction

The `ForsScience` actor will execute the FORS2 pipeline recipe `fors_science` to apply sky subtraction and extract the spectra. Please refer to the FORS2 pipeline user manual (Izzo et al. 2012: Sections 9 and 10) for the details of this recipe and the extraction algorithms employed. The FORS2 workflow will flux-calibrate the science observation using the instrument response curve derived from the standard star observation if it exists in the current `DataSet`. If no standard star observation exists in the current `DataSet`, then the science observation will not be flux-calibrated.

6.3.5 Step 5: Output Organisation

After having processed the input data for a `DataSet`, the workflow highlights and executes the `ProductRenamer` actor, which, by default, will copy the defined final products of the `ForsScience` actor to the directory specified by `END_PRODUCTS_DIR` and rename them with names derived from the values of certain FITS header keywords. Specifically, final products are renamed by default with names of the form `<HIERARCH.ESO.OBS.NAME>_<HIERARCH.ESO.PRO.CATG>.fits`, with `<HIERARCH.ESO.OBS.NAME>` and `<HIERARCH.ESO.PRO.CATG>` representing the values of the corresponding FITS header keywords.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	25 of 45

These names are fully configurable by right-clicking on the `Product Renamer` actor, selecting `Configure Actor`, and then editing the string as appropriate.

For MOS data the final products that are copied and renamed are:

- `<HIERARCH.ESO.OBS.NAME>_REDUCED_SCI_MOS.fits` 1-dimensional extracted spectra
- `<HIERARCH.ESO.OBS.NAME>_REDUCED_ERROR_SCI_MOS.fits` error of 1-dimensional extracted spectra
- `<HIERARCH.ESO.OBS.NAME>_REDUCED_FLUX_SCI_MOS.fits` 1-dimensional extracted and flux-calibrated spectra (if photometric calibration is performed)
- `<HIERARCH.ESO.OBS.NAME>_REDUCED_FLUX_ERROR_SCI_MOS.fits` error of 1-dimensional extracted and flux-calibrated spectra (if photometric calibration is performed)
- `<HIERARCH.ESO.OBS.NAME>_REDUCED_SKY_SCI_MOS.fits` 1-dimensional extracted sky spectra
- `<HIERARCH.ESO.OBS.NAME>_UNMAPPED_SCI_MOS.fits` (not for LSS) 2-dimensional SCIENCE frame, sky-subtracted, neither wavelength calibrated nor distortion corrected
- `<HIERARCH.ESO.OBS.NAME>_UNMAPPED_SKY_SCI_MOS.fits` (not for LSS) 2-dimensional frame with fitted sky background, neither wavelength calibrated nor distortion corrected
- `<HIERARCH.ESO.OBS.NAME>_MAPPED_ALL_SCI_MOS.fits` 2-dimensional SCIENCE frame, wavelength calibrated and distortion corrected
- `<HIERARCH.ESO.OBS.NAME>_MAPPED_FLUX_SCI_MOS.fits` 2-dimensional SCIENCE frame, sky-subtracted, wavelength calibrated and distortion corrected, flux-calibrated (if photometric calibration is performed)
- `<HIERARCH.ESO.OBS.NAME>_MAPPED_SCI_MOS.fits` 2-dimensional SCIENCE frame, sky-subtracted, wavelength calibrated and distortion corrected
- `<HIERARCH.ESO.OBS.NAME>_MAPPED_SKY_SCI_MOS.fits` 2-dimensional frame with fitted sky background, wavelength calibrated and distortion corrected
- `<HIERARCH.ESO.OBS.NAME>_OBJECT_TABLE_SCI_MOS.fits` table with position information for detected spectra
- `<HIERARCH.ESO.OBS.NAME>_DISP_COEFF_SCI_MOS.fits` dispersion coefficients after adjusting to sky line positions (if sky alignment is requested)
- `<HIERARCH.ESO.OBS.NAME>_SKY_SHIFTS_SLIT_SCI_MOS.fits` shifts in wavelength derived from sky line positions (only for MOS-like¹¹ data if sky alignment is requested)
- `<HIERARCH.ESO.OBS.NAME>_WAVELENGTH_MAP_SCI_MOS.fits` 2-dimensional frame with pixel value=wavelength of pixel (if sky alignment is requested)

¹¹For long-slit and LSS-like MOS data this would be `<HIERARCH.ESO.OBS.NAME>_SKY_SHIFTS_LONG_SCI_MOS.fits`.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	26 of 45

The product filenames for the other modes are similar but with the MOS suffix replaced by MXU, LONG_MOS or LSS as appropriate.

All products with `<HIERARCH.ESO.OBS.NAME>_REDUCED` are created only if spectra are identified and can be extracted. All products `<HIERARCH.ESO.OBS.NAME>_<type>_FLUX` are created only if flux standard star observations for the upper chip are provided and flux calibration is requested (default).

The following actors in this step of the workflow are concerned with the termination of the data flow for the current DataSet and will highlight briefly as they are executed.

Finally, the `Provenance Explorer` window will appear as shown in Fig. 4.5 with a list of datasets on the left menu. By unfolding the menu under each dataset, all the renamed products appear, and if one is interested in the files, including all intermediate steps, that are used to produce that final product, just click on it and a dependency tree will show the whole reduction chain.

