



# 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 UVES Tutorial

VLT-MAN-ESO-19540-5296

Issue 6.0

Date 2013-05-06

Prepared: Daniel Bramich, Sabine Moehler 2013-05-06  
Name Date Signature

Approved: W.Freudling  
Name Date Signature

Released: M. Sterzik  
Name Date Signature

This page was intentionally left blank

|            |                             |        |                         |
|------------|-----------------------------|--------|-------------------------|
| <b>ESO</b> | <b>Reflex UVES Tutorial</b> | Doc:   | VLT-MAN-ESO-19540-5296  |
|            |                             | Issue: | Issue 6.0               |
|            |                             | Date:  | Date 2013-05-06         |
|            |                             | Page:  | 3 of <a href="#">44</a> |

### Change record

| Issue/Rev. | Date       | Section/Parag. affected | Reason/Initiation/Documents/Remarks |
|------------|------------|-------------------------|-------------------------------------|
| 1.1        | 15/01/2011 | All                     | Public release                      |
| 2.0        | 26/04/2011 | Updated details         | Update                              |
| 3.0        | 05/04/2012 | All                     | Updated for Reflex 2.0              |
| 5.0        | 22/05/2012 | All                     | Updated for Reflex 2.1              |
| 5.1        | 21/12/2012 | All                     | Updated for Reflex 2.3              |
| 6.0        | 06/05/2013 | All                     | Updated for Reflex 2.4              |

This page was intentionally left blank

|            |                             |        |                        |
|------------|-----------------------------|--------|------------------------|
| <b>ESO</b> | <b>Reflex UVES Tutorial</b> | Doc:   | VLT-MAN-ESO-19540-5296 |
|            |                             | Issue: | Issue 6.0              |
|            |                             | Date:  | Date 2013-05-06        |
|            |                             | Page:  | 5 of 44                |

## Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction And Scope</b>                               | <b>6</b>  |
| <b>2</b> | <b>Software Installation</b>                                | <b>7</b>  |
| <b>3</b> | <b>Demo Data</b>  | <b>8</b>  |
| <b>4</b> | <b>Quick Start: Reducing The Demo Data</b>                  | <b>9</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 UVES Workflow</b>                                    | <b>14</b> |
| 6.1      | Workflow Canvas Parameters . . . . .                        | 14        |
| 6.2      | Workflow Actors . . . . .                                   | 14        |
| 6.2.1    | Simple Actors . . . . .                                     | 14        |
| 6.2.2    | Composite Actors . . . . .                                  | 15        |
| 6.2.3    | Recipe Execution within Composite Actors . . . . .          | 17        |
| 6.2.4    | Lazy Mode . . . . .   | 19        |
| 6.3      | Workflow Steps . . . . .                                    | 20        |
| 6.3.1    | Step 1: Data Organisation And Selection . . . . .           | 20        |
| 6.3.2    | Step 2: Creation Of Master Calibration Files . . . . .      | 23        |
| 6.3.3    | Step 3: Wavelength And Response Calibration . . . . .       | 23        |
| 6.3.4    | Step 4: Spectrum Extraction . . . . .                       | 24        |
| 6.3.5    | Step 5: Output Organisation . . . . .                       | 24        |
| <b>7</b> | <b>Optimising Your Results Through Workflow Interaction</b> | <b>27</b> |
| <b>8</b> | <b>Frequently Asked Questions</b>                           | <b>38</b> |
| <b>9</b> | <b>Troubleshooting</b>                                      | <b>40</b> |

|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 6 of 44                |

## 1 Introduction And Scope

Reflex is the ESO Recipe Flexible Execution Workbench, an environment to run ESO VLT pipelines which employs a workflow engine (Kepler<sup>1</sup>) to provide a real-time visual representation of a data reduction cascade, called a workflow, which can be easily understood by most astronomers.

Reflex and the data reduction workflows have been developed at ESO 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 UVES 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 tool<sup>2</sup> 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 through 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 and employing user-configurable file names.

This tutorial deals with the reduction of **blue and red arm UVES echelle data for point source observations only** via the UVES Reflex workflow. The user is referred<sup>3</sup> to the UVES user manual (Kaufer et al. 2010) for more information on the instrument itself, and the UVES pipeline user manual (Larsen et al. 2012) for the details of UVES pipeline recipes. The quick start section (see Section 4) describes the minimum effort to get started, and it makes up only two pages of this tutorial.

---

<sup>1</sup><http://kepler-project.org>

<sup>2</sup><http://www.eso.org/sci/archive/calselectorInfo.html>

<sup>3</sup>These documents are available from <http://www.eso.org/sci/facilities/paranal/instruments/uves/doc/index.html>

|            |                             |        |                        |
|------------|-----------------------------|--------|------------------------|
| <b>ESO</b> | <b>Reflex UVES Tutorial</b> | Doc:   | VLT-MAN-ESO-19540-5296 |
|            |                             | Issue: | Issue 6.0              |
|            |                             | Date:  | Date 2013-05-06        |
|            |                             | Page:  | 7 of 44                |

## 2 Software Installation

The software pre-requisites for Reflex 2.4 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 2.4 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_reflex
```

2. Make the installation script executable:

```
chmod u+x install_reflex
```

3. Execute the installation script:

```
./install_reflex
```

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 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.
5. To start Reflex, issue the command:

```
<install_dir>/bin/reflex
```

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 UVES Tutorial</b> | Doc:   | VLT-MAN-ESO-19540-5296 |
|            |                             | Issue: | Issue 6.0              |
|            |                             | Date:  | Date 2013-05-06        |
|            |                             | Page:  | 8 of 44                |

### 3 Demo Data

Together with the pipeline you will also receive a demo data set, that allows you to run the `Reflex UVES` 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  $\sim 1.2$  GB,  $\sim 0.9$  GB and  $\sim 2.3$  GB of free disk space for the directories `<download_dir>`, `<install_dir>` and `<data_dir>`, respectively, at least  $\sim 2.0$  GB of total memory (physical + swap), and a screen with a resolution better than 1024x800 pixels .

The UVES demo data include five blue arm science observations, and three red arm science observations, covering a range of signal-to-noise and object types. In addition we provide one incomplete `RED` data set taken with central wavelength 760 nm to illustrate how an incomplete `DataSet` looks in the “Select DataSets” window (see Fig. ??).



|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 9 of 44                |

## 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 UVES demo data set supplied with the Reflex 2.4 release. By following these steps, the user should have enough information to attempt a reduction of his/her own data without any further reading:

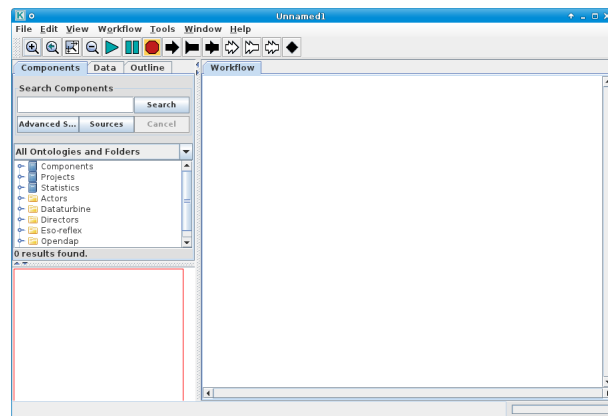


Figure 4.1: *The empty Reflex canvas.*

1. Start the Reflex application:


