EUROPEAN SOUTHERN OBSERVATORY
Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral
Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

UVES Workflow Tutorial

Author: Daniel Bramich (dbramich@eso.org)
Department: Science Data Products Group
Version: 2.0
Date: January 24, 2011

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[1]) to provide a real-time visual representation of a data reduction cascade, called a workflow, which can be easily understood by most astronomers. 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 or as downloaded from the archive, and organises them into groups of files called LoSOs (List of Science Observations), where each LoSO contains one science object observation and all associated raw 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 LoSOs selected by the user for reduction are fed through the workflow which executes the relevant pipeline recipes (or stages) in the correct order, providing optional user interactivity at key data reduction points with the aim of enabling the iteration of certain recipes in order to obtain better results. 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[2] 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. 2010) for the details of UVES pipeline recipes. The quick start section (see Section 3) describes the minimum effort to get started, and it makes up only two pages of this tutorial.

User support for this software is available by sending enquiries to `usd-help@eso.org`.

[1] https://kepler-project.org
[2] These documents are available from http://www.eso.org/sci/facilities/paranal/instruments/uves/doc/index.html

2 Software Installation

To install the Reflex 1.1 software and test data, please follow these instructions:

1. From any directory, download the installation script:

   ```
   wget ftp://ftp.eso.org/pub/dfs/reflex/install_uves_reflex
   ```

2. Make the installation script executable:

   ```
   chmod u+x install_uves_reflex
   ```

3. Execute the script with three arguments, the download directory <download_dir>, the software installation directory <install_dir>, and the directory to be used to store the test data <data_dir>:

   ```
   ./install_uves_reflex <download_dir> <install_dir> <data_dir>
   ```

   If the script is executed with no arguments, then the default directories download, software, and data will be created in the current directory.

4. 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.

Note that you will need a minimum of ∼1.3 GB, ∼400 MB and ∼5 GB of free disk space for each of the directories <download_dir>, <install_dir> and <data_dir>, respectively.

3 Quick Start: Reducing The Test 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 nine data sets in the standard set of UVES data supplied with the Reflex 1.1 release. By following these steps, the user should have enough information to attempt a reduction of his/her own data without any further reading:

1. Start the Reflex application:

   ```
   reflex &
   ```

   The empty Reflex canvas as shown in Figure 1 will appear.