6.4 Interactive Windows

The FORS2 workflow contains three interactive windows that allow the user to iterate on the processing of their data. They are described below. For tips how to improve the results in specific cases see Sect. 7.

6.4.1 fors_calib

The interactive window shown in Fig. 4.4 (p. 13) provides information about the quality of the wavelength calibration (left column), distortion correction (top right and center plot) and the flat field combination and normalization (bottom right plot). The plots contain in detail:

Top Left *Wavelength-calibrated arc lamp frame:* In this plot the arc lamp lines should run straight from top to bottom without any empty rows between them. Some arc lines may show gaps due to the placement of the slits, but empty rows without any lines point towards problems with the detection of the arc lamp lines.

Center Left *Positions of detected arc lines in rectified frame* Here the x/y positions of the detected (black) and identified (green) arc lines in the rectified frame are plotted. Dark/light green mark lines identified in the first attempt and during the search around identified lines (`wradius>0`). Again, there should be no empty rows and most of the detected lines should be identified. Unidentified lines at the edges of the frame are especially problematic, as they cause unnecessary extrapolations of the dispersion relation.

Bottom Left *Residuals between predicted and detected arc line position* The residuals should generally be below 0.5 pixel. If they show systematic variations the polynomial degree used to fit the dispersion relation may be too low (or in rare cases too high). If the scatter appears very large one should zoom in, because there are often only a few outliers and the majority of the residuals are within ± 0.5 pixels.

Top Right *Spatial map* The spatial map has as pixel values the distance of a pixel from the bottom of the respective slitlet. The regions of the slitlets should not be strongly curved nor should regions of different slitlets overlap with each other.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	27 of 45

Center Right *First raw flat* This plot is mostly of interest in comparison to the **Bottom Right** one, as the number of slitlets and the areas covered by them should be identical. The red lines show the traces of the slitlet edges (in case of MOS/MXU data). They should therefore follow the slit edges and not cut across slitlets. All slitlets should be detected and there should be no spurious detections (e.g. one slitlet detected as several). For LSS data the illuminated region along the y-axis has to be identical between this plot the bottom right one. Otherwise part of the exposed area is lost.

Bottom Right *Normalized master flat* The normalized master flat field should have the same number of slitlets as the first raw flat and their areas should also be identical. Depending on the normalization method residual gradients may be validly present (e.f. `sradius=-1` as recommended for LSS data).

6.4.2 for_scienc (Response)



Figure 6.4: The interactive window of the Response Curve actor for the first demo DataSet..

The interactive window shown in Fig. 6.4 provides information about the quality of the response curve fit:

Top *Extracted standard star spectrum*: The extracted standard star spectrum should shows no jumps or sky

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	28 of 45

emission lines. Strong gradients due to order separating filters are valid but may cause problems with the fit of the response curve.

Center *Raw response and fit:* The dots show the raw response (ratio of reference spectrum and observed spectrum integrated over same bins as reference spectrum) and the blue line shows the corresponding fit. Blue dots are masked (`ignore_resp_mode`, `ignore_resp_points`) and not used for the fit.

Bottom *Flux-calibrated standard star spectrum and reference:* The red line marks the observed standard star spectrum calibrated with its own response curve and the green and blue points indicate the reference data (blue points were masked during the fitting of the response). Differences between the green points and the red line indicate a problem with the flux calibration, usually features on a scale smaller than the bins of the reference data. Differences between the red line and blue points indicate problems with inter- and/or extrapolation.

6.4.3 fors_science (Science)



Figure 6.5: The interactive window of the ForsScience actor for the first demo DataSet..

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	29 of 45

The interactive window shown in Fig. [6.5](#) provides information about the quality of the sky subtraction and spectrum extraction. It shows the spectra in ADU/sec, i.e. not flux-calibrated:

Top *Mapped sky-subtracted 2-dimensional spectrum:* The wavelength-calibrated, rectified frame is shown after sky subtraction. The yellow and red lines mark the lower/upper extraction limits of the detected spectra. Right-clicking on such a range will plot the extracted spectrum in the **bottom** plot. For LSS data you should check in this window that your spectra are horizontal (see also Frequently Asked Questions, p. [39](#)). This is most easily done by zooming onto a spectrum, selecting the full range in wavelength but only a small range along the y-axis.

Bottom *Extracted science spectrum:* The spectrum should not show strong residuals of sky lines.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	30 of 45

7 Improving Your Results Through Workflow Interaction

In this section we provide information on how to improve your results by changing the parameters of the `fors_calib` and `fors_science` recipes.

7.1 Demo Data

DataSet 1: LSS GRIS_600RI The interactive window of the `ForsScience` actor shows that the sky subtraction did not work well. Setting `skylocal` to `FALSE` and `skyglobal` to `TRUE` will improve the situation.

DataSet 4: LSS GRIS_1400V In this dataset the stellar features of the standard star do not cause problems in the response, so one may remove `stellar_absorption` from `resp_ignore_mode`. Setting `skyglobal` to `TRUE` and `skylocal` to `FALSE` improves the sky correction. In this case also `cosmics = TRUE` improves the results.