```
reflex &
```

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

2. Now open the UVES workflow by clicking on File -> Open File, selecting first `uves-5.2.0` and then the file `Uves.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 data<sup>4</sup>, 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.

<sup>4</sup>If 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 UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 10 of 44               |

- Click the  button to start the workflow
- The workflow will highlight the Data Organiser actor which has recursively scanned the raw data directory (specified by the parameter RAWDATA\_DIR under “Setup Directories” in the workflow canvas) and constructs the DataSets. Note that the calibration and reference 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.
- The Data Set Chooser actor will be highlighted next and will display a “Select Datasets” window (see Figure 4.3) that lists the DataSets along with the values of a selection of useful header keywords<sup>5</sup>. 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.
- 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.

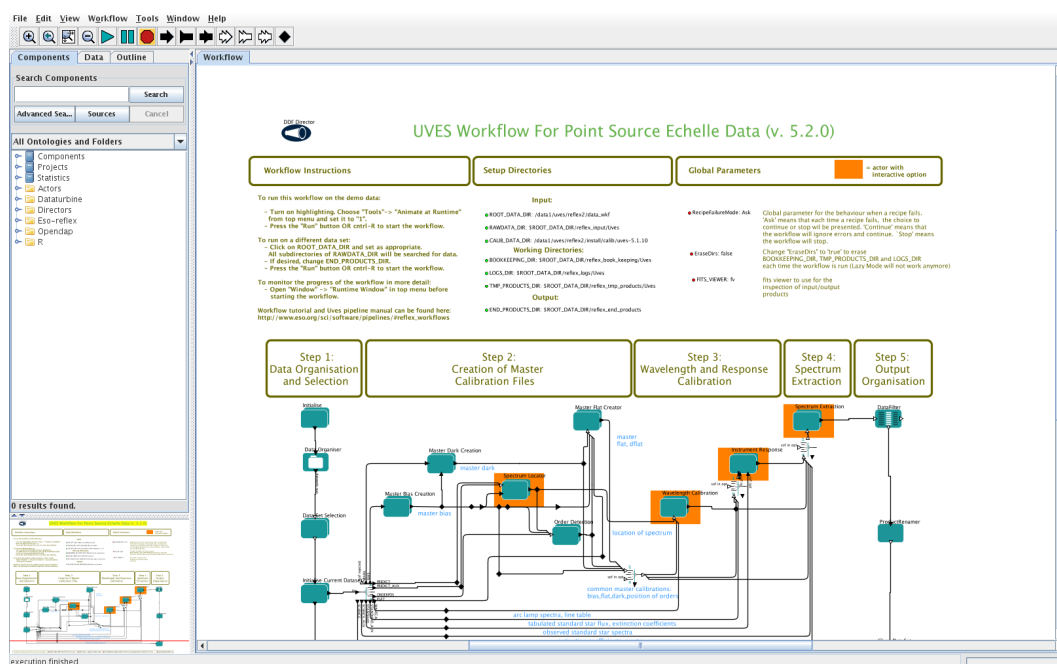


Figure 4.2: UVES workflow general layout.

- When the workflow has finished executing the final pipeline recipe `uves_obs_scired` in the Spectrum Extraction actor for the first DataSet, an interactive window will appear (see Figure 4.4) which shows a plot of the extracted and merged spectrum in the top panel, and the flux-calibrated version in the panel below. Using the buttons at the top of this window, one may pan and zoom in on the spectrum in order

<sup>5</sup>The 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. Make sure that the Lazy Mode is not active and then click on Commit to save the change.

|            |                             |        |                        |
|------------|-----------------------------|--------|------------------------|
| <b>ESO</b> | <b>Reflex UVES Tutorial</b> | Doc:   | VLT-MAN-ESO-19540-5296 |
|            |                             | Issue: | Issue 6.0              |
|            |                             | Date:  | Date 2013-05-06        |
|            |                             | Page:  | 11 of 44               |

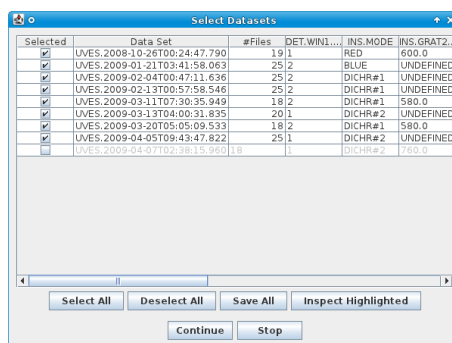


Figure 4.3: The “Select Datasets” pop-up window. On purpose the demo includes an incomplete (grey) DataSet.

to inspect absorption/emissions lines and other interesting spectral features. Of the other plots in this window, the most important one is the extracted S/N of the spectrum as a function of the spectral order as determined from the CCD noise model (and extraction procedure).

10. When the reduction of the current DataSet finishes, a pop-up window will appear showing the directory where the final products have been saved.
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 UVES workflow that merit a look at the rest of this tutorial.

|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 12 of 44               |

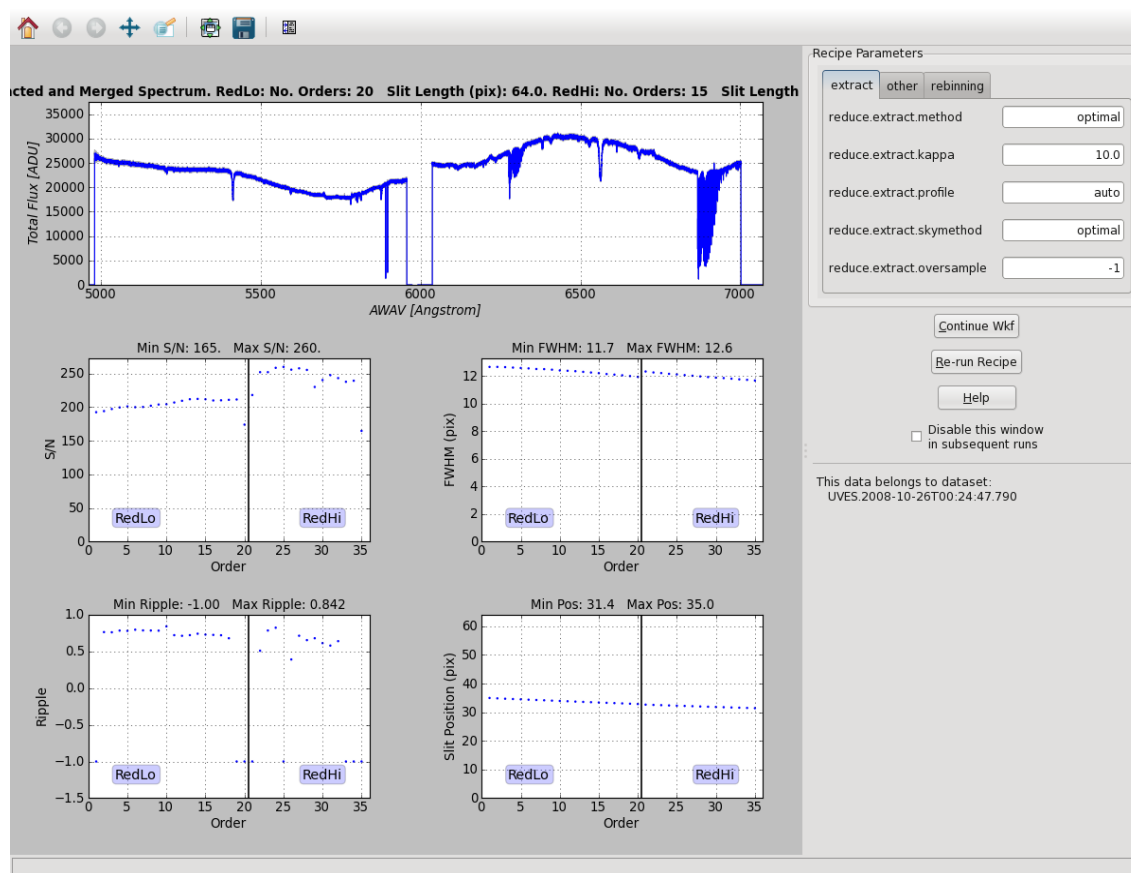


Figure 4.4: The interactive pop-up window for the Spectrum Extraction actor and UVES pipeline recipe `uves_obs_scired`. The extracted and merged spectrum for the first DataSet is displayed in the top panel, and the flux-calibrated spectrum is displayed in the panel below.

|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 13 of 44               |








## 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. Note that after clicking the  button, it is possible that more than one actor is executed, since this behaviour depends on the workflow scheduling. For instance, if there are two actors in parallel, and you pause the workflow while one is being executed, then both of them will be executed before the workflow is actually paused. After pausing, the workflow may be resumed by clicking the  button again.

|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 14 of 44               |

## 6 The UVES Workflow

The UVES 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 `ROOT_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. Raw data should be stored in a subdirectory of `ROOT_DATA_DIR`, defined by the parameter `RAWDATA_DIR`, which is recursively scanned by the `Data Organiser` actor for input raw data. The directory `CALIB_DATA_DIR`, which is 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.





### 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:




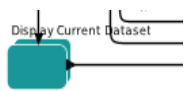


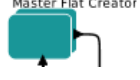


|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 15 of 44               |

-  - The Data Set Chooser actor.
-  - The Fits Router actor
-  - The Product Renamer actor.
-  - The Data Filter actor.

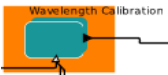


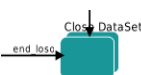
Access to the parameters for a simple actor is achieved by right-clicking on the actor and selecting `Configure Actor`. This will open an “Edit parameters” window. Note that the `Product Renamer` actor is a `jython` script (Java implementation of the Python interpreter) meant to be customised by the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

### 6.2.2 Composite Actors

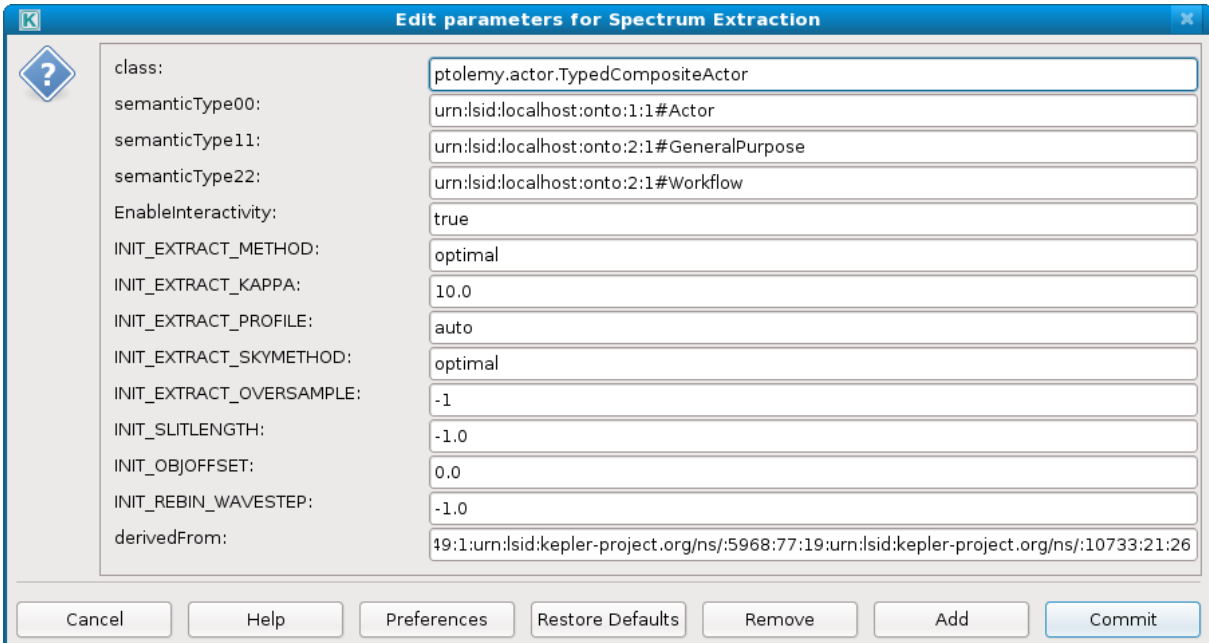
Composite actors have workflow symbols that consist of multiply-layered green-blue rectangles. They generally do not have a logo within the rectangle. A composite actor represents a combination of more simple or composite actors which hides over-complexity from the user in the top-level workflow. An orange box marks an actor with an associated interactive window. In the UVES workflow, the following actors are composite actors:

-  - The Initialise actor.
-  - The Display Current DataSet actor.
-  - The Master Bias Creation actor.
-  - The Master Dark Creation actor.
-  - The Master Flat Creation actor.
-  - The Spectrum Locator actor.
-  - The Order Detection actor.

|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 16 of 44               |

-  - The Wavelength Calibration actor.
-  - The Instrument Response actor.
-  - The Spectrum Extraction actor.
-  - The Close DataSet actor.

Access to the parameters for a composite actor is achieved by right-clicking on the actor and selecting **Configure Actor**. This will open an “Edit parameters” window (see Figure 6.1).



| Parameter                | Value   |
|--------------------------|---|
| class:                   | ptolemy.actor.TypedCompositeActor   |
| semanticType00:          | urn:lsid:localhost:onto:1:1#Actor   |
| semanticType11:          | urn:lsid:localhost:onto:2:1#GeneralPurpose  |
| semanticType22:          | urn:lsid:localhost:onto:2:1#Workflow  |
| EnableInteractivity:     | true  |
| INIT_EXTRACT_METHOD:     | optimal   |
| INIT_EXTRACT_KAPPA:      | 10.0  |
| INIT_EXTRACT_PROFILE:    | auto  |
| INIT_EXTRACT_SKYMETHOD:  | optimal   |
| INIT_EXTRACT_OVERSAMPLE: | -1  |
| INIT_SLITLENGTH:         | -1.0  |
| INIT_OBJOFFSET:          | 0.0   |
| INIT_REBIN_WAVESTEP:     | -1.0  |
| derivedFrom:             | 49:1:urn:lsid:kepler-project.org/ns/:5968:77:19:urn:lsid:kepler-project.org/ns/:10733:21:26 |

Figure 6.1: The “Edit parameters” window for a typical composite 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 `RecipeExecutor`.



|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 17 of 44               |

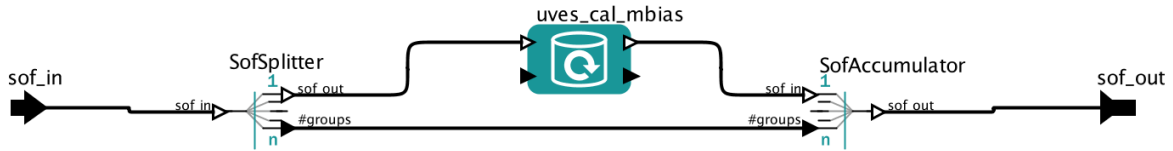


Figure 6.2: This is the window you get when you choose Open Actor for the composite actor Master Bias Creation. This is the most simple case of a composite actor in the UVES workflow.

### 6.2.3 Recipe Execution within Composite Actors

The UVES workflow contains composite actors to run the pipeline recipes. This is in the most simple case due to the `SoF Splitter/Accumulator`<sup>6</sup>, which allows to process calibration data from different settings within one given DataSet (e.g. bias frames with different binnings, dark frames with different DIT, etc.). More complex composite actors contain several actors.

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 Figure 6.2, which is the most simple version of a composite actor in the UVES workflow. 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 the SoFs independently. 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 bias frames. The different bias frames are to be used to de-bias different frames, e.g. the science frames, the various arc lamp and flat field frames and the standard star frames. The `Data Organiser` determines and records this “purpose” of each bias, and this information is included in the DataSet and each SoF created from this DataSet. The `FitsRouter` directs all raw biases to the `Master Bias Creation` composite actor. The `SofSplitter` then creates SoFs, one for the biases to be used for the science images, and (possibly) separate ones for the biases to be used for the calibration images and flat fields. The `uves_cal_mbias` recipe creates one master bias for each SoF, and the `SoAccumulator` then creates a SoF that contains all the master biases. Table 6.2.3 shows which actors contain which recipes and what they do.

A `RecipeExecutor` actor is used in the workflow to run a single UVES pipeline recipe (e.g. the “Master Bias Creation” actor runs the `uves_cal_mbias` pipeline recipe). In order to configure a `RecipeExecutor`, one has to first use `Open Actor` (possibly repeatedly) to get to the level of the `RecipeExecutor` actors (see Figures 6.3 & 6.4).

<sup>6</sup>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\_UVB)

|            |                             |        |                        |
|------------|-----------------------------|--------|------------------------|
| <b>ESO</b> | <b>Reflex UVES Tutorial</b> | Doc:   | VLT-MAN-ESO-19540-5296 |
|            |                             | Issue: | Issue 6.0              |
|            |                             | Date:  | Date 2013-05-06        |
|            |                             | Page:  | 18 of 44               |

Table 6.1: The UVES pipeline actors and their contents

| actor                  | recipes           | description                        |
|------------------------|-------------------|------------------------------------|
| Master Bias Creation   | uves_cal_mbias    | create master bias                 |
| Master Dark Creation   | uves_cal_mdark    | create master dark                 |
| Spectrum Locator       | uves_cal_predict  | create line and order guess tables |
| Order Detection        | uves_cal_orderpos | determine centers of orders        |
| Master Flat Creation   | uves_cal_mflat    | create master flat                 |
| Wavelength Calibration | uves_cal_wavecal  | determine wavelength solution      |
| Instrument Response    | uves_cal_response | determine response function        |
| Spectrum Extraction    | uves_obs_scired   | reduce science data                |