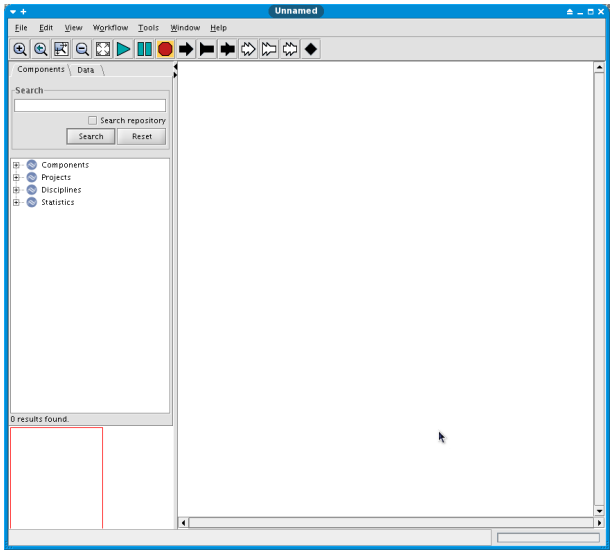


Figure 1: *The empty* `Reflex` *canvas.*

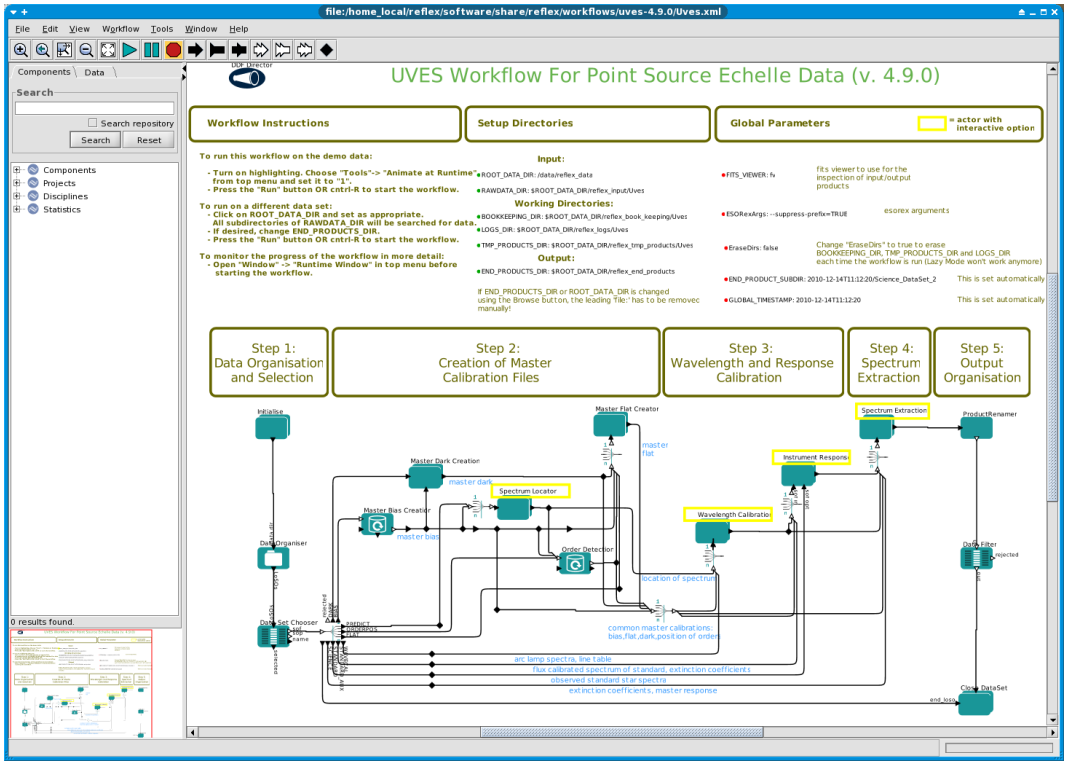


Figure 2: *UVES workflow general layout.*

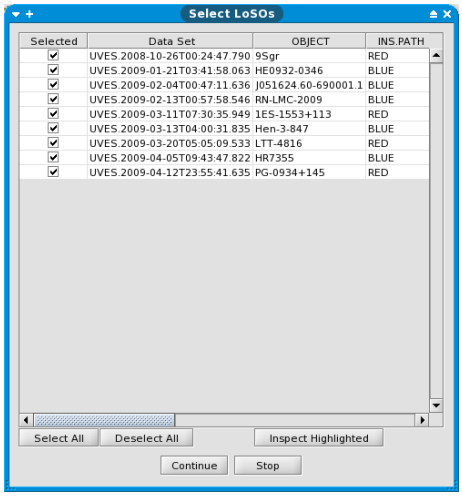


Figure 3: *The "Select LOSOs" pop-up window.*

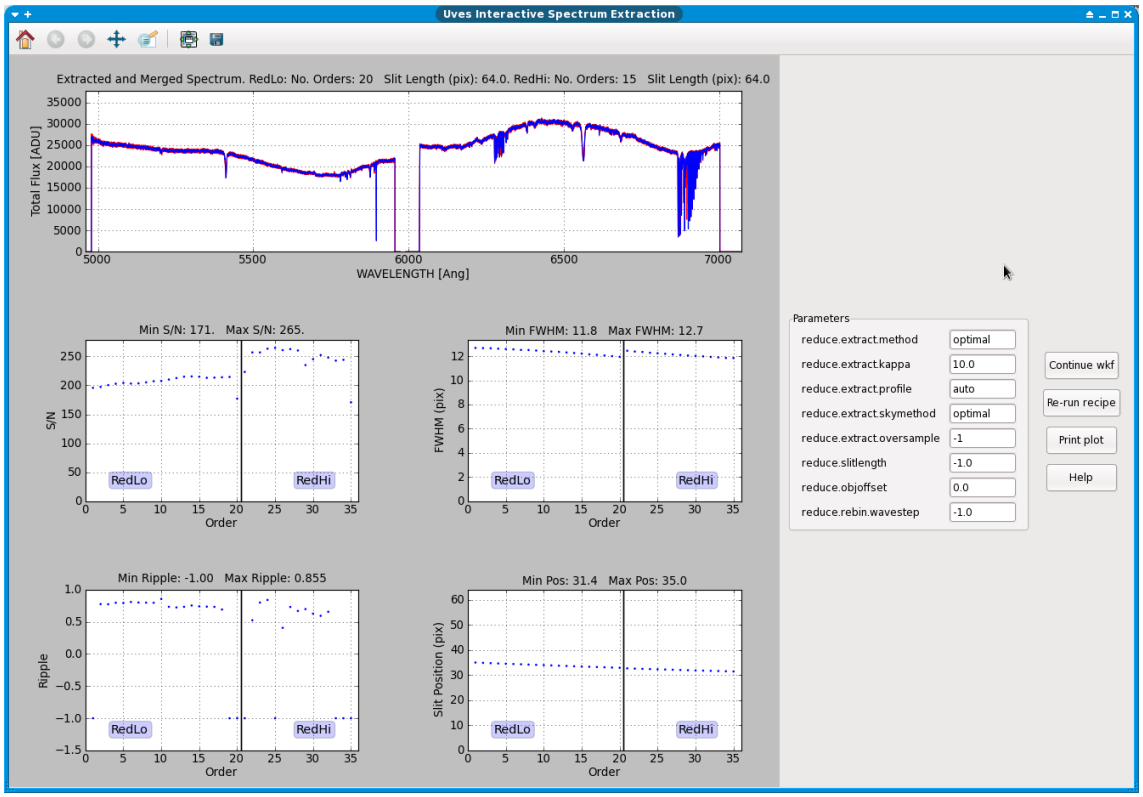


Figure 4: *The interactive pop-up window for the* `Spectrum Extraction` *actor and UVES pipeline recipe* `uves_obs_scired`*. The extracted and merged spectrum for the first data set is displayed in the top panel.*

2. Now open the UVES workflow by clicking on `File -> Open File`, selecting the file `Uves.xml` in the file browser[3], and clicking `Open` (this file is located in `<install_dir>/share/reflex/workflows/uves-4.9.0/Uves.xml` in the pipeline installation). You will be presented with the workflow canvas shown in Figure 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 (1 ms is recommended), and click `OK`.

4. Under "Setup Directories" in the workflow canvas there are six 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 test data[4], 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 `Browse` button to select the directory from a file browser (in this case, remove the leading `file:` string). When you have finished, click `OK` to save your changes.

5. Click the `Run` button  .

6. The workflow will highlight the `Data Organiser` actor which recursively scans the raw data directory (specified by the parameter `RAWDATA_DIR` under "Setup Directories" in the workflow canvas) and constructs the set of LoSOs. Note that the calibration and reference data must be present.

7. The `Data Set Chooser` actor will be highlighted next and will display a "Select LOSOs" window (see Figure 3) that lists the set of LoSOs along with the values of a selection of useful header keywords including the object name, UVES arm, detector binning and readout speed, instrument grating, and slit width. The first column consists of a set of tick boxes which allow the user to select the LoSOs to be processed, and by default all LoSOs are selected.

8. Click the `Continue` button and watch the progress of the workflow by following the red highlighting of the actors. You will notice that the highlighting dances around the workflow between the execution of each pipeline recipe. This is perfectly normal and is a consequence of the workflow director scheduling the workflow execution.

9. When the workflow has finished executing the final pipeline recipe `uves_obs_scired` in the `Spectrum Extraction` actor for the first data set, an interactive window will appear (see Figure 4) which shows a plot of the extracted and merged spectrum in the top panel (and the flux-calibrated version, if a suitable instrument response curve exists, in the panel below). Using the buttons at the top of this window, one may pan and zoom in on the spectrum in order 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. Click on the `Continue wkf` button and the workflow will write out the important products of the reduction cascade to the end products directory (specified by the parameter `END_PRODUCTS_DIR`

[3] If you used the install script `install_uves_reflex`, then the file browser will open in the directory `workflows` containing the UVES workflow in `uves-4.9.0/Uves.xml`.

[4] If you used the install script `install_uves_reflex`, then the value of the parameter `ROOT_DATA_DIR` will already be set correctly to the directory where the standard UVES test data was downloaded.

under “Setup Directories” in the workflow canvas), which includes the extracted and merged spectrum.

11. The workflow will automatically move on to the next data set repeating the reduction cascade and displaying the interactive window with the extracted and merged spectrum. For each data set the procedure is the same; simply inspect the science spectrum and then continue until all nine data sets are processed.

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.

4 About The Reflex Canvas

4.1 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 -> Save 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 File`.

4.2 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.
-  - Full screen (press `Esc` to exit).
-  - Run (or resume) the workflow.
-  - Pause the workflow execution.
-  - Stop the workflow execution.

Note that the full screen mode is not recommended because the workflow canvas does not show any menu or buttons. The remainder of the buttons (not shown here) are not relevant to the workflow execution.

4.3 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 `Run`, `Pause` and `Stop` buttons, respectively. A workflow is executed by clicking the `Run` button. Subsequently the workflow and any running pipeline recipe may be stopped immediately by clicking the `Stop` button, or the workflow may be paused by clicking the `Pause` button which will allow the current actor/recipe to finish execution before the workflow is actually paused. Note that after clicking the `Pause` 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 `Run` button again.