DataSet 5: MOS GRIS_300V With default parameters `fors_calib` gives the error message “Recipe error: The wavelength solution at row 96 does not increase monotonically, which is physically impossible. Try with new parameters”. Changing `wdegree` from 5 to 4 shows some spurious lines between the two bottom slits that cause the problem. Increasing `peakdetection` to 500 allows to set `wdegree` again to 5.

The response curve obviously provides an unsatisfactory fit at the blue end ($\lambda < 4800\text{\AA}$) and in some of the telluric regions as seen in the plot of the flux-calibrated standard star at the bottom of the interactive window (see Fig. 7.1). A closer inspection of the telluric regions in the central plot shows that only a few data points deviate. Removing `telluric` from `resp_ignore_mode` and adding 7550–7700, 8150, 8200 to `resp_ignore_points` will improve the fit in the telluric regions (see Fig. 7.2). To improve the fits also in the blue an increase of `resp_fit_nknots` to 40 is necessary (see Fig. 7.3).

The `ForsScience` interactive window shows that some spectra were not detected. This problem is known and may be solved in a later release. Until then you have to use some other software of your choice (e.g. MIDAS, IRAF) to extract the missing spectra.

7.2 Wavelength Calibration

To verify the quality of the wavelength calibration the user should check the displays in the interactive window, which shows up after the execution of the `fors_calib` recipe (see Sect. 6.4.1 and Fig. 4.4, p. 13 for an example).

While the parameters in the grism tables should generally provide good results some grisms are sensitive to the positions of the slits on the CCD, the line intensities, etc. In these cases it may make sense to try changing certain parameters to improve the results. For problematic cases setting `wmode` (LSS/LSS-like) or `wmosmode` (MOS/MXU) to 0 can be helpful as then only lines that are actually found and detected are used, without interpolation along the spatial axis. For some grisms additional lines are listed in the line catalogues, that may

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	31 of 45

be useful to process slitlets with large offsets from the center along the dispersion axis. They can be included by setting `used_linesets` to `standard`, `extended`.

Below we list suggestions to solve problems that may occur during wavelength calibration:

Non-monotonic dispersion relation

The pipeline may sometimes fail with the error message *The wavelength solution at row <number> does not increase monotonically, which is physically impossible. Try with new parameters*". In such cases the user should try to decrease `wdegree` and/or increase `peakdetection`. A further possibility is to decrease `wradius`.

Few slitlets

For MOS/MXU observations with few (usually three or fewer) slitlets the slit identification may fail. Using `slit_ident=FALSE` allows the user to process such data.

GRIS_150I+OG590

This grism may need tailored parameter settings for each setup. Increasing `peakdetection` to 300–400 and/or changing `wradius` to higher or lower values may provide better results for certain masks. Changing the `dispersion` within the range 2.9 to 3.8 may be helpful as well.

GRIS_300V+GG435

Reducing `wdegree` from 5 to 4 produces in some cases more stable results. In some cases increasing `peakdetection` to 350 can improve the slit tracing.

GRIS_300I+OG590

Bad slit tracing can often be improved by increasing `peakdetection` to 400. Changing `wdegree` to higher or lower values can also help.

GRIS_600B

Setting `wdegree` to 5 sometimes gives better results for both wavelength calibration and slit tracings. Also changing `wradius` to lower or higher values can help.

GRIS_600RI+GG435

In some cases the solution can be improved by changing `wdegree` to 3 or 5.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	32 of 45

GRIS_1200B

For this grism it is sometimes impossible to find a solution for all slitlets, especially if the slitlets are widely distributed along the dispersion direction. The user may have to process the data twice, once optimizing for the redder slitlets and once for the bluer ones. In that case varying dispersion between 0.32 (blue slits) and 0.38 (red slits) may give good results for extreme slits. The wavelength calibration reacts very sensitively to the presence or absence of the line 4077 Å. If results stay unstable the user should ignore one or more of the following lines via the `ignore_lines` parameter: 3466, 3611, 3965, 4026, 5331, 5401.

GRIS_1400V

Decreasing `wradius` to 2 and/or increasing `peakdetection` to 90 or 120 can help to improve unstable solutions. Some solutions especially for MOS/MXU data can be improved by increasing `wdegree` to 4.

GRIS_1028z+OG590

The solution can be improved in some cases by increasing `wdegree` to 4. Decreasing `wradius` to 2 may improve unstable solutions.

7.3 Flat Field Normalization

In general the user will want to remove all traces of the flat field lamp and the slit profile (spatial illumination) and keep only the pixel-to-pixel variation of the detector. The default parameters of the flat field normalization are optimized for this situation. The spline fit along the dispersion axis is not well suited for data with very steep slopes due to order separation filters (e.g. GRIS_150I+OG590).

For LSS data it can be useful not to normalize the flat field along the spatial axis (i.e. setting both `sradius` and `s_degree` to -1), but instead keep the illumination profile caused by the slit to correct the illumination profile of the science data and thereby facilitate the fitting of the sky background.

7.4 Response Determination

The reference data for the flux standard stars have entries that identify the presence of strong stellar features (column `STLLR_ABSORP` set to T(rue)). These regions will be ignored during the response fit if `resp_ignore_mode` is set to `stellar_absorption`. Depending on the resolution of the grism such masking may or may not be necessary. If additional regions need to be ignored `resp_ignore_mode` set to `stellar_absorption`, `command_line` allows to specify the additional points (e.g. 4750, 4900) and/or regions (e.g. 8900–9850) with `resp_ignore_points`.