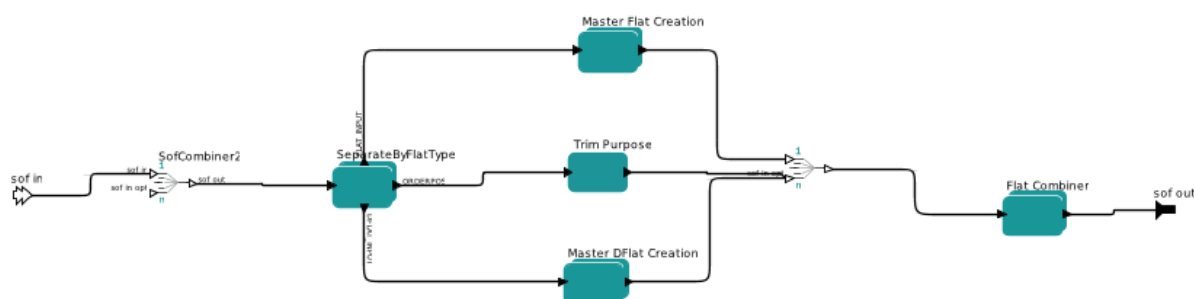


Figure 6.3: The window you get when you choose Open Actor for the composite actor Master Flat Creation. Using Open Actor on Master Flat Creation gives you Figure 6.4.

In Figure 6.5 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 UVES 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 the next section.

|            |                             |        |                        |
|------------|-----------------------------|--------|------------------------|
| <b>ESO</b> | <b>Reflex UVES Tutorial</b> | Doc:   | VLT-MAN-ESO-19540-5296 |
|            |                             | Issue: | Issue 6.0              |
|            |                             | Date:  | Date 2013-05-06        |
|            |                             | Page:  | 19 of 44               |



Figure 6.4: This is the window you get when you choose Open Actor for the composite actor Master Flat Creation *within* Master Flat Creator.

- 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 is the default option.
- The set of parameters, which start with “recipe\_param\_” and end with a number, corresponds to the parameters of the relevant UVES 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 value. For more information on the UVES pipeline recipe parameters, the user should refer to the UVES pipeline user manual (Larsen et al. 2012<sup>7</sup>).

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 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

<sup>7</sup> Available from <http://www.eso.org/sci/facilities/paranal/instruments/uves/doc/index.html>

|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 20 of 44               |

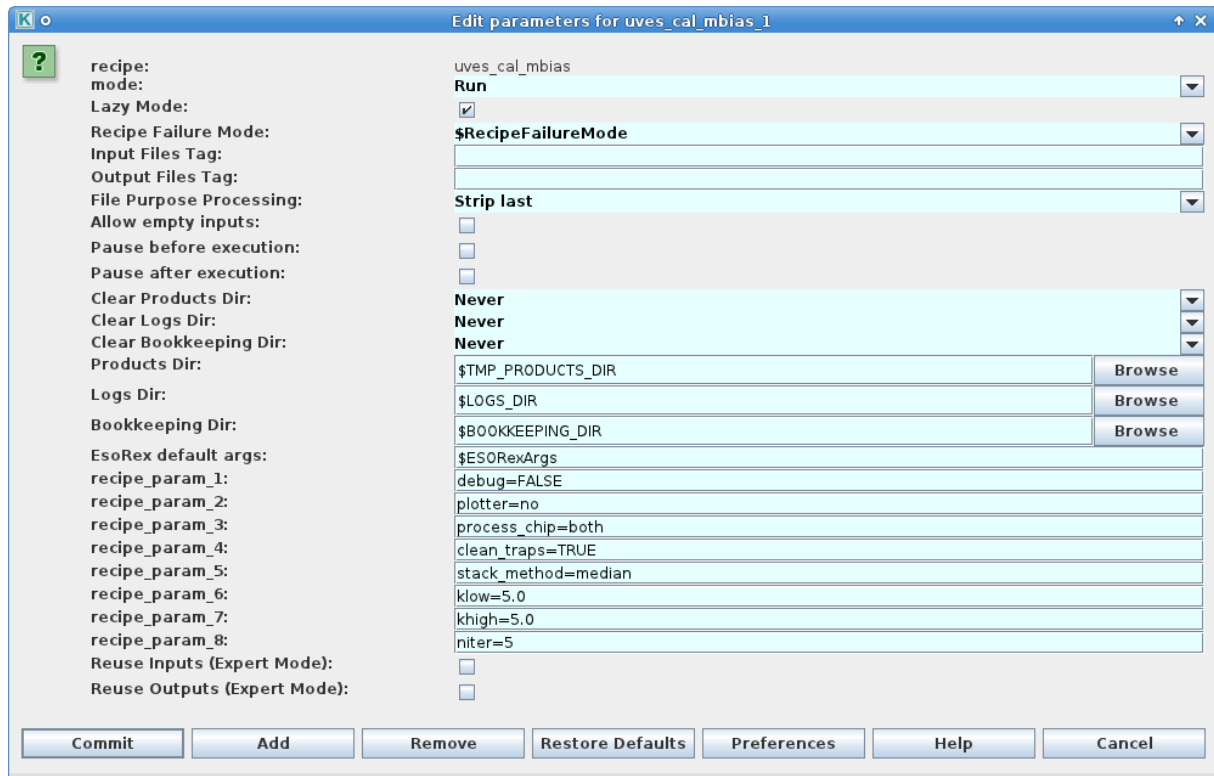



Figure 6.5: The “Edit parameters” window for a typical RecipeExecutor actor; the `uves_cal_mbias` actor which runs the `uves_cal_mbias` pipeline recipe.

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.

## 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).

|            |                             |        |                        |
|------------|-----------------------------|--------|------------------------|
| <b>ESO</b> | <b>Reflex UVES Tutorial</b> | Doc:   | VLT-MAN-ESO-19540-5296 |
|            |                             | Issue: | Issue 6.0              |
|            |                             | Date:  | Date 2013-05-06        |
|            |                             | Page:  | 21 of 44               |

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 “Data Sets”. 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`.

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 OCA<sup>8</sup> rules to find the files to include in a `DataSet`, as well as their categories and purposes. The file category identifies different types of files. A category could for example be `FLAT_BLUE`, `DARK_RED`, or `SCIENCE_BLUE`. 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 `MASTER_BIAS_BLUE` and `EFLAT_ECH_BLUE`. The former creates a master bias from raw biases, and the latter creates a master flat from raw flats. The `EFLAT_ECH_BLUE` action needs raw flats and the master bias as input. In this case the biases will have the purpose `MASTER_BIAS_BLUE/EFLAT_ECH_BLUE`. The same `DataSet` might also include biases with a different purpose, e.g. `MASTER_BIAS_BLUE/MASTER_DARK_BLUE`. Irrespective of their purpose the file category for all these biases will be `BIAS_BLUE`.

2. Next the `Data Set Chooser` displays the available `DataSets` in the “Select Datasets” window<sup>9</sup>, activating a vertical scroll bar on the right if necessary (see Figure ??). 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 second column (see Figure 6.6). 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 6.6), 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”

<sup>8</sup>OCA stands for OrganisationClassificationAssociation and refers to rules, which allow to classify the raw data according to the contents of the header keywords, organise them in appropriate groups for processing, and associate the required calibration data for processing. They can be found in the directory `<install_dir>/share/esopipes/<pipeline-version>/reflex/`, carrying the extension `.oca`

<sup>9</sup>If you run the `Data Organiser` in Lazy Mode, changes in the Keywords to be displayed list will have no effect on the output shown in the `Data Set Chooser`.

|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 22 of 44               |

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 RecipeExecutor (see Sect. 6.2.3). The file tags are used by the FitsRouter to send files to particular processing steps or branches of the workflow (see below). The purpose is used by the SofSplitter and SofAccumulator to generate input SoFs for the RecipeExecutor. 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.

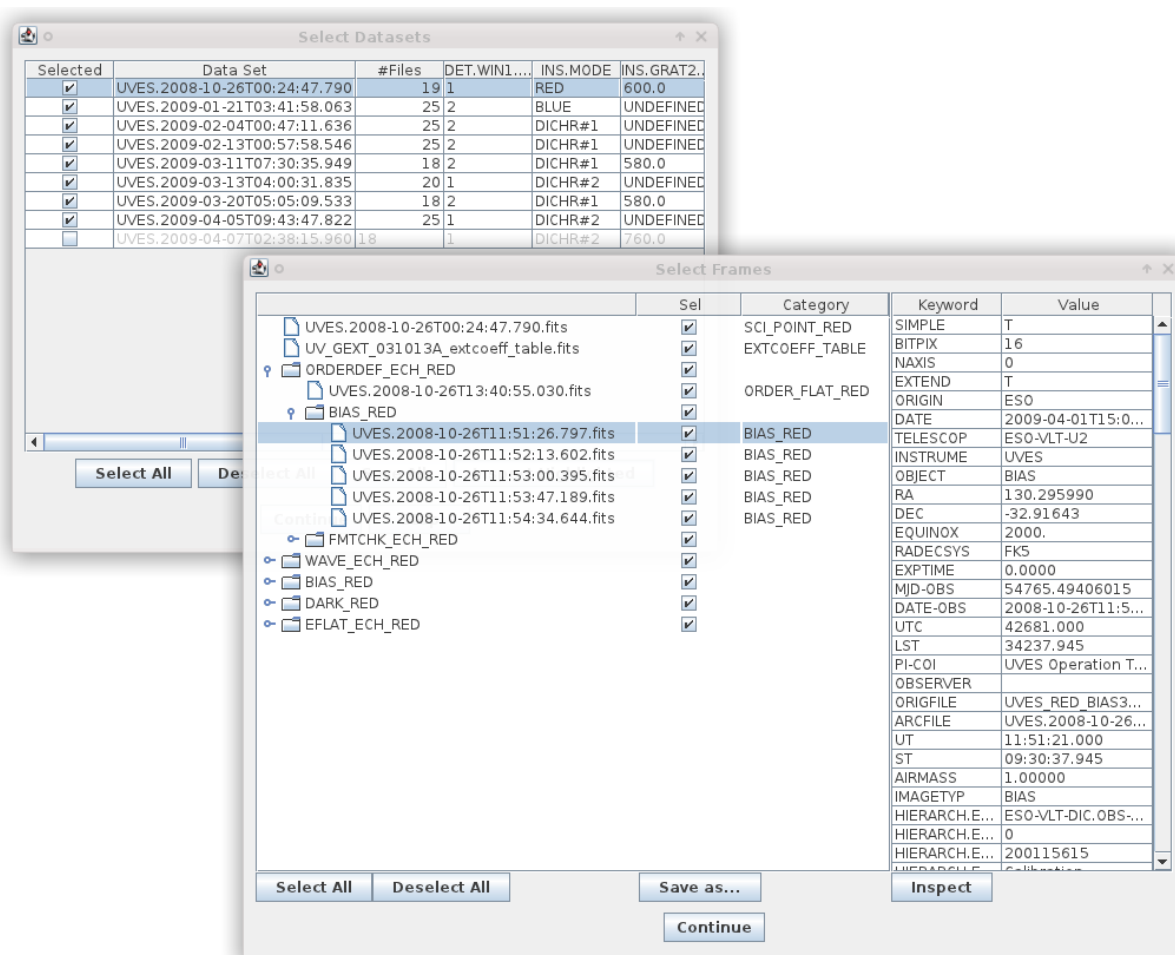


Figure 6.6: The “Select Frames” window with a single file from the current DataSet highlighted in blue, and the corresponding FITS header displayed in the text box on the right. Hidden partially behind the “Select Frames” window is the “Select Datasets” window with the currently selected DataSet highlighted in blue.

|            |                             |        |                        |
|------------|-----------------------------|--------|------------------------|
| <b>ESO</b> | <b>Reflex UVES Tutorial</b> | Doc:   | VLT-MAN-ESO-19540-5296 |
|            |                             | Issue: | Issue 6.0              |
|            |                             | Date:  | Date 2013-05-06        |
|            |                             | Page:  | 23 of 44               |

### 6.3.2 Step 2: Creation Of Master Calibration Files

In this step of the workflow, the following UVES recipes are executed in the order listed below. Please refer to the UVES pipeline user manual ([Larsen et al. 2012](#): Sections 9, 10 & 11) for the details of each recipe and the algorithms employed:

1. The `Master Bias Creation` actor will execute the UVES pipeline recipe `uves_cal_mbias` in order to create a combined master bias frame from the set of raw bias frames.
2. The `Master Dark Creation` actor will execute the UVES pipeline recipe `uves_cal_mdark` in order to create a combined master dark frame from the set of raw dark frames. Note that this actor will be skipped if there are no raw dark frames in the current DataSet.
3. The `Spectrum Locator` actor will execute the UVES pipeline recipe `uves_cal_predict` in order to compute initial guesses for the wavelength solution and order positions by employing a physical model of UVES combined with the information on the atmospheric pressure, temperature, and corresponding instrument setting stored in the FITS header of the input arc-lamp frame.
4. The `Order Detection` actor will execute the UVES pipeline recipe `uves_cal_orderpos` in order to fully detect and define the order positions on the detector from an input flat-field frame.
5. The `Master Flat Creation` actor will execute the UVES pipeline recipe `uves_cal_mflat` in order to create a combined master flat frame from the set of raw flat frames.

Note that the workflow supports the processing of deuterium lamp flats (`DPR TYPE = LAMP, DFLAT`) in addition to the usual lamp flats (`DPR TYPE = LAMP, FLAT`). The deuterium lamp flats are recommended for the spectral region shortwards of 350 nm and they are taken as supplementary calibrations for the UVES blue arm central wavelength setting of 346 nm. If deuterium flats are present in the DataSet, then they will be combined with the usual lamp flats into a combined master flat that provides much better S/N shortwards of 350 nm.

### 6.3.3 Step 3: Wavelength And Response Calibration

In this step of the workflow, the following UVES recipes are executed in the order listed below. Please refer to the UVES pipeline user manual ([Larsen et al. 2012](#): Sections 9, 10 & 11) for the details of each recipe and the algorithms employed:

1. The `Wavelength Calibration` actor will execute the UVES pipeline recipe `uves_cal_wavec` in order to create a wavelength calibration solution from an input arc-lamp frame.
2. The `Instrument Response` actor will execute the UVES pipeline recipe `uves_cal_response` in order to create an instrument response curve from the observation of a standard star, which will subsequently be used to flux-calibrate the science observation. Note that this actor will be skipped if there are no observations of a standard star in the current DataSet.

One should also note that a standard star observation is only included in the DataSet for a science observation if it was taken on the same night. Generally, UVES data are supplied with master instrument response curves



|            |                             |        |                        |
|------------|-----------------------------|--------|------------------------|
| <b>ESO</b> | <b>Reflex UVES Tutorial</b> | Doc:   | VLT-MAN-ESO-19540-5296 |
|            |                             | Issue: | Issue 6.0              |
|            |                             | Date:  | Date 2013-05-06        |
|            |                             | Page:  | 24 of 44               |

which will also be included in the DataSets for a science observation. The UVES 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, and failing this, the master instrument response curve will be used if it exists in the current DataSet. If neither a standard star observation nor a master instrument response curve exist in the current DataSet, then the science observation will not be flux-calibrated.

#### 6.3.4 Step 4: Spectrum Extraction

In this step of the workflow, the `Spectrum Extraction` actor will execute the UVES pipeline recipe `uves_obs_scired` in order to extract and merge the science spectrum from the science observation. Please refer to the UVES pipeline user manual ([Larsen et al. 2012](#): Sections 9, 10 & 11) for the details of this recipe and the extraction algorithms employed. As described in the previous section, the flux-calibration of the science spectrum will be carried out if an appropriate instrument response curve (master or otherwise) has been used as an input to this actor.

After the spectrum extraction, the actor will display an interactive window to allow the user to inspect the extracted spectrum and assess the extraction quality (see Figure 4.4).

#### 6.3.5 Step 5: Output Organisation

After having processed the input data for a DataSet, the `Data Filter` actor allows the user to inspect the results and select which files to save. By default this actor is set to skip and all products will be saved. Next the workflow highlights and executes the `Product Renamer` actor, which, by default, will copy the defined final products of the UVES pipeline recipe `uves_obs_scired` 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
```