4.4 The Runtime Window

You may find the runtime window a useful aid in monitoring the reduction progress of your data. This window may be started by clicking `Workflow -> Runtime Window` from the `Reflex` canvas menu, and Figure 5 shows how the runtime window looks after having reduced the data sets supplied with `Reflex 1.1`. You will notice that on the left-hand side the runtime window has buttons allowing the control of the workflow (`Go`, `Pause`, `Resume`, `Stop`) and text boxes for controlling workflow parameters such as the working data directory etc.

On the right-hand side of the runtime window is a text box with the title "Recipe Status" which lists the current status of each pipeline recipe and the reduction status of each data set. A recipe may have the following status values:

- `Not Running` - The recipe has not yet been run for any data set so far.
- `Executing` - The recipe is currently executing for a data set.
- `Done` - The last execution of the pipeline recipe was successful.
- `Failed` - The recipe failed on the last data set.
- `Skip` - The recipe was skipped for the last data set.
- `Disabled` - The recipe was disabled for the last data set.
- `Stopped` - The workflow was stopped during the reduction of a data set.

Below the list of recipe status values is a detailed list of input and output files used for each data set within each recipe execution. This information is sometimes very useful for the user who wants to know exactly which files were used as input for a particular data set for a given pipeline recipe, and where the relevant output files were written.

5 The UVES Workflow

The UVES workflow canvas is organised into a number of areas (see Figure 2). 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 green boxes describing the workflow general processing steps in order from left to right, and below this the workflow actors themselves are organised as per the workflow general steps.

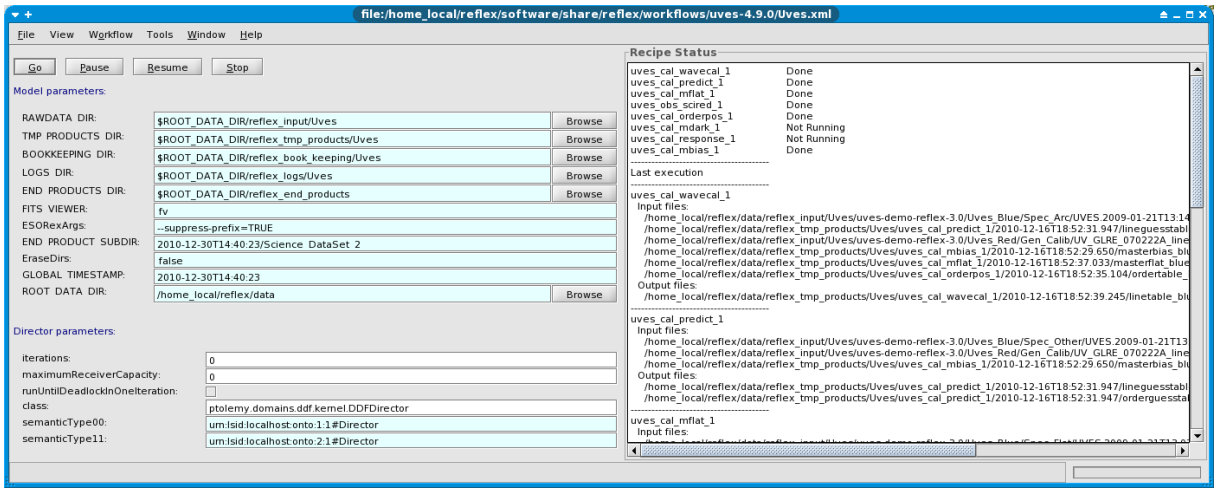


Figure 5: *The runtime window after having reduced the data sets supplied with* `Reflex 1.1`.

5.1 Workflow Canvas Parameters

The workflow canvas displays a number of parameters that may be set by the user (see Figure 2). Under "Setup Directories" the user is only required to set the `ROOT_DATA_DIR` to the working directory for the data set(s) to be reduced, which, by default, is set to the directory containing the standard UVES test 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. 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; Forchì 2010).

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` 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 5.2.4), reusing the previous pipeline recipe outputs where input files and parameters are the same as for the previous execution, which saves considerable processing time. Sometimes it is desirable to set the `EraseDirs` parameter to `true`, which forces the workflow to recursively delete the contents of the directories specified by `BOOKKEEPING_DIR`, `LOGS_DIR` and `TMP_PRODUCTS_DIR`. This is useful for keeping disk space usage to a minimum and will force the workflow to fully rereduce the data each time the workflow is run.

The remaining two global parameters are set automatically by the workflow itself and should not be modified.

5.2 Workflow Actors

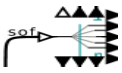
5.2.1 Simple Actors

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. In the UVES workflow, the

following actors are simple actors:

- 

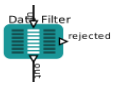
- The `Data Organiser` actor.
- 

- The `Data Set Chooser` actor.
- 

- The `Fits Router` actor
- 

- The `Master Bias Creation` actor.
- 

- The `Order Detection` actor.
- 

- The `Product Renamer` actor.
- 

- The `Data Filter` actor.

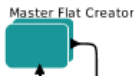
Access to the parameters for a simple actor is achieved by right-clicking on the actor and selecting `Configure Actor`. This will open an "Edit parameters" window (see Figure 6). Note that the `Product Renamer` actor is a jython script (Java implementation of the Python interpreter) meant to be customized by the user (by double-clicking on it).

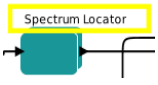
5.2.2 Composite Actors

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 subworkflow of more simple or composite actors which hides over-complexity from the user in the top-level workflow. In the UVES workflow, the following actors are composite actors:

- 

- The `Initialise` actor.
- 

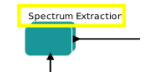
- The `Master Dark Creation` actor.
- 

- The `Master Flat Creation` actor.
- 

- The `Spectrum Locator` actor.
- 

- 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).

Composite actors may also be expanded to inspect the subworkflow that they represent. To do this, right-click on the actor and select `Open Actor`, which will open the subworkflow in a new `Reflex` canvas window. If the composite actor corresponds to a pipeline recipe, then the corresponding recipe executor actor will be present as a simple actor within the subworkflow, and its parameters are accessible as for any other simple actor.

5.2.3 Recipe Executor Actors

A recipe executor 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 Figure 6 we show the "Edit parameters" window for a typical recipe executor 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 recipe executor 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 5.2.4). 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 recipe executor actor will run in Lazy mode or not. A full description of Lazy mode is provided in the next section.
- 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. This option is the default option.
  - `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.