In order to avoid the fitting of regions affected by telluric absorption the static calibration table `fors2_telluric_regions.fits` contains such regions (per grism, as their width depends on spectral resolution). These can be masked by setting `resp_ignore_mode` to `telluric`. If additional regions need to be ignored `resp_ignore_mode` set to `telluric`, `command_line` allows to specify the additional

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	33 of 45

points/regions with `resp_ignore_points`. This is especially useful for data with very strong variations at the edges of the response (e.g. the 4th demo dataset, see above).

By default `resp_ignore_mode` is set to `stellar_absorption`, `telluric`, `command_line`.

To verify the quality of the response determination the user should check the following displays in the interactive window, which shows up after the execution of the `fors_science` recipe in the `ResponseCurve` actor (see Fig. 7.1):

Raw and fitted response The blue curve (fit) in the central plot should follow the green dots (unmasked raw response) closely. If the fit deviates strongly in the masked regions (blue dots) the user may try to remove `stellar_absorption` and/or `telluric` from the parameter `resp_ignore_mode` and mask only smaller regions via `resp_ignore_points`¹². Alternatively or in addition one may change the fit method and/or degree via `resp_fit_degree` (polynomial degree) or `resp_fit_nknots` (number of spline knots).

One should keep in mind that interpolating across a larger region (e.g. covered by telluric lines) provides at best a probable response curve, but may also show strong deviations. This is even more true for extrapolations, e.g. if a response curve has telluric regions at its edges.

Flux-calibrated standard star The red curve (flux-calibrated standard star spectrum) in the lower plot should coincide with the green dots (tabulated standard star flux; blue dots were ignored during the response fit). Deviations at the very blue end may results from problems with atmospheric extinction and deviations at wavelengths above 6000 Å may point to problems with telluric absorption.

In order to take small scale variations into account when fitting the response the pipeline uses a spline fit with the number of knots defined with `resp_fit_nknots`. For grisms with a small wavelength range and/or only moderate large-scale variation in their response a small number of knots can be sufficient. For grisms with small-scale variations a higher number of knots may be necessary (e.g. fifth demo dataset, see above). One should keep in mind that the maximum number of knots for the spline is the number of unmasked data points – 2 (entering a higher number will cause the pipeline to reduce it to the allowed maximum). Since the knots are distributed at equal distances this means that the distance between two knots is always larger than the distance between two data points. This explains why even at a maximum number of knots the fit may not go through all data points.

¹²Please note than an empty field for `resp_ignore_mode` the default value `stellar_absorption`, `telluric`, `command_line` is taken. As a work-around, set `resp_ignore_mode` to `command_line` and `resp_ignore_points` to empty, in order to express that no masking should be applied.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	34 of 45



Figure 7.1: The interactive window of the Response Curve actor for the 5th demo DataSet with default parameters.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	35 of 45