where `<HIERARCH.ESO.OBS.NAME>` and `<HIERARCH.ESO.PRO.CATG>` represent the values of the corresponding FITS header keywords. These names are fully configurable by right-clicking on the `Product Renamer` actor and selecting `Configure Actor`, and then editing the `Rename_keywords` parameter as appropriate.

The final products that are copied and renamed are:

- `<HIERARCH.ESO.OBS.NAME>_RED_SCI_POINT_<ARM>.fits` - The extracted and merged science spectrum.
- `<HIERARCH.ESO.OBS.NAME>_ERRORBAR_SCI_POINT_<ARM>.fits` - The uncertainty on the extracted and merged science spectrum.
- `<HIERARCH.ESO.OBS.NAME>_FLUXCAL_SCI_POINT_<ARM>.fits` - The extracted, merged, and flux-calibrated science spectrum. This product is only generated if an appropriate instrument response curve (master or otherwise) was used as an input to the `Spectrum Extraction` actor.



|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 25 of 44               |

- `<HIERARCH.ESO.OBS.NAME>_FLUXCAL_ERRORBAR_SCI_POINT_<ARM>.fits` - The uncertainty on the extracted, merged, and flux-calibrated science spectrum. This product is only generated if an appropriate instrument response curve (master or otherwise) was used as an input to the Spectrum Extraction actor.
- `<HIERARCH.ESO.OBS.NAME>_ORDER_EXTRACT_QC_<ARM>.fits` - A table of useful statistics from the spectrum extraction procedure which are used to make the plots in the interactive window for the Spectrum Extraction actor (see Figure 4.4).

The remaining 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.