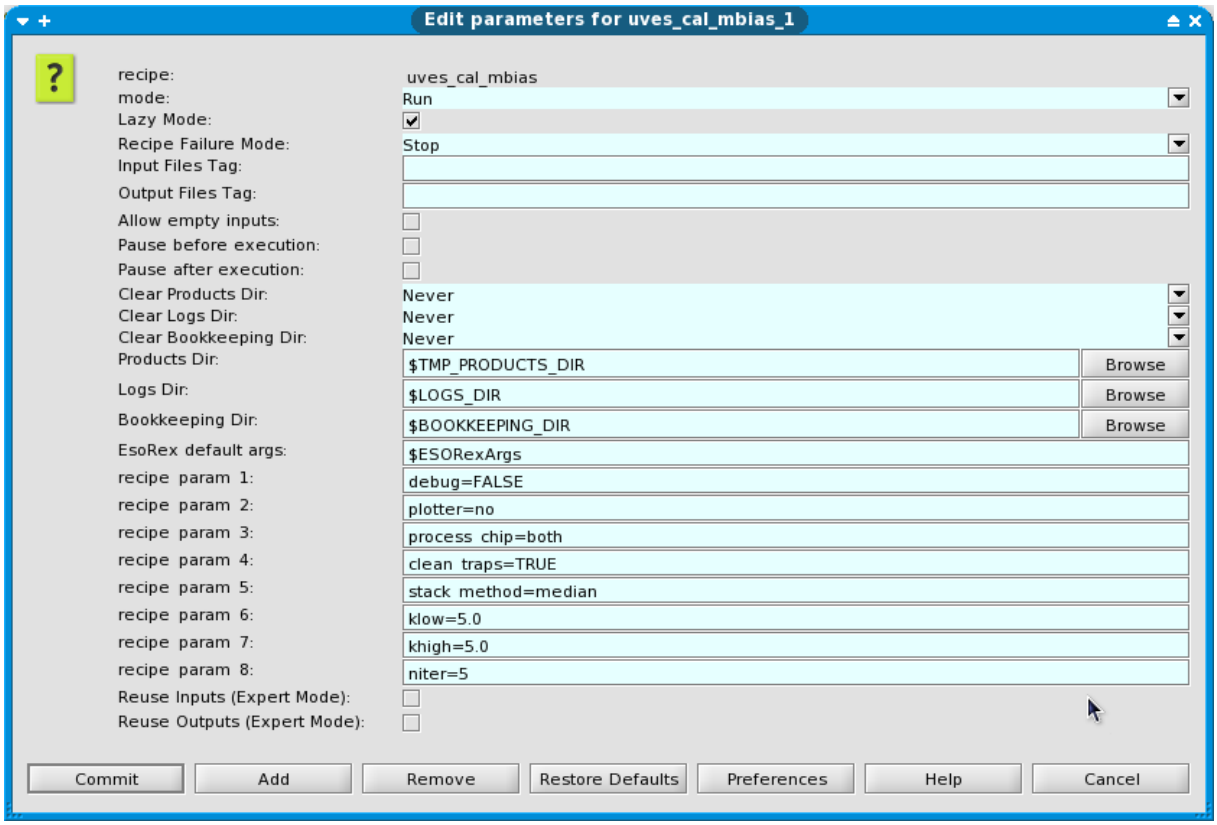


Figure 6: *The "Edit parameters" window for a typical recipe executor actor, the "Master Bias Creation" actor which runs the* `uves_cal_mbias` *pipeline recipe.*

- The set of parameters which start with "recipe param" and end with a number correspond to the parameters of the relevant UVES pipeline recipe. By default in the recipe executor 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. 2010[5]).

The description of the remainder of the recipe executor actor parameters are outside the scope of this tutorial, and the interested user is referred to the Reflex User Manual for further details (Forchì 2010). 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.

5.2.4 Lazy Mode

By default, all recipe executor actors in the UVES 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

[5] Available from http://www.eso.org/sci/facilities/paranal/instruments/uves/doc/index.html

is the case, then the actor will not execute the pipeline recipe, and instead it will simply broadcast the previously generated products to the output port. The purpose of the Lazy mode is therefore to minimise any reprocessing of data by avoiding data rereduction where it is not necessary.

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

The forced rereduction of data at each execution may of course be desirable. To force a rereduction of all data for all recipe executor 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`. To force a rereduction of data for any single recipe executor actor in the workflow, right-click the actor, select `Configure Actor`, and uncheck the Lazy mode parameter tick-box in the "Edit parameters" window that is displayed. For a composite actor, you will first need to open the subworkflow by right-clicking on the composite actor and selecting `Open Actor`.

5.3 Workflow Steps

5.3.1 Step 1: Data Organisation And Selection

On clicking the `Run` 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 5.1). The `Data Organiser` will be executed next which recursively scans the directory specified by `RAWDATA_DIR` and constructs the set of LoSOs from the data it finds.

The next action in this step is the execution of the `Data Set Chooser` which displays the set of LoSOs available in the "Select LOSOs" window, activating a vertical scroll bar on the right if necessary (see Figure 3). Sometimes you will want to reduce a subset of these LoSOs rather than all LoSOs, and for this you may individually select (or de-select) LoSOs 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 LoSO 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 LoSO including the full filename and path for each file, the file category (from the FITS header), and a selection tick box in the first column (see Figure 7). The tick boxes allow you to edit the set of files in the LoSO 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 LoSO may also be saved to disk as an `ASCII` file by clicking on `Save Selected 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 7), 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 LOSOs" window, click either `Continue` in order to continue with the workflow reduction, or `Stop` in order to stop the workflow.

On continuing the workflow, the `Fits Router` actor will be executed which simply sends the relevant files from the `Data Set Chooser` for the current LoSO along the correct paths in the workflow.

5.3.2 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. 2010: Sections 9, 10 & 11) for the details of each recipe and the algorithms employed:

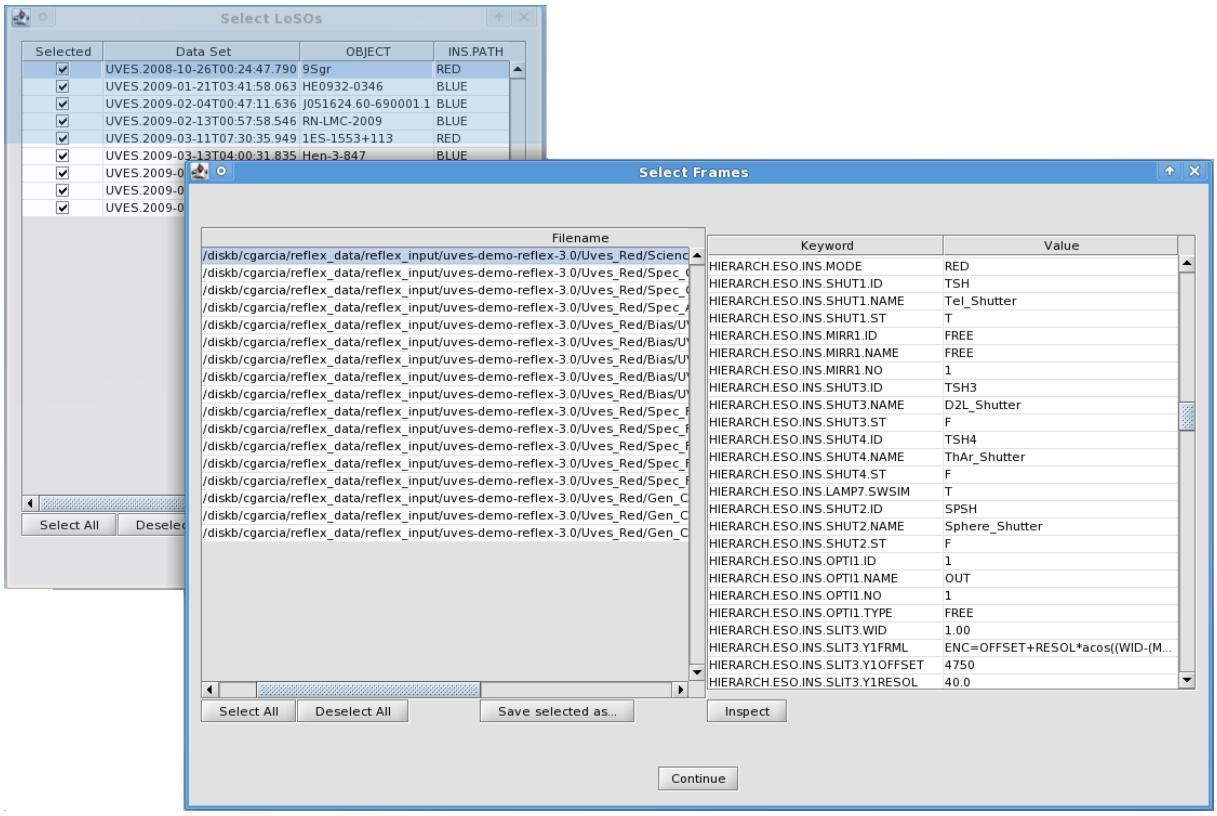


Figure 7: *The "Select Frames" window with a single file from the current LoSO highlighted in blue, and the corresponding FITS header displayed in the text box on the right. Hidden partially behind the "Select Frames" window is the "Select LOSOs" window with the currently selected LoSO highlighted in blue.*

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 LoSO.

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 LoSO, 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.

5.3.3 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. 2010: 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_wavecal` 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 LoSO.

One should also note that a standard star observation is only included in the LoSO for a science observation if it was taken on the same night. Generally, UVES PI-Packs are supplied with master instrument response curves which will also be included in the LoSO 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 LoSO, and failing this, the master instrument response curve will be used if it exists in the current LoSO. If neither a standard star observation nor a master instrument response curve exist in the current LoSO, then the science observation will not be flux-calibrated.

5.3.4 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. 2010: 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).

5.3.5 Step 5: Output Organisation

After having processed the input data for a data set, the workflow highlights and executes the `Product Renamer` actor, which, by default, will copy the most important 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 double-clicking on the `Product Renamer` actor and editing the string 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.
- `<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).

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

6 Optimising Your Results Through Workflow Interaction

In this section, we use the information from Sections 4 & 5 along with the standard UVES data supplied with `Reflex 1.1` 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 nine data sets as described in Section 3, although this is not a pre-requisite to following this section. By doing this, the user will be

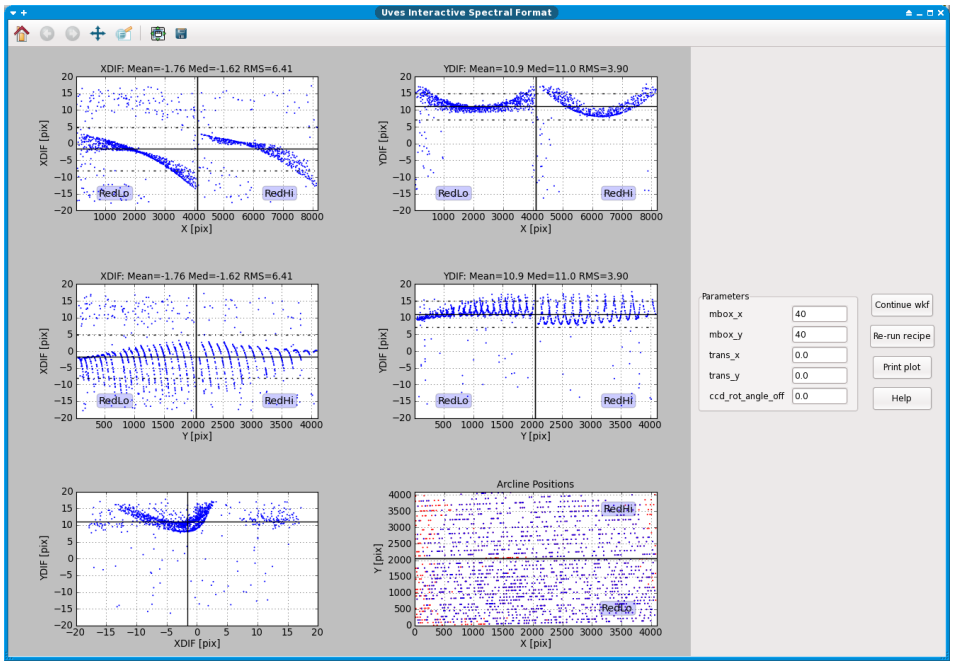


Figure 8: *The interactive window for the* `Spectrum Locator` *actor for the first LoSO in the data supplied with* `Reflex 1.1`.

