

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

FLAMES-UVES Pipeline User Manual

VLT-MAN-ESO-19500-3016

Issue 6.6.2

Date 2026-05-22

Prepared:	ESO UVES Pipeline Team	2026-05-22	
	Name	Date	Signature
Approved:	P.Ballester		
	Name	Date	Signature
Released:	M. Peron		
	Name	Date	Signature

This page was intentionally left blank

Change record

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
5.0	28/11/2008		Documentation describing CPL based release 4.3.0
6.0	03/11/2009		Update to 4.4.8
8.0	29/07/2010		Update to 4.7.8
9.0	24/01/2011		Update to 4.9.0
10.0	15/04/2011		Update to 4.9.5
11.0	31/10/2011		Update to 4.9.8
12.0	22/05/2012		Update to 5.0.9
13.0	15/10/2012		Update to 5.0.17
14.0	21/12/2013		Update to 5.1.0
15.0	06/05/2013		Update to 5.2.0
16.0	09/09/2013		Update to 5.2.5
17.0	15/11/2013		Update to 5.3.0
18.0	25/03/2014		Update to 5.4.0
18.1	17/04/2014		Update to 5.4.1
18.2	25/06/2014		Update to 5.4.3
18.3	18/03/2015		Update to 5.5.2
18.4	20/07/2015		Update to 5.5.5
18.5	14/09/2015		Update to 5.5.7
18.6	01/03/2016		Update to 5.7.0
18.7	03/03/2016		Update to 5.7.1
18.9	19/10/2017		Update to 5.8.3
6.1.7	15/03/2022		Update to 6.1.7
6.1.8	19/05/2022		Update to 6.1.8
6.6.2	05/22/2026		Update to 6.6.2

This page was intentionally left blank

Contents

1 Introduction **11**

1.1 Purpose . . . 11

1.2 Acknowledgements . . . 11

1.3 Scope . . . 11

1.4 Conventions . . . 12

1.5 Reference documents . . . 12

2 Overview **14**

3 What's new in pipeline release 6.6.2 **15**

4 FLAMES Instrument Description **16**

4.1 Instrument overview . . . 16

5 Quick start **19**

5.1 FLAMES-UVES pipeline recipes . . . 19

5.2 An introduction to Reflex, Gasgano and EsoRex . . . 19

5.2.1 Using *Reflex* . . . 20

5.2.2 Example of data reduction using the Reflex-based FLAMES-UVES workflow . . . 22

5.2.3 Using Gasgano . . . 25

5.2.4 Using EsoRex . . . 30

5.3 Example of data reduction using EsoRex . . . 32

6 Known problems **43**

7 Instrument Data Description **44**

7.1 Bias frames . . . 44

7.2 Dark frames . . . 44

7.3 Slit flat-field frames . . . 44

7.4 Formatcheck frames . . . 45

7.5 Single fibre order definition frames . . . 45

7.6 Fibre Flat-Field (odd-even-all) frames . . . 45

7.7 Wavelength calibration frames . . . 46
7.8 Supported raw frames (keyword identifiers) . . . 47

8 Static Calibration Data **54**
8.1 Line reference table . . . 54
8.2 Table to monitor line intensity . . . 54

9 Data Reduction **55**
9.1 Data reduction overview . . . 55
9.2 Required input data . . . 55
9.3 Reduction cascade . . . 56
9.3.1 Data reduction peculiarities . . . 57

10 Pipeline Recipes Interfaces **59**
10.1 uves_cal_mbias . . . 60
10.1.1 Input . . . 60
10.1.2 Output . . . 60
10.1.3 Quality control . . . 60
10.1.4 Parameters . . . 61
10.2 uves_cal_mdark . . . 61
10.2.1 Input . . . 61
10.2.2 Output . . . 61
10.2.3 Quality control . . . 62
10.2.4 Parameters . . . 62
10.3 flames_cal_mkmaster . . . 63
10.3.1 Input . . . 63
10.3.2 Output . . . 63
10.3.3 Quality control . . . 63
10.3.4 Parameters . . . 64
10.4 flames_cal_predict . . . 64
10.4.1 Input . . . 64
10.4.2 Output . . . 65

10.4.3 Quality control . . . 66
10.4.4 Parameters . . . 66
10.5 flames_cal_orderpos . . . 67
10.5.1 Input . . . 67
10.5.2 Output . . . 67
10.5.3 Quality control . . . 68
10.5.4 Parameters . . . 68
10.6 flames_cal_prep_sff_ofpos . . . 69
10.6.1 Input . . . 69
10.6.2 Output . . . 70
10.6.3 Quality control . . . 70
10.6.4 Parameters . . . 71
10.7 flames_cal_wavecal . . . 71
10.7.1 Input . . . 71
10.7.2 Output . . . 71
10.7.3 Quality control . . . 73
10.7.4 Parameters . . . 74
10.8 flames_obs_scired . . . 76
10.8.1 Input . . . 76
10.8.2 Output . . . 77
10.8.3 Quality control . . . 78
10.8.4 Parameters . . . 78
10.9 flames_obs_redchain . . . 79
10.9.1 Input . . . 79
10.9.2 Output . . . 79
10.9.3 Quality control . . . 79
10.9.4 Parameters . . . 79