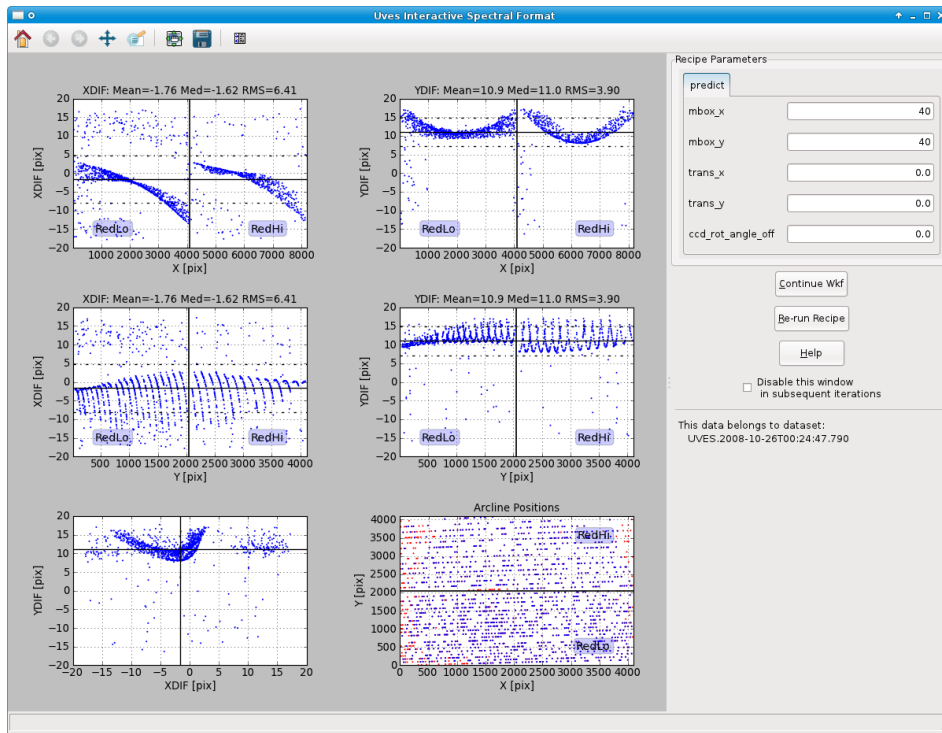


Figure 6.7: The interactive window for the Spectrum Locator actor for the first DataSet in the data supplied with Reflex 2.4..

|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 26 of 44               |

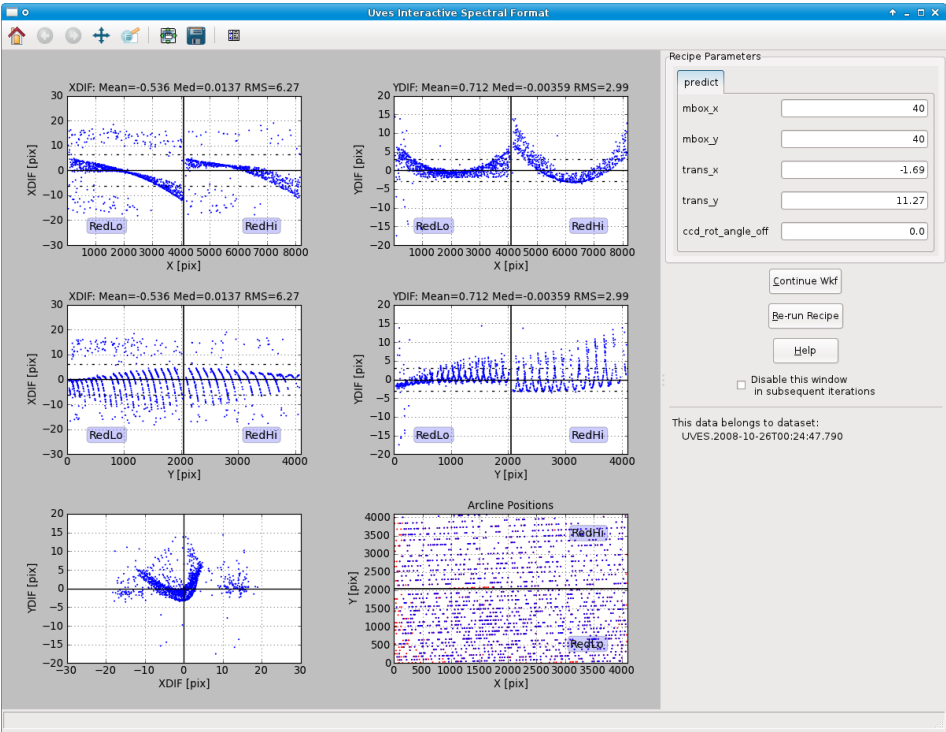


Figure 6.8: Same as Figure 6.7 but after iteration of the corresponding `uves_cal_predict` pipeline recipe.

|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 27 of 44               |


## 7 Optimising Your Results Through Workflow Interaction

In this section, we use the information from previous sections along with the UVES demo data supplied with `Reflex 2.4` to illustrate how to optimise the scientific products in terms of quality and S/N. This is achieved by interaction with the workflow actors via interactive windows displayed at key data reduction points in the data flow, which enable iteration of certain recipes in order to obtain better results. For instance, the physical model prediction calculated during execution of the `Spectrum Locator` actor may be inaccurate and need correcting, or the wavelength solution may need tweaking, or the user may want to clip more cosmic rays from the extracted spectrum.

We recommend that the user has already carried out the reductions for all the data sets as described in Section 4, although this is not a pre-requisite to following this section. By doing this, the user will be taking advantage of the workflow Lazy mode, with minimal waiting time between various pipeline recipe executions.

Please notice that the numbers given in the rest of the section were obtained in a Fedora 15, 64 bit machine. Your values may differ slightly depending on you operating system and platform.

Please follow these steps in order to optimise the reductions for the UVES demo data supplied with `Reflex 2.4`:

1. Carry out the first four steps described in the quick start Section 4.
2. In the UVES workflow, the four interactive actors (`Spectrum Locator`, `Wavelength Calibration`, `Instrument Response`, `Spectrum Extraction`) are identifiable by an orange rectangle encompassing the actor name. For all of these composite actors except the `Spectrum Extraction` actor, you will need to enable their interactive mode. Do this by double-clicking on the composite actor, setting the “EnableInteractivity” parameter to `true`, and clicking `Commit` to save the changes to the workflow.
3. Click the  button.
4. Click the `Continue` button in the “Select Datasets” window.
5. A “Current DataSet” window pops up that you can minimise.
6. `Spectrum Locator`: During the execution of the `Spectrum Locator` actor, an interactive window will appear as shown in Figure 6.7. The interactive window consists of six plots, five pipeline recipe parameters that may be changed, three buttons on the right, and a panel of buttons at the top-left. Help for each of these features may be accessed by leaving the mouse pointer for about one second over the plot/parameter/button for which information is required. Note that the recipe parameters are grouped via labels at the top of the “Recipe Parameters” box that may be left-clicked on to reveal the associated parameters.






The `Spectrum Locator` actor executes the UVES pipeline recipe `uves_cal_predict`, taking as input a format-check arc-lamp frame. The purpose of the recipe is to compute an initial guess for the wavelength solution and order positions by employing a physical model of UVES combined with the information on the atmospheric pressure, temperature, and corresponding instrument setting in the FITS header of the input arc-lamp frame. For quality control purposes, this initial guess wavelength solution



|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 28 of 44               |

may be compared with the actual detected line positions (shown in the bottom-right plot), which also allows the identification of cases where the physical model prediction is inaccurate.

The plots in the interactive window provide such a comparison by showing the differences XDIF and YDIF between the predicted and detected line positions along the  $x$  and  $y$  image axes, respectively, as functions of  $x$ ,  $y$  and each other. If the physical model prediction is good, then these plots will show clear trends with sets of scattered outliers, and the statistics shown in the plot titles should report median values for XDIF and YDIF that are in the range  $\pm 10$  pix. If the physical model prediction is poor, then these plots will show only scattered points with no trends, although this happens very rarely (e.g. for data taken after an earthquake event).

- The panel of buttons at the top-left of the interactive window may be used to manipulate the displayed plots. The buttons have the following functions:

-  - Reset all the plot ranges to their original values.
-  - Undo the last modification of the plot ranges (if possible).
-  - Redo the next modification of the plot ranges (if possible).
-  - Selecting this button allows the user to use the mouse to shift the plot ranges by left-clicking on the target plot canvas and then dragging the mouse around while keeping the left mouse button held down, and releasing when ready.
-  - Selecting this button allows the user to zoom in on each plot by left-clicking on the target plot canvas to mark the top-left corner of a rectangle and then dragging the mouse to the bottom-right corner of the rectangle and releasing. The plot ranges will then be modified to match the rectangle that was defined. The following constraints can be made by holding simultaneously a key while pressing the left mouse button:
 

|                              |      |                   |
|------------------------------|------|-------------------|
| Constrain pan/zoom to x axis | hold | <span>X</span>    |
| Constrain pan/zoom to y axis | hold | <span>Y</span>    |
| Preserve aspect ratio        | hold | <span>Ctrl</span> |
-  - Clicking this button opens a “Configure subplots” window that allows the user to adjust the spacing and positioning of the individual plots.
-  - Clicking this button opens a “Save to file” window which allows the user to save a screenshot of the current interactive window.

Use these buttons to inspect the plots in the interactive window in more detail.

- In some cases it is worth adjusting the initial guess wavelength solution to better match the detected line positions in order for the subsequent wavelength calibration recipe to produce a more robust (and possibly better) wavelength solution, especially if the median XDIF and/or YDIF values are outside the range  $\pm 5$  pix. This may be done by adding the median XDIF and YDIF values to the values in the text boxes corresponding to the pipeline recipe parameters `trans_x` and `trans_y`, respectively, on the right-hand side of the interactive window, and then clicking the Re-run recipe button. The `uves_cal_predict` pipeline recipe will be executed again with the new parameter values, and the new results will appear in the interactive window. This time the median XDIF and YDIF values should be closer to zero, and the corresponding root-mean-square (RMS) values will most likely have decreased.

|            |                             |        |                        |
|------------|-----------------------------|--------|------------------------|
| <b>ESO</b> | <b>Reflex UVES Tutorial</b> | Doc:   | VLT-MAN-ESO-19540-5296 |
|            |                             | Issue: | Issue 6.0              |
|            |                             | Date:  | Date 2013-05-06        |
|            |                             | Page:  | 29 of 44               |

The adjustment of the `trans_x` and `trans_y` parameters may be iterated until the median XDIF and YDIF values are in the range  $\pm 0.1$  pix (usually two or three iterations are required).

Carry out this procedure for the current DataSet. The first values for `trans_x` and `trans_y` should be -1.62 and 11.00 pix respectively for this DataSet, with RMS statistics of 6.41 and 3.90 pix for XDIF and YDIF respectively. Further iteration should converge to values of  $\sim -1.69$  and  $\sim 11.27$  pix for `trans_x` and `trans_y` respectively, with RMS statistics of 6.27 and 2.99 pix for XDIF and YDIF respectively. At this point, the interactive window should look like the one shown in Figure 6.8.

- Before we move on, we note that there are three other pipeline recipe parameters on the right-hand side of the current interactive window. The parameters `mbox_x` and `mbox_y` are used to define a search box for matching the physical model line positions with the positions of the lines detected in the input arc-lamp frame. If the physical model prediction is poor, as shown by the lack of clear trends in the plots in the interactive window, then one should try larger values of `mbox_x` and `mbox_y`, such as 60 or 80 pix, in order to aid the `uves_cal_predict` recipe in determining the initial guess wavelength solution. Subsequent adjustment of `trans_x` and `trans_y` should then be performed (as described in the previous step), and then `mbox_x` and `mbox_y` should be set back to 40 pix, and one final adjustment of `trans_x` and `trans_y` should be performed to arrive at the best initial guess wavelength solution.

The parameter `ccd_rot_angle_off` may be used to help “tighten” any less-well defined trends in the plots in the interactive window. This adjustment is usually not necessary, but if it is, then rotations of the order of 0.01-0.1 in either sense may make improvements.