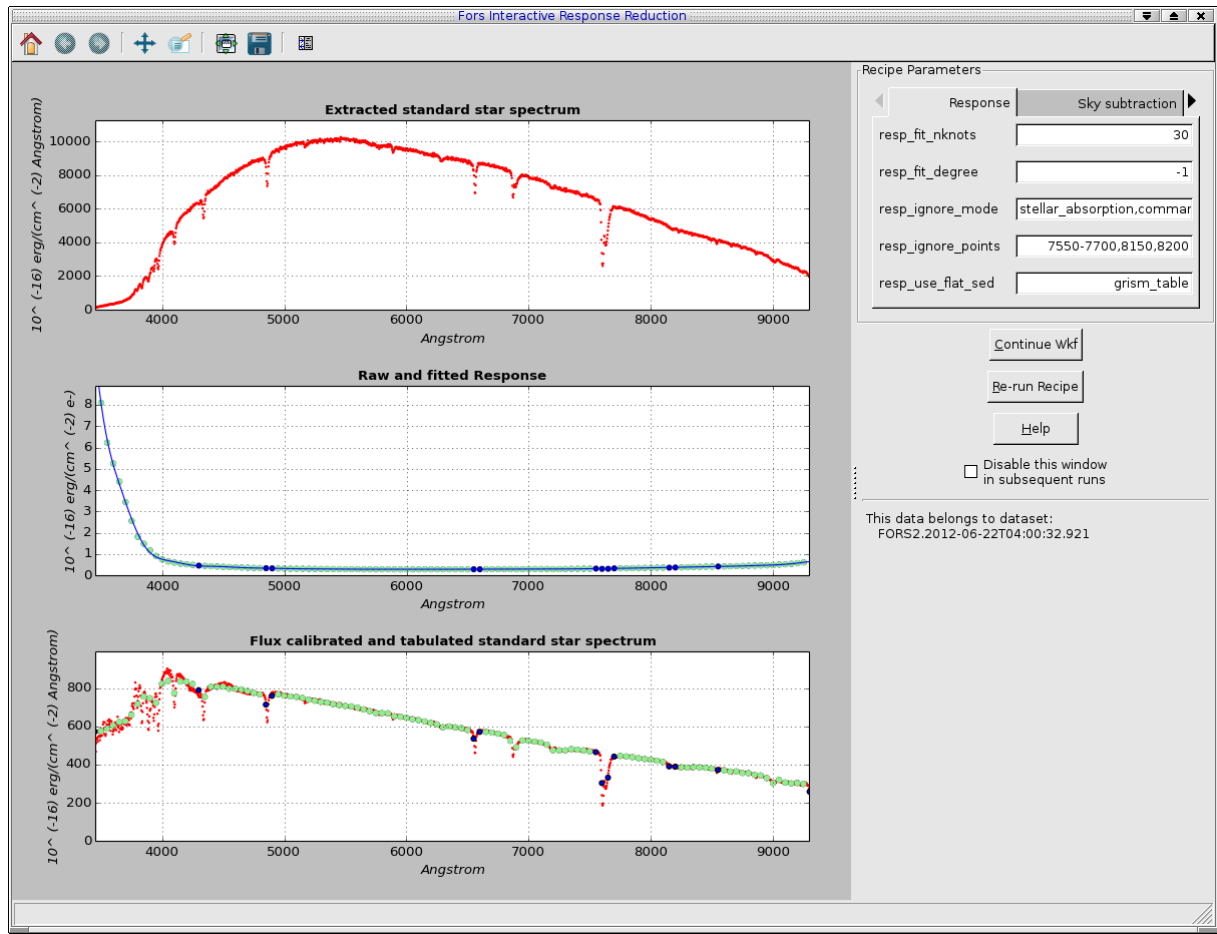


Figure 7.2: The interactive window of the Response Curve actor for the 5th demo DataSet after changing the masking of the telluric regions.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	36 of 45

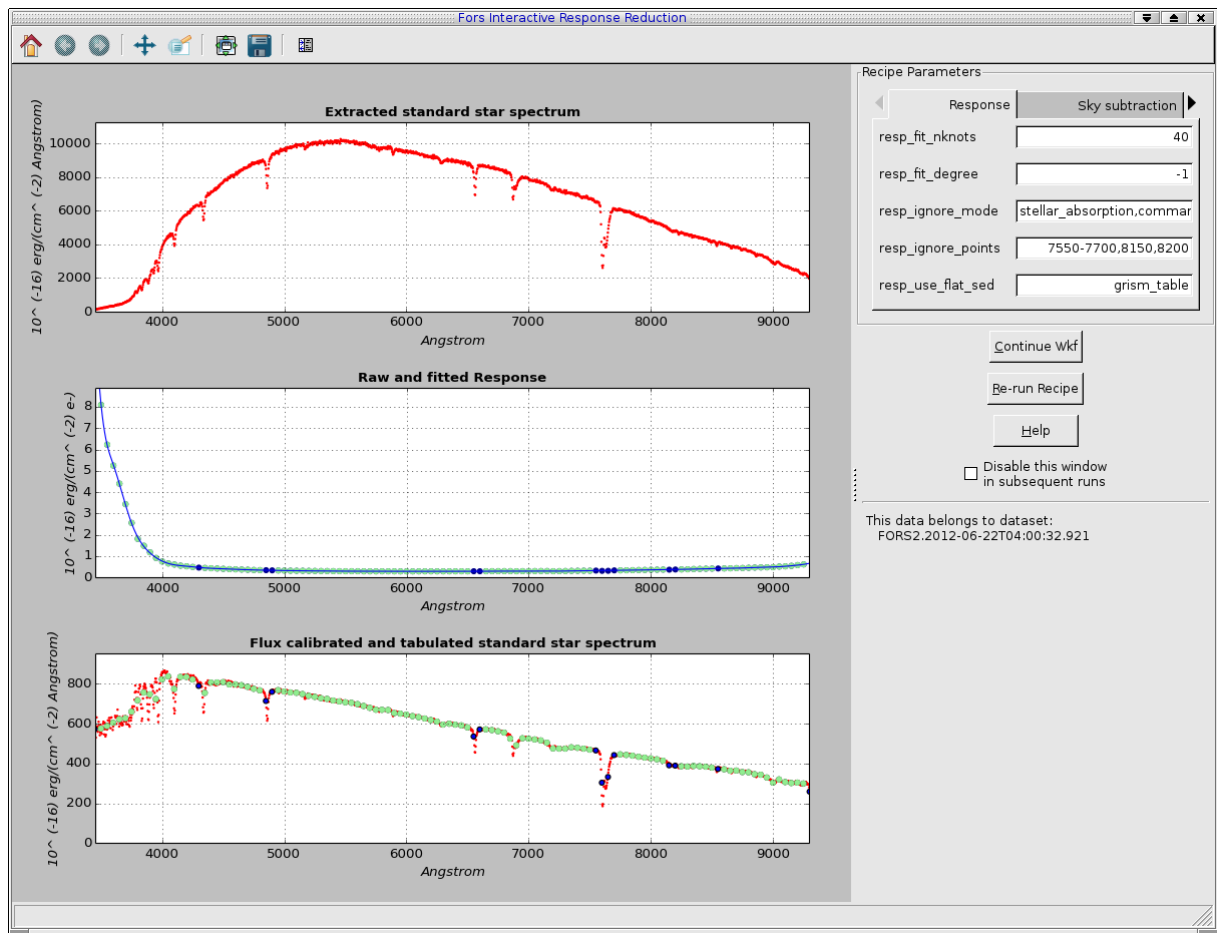


Figure 7.3: The interactive window of the Response Curve actor for the 5th demo DataSet after changing the masking of the telluric regions and increasing the number of knots for the spline fit..