11 Algorithms and recipe details **85**

11.1 Algorithms . . . 85
11.1.1 Error model . . . 85

11.1.2 Physical model offset setting . . . 85
11.1.3 Fibre frames preparation . . . 85
11.1.4 Slit frames preparation . . . 86
11.1.5 Lit fibres flag setting . . . 86
11.1.6 Inter-order background determination . . . 86
11.1.7 Fibre shift determination . . . 87
11.1.8 Generation of UVES related fibre information . . . 88
11.1.9 Frames preparation . . . 88
11.1.10 Bad pixel map preparation . . . 89
11.1.11 Fibre order tracing . . . 89
11.1.12 Iter order background determination . . . 90
11.1.13 Measure of flat part size of Slit frames . . . 90
11.1.14 Slit frames preparation . . . 90
11.1.15 Fiber frames preparation . . . 90
11.1.16 Fiber frames normalization . . . 91
11.1.17 Fiber frames extraction . . . 92
11.1.18 Fiber frames merging . . . 92
11.1.19 Interorder background subtraction . . . 93
11.1.20 Hough transform . . . 93
11.1.21 Order tracing . . . 94
11.1.22 Line Search . . . 94
11.1.23 Wavelength calibration first solution determination . . . 94
11.1.24 Line identification . . . 95
11.2 Recipes . . . 95
11.2.1 Master bias generation . . . 96
11.2.2 Master dark generation . . . 96
11.2.3 Spectral format guess . . . 96
11.2.4 Single fibre order table generation . . . 96
11.2.5 Master slit flat field generation . . . 96
11.2.6 Slit flats preparation, odd-even-all fibre flats preparation, normalization and fibre-order table determination . . . 96

11.2.7 Wavelength Calibration . . . 97
11.2.8 Science data reduction . . . 97
11.2.9 Full UVES-FIBRE data reduction . . . 98

A Workflow technicalities **100**
A.1 Data requirements and optional inputs . . . 100
A.2 Overall layout . . . 100
A.3 OCA rules . . . 101
A.4 Workflow fine tuning and hints . . . 102

B Installation **104**
B.1 Supported platforms . . . 104
B.2 Requirements . . . 104
B.3 Building the UVES pipeline . . . 105

C FLAMES-UVES pipeline data reduction parameters **107**

D Abbreviations and acronyms **114**

1 Introduction

1.1 Purpose

The FLAMES-UVES pipeline is a subsystem of the *VLT Data Flow System* (DFS). Its target user is ESO *Data Flow Operations* (DFO) in the generation of master calibration data, in the reduction of scientific exposures, and in the data quality control. It should also serve as a quick look tool for *Paranal Science Operations* (PSO). Additionally, the FLAMES-UVES pipeline recipes are made public to the user community, to allow a more personalised processing of the data from the instrument.

This manual is a complete description of the data reduction recipes implemented by the CPL based FLAMES-UVES pipeline, reflecting the status of the FLAMES-UVES pipeline as of May 22, 2026 (version 6.6.2). Release 6.6.2 supports the reduction of UVES frames obtained when the UVES slit is fed by the fiber link to FLAMES. [1]

1.2 Acknowledgements

The FLAMES-UVES pipeline has been initially (2001-2003) developed as a MIDAS package by Andrea Modigliani, as a wrapper of the FLAMES-UVES Data Reduction Software developed by Giacomo Mulas, Ignazio Porceddu and Francesco Damiani of the Ital-FLAMES consortium. Starting in winter 2006, to standardize implementation across all ESO pipelines, the FLAMES-UVES pipeline has been ported to CPL by Andrea Modigliani and Jonas Møller Larsen. Thanks to John Pritchard, from the Data Products Department [2], and Domenic Naef from Paranal Science Operations for the feedback provided to improve this release.

1.3 Scope

This document describes the CPL based FLAMES-UVES pipeline used at ESO-Garching and ESO-Paranal for the purpose of data assessment and data quality control. Updated versions of the present document may be found on [12]. For general information about the current instrument pipeline status we remind the user of [5]. Quality Control information are at [4]. Additional information on the Common Pipeline Library (CPL) and EsoRex can be found respectively at [11], [11]. The Reflex front end is described in [20]. The Gasgano front end is described in [13]. Description of the instrument are in [6],[8]. The FLAMES-UVES instrument user manual is in [8]. The FLAMES-UVES calibration plan is in [3], while results of Science Verifications (SV) are at [7]. Additional information on the DFS and VLT data interfaces are in [2], [10], and [17]. A brief description of the FLAMES-UVES data reduction software design is at [1]. First results obtained with the FLAMES-UVES data reduction software are described in [19]. The UVES user manual can be found at [9]. The UVES pipeline user manual can be found at [14]. A clear and compact description of the FLAMES-UVES pipeline is in [16].