- Click the `Continue wkf` button in the interactive window.
- Wavelength Calibration:** The next interactive window that will appear (see Figure 7.1) is associated with the execution of the Wavelength Calibration actor, which executes the UVES pipeline recipe `uves_cal_wavecal`. This pipeline recipe determines the wavelength calibration solution from an appropriate arc-lamp frame and a line reference catalogue. The top two plots in the interactive window show the wavelength calibration residuals for arc lines that were identified in the input line reference catalogue as a function of wavelength and spectral order. Blue points represent arc lines that were used in the final solution, and red points represent arc lines that were excluded from the final solution. The bottom-left plot shows the arc line FWHM as a function of wavelength, and the bottom-right plot displays the positions of all detected arc lines on the detector.
- The wavelength calibration solutions produced by the UVES pipeline are generally robust and of good quality, showing very few systematic errors in the wavelength residuals and demonstrating the correct identification and clipping of outliers. The main purpose of this interactive window is therefore to allow the user to check that all is well with the wavelength solution, and to allow some further refinement if so required. Most users will simply accept the solution that is presented by clicking on the `Continue wkf` button.

The interactive window gives access to seven pipeline recipe parameters that may be adjusted in order to improve the wavelength calibration solution, where an improvement is characterised by a smaller RMS statistic for the wavelength residuals (reported in the upper plot titles) but with a reasonable number of arc lines used in the final solution (reported as “Final no. of fitted lines”). We suggest that you attempt to improve the wavelength calibration solution for the first DataSet, based on the parameter descriptions that follow:

|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 30 of 44               |

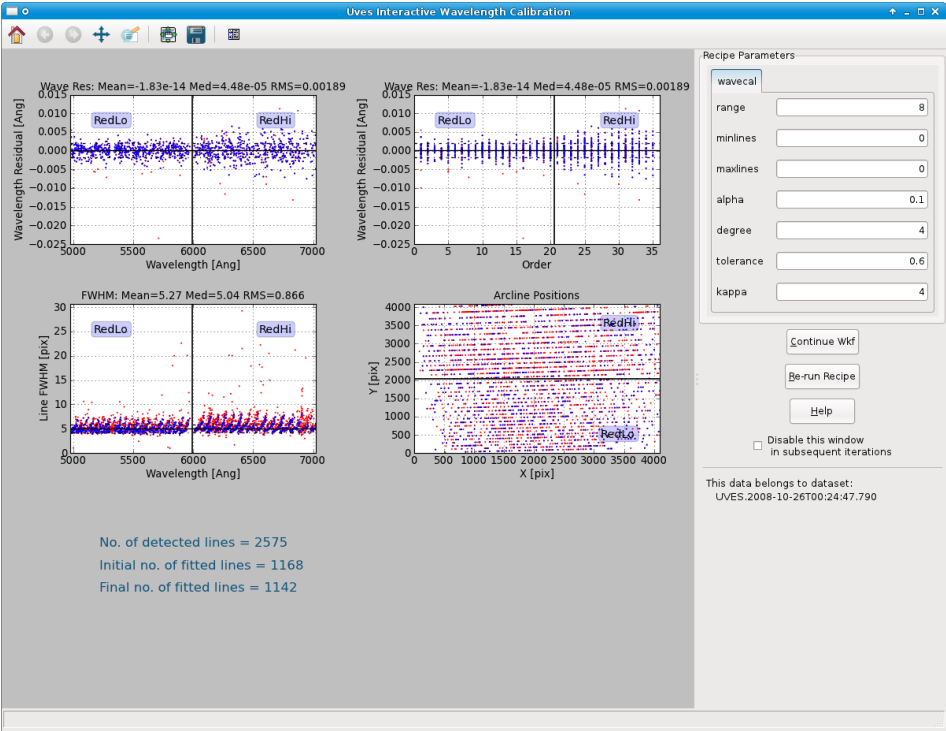


Figure 7.1: The interactive window for the Wavelength Calibration actor for the first DataSet in the data supplied with Reflex 2.4.

|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 31 of 44               |

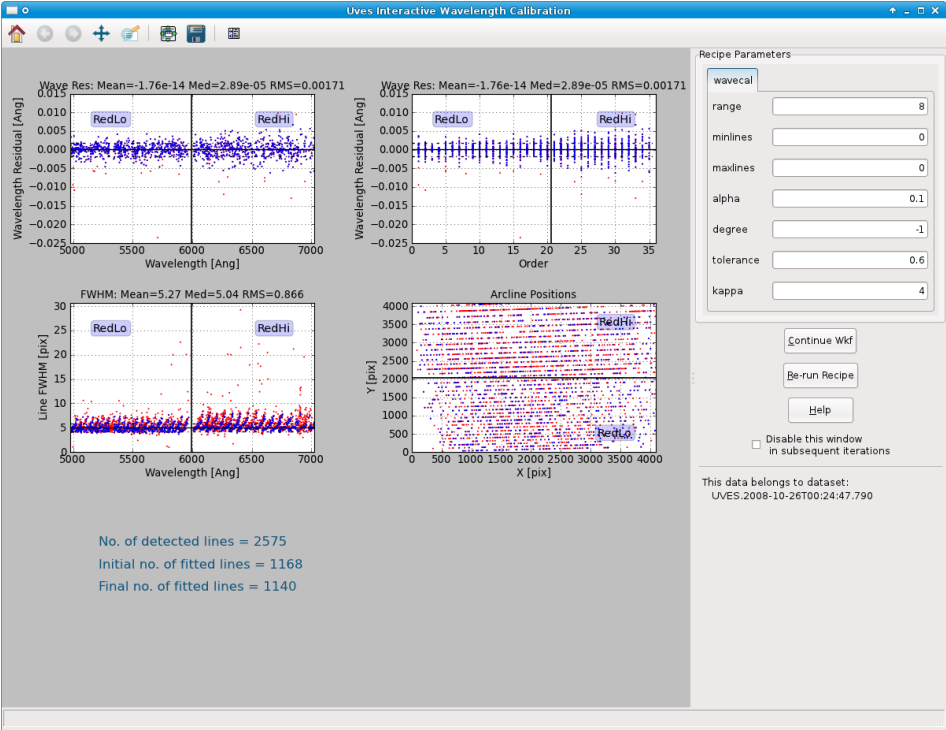


Figure 7.2: Same as Fig. 7.1 but after iteration of the corresponding uves\_cal\_wavecalf pipeline recipe..



|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 32 of 44               |

- `range` - Twice the value of this parameter is the window width used for arc line detection (pix). Adjusting this parameter will influence the number of arc lines that are detected. Varying this parameter in the range 4-12 pix is sometimes useful for optimising the number of detected arc lines.
- `minlines` - The minimum number of arc lines to be detected. This parameter should only be adjusted in conjunction with the parameter `maxlines`. Adjusting this parameter will force the pipeline recipe to find a minimum number of arc lines, if possible. This parameter is useful for controlling the number of lines used in the wavelength calibration solution.
- `maxlines` - The maximum number of arc lines to be detected. This parameter should only be adjusted in conjunction with the parameter `minlines`. Adjusting this parameter will force a maximum on the number of arc lines the pipeline recipe finds. This parameter is useful for controlling the number of lines used in the wavelength calibration solution.
- `alpha` - This parameter ranges from 0.0 to 1.0, and is used to control the line matches between the detected arc lines and those in the reference line catalogue by rejecting line matches where the distance to the next nearest neighbour (in the spectrum as well as in the catalogue) is smaller than  $\Delta\lambda/\alpha$ , where  $\Delta\lambda$  is the wavelength residual for the current line match candidate. This parameter is best adjusted in conjunction with the parameter `tolerance`. Adjusting `alpha` in the range  $\sim 0.03$ -0.3 is useful for controlling the number of line identifications and hence the number of lines used in the wavelength calibration solution.
- `degree` - The degree of the polynomial model to be used in the wavelength calibration process. Usually the default value of 4 gives acceptable wavelength residuals with no systematics, but sometimes higher degrees such as 5 or 6 give better results. It is also worth trying the value -1, which instructs the pipeline recipe to determine the model degree based on the wavelength residuals.
- `tolerance` - This parameter is used to reject line identifications with wavelength residuals worse than `tolerance` from the wavelength calibration solution. This parameter is best adjusted in conjunction with the parameter `alpha`. Varying `tolerance` in the range 0.1-1.0 is sometimes useful for optimising the number of lines used in the wavelength calibration solution.
- `kappa` - The level of sigma-clipping to be performed in the final iterations of the fit of the wavelength calibration model. Suggested values for this parameter are in the range 3.0-5.0.

We find that for the first DataSet, setting `degree = -1` and leaving the other parameters at their default values, is sufficient to reduce the RMS of the residuals from  $1.89 \times 10^{-3}$  to  $1.71 \times 10^{-3}$  Ang while only reducing the final number of fitted lines from 1142 to 1140 (see Figure 7.2).

13. Click the `Continue wkf` button in the interactive window.
14. **Instrument Response:** The standard DataSets supplied with Reflex 2.4 do not include standard star observations. Hence the `Instrument Response` actor will not be executed, and the corresponding interactive window will not appear. In Figure 7.3 we show what this interactive window looks like. The usage and purpose of the `Instrument Response` interactive window are the same as for the `Spectrum Extraction` interactive window which we describe in detail in the next few steps, and so no more information on this window is given here in the tutorial.
15. **Spectrum Extraction:** In Figure 7.4, we show the interactive window associated with the `Spectrum Extraction` actor, which executes the UVES pipeline recipe `uves_obs_scired`. This pipeline recipe performs the extraction of the science object spectrum using optimal extraction (Horne 1986),



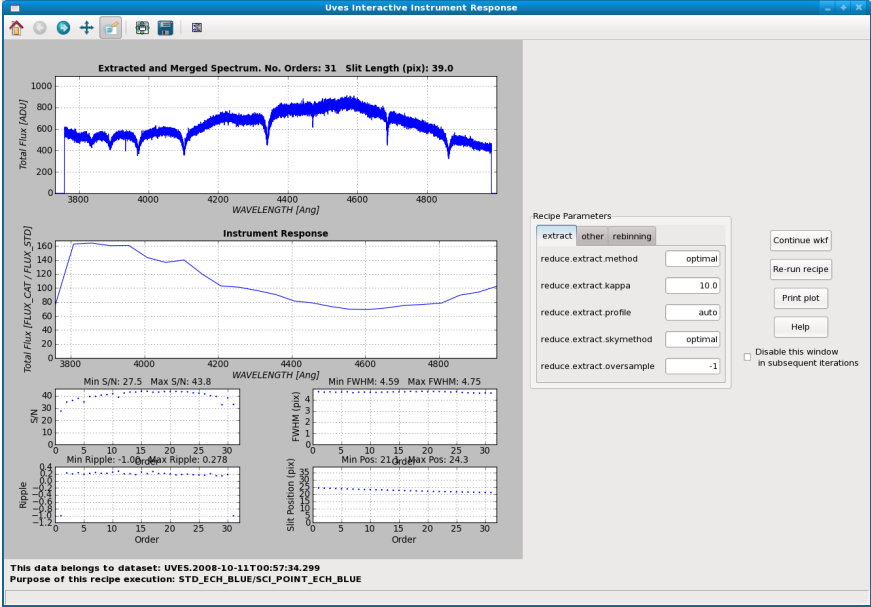


Figure 7.3: The interactive window for the Instrument Response actor.