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	37 of 45

8 Frequently Asked Questions

- **Where are my intermediate pipeline products?** Intermediate pipeline products are stored in the directory `<TMP_PRODUCTS_DIR>` (defined on the workflow canvas) 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 (Forchi (2012)). 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:

```
reflex -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). 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 (Forchi (2012)) 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 (Forchi (2012)) for more information.
- **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.txt`. Alternatively, to ensure that the path to `esorex` is the correct one, the user can execute

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

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	38 of 45

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.txt` script.

- If I enter "-" into an empty parameter string it is automatically completed to "-1". Why?

The parameters are validated for correctness according to their type (e.g. string, integer, float). In the case of an integer or float parameter "-" alone is considered an invalid input and is therefore automatically completed to "-1". This is part of the validation of input done by the WxPython library.

8.1 FORS specific questions

- Some datasets are greyed out and a mouse-over yields “Missing GLOBAL_DISTORTION_TABLE”. What can I do?

First of all, you can process the data also without this calibration by clicking on the box in the “Select” column of the Data Set Chooser or using the `Select all` button. Then the spatial distortion of your LSS data is not corrected, which is of importance mainly if you are interested in the spatial information about your targets. The extraction quality of the spectra is generally affected only at the very edges of the field.

The `GLOBAL_DISTORTION_TABLES` do not exist for the FORS2 grisms `GRIS_200I`, `GRIS_600V`, `GRIS_600R` nor for any FORS1 grism. It exists for the grisms `GRIS_300V` and `GRIS_300I` that have been in use in FORS2 since April 2009. To use the `GLOBAL_DISTORTION_TABLES` of these two grisms for older data you need to change the OCA rules (located at `<install_dir>/share/esopipes/<pipeline-version>/reflex/fors_spec_wkf.oca`). You have to change the part that associates the `GLOBAL_DISTORTION_TABLE`

```
select file as GLOBAL_DISTORTION_TABLE from calibFiles where
PRO.CATG=="GLOBAL_DISTORTION_TABLE" and
inputFile.INS.GRIS1.ID==INS.GRIS1.ID and
inputFile.DET.CHIP1.X==DET.CHIP1.X and
inputFile.DET.CHIP1.Y==DET.CHIP1.Y and
inputFile.DET.CHIP1.PSZX==DET.CHIP1.PSZX and
inputFile.DET.CHIP1.PSZY==DET.CHIP1.PSZY and
inputFile.INS.COLL.NAME==INS.COLL.NAME and
inputFile.INSTRUME==INSTRUME;
```

replacing

```
inputFile.INS.GRIS1.ID==INS.GRIS1.ID
```

by

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	39 of 45