The examples on running individual pipeline recipes in this manual use the *EsoRex* command and manually created list of input files. Several interfaces to automatically organise the data, create the list of input

[1] While the pipeline will generally cope with data acquired in non-standard modes ESO offers NO quality control over such data, and therefore it is responsibility of the user to verify the scientific efficacy of any calibration and/or science products produced by the pipeline for non-standard modes

[2] formerly Data Flow Operation Department

files and execute the pipeline recipes in the proper sequence are available, see the ESO Pipeline web page *https://www.eso.org/pipelines* for details.

Please note that the use of Gasgano as a GUI for processing data is deprecated. Its use is no longer recommended and the related section in this manual, as well as support for Gasgano as a data processing GUI application in general will be dropped entirely in a future release.

1.4 Conventions

Throughout this document, we use several conventions to help simplify the text:

- For generic filenames that refer to an unspecified *arm* of the UVES instrument, the `_<arm>` suffix may indicate either `_blue` or `_red` (example: `/path_raw/uves_bias_<arm>.fits`)
- For generic filenames that refer to an unspecified *detector chip* in the UVES instrument, the `_<chipid>` suffix may indicate either `_blue`, `_redl`, or `_redu` (example: `masterdark_<chipid>.fits`)
- For generic `PRO.CATG` TAGs that refer to an unspecified *arm*, the `_<ARM>` suffix may indicate either `_BLUE` or `_RED` (example: `DARK_<ARM>`)
- For generic `PRO.CATG` TAGs that refer to an unspecified *detector chip*, the `_<CHIPID>` suffix may indicate either `_BLUE`, `_REDL`, or `_REDU` (example: `MASTER_BIAS_<CHIPID>`)

1.5 Reference documents

[1] *FLAMES-UVES Data Reduction Software Design and Architecture*. *https://www.eso.org/sci/facilities/paranal/instruments/flames/tools/drs.html*. 11, 57

[2] ESO. *VLT Data Flow System Specifications for Pipeline and Quality Control*. VLT-SPE-ESO-19600-1233. 11

[3] ESO, . *VLT FLAMES Calibration Plan*. VLT-PLA-ESO-13200-1567. 11

[4] ESO/DMO/DFO, *http://www.eso.org/observing/dfo/quality/*. *ESO-Data Flow Operation home page*. 11

[5] ESO/DMO/DFO, *http://www.eso.org/sci/software/pipelines/*. *FLAMES-UVES Pipeline Current Status*. 11

[6] ESO/INS, *http://www.eso.org/sci/facilities/paranal/instruments/flames*. *FLAMES instrument home page*. 11, 47

[7] ESO/INS, *http://www.eso.org/science/vltsv/flamessv*. *FLAMES SV home page*. 11

[8] ESO/INS, *http://www.eso.org/sci/facilities/paranal/instruments/flames/doc*. *FLAMES User Manual*. VLT-MAN-ESO-13700-2994. 11, 47

[9] ESO/INS, *http://www.eso.org/sci/facilities/paranal/instruments/uves/doc*. *UVES User Manual*. VLT-MAN-ESO-13200-1825. 11

[10] ESO/SDD/DFS. *DFS Pipeline & Quality Control – User Manual*. VLT-MAN-ESO-19500-1619. 11

[11] ESO/SDD/DFS, *http://www.eso.org/cpl/esorex.html*. *ESOREX home page*. 11, 31

[12] ESO/SDD/DFS, *http://www.eso.org/pipelines*. *FLAMES-UVES Pipeline Users' Manual*. VLT-MAN-ESO-19500-3016. 11, 57

[13] ESO/SDD/DFS, *http://www.eso.org/gasgano/*. *Gasgano User's Manual*. VLT-PRO-ESO-19000-1932. 11, 14, 28

[14] ESO/SDD/DFS, *http://www.eso.org/pipelines*. *UVES Pipeline User Manual*. VLT-MAN-ESO-19500-2964. 11, 96, 97

[15] ESO/SDD/SPD, *https://ftp.eso.org/pub/dfs/pipelines/instruments/uves/uves-fibre-reflex-tutorial-6.6.2.pdf*. *Reflex FLAMES-UVES Tutorial*, 2013. VLT-MAN-ESO-19500-XXXX. 20, 22

[16] A. Modigliani et al. The FLAMES-UVES Pipeline. ESO/SDD/PSD, *http://www.eso.org/sci/publications/messenger*. The Messenger, **118**, 8, 2004. 11

[17] *http://archive.eso.org/cms/tools-documentation/eso-data-interface-control.html*. *ESO DICB – Data Interface Control Document*. ESO/DMO. GEN-SPE-ESO-19400-0794 (5.0). 11

[18] Kepler Project, code.kepler-project.org/code/kepler-docs/trunk/outreach/documentation/shipping/getting-started-guide.pdf. *Kepler Getting Started Guide*. 20

[19] Andrea Modigliani. *Automatic data reduction in support of the FLAMES-UVES VLT facility*. ESO/SDD/PSD, ADS. SPIE, **4844**, 310-320, 2002. 11, 57

[20] Reflex Project, *http://www.eso.org/sci/software/reflex/*. *Reflex User Manual*. 11

2 Overview

In collaboration with instrument consortia, the Pipeline Systems Department (PSD) of the Software Development Division is implementing data reduction pipelines for the most commonly used VLT/VLTI instrument modes. These data reduction pipelines have the following three main purposes:

Data quality control: pipelines are used to produce the quantitative information necessary to monitor instrument performance.