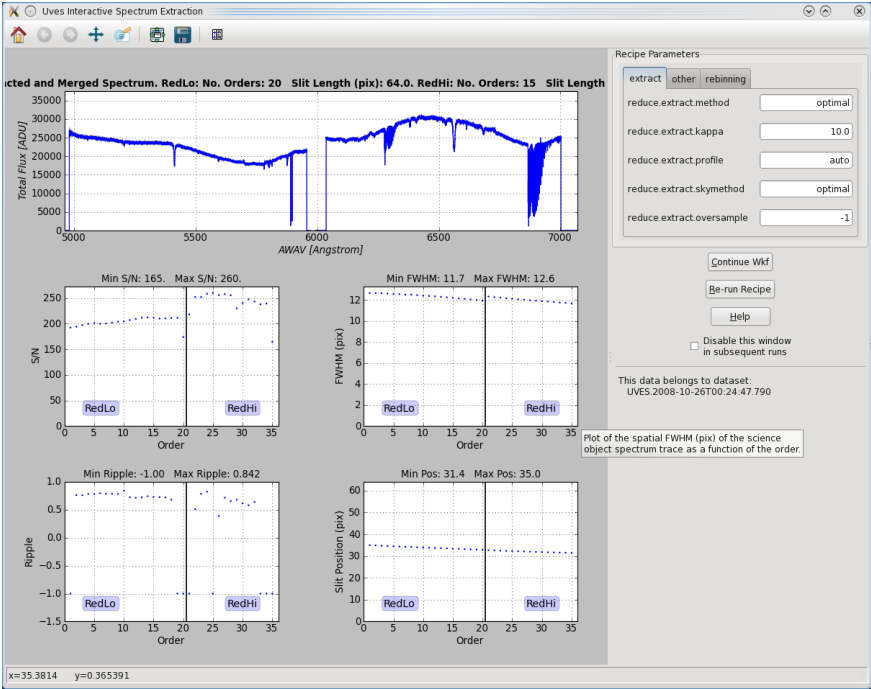


Figure 7.4: The interactive window for the Spectrum Extraction actor for the first DataSet in the data supplied with Reflex 2.4.

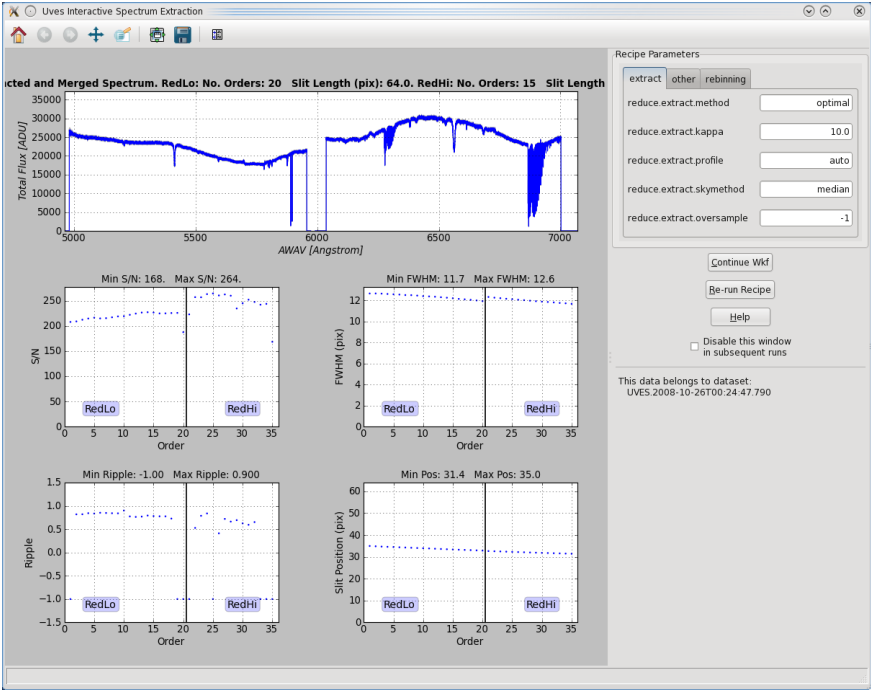


Figure 7.5: Same as Fig. 7.4 but after iteration of the corresponding `uves_obs_scired` pipeline recipe.

|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 35 of 44               |

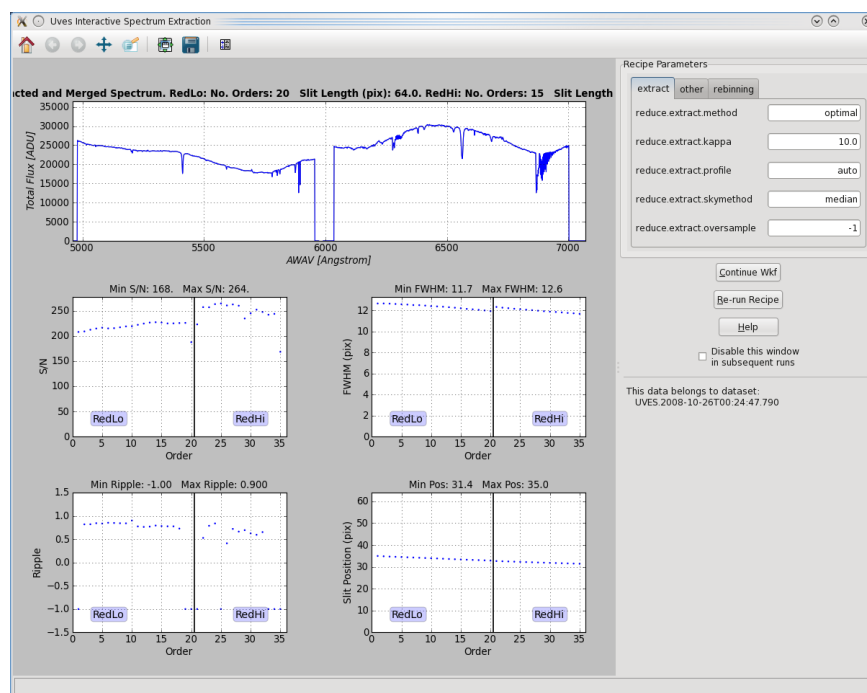


Figure 7.6: The interactive window for the Spectrum Extraction actor for the first DataSet in the data supplied with Reflex 2.4 after iteration of the corresponding uves\_obs\_scired pipeline recipe and rebinning to a resolution of 1 Ang.

|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 36 of 44               |

automatically selecting an analytical spatial profile (Gaussian or Moffat) in the case of low S/N spectra ( $S/N < 10$ ) or an oversampled empirical spatial profile otherwise, and automatically determining the degrees of the low-order polynomials involved in the extraction process. The top plot in the interactive window shows the extracted (using optimal extraction) and merged one-dimensional spectrum of the science object (blue) with  $\pm 1$ -sigma uncertainties plotted as the light blue region encompassing the object spectrum (and bounded by the light grey spectra), and the plot below this shows the flux-calibrated version.

Below the plots displaying the spectra are four smaller plots which show the S/N of the extracted spectrum (upper-left), the spatial FWHM (upper-right), the “Ripple” parameter (lower-left), and the science object position on the slit (lower-right), all as a function of the spectral order. The ripple parameter is a measure of any systematic oscillations in the extracted spectrum, and should have values less than  $\sim 5$  over all orders to indicate that such oscillations are not present.

Use the buttons at the top of the interactive window to inspect these plots in detail, especially the extracted spectra. For instance, you could zoom in on the absorption feature at 5412 Ang.

16. The current interactive window provides eight pipeline recipe parameters with the aim of allowing the user to optimise the extraction for quality and S/N. Try to improve the spectrum extraction for the first DataSet, based on the parameter descriptions that follow:

- `reduce.extract.method` - The method to be used for extracting the science spectrum, and it may be set to `optimal`, `average`, or `linear` for our purposes. In most cases, the default `optimal` extraction is the desired method.
- `reduce.extract.kappa` - The level of sigma-clipping to be performed when rejecting pixels suspected of being contaminated by cosmic ray hits. The default value of 10.0 should be reduced to smaller values (down to  $\sim 3.0$ ) when an extracted spectrum shows sharp emission spikes that should not be present
- `reduce.extract.profile` - The spatial profile model to be used when performing optimal extraction, and it may be set to `auto`, `virtual`, `gauss`, or `moffat` for our purposes. The default parameter value `auto` instructs the pipeline recipe to automatically decide on the type of spatial profile model to employ based on the S/N of the science spectrum. Sometimes it is worth testing the other spatial profiles that are available.
- `reduce.extract.skymethod` - The method for calculating the sky level may be set to one of two values; namely `optimal` or `median`. When set to `optimal`, the sky level is included in the optimal extraction model, and when set to `median`, the sky level is estimated via the median and subtracted from the science spectrum before extraction. Sometimes the non-default `median` option can give slightly better S/N than the default `optimal` option.
- `reduce.extract.oversample` - The oversampling factor to be used in the empirical spatial profile. When the empirical spatial profile is employed by the pipeline recipe, the oversampling factor is set by default to 5 for  $S/N < 200$ , and to 10 for  $S/N > 200$ . Sometimes a larger value for the oversampling factor may give a higher S/N extracted spectrum, especially for very high S/N spectra.
- `reduce.slitlength` - This parameter may be used to control the slit length used to define the spatial profile in the spectrum extraction. The default negative value instructs the pipeline recipe to read the slit length from the science object FITS header. Setting this parameter to smaller values

|            |                             |        |                        |
|------------|-----------------------------|--------|------------------------|
| <b>ESO</b> | <b>Reflex UVES Tutorial</b> | Doc:   | VLT-MAN-ESO-19540-5296 |
|            |                             | Issue: | Issue 6.0              |
|            |                             | Date:  | Date 2013-05-06        |
|            |                             | Page:  | 37 of 44               |

than the slit length in the header may sometimes be useful in avoiding problems at the order edges, with the disadvantage that the maximum achievable S/N will be smaller (less pixels involved in the extraction).

- `reduce.objoffset` - This parameter may be used in conjunction with the parameter `reduce.slitlength` in order to extract the spectra of more than one object observed on the slit in the same science image. Please see the section on “Optimal Extraction” in the chapter on “Algorithms And Recipe Details” in the pipeline user manual ([Larsen et al. 2012](#)) for more details.
- `reduce.rebin.wavestep` - The rebinning step size in wavelength units for the extracted science spectrum. The default negative value instructs the pipeline recipe to use the optimal resolution. Setting this parameter to a positive value is useful for rebinning the spectrum to a lower resolution while gaining a substantial increase in S/N. Hence this parameter is more concerned with spectrum post-processing rather than extraction optimisation.