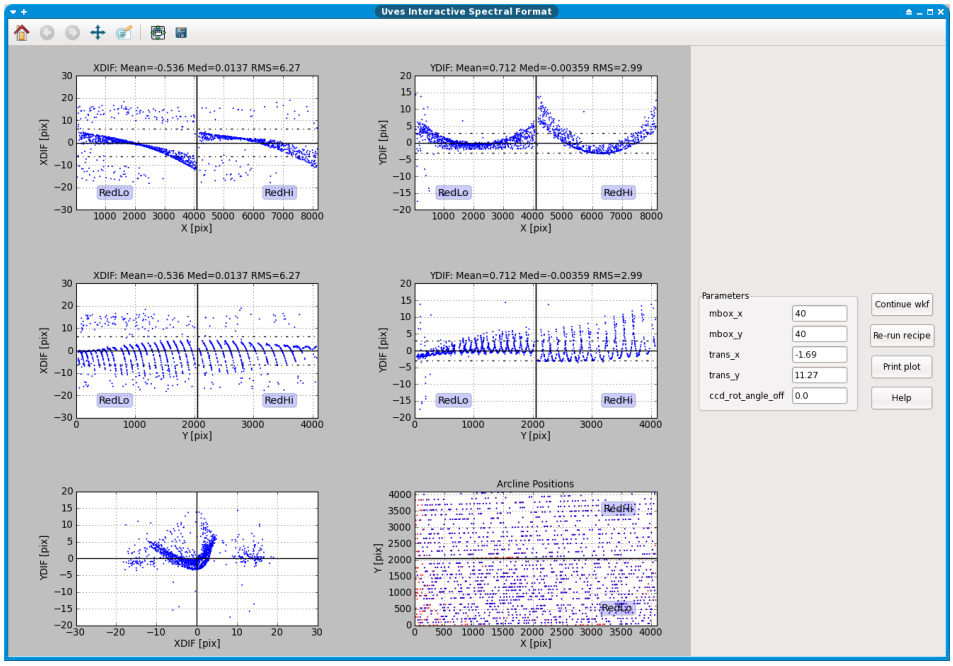


Figure 9: *Same as Figure 8 but after iteration of the corresponding* `uves_cal_predict` *pipeline recipe.*

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 where obtained in a Fedora 11 32 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 standard UVES data supplied with `Reflex 1.1`:

1. Carry out the first four steps described in the quick start Section 3.
2. In the UVES workflow, the four interactive actors (`Spectrum Locator`, `Wavelength Calibration`, `Instrument Response`, `Spectrum Extraction`) are identifiable by a yellow 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 `Run` button  .
4. Click the `Continue` button in the "Select LoSOs" window.
5. `Spectrum Locator:` During the execution of the `Spectrum Locator` actor, an interactive window will appear as shown in Figure 8. The interactive window consists of six plots, five pipeline recipe parameters that may be changed, four 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.

   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 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).
6. 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.
-  - 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 screen-shot of the current interactive window.

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

7. 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. 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 LoSO. The first values for `trans_x` and `trans_y` should be -1.62 and 11.00 pix respectively for this data set, 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 9.

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.

9. Click the `Continue wkf` button in the interactive window.

10. `Wavelength Calibration`: The next interactive window that will appear (see Figure 10) 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

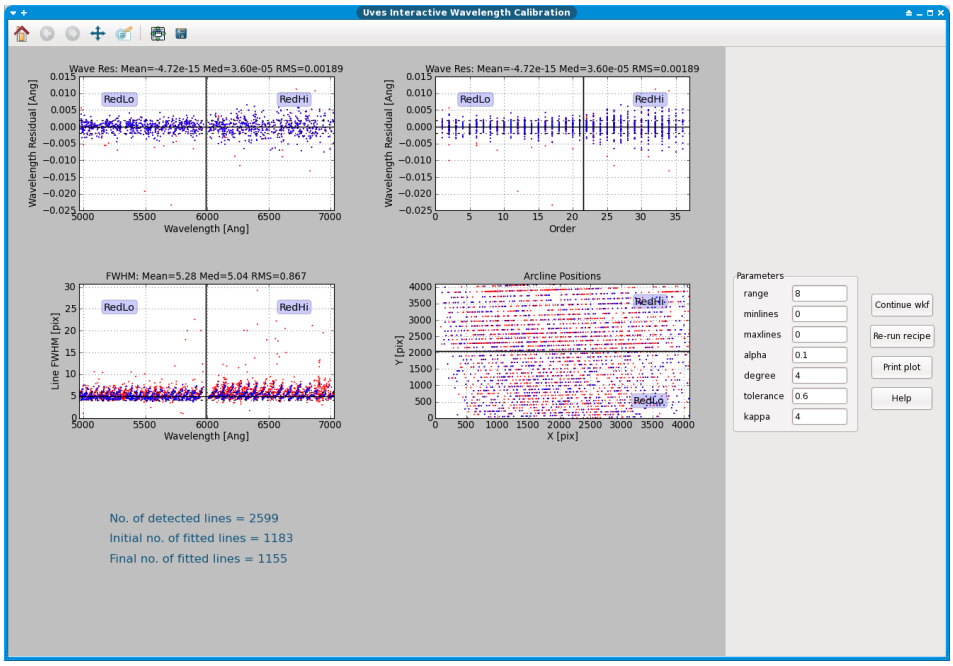


Figure 10: *The interactive window for the* `Wavelength Calibration` *actor for the first LoSO in the data supplied with* `Reflex 1.1`.

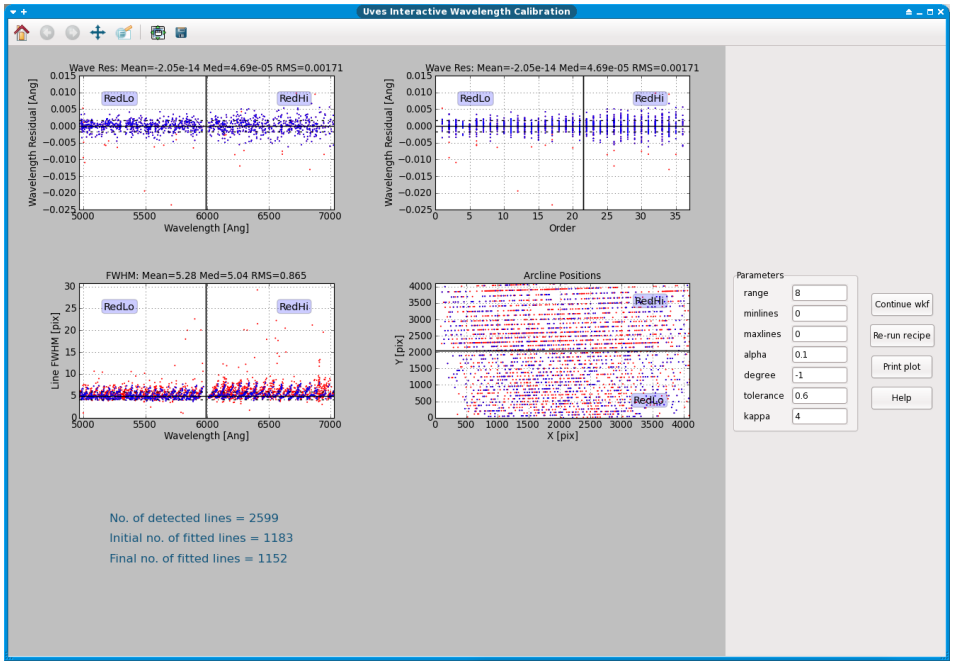


Figure 11: *Same as Fig. 10 but after iteration of the corresponding* `uves_cal_wavecal` *pipeline recipe.*

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.