Master calibration product creation: pipelines are used to produce master calibration products (*e.g.*, combined bias frames, super-flats, wavelength dispersion solutions).

Science product creation: using pipeline-generated master calibration products, science products are produced for the supported instrument modes (*e.g.*, combined ISAAC jitter stacks; bias-corrected, flat-fielded FORS images, wavelength-calibrated UVES spectra). The accuracy of the science products is limited by the quality of the available master calibration products and by the algorithmic implementation of the pipelines themselves. **In particular, adopted automatic reduction strategies may not be suitable or optimal for all scientific goals.**

Instrument pipelines consist of a set of data processing modules that can be called from the command line, from the automatic data management tools available on Paranal or from Gasgano.

ESO offers two front-end applications for launching pipeline recipes, *Gasgano* [13] and *EsoRex*, both included in the pipeline distribution (see Appendix B, page 104). These applications can also be downloaded separately from `www.eso.org/gasgano` and `www.eso.org/cpl/esorex.html`. An illustrated introduction to Gasgano is provided in Section 5.

The FLAMES facility and the different types of FLAMES-UVES raw frames and auxilliary data are described in Sections 4, 7, and 8.

A brief introduction to the usage of the available reduction recipes using Gasgano or EsoRex is presented in Section 5. In section 6 we advise the user about known data reduction problems.

An overview of the data reduction, the input data, and the recipes involved in the calibration cascade is provided in section 9.

More details on inputs, products, quality control measured quantities, and controlling parameters of each recipe is given in section 10.

More detailed descriptions of the data reduction algorithms used by the individual pipeline recipes can be found in Section 11.

In Appendix B the installation of the FLAMES-UVES pipeline recipes is described, in Appendix C the full list of relevant data reduction parameters is provided, and in Appendix D a list of used abbreviations and acronyms is given.

3 What's new in pipeline release 6.6.2

The following major changes has been realised on the FLAMES-UVES pipeline:

- Updated to CPL7.3
- Document a rare numerical instability in **flames_cal_prep_ofpos** that generates Singular Matrix errors
- Expose slit geometry parameter **reduce.objslit** missing from the interactive Python workflow actors
- Expose the recently added **reduce.rebin.wavestep_redu** parameter in the appropriate interactive Python workflow actors
- Adopt the `_<arm>`, `_<ARM>`, `_<chipid>`, and `_<CHIPID>` conventions for all tutorials and manuals and add a Conventions section to describe them
- Other minor documentation corrections

4 FLAMES Instrument Description

FLAMES is the multi-object, intermediate and high resolution spectrograph of the VLT. Mounted at UT2, FLAMES can access targets over a field of view 25 arcmin in diameter. FLAMES feeds two different spectrographs covering the whole visual spectral range: GIRAFFE and UVES. GIRAFFE allows the observation of up to 130 targets at the time or to do integral field spectroscopy, with intermediate resolution (either R $\approx$ 25000 or R $\approx$ 10000). UVES provides the maximum possible resolution (R=47000) but can access only up to 8 objects at the time. The instrument has been made available to the community and started operations in Paranal on April 1^{st}, 2000.

In this chapter a brief description of the FLAMES-UVES fiber link and the UVES instrument is given. A more complete documentation can be found in the FLAMES and UVES User Manuals [10,18].



Figure 4.1: A photo of FLAMES mounted at the Nasmyth A focus of Kuyen (VLT-UT2).

4.1 Instrument overview

FLAMES consists of three main components:

- A Fibre Positioner (OzPoz) hosting two plates: while one plate is observing the other positions the fibres for the subsequent observations, therefore limiting the dead time between one observation and the next to less than 15 minutes, including the telescope preset and the acquisition of the next field.

- A medium-high resolution optical spectrograph, GIRAFFE, with three types of fibre-fed systems : MEDUSA (132 individually deployable 1.2 arcec entrance aperture fibres), IFU (15 deployable 2x3 arcmin aperture IFUs), ARGUS (an 11.5 or 6.6x4.2 arcsec Integral Field Unit).
- A link to the UVES spectrograph (Red Arm) via 8 single fibres of 1 arcsec entrance aperture.

Special observing software (FLAMES OS) coordinates the operation of the different subsystems, also allowing simultaneous acquisition of UVES and GIRAFFE observations. For combined observations, the exposure times for UVES and GIRAFFE do not need to be the same. Note that it is not possible to observe simultaneously in two GIRAFFE modes, or to observe the same target simultaneously with the two spectrographs.