We find that for the first DataSet, setting `reduce.extract.skymethod = median` and `reduce.extract.oversample = 10`, and leaving the other parameters at their default values, increases the maximum spectrum S/N from 260 to 264 (see Figure 7.5).

17. For illustrative purposes, set `reduce.rebin.wavestep = 1.0 Ang` for the first DataSet, and re-run the pipeline recipe. In Figure 7.6 we display the interactive window with the rebinned spectrum. Notice that the S/N has been greatly improved at the cost of a lower resolution. Since the spectrum rebinning is a post-processing operation, the S/N per order reported in the interactive window corresponds to the S/N before rebinning.
18. Click the `Continue wkf` button in the interactive window.
19. As noted in the quick start Section 4, the workflow will write out the extracted spectrum and other important products to the end products directory (specified by the parameter `END_PRODUCTS_DIR` under “Setup Directories” in the workflow canvas).
20. The workflow will now repeat the above reduction steps for the next DataSet, looping over each DataSet until all of them have been processed. The other demo DataSets are provided to allow you to experiment with optimising the reductions. We advise you to follow the above interactive procedures until you feel that you have mastered the functionality of the pipeline parameters that are provided in the interactive windows.

|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 38 of 44               |

## 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 (Forchì (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 launch Reflex from the command line?** Yes, use the command:

```
reflex -runwf -nocache -nogui <workflow_path>/<workflow>.xml
```

Note that this mode is not fully supported, and the user should be aware of two points. Firstly, the execution prompt is not returned after the workflow finishes, and therefore Reflex must be manually killed. Secondly, all the interactive windows will still appear (if activated in the workflow), so it is not suitable for batch processing.

- **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 (Forchì (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 (Forchì (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/UVES/<recipe_name>_<number>` (for instance `reflex_book_keeping/UVES/bias_1/`). 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="REFLEX_INST/etc/esorex.rc REFLEX_INST/bin/esorex" -recipe-config=<recipe>.rc <recipe> data.sof`, where `REFLEX_INST` is the directory where Reflex and the pipelines were installed. If the user knows the name of the input raw files

|            |                             |        |                        |
|------------|-----------------------------|--------|------------------------|
| <b>ESO</b> | <b>Reflex UVES Tutorial</b> | Doc:   | VLT-MAN-ESO-19540-5296 |
|            |                             | Issue: | Issue 6.0              |
|            |                             | Date:  | Date 2013-05-06        |
|            |                             | Page:  | 39 of 44               |

for the recipe, the correct directory among the many time stamps can be found via `grep <raw_file> */data.sof`. Afterwards the procedure is the same as 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.

- **I have many DataSets in my data directory. How can I reduce them interactively without having to wait a long time between interactive windows being displayed?** Reduce all the DataSets at once with the interactive windows disabled for all interactive actors. When this reduction has finished, you should re-enable the interactive windows that you require, and run the workflow again. The workflow will run in Lazy mode and no time will be spent on pipeline reductions, unless you specifically change a parameter in one of the interactive windows.

Note that Lazy mode will not work if the workflow parameter `EraseDirs` is set to `true`.

- **I have many DataSets of extended objects. How can I reduce them?** UVES observations of extended object can be reduced with the normal workflow. We have not verified that the corresponding products are science data products.
- **I have many DataSets of objects observed with the Image Slicer. How can I reduce them?** UVES observations of object observed with the Image Slicer can be reduced with the normal workflow. We have not verified that the corresponding products are science data products.
- **I have many DataSets of objects observed with the the Iodine Cell filter. How can I reduce them?** UVES observations of object observed with the Iodine Cell filter is not yet supported.

|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 40 of 44               |

## 9 Troubleshooting

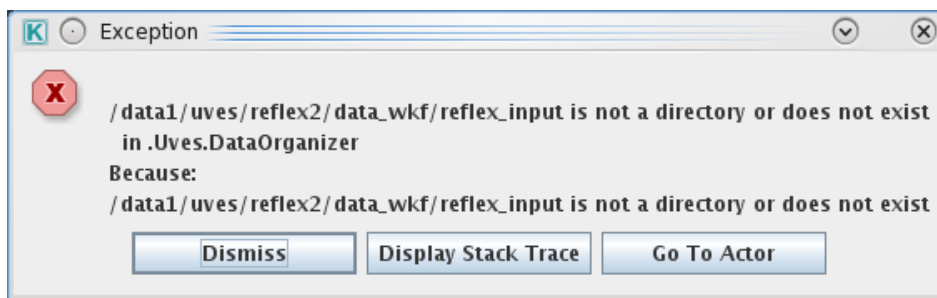


Figure 9.1: *TheDataOrganizer* interactive window reports an error “:No Datasets have been created, check the DataSet and the OCA rules.”.

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**

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) **it fails with 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 executing Reflex.

- (c) **all DataSets are greyed out in the DataSets interactive window.**

The ESO archive used with CalSelector does not always supply all static calibration files. As a consequence some/all DataSets are greyed out because they were missing such required data.

Missing static calibration should be found by reflex in

`<install_directory>/calib/<pipeline_version>/cal`.

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 that 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 report the category of the calibration products that are missing (e.g. `MASTER_FLAT_REDL_WKF`, `MASTER_FLAT_REDU_WKF`, `ORDER_GUESS_TAB_REDL_WKF`, `ORDER_GUESS_TAB_REDU_WKF`, `LINE_GUESS_TAB_REDL_WKF`, `LINE_GUESS_TAB_REDU_WKF`). From this the user has then to determine the missing raw data (in this case formatcheck and flat frames for the RED arm). If static calibrations are missing the mechanism unfortunately does not work, but should be found by reflex in `<install_directory>/calib/<pipeline_version>/cal`



|     |                      |        |                        |
|-----|----------------------|--------|------------------------|
| ESO | Reflex UVES Tutorial | Doc:   | VLT-MAN-ESO-19540-5296 |
|     |                      | Issue: | Issue 6.0              |
|     |                      | Date:  | Date 2013-05-06        |
|     |                      | Page:  | 41 of 44               |

A faster way to identify which kind of data are missing is moving the mouse over the grey DataSet (see Fig. 9.2).

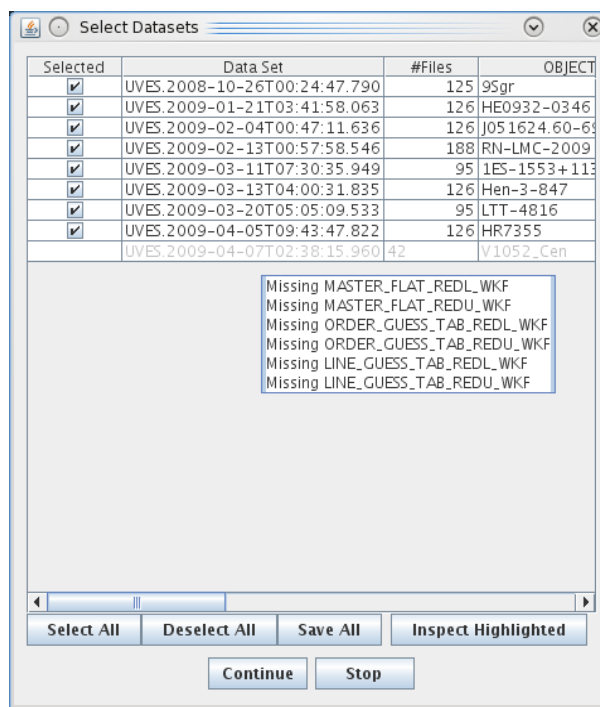


Figure 9.2: This shows how information about missing files in an incomplete DataSet is displayed when moving the mouse over it.

3. **I downloaded the data from the ESO archive and put them into a new directory, tried to run Reflex on them, but all DataSets are greyed out in the DataSets interactive window.**

The ESO archive does not supply as default general calibration files such as the LINE\_REFER\_TABLE. As a consequence all DataSets are greyed out because they were missing such required data.

The user may solve this problem remembering that any reference data file required for the data reduction is also provided by the pipeline distribution kit.

4. **I have only DFLATS for a given DataSet, which is shown as incomplete by the DataSetChooser. Is there support for this setup?**

Yes, there is support. Use the `Uves_dflats.xml` workflow instead of the usual `Uves.xml` workflow.

5. **The workflow reports to me that there are no DataSets in my data directory. How can this be?**

Remember that the UVES workflow supports data for point source observations only. It is possible that your data consists entirely of extended source, image slicer, or multi-object spectrograph observations, in which case the Data Organiser actor will not construct any DataSets.

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

|            |                             |        |                          |
|------------|-----------------------------|--------|--------------------------|
| <b>ESO</b> | <b>Reflex UVES Tutorial</b> | Doc:   | VLT-MAN-ESO-19540-5296   |
|            |                             | Issue: | Issue 6.0                |
|            |                             | Date:  | Date 2013-05-06          |
|            |                             | Page:  | 42 of <a href="#">44</a> |

If a DataSet in the “Select Datasets” window is greyed out, then it means that the DataSet that 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 report the file category of the calibration(s) that are missing.

7. **In some platforms, when I click on the “Configure subplots” button in the interactive window (see Section [7](#), step 6), the window that appears is empty. Can this be corrected?**

This is a known problem that will be addressed in future releases.

|            |                             |        |                          |
|------------|-----------------------------|--------|--------------------------|
| <b>ESO</b> | <b>Reflex UVES Tutorial</b> | Doc:   | VLT-MAN-ESO-19540-5296   |
|            |                             | Issue: | Issue 6.0                |
|            |                             | Date:  | Date 2013-05-06          |
|            |                             | Page:  | 43 of <a href="#">44</a> |

## Acknowledgements

The `Reflex` team in alphabetical order consists of Pascal Ballester, Daniel Bramich, Vincenzo Forchì, Wolfram Freudling, César Enrique García, Maurice Klein Gebbinck, Andrea Modigliani, Sabine Moehler & Martino Romaniello. Our thanks go to R. Figuera Jaimes for volunteering to be our first external test user.

|            |                             |        |                        |
|------------|-----------------------------|--------|------------------------|
| <b>ESO</b> | <b>Reflex UVES Tutorial</b> | Doc:   | VLT-MAN-ESO-19540-5296 |
|            |                             | Issue: | Issue 6.0              |
|            |                             | Date:  | Date 2013-05-06        |
|            |                             | Page:  | 44 of 44               |

Forchì V., 2012, Reflex User Manual, VLT-MAN-ESO-19000-5037, Issue 3.4 [14](#), [19](#), [38](#)

Horne K., 1986, PASP, 98, 609 [32](#)

Kaufer A. et al., UV-Visual Echelle Spectrograph User Manual, VLT-MAN-ESO-13200-1825, Issue 86 [6](#)

Larsen J.M., Modigliani A. & Bramich D.M., UVES Pipeline User Manual, VLT-MAN-ESO-19500-2965, Issue 22.3 [6](#), [19](#), [23](#), [24](#), [37](#)