11. 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 LoSO, based on the parameter descriptions that follow:

    - `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 aswell 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 LoSO, setting `degree = -1` and leaving the other parameters at their default values, is sufficient to reduce the RMS of the residuals from 1.89×10^{-3} to 1.71×10^{-3} Ang while only reducing the final number of fitted lines from 1155 to 1152 (see Figure 11).

12. Click the `Continue wkf` button in the interactive window.

13. `Instrument Response`: The standard data sets supplied with `Reflex 1.1` 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 12 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.

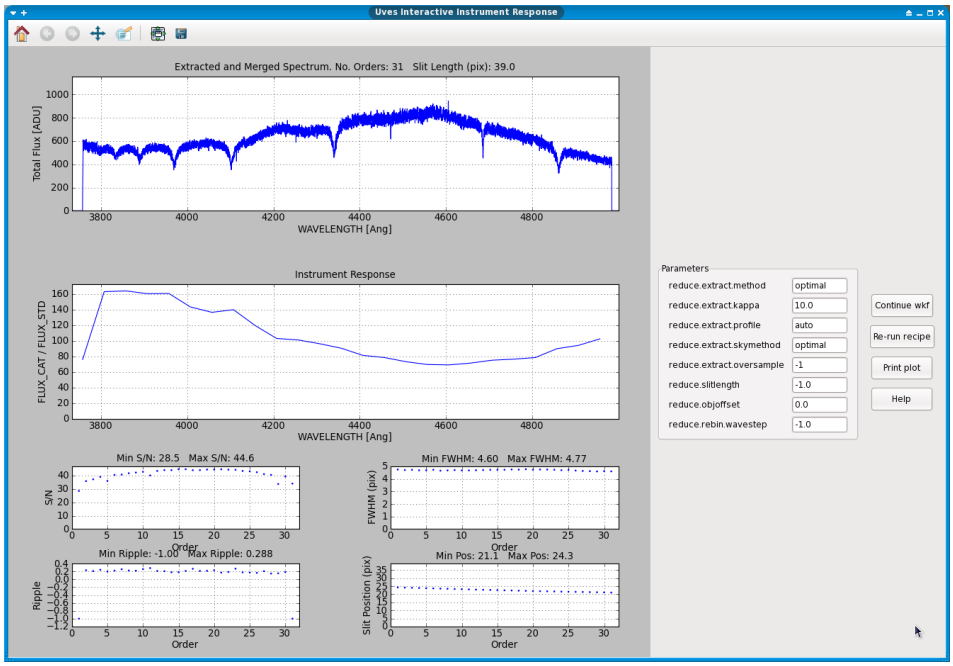


Figure 12: *The interactive window for the* `Instrument Response` *actor.*

14. `Spectrum Extraction`: In Figure 13, 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), 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 and merged spectrum of the science object (blue), and the plot below this shows the flux-calibrated version. Error bars on the spectrum are plotted as 1-sigma limits in red.

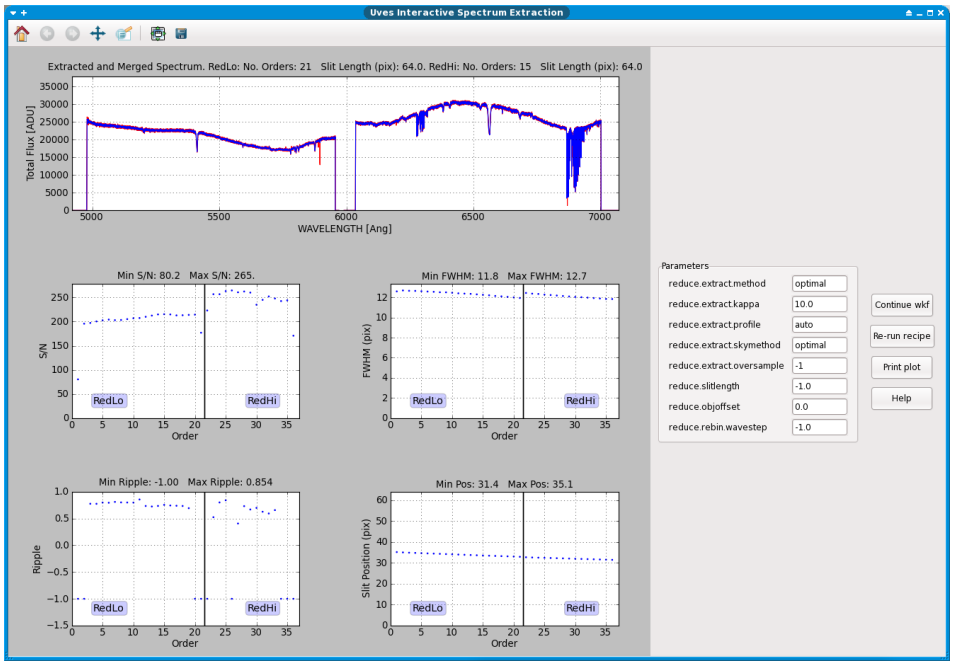


Figure 13: *The interactive window for the* `Spectrum Extraction` *actor for the first LoSO in the data supplied with* `Reflex 1.1`.

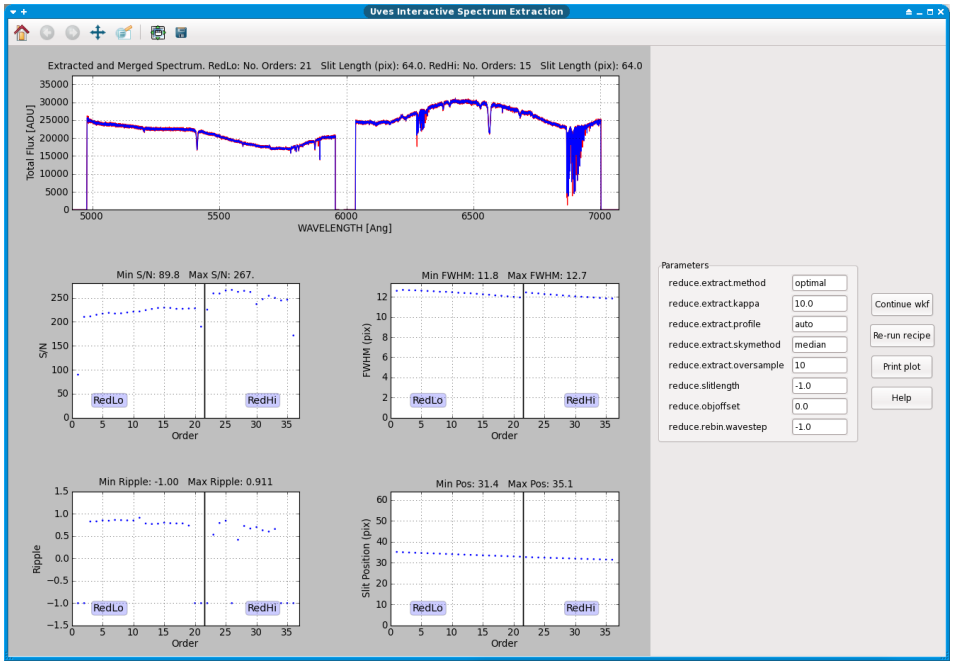


Figure 14: *Same as Fig. 13 but after iteration of the corresponding* `uves_obs_scired` *pipeline recipe.*