UVES is the high resolution spectrograph at UT2 of the VLT (see Section 6.4). It was designed to work in long slit mode but it has been possible to add a fibre mode (6 or 8 fibres, depending on setup and/or mode)[3] fed by the FLAMES positioner to its Red Arm only. Only three standard UVES Red setups are offered, with central wavelength of 520, 580 and 860 nm respectively (see the manual for details). The standard readout mode of FLAMES-UVES is 225 kHz (unbinned) which ensures low readout noise. As of P76 a high-speed readout mode (625 kHz, unbinned, low gain) with increased readout noise but less overheads is offered in visitor mode only. No pipeline support is available in this mode. With an aperture on the sky of 1 arcsec, the fibres project onto a resolution spot of diameter 5 UVES pixels giving a resolving power of 47000. For faint objects and depending on the spectral region, one or more fibres can be devoted to recording the sky contribution. In addition, for the 580 nm setup only, a separate calibration fibre is available to acqire simultaneous ThAr calibration spectra. This allows very accurate radial velocity determinations. In this configuration, 7 fibres remain available for targets on sky.

Spectrograph	Mode	N. of Objects N. of Objects	Aperture (arcsecs)	Resolving power	Spectral Band [nm]	Central wavelength	Observing Mode
UVES	Red arm	8	1.0	47000	200	520/580/860	OzPoz
UVES	Red arm	7+1(calibration)	1.0	47000	200	580	SimCal

Table 4.1: This table summarize main features of UVES fibres observations

[3] Pls note that FIBER mode data obtained in plate2 after 9 September 2012 the fiber assigned to button 37 is not connected