```
inputFile.INS.GRIS1.NAME==INS.GRIS1.NAME
```

While the old and new grisms should be very similar there is, however, no guarantee and you should check your data carefully.

- **I get a warning “Column STLLR_ABSORP not found in std star table. Value ‘stellar_absorption’ in ‘resp_ignore_mode’ is ignored” if I use the standard star tables that have been downloaded from the archive with CalSelector.**

For the time being the ESO archive contains old versions of the flux standard star tables. Therefore, until the archive is updated, you should remove these files from your input files. Since the static calibrations are also delivered together with the latest version of the pipeline, you don’t really need them, Reflex will take the ones under directory `CALIB_DATA_DIR`.

- **There are no extracted spectra - why?**

While this may happen for observations of extremely faint targets the most common case for this behaviour are LSS observations from the lower chip, as the main target is usually placed close to the centre of the field-of-view, which is on the upper chip.

- **The spectra in my rectified LSS data are tilted - why?**

In rare cases the use of the `GLOBAL_DISTORION_TABLE` results in tilted instead of horizontal spectra in the `MAPPED...` data. We are still looking into the causes for this behaviour, so please report the problem (and the data for which you see it) to `usd-help@eso.org`.

- **I am using the maximum number of knots for the spline fit of the response but the fit still does not go through all data points.**

The maximum number of knots for the spline is the number of unmasked data points – 2 (entering a higher number will cause the pipeline to reduce it to the allowed maximum). Since the knots are distributed at equal distances this means that the distance between two knots is always larger than the distance between two data points. This explains why even at a maximum number of knots the fit may not go through all data points. A polynomial of very high degree might achieve that, but is rather unstable.

- **The fit to the response curve shows very strong deviations in the masked regions - how can I avoid that?**

Some grisms have a very steep slope and/or cover small-scale variations of the detector response so that a spline fit with a high number of knots is needed to fit their response. This fit is unconstrained in the masked regions, which can result in very strong deviations. The masking regions have been defined very conservatively. It may well be that the response in the masked regions does not show any significant deviations or only at very few points. In such a case set `resp_ignore_mode` to `command_line` only and add the deviating points at `resp_ignore_points`. Alternatively you can try to exclude the steep part of the response via `resp_ignore_points` and/or use a polynomial fit of lower order, that will not take into account small-scale variations of the response (via `resp_fit_degree` instead of `resp_fit_nknots`).

- **I used `-use_flat_sed=true` for my GRIS_1200B data but the flux-calibrated data look very bad.**

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	40 of 45

Unfortunately the spectral energy distribution of the flat field lamp use for GRIS_1200B data can vary considerably on short timescales. In such cases the `FLAT_SED_<mode>` used to correct the effect of the position-dependent response will differ even if they were taken at the same position on the CCD. So it will be impossible to correct the effect. If you have LSS data it is very likely that the flux standard star was observed at the same position on the CCD as your science target, so that you may use `-use_flat_sed=false`. In the case of MOS/MXU data there is unfortunately no solution for this problem. With `-use_flat_sed=true` you can correct the small scale variations of the detector response at the price of introducing spurious flux variations. Setting `-use_flat_sed=false`, however, will not correct for the position dependency of the response, which can severely distort the SED of your targets, up to reversing the slope of the spectra (i.e. making a hot spectrum look cool and vice versa).

- **Why do my flux-calibrated data not agree with independent photometric measurements?**

In order to have a true absolute flux calibration several requirements need to be fulfilled:

1. The fraction of the total flux of an object that is contained in the slit depends on the shape of the object, the width and orientation of the slit, and the seeing. Absolute flux calibration requires that all the flux of both the object and the standard star has been collected.
2. The flux that arrives at the telescope depends on the transparency of the sky. Absolute flux calibration requires the same transparency for the observations of the target and the standard star.

A change between the flux standard observation and the science object observations of any of the parameters mentioned above will change the flux scale in the final spectrum. To compare the flux calibrated spectrum to other measurements, differences in slit losses and atmospheric conditions have to be taken into account.

With respect to the fraction of flux contained within the slit one should keep in mind that the flux standard stars are observed with a 5'' wide slit, while science data are typically observed with slit widths of 0.8'' to 1''. For a point source a slit width of 0.8''/1.0'' used with a seeing of 0.8'' means that some 33%/24%, respectively, of the target flux are lost (see also Fig. 8.1). This results in a too low flux for the flux calibrated spectrum of the target.