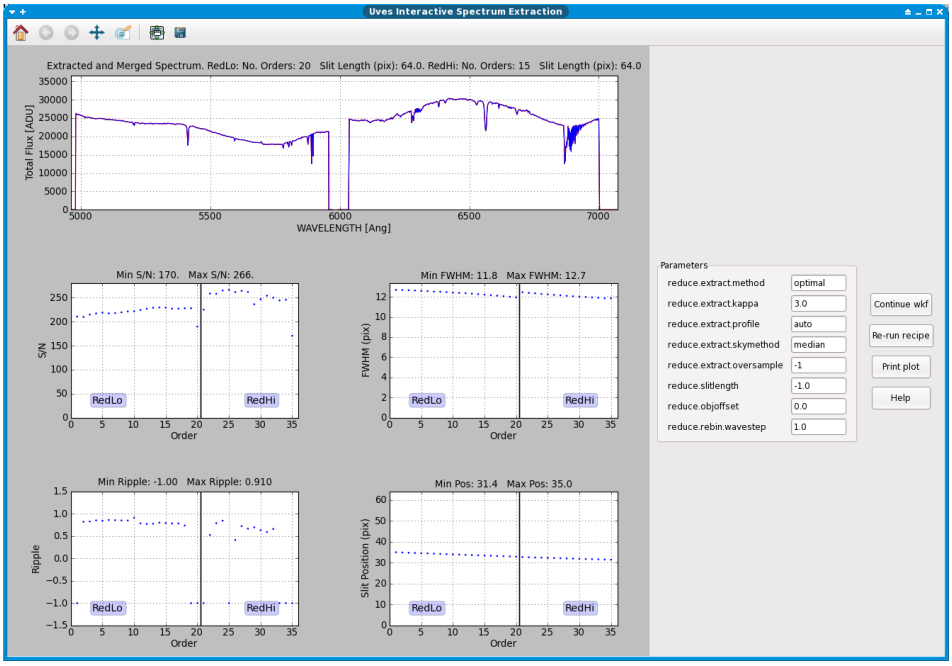


Figure 15: *The interactive window for the* `Spectrum Extraction` *actor for the first LoSO in the data supplied with* `Reflex 1.1` *after iteration of the corresponding* `uves_obs_scired` *pipeline recipe and rebinning to a resolution of 1 Ang.*

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.

15. 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 LoSO, 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 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. 2010) 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 LoSO, 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 265 to 267 (see Figure 14).

16. For illustrative purposes, set `reduce.rebin.wavestep = 1.0 Ang` for the first LoSO, and re-run the pipeline recipe. In Figure 15 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.

17. Click the `Continue wkf` button in the interactive window.

18. As noted in the quick start Section 3, 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).

19. The workflow will now repeat the above reduction steps for the next LoSO, looping over each LoSO until all nine have been processed. The other test data sets 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.

7 Frequently Asked Questions

1. **Why does the actor highlighting jump around the workflow so much?**

   As mentioned in the quick start Section 3, this is a perfectly normal consequence of the workflow director scheduling the workflow execution.

2. **Where are my intermediate pipeline products?**

   Intermediate pipeline products are stored in the directory `<TMP_PRODUCTS_DIR>/UVES` and organised further in directories by pipeline recipe.

3. **I have many LoSOs in my data set (or PI-Pack). How can I reduce them interactively without having to wait a long time between interactive windows being displayed?**

   Reduce all the LoSOs at once with the interactive windows disabled for all interactive actors (right-click the corresponding composite actor, select `Configure Actor`, set the "EnableInteractivity" parameter to `false`, and click `Commit` to save the change). 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`.

4. **Can I use different sets of bias frames to calibrate my flat frames and science data?**

   Currently, the `Reflex` platform only supports the creation of a single masterbias frame (or any other calibration) per LoSO. This means that the same masterbias frame will be used to calibrate the flat frames and science data from the same LoSO. Future releases of `Reflex` will aim to address this limitation.

5. **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.

6. **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 `Projects -> ESO` 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.RecipeExecuter` and in the pop-up window choose the required recipe from the pull-down menu. To connect the ports of the actor, click on the source port, holding down the left mouse button, and release the mouse button over the destination port. Please consult the Reflex User Manual (Forchì 2010) for more information.

7. **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 connections 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 connection from the output port to the diamond. Then make connections from the input ports to the diamond. Please note that you cannot click to start a connection from the diamond itself. Please consult the Reflex User Manual (Forchì 2010) for more information.

8. **Which instruments will be supported in the future?**

   The upcoming workflow releases for next year include AMBER, MIDI, HAWK-I, XSHOOTER and FORS.

8 Troubleshooting

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

   Yes, there is support. However, the OCA rules have to be changed. Right click on the DataOrganizer -¿ Configure Actor. In OCA File substitue uves_wkf.oca by uves_wkf_onlydflats.oca. Then, execute the workflow normally. Notice should be taken that LoSOs with only normal flats will appear now as incomplete.

2. **The workflow reports to me that there are no LoSOs 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 LoSOs.

3. **The "Select LoSOs" window displays my LoSOs, but some/all of them are greyed out. What is going on?**

If a LoSO in the "Select LoSOs" window is greyed out, then it means that the LoSO that was constructed is missing some key calibration(s) (i.e. the LoSO is incomplete). To find out what calibration(s) are missing from a greyed out LoSO, click on the LoSO 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.

4. **I am getting an error message when I start a second instance of `Reflex`. What should I do?**

   When a second instance of `Reflex` is started on the same machine (under the same account), a window with the title "Command failed" will appear. This will also happen if a previous execution of `Reflex` did not finish cleanly. To fix this, close the other instance of `Reflex`, or if it is not alive anymore, kill java.

5. **When I click on the "Configure subplots" button in the interactive window (see Section 6, step 6), the window that appears is empty. Can this be corrected?**

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

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 & Martino Romaniello. Our thanks go to R. Figuera Jaimes for volunteering to be our first external test user.

References

Forchì V., Reflex User Manual, VLT-MAN-ESO-19000-5037, Issue 0.7

Horne K., 1986, PASP, 98, 609

Kaufer A. et al., UV-Visual Echelle Spectrograph User Manual, VLT-MAN-ESO-13200-1825, Issue 86

Larsen J.M., Modigliani A. & Bramich D.M., UVES Pipeline User Manual, VLT-MAN-ESO-19500-2965, Issue 14.0