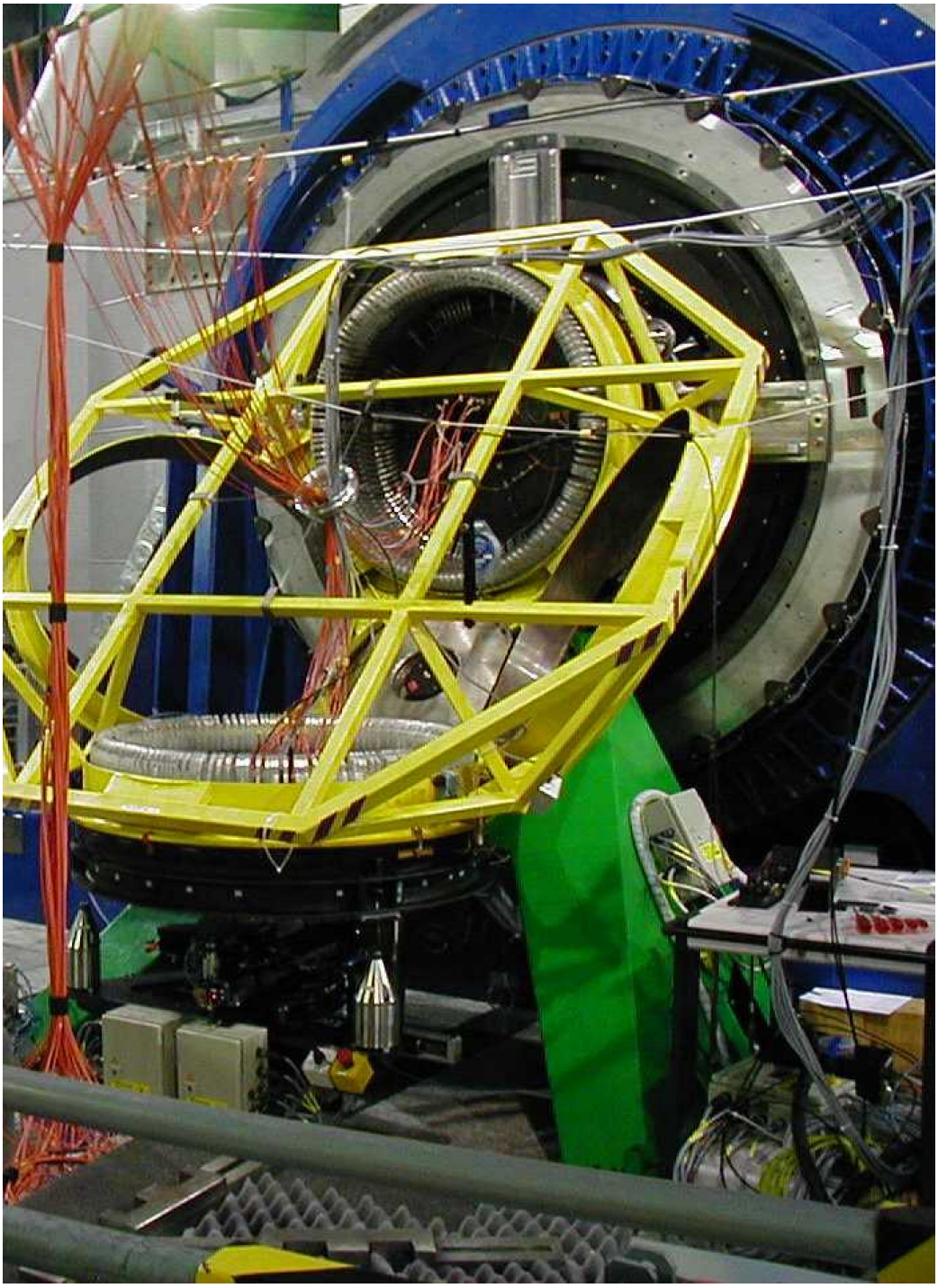


Figure 4.2: A photo of the OzPoz fibre positioner. While one plate is observing, the other one is positioning the fibres for the subsequent observations.

5 Quick start

This section describes the most immediate usage of the FLAMES-UVES pipeline recipes.

5.1 FLAMES-UVES pipeline recipes

The current FLAMES-UVES pipeline is based on a set of 7 stand-alone recipes plus one utility recip (and uses 2 recipes from the UVES pipeline to combine biases and darks) involved in the data reduction cascade:

uves_cal_mbias creates a master bias frame.

uves_cal_mdark creates a master dark frame.

flames_cal_predict implements the UVES physical model (and applies a shift to reflect the offset between the calibration fibre and the order centre).

flames_cal_orderpos defines the calibration fibre order positions.

flames_cal_mkmaster creates master slit flat-field frames.

flames_cal_prep_sff_ofpos process master slit flats and odd-even-all fibre flats to determine the a table tracing all fibres and creates additional frames, needed to extract the science data.

flames_cal_wavecal performs the wavelength calibration.

flames_obs_scired reduces a science frame.

flames_utl_unpack uppack products relative to each fibre from the products usually delivered by the Quality Control Group of ESO, Garching.

5.2 An introduction to Reflex, Gasgano and EsoRex

Before being able to call pipeline recipes to process a set of data, the data must be correctly classified, and associated with the appropriate calibrations. The *Data Classification* consists of tasks such as: "What kind of data am I?", *e.g.*, BIAS, "to which group do I belong?", *e.g.*, to a particular Observation Block or observing template. *Data Association* is the process of selecting appropriate calibration data for the reduction of a set of raw science frames. Typically, a set of frames can be associated if they share a number of properties, such as instrument and detector configuration. Since all the required information is stored in the FITS headers, data association is based on a set of header keywords (called "association keywords") and the process is specific to each type of calibration. The process of data classification and association is known as data organisation.

An instrument pipeline consists of a set of data processing modules that can be called from different host applications, namely:

- *Reflex* is a graphical tool that helps the user to execute data reduction workflows which contain several recipes. This dramatically decreases the time the user needs to run a whole reduction chain, from calibration and raw data down to the final products. *Reflex* takes care of grouping the different data sets,

associating the calibration frames and managing the interdependencies between recipes in the calibration cascade. **Reflex is the recommended software tool for reducing your data.**

- *Gasgano* is an alternative data management tool that simplifies the data organization process. In addition, *Gasgano* allows the user to execute directly the pipeline recipes on a set of selected files.
- *EsoRex* is a command line tool used to run the pipeline recipes. *EsoRex* commands can be easily scripted.
- The Paranal observatory implements automatic data management tools that trigger the execution of pipeline recipes. This aspect is not covered in this manual.

5.2.1 Using *Reflex*

Reflex is the recommended tool to reduce complete data sets that include all the calibration frames. It is an advanced tool, and yet easy to use, that is geared towards maximum scientific return. It is based on the workflow engine *Kepler* [18].

This manual does not cover the installation of *Reflex* . Please refer to [15] for the installation procedure which also contains a detailed description of the *Reflex* application. What follows is a very brief summary of it.

Once installed, *Reflex* can be executed with the command:

```
user@host# reflex &
```