If the standard star or the target or both are observed under non-photometric conditions (e.g. CLR or THN) their observed flux will be lower than it should be. If the standard star is observed under photometric conditions but the science target is not the flux in the flux calibrated target spectrum will be too low. The opposite happens if the target is observed under photometric conditions but the standard star is not. CLR/THN conditions allow for transparency variations of 10%/20%, respectively.

- **Does the pipeline combine different detectors chips into a common product?**

No. The spectroscopic pipeline and Reflex workflow works only on files from the same detector chip. Files from different detectors must not be mixed in the same sof.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	41 of 45

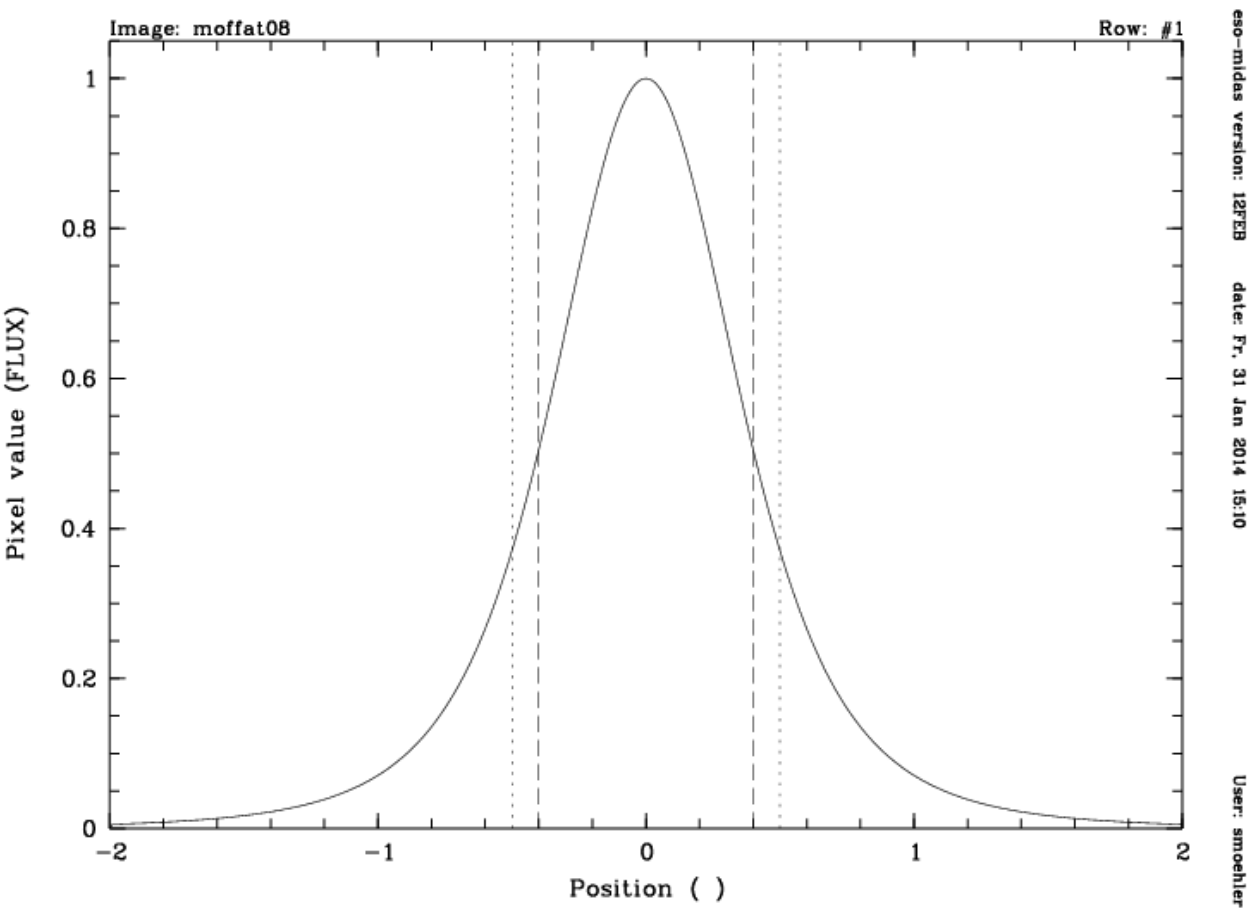


Figure 8.1: Here a Moffat profile with a FWHM of 0.8'' is shown. The dashed and dotted lines mark the limits of a 0.8''/1.0'' slit..

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	42 of 45

9 Troubleshooting



Figure 9.1: *TheDataOrganizer* interactive window reports an error “:No DataSets have been created, check the data set and the OCA rules.”.

1. **I downloaded the data from the ESO archive, put them into a new directory, tried to run `Reflex` on them, but**

- (a) **it crashes**

The current release of FORS includes some additional data in the static calibration frames. The recipes would chock if this data is not present. However, the ESO archive with CalSelector will associate calibration data which is old and Reflex will pick the files either from the installed pipeline static data or from the CalSelector in a non-deterministic way. In order to solve the issue, remove the static calibration data downloaded from the archive (all the files starting with M.FORS2).

This may happen if one of the files was downloaded only partially (check for a file with the extension `fits.Z.part`. You will have to download that file again in order to have an uncorrupted file (and remove the partial one).

- (b) **The DataOrganiser fails with the error message “:No DataSets have been created, check the data set and the OCA rules.”(see Figure 9.1.)**

This error may be due to the fact that the data provided by the ESO archive are compressed (`<filename>.fits.Z`). Please remember to uncompress the data before running the workflow in Reflex.

Also, please remember that the FORS2 workflow supports only spectroscopic data (LSS/MOS/MXU, no PMOS). It is possible that your data consists entirely of IMG/IPOL/PMOS observations, in which case the `Data Organiser` actor will not construct any DataSets, showing the mentioned error message.

2. **The “Select DataSets” window displays my DataSets, but some/all of them are greyed out. What is going on?**

If a DataSet in the “Select DataSets” window is greyed out, then it means that the DataSet which was constructed is missing some key calibration(s) (i.e. the DataSet is incomplete). To find out what calibration(s) are missing from a greyed out DataSet, click on the DataSet in question to highlight it in blue, and then click on the button `Inspect Highlighted`. The “Select Frames” window that appears will

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	43 of 45

report the category of the calibration products that are missing (e.g. MASTER_BIAS). From this the user has then to determine the missing raw data (in this case bias frames). If static calibrations are missing the mechanism unfortunately does not work, but such data should be found by `reflex` in `<install_directory>/calib/<pipeline_version>/cal`

3. **The pipeline fails with the error message “The wavelength solution at row <number> does not increase monotonically, which is physically impossible. Try with new parameters”.**

Non-monotonic dispersion relations are often due to spurious detections. In such cases try to decrease `wdegree` and/or increase `peakdetection`. A further possibility is to decrease `wradius`.

4. **The pipeline fails with the error message “Missing columns WLEN CHEMICAL_ION LINE_SET in input line catalog”**

This is due to the use of an old MASTER_LINECAT delivered by `calSelector`, which does not contain the column `CHEMICAL_ION`. If you remove the corresponding file and rerun the workflow it should pick the correct file from the pipeline installation.

5. **If I set parameter `resp_ignore_mode` to empty in the response interactive window, the value `stellar_absorption,telluric,command_line` is taken**

This is a known bug in Reflex that will be solved in next release. As a work-around, set `resp_ignore_mode` to `command_line` and `resp_ignore_points` to empty, in order to express that no masking has to be done.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	44 of 45

Acknowledgements

The `Reflex` team in alphabetical order consists of Pascal Ballester, Lodovico Coccato, Vincenzo Forchì, Wolfram Freudling, César Enrique García, Maurice Klein Gebbinck, Martino Romaniello.

ESO	Reflex FORS2 Tutorial	Doc:	VLT-MAN-ESO-19500-....
		Issue:	Issue 1.8
		Date:	Date 2015-06-24
		Page:	45 of 45

Forchì V., 2012, Reflex User Manual, VLT-MAN-ESO-19000-5037, Issue 0.7,
<ftp://ftp.eso.org/pub/dfs/reflex/ReflexUserManual-3.1.pdf> 16, 20, 37

Saviane I. et al., FORS User Manual, VLT-MAN-ESO-13100-1543, Issue 91.0 7

Izzo C. et al., FORS Pipeline User Manual, VLT-MAN-ESO-19500-4106, Issue 4.3 7, 20, 24