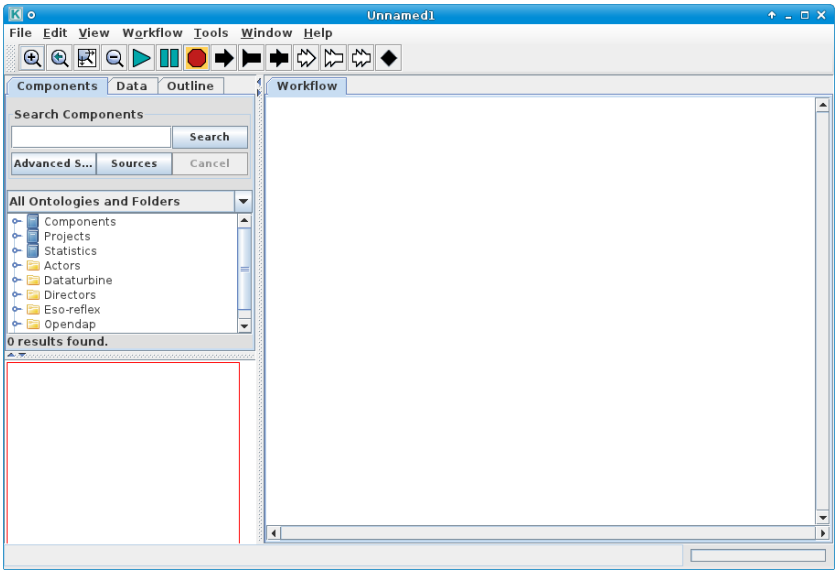


Figure 5.1: *Fresh Reflex canvas.*

Reflex main concepts are workflows and actors. Workflows are canvasses which show the interdependence of the pipeline recipes, allowing the user to easily obtain an overview of the reduction steps. Workflows have the advantage of requiring a small learning curve in order to get the pipeline running.

Actors are the entities which actually perform some kind of operation. In *Reflex*, to each main actors correspond the pipeline recipes themselves, which perform the data reduction steps, but there are other actors such as the DataOrganizer, or the FitsRouter that are useful to manage the data files. Each actor can be configured by right-clicking on it and selecting *Configure Actor* as shown in Figure 5.2. In the case of the recipe actors, the recipe parameters are part of the actor and make up the second group of parameters.

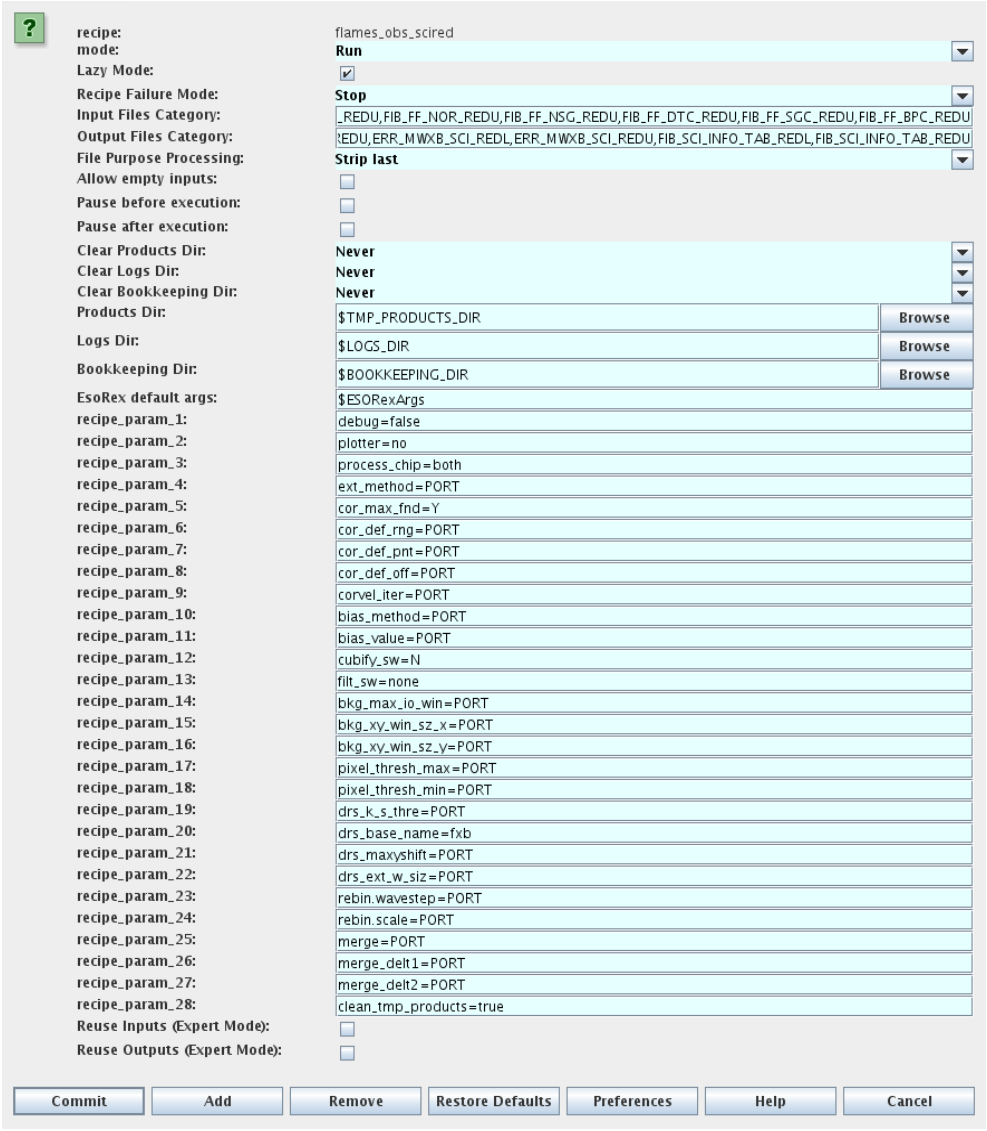


Figure 5.2: *Parameters of a recipe actor. The first group of parameters affect the execution of the pipeline recipe and are common to all recipe actors. The second group of parameters are specific to the pipeline recipe to be called and they are identical to those that can be configured in EsoRex (see 5.2.4).*

In addition to those elements, the workflow contains variables that contain the most important settings, such as the directories where data is located and will be saved.

To start using *Reflex* with this pipeline, please refer to [15].

5.2.2 Example of data reduction using the Reflex-based FLAMES-UVES workflow

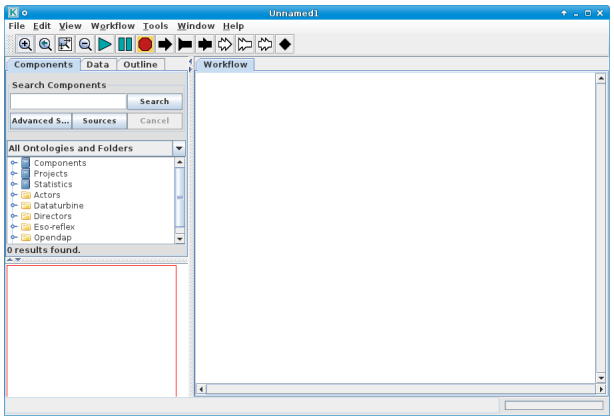


Figure 5.3: *The empty* `Reflex` *canvas.*

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 FLAMES-UVES demo data set supplied with the `Reflex 2.11.5` 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 5.1 will appear.

2. Now open the UVES workflow by clicking on `File -> Open File`, selecting first `uves-6.6.2` and then the file `flames-uves.xml` in the file browser. You will be presented with the workflow canvas shown in Figure 5.4. 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 `OK`.

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[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`

[4]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.

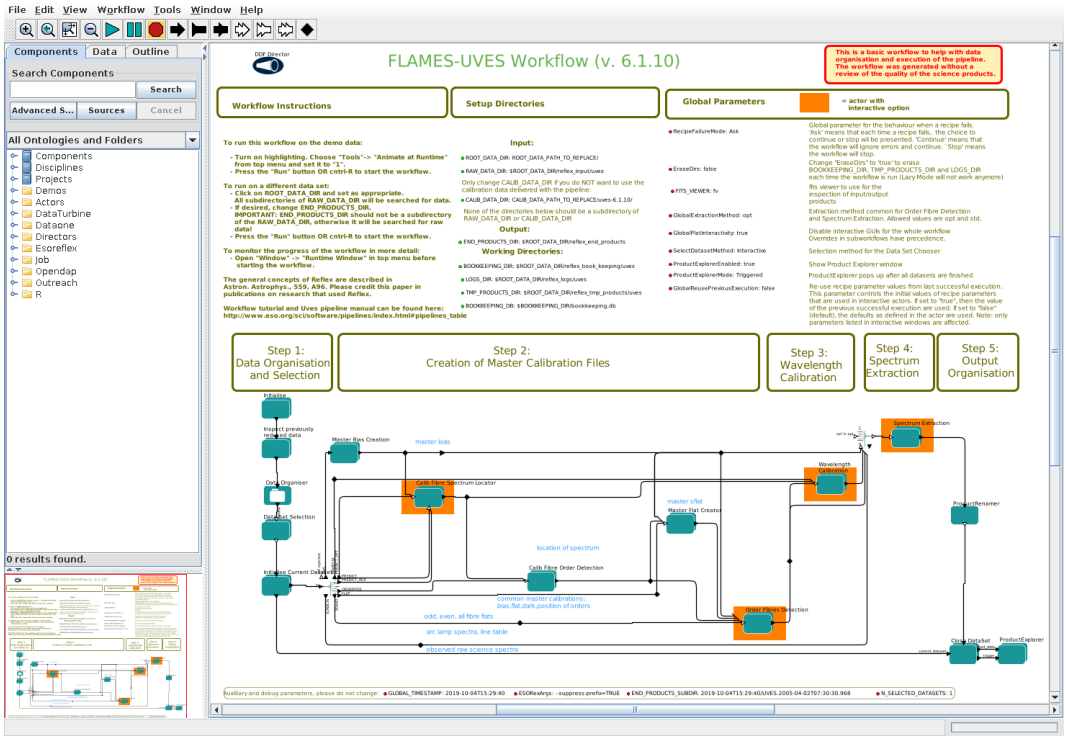


Figure 5.4: *FLAMES-UVES workflow general layout.*

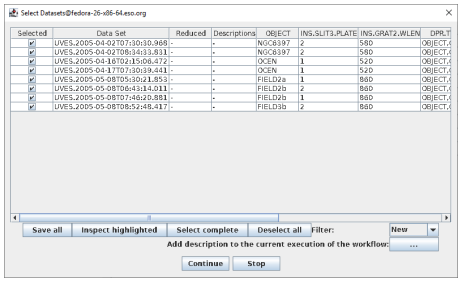


Figure 5.5: *The "Select Datasets" pop-up window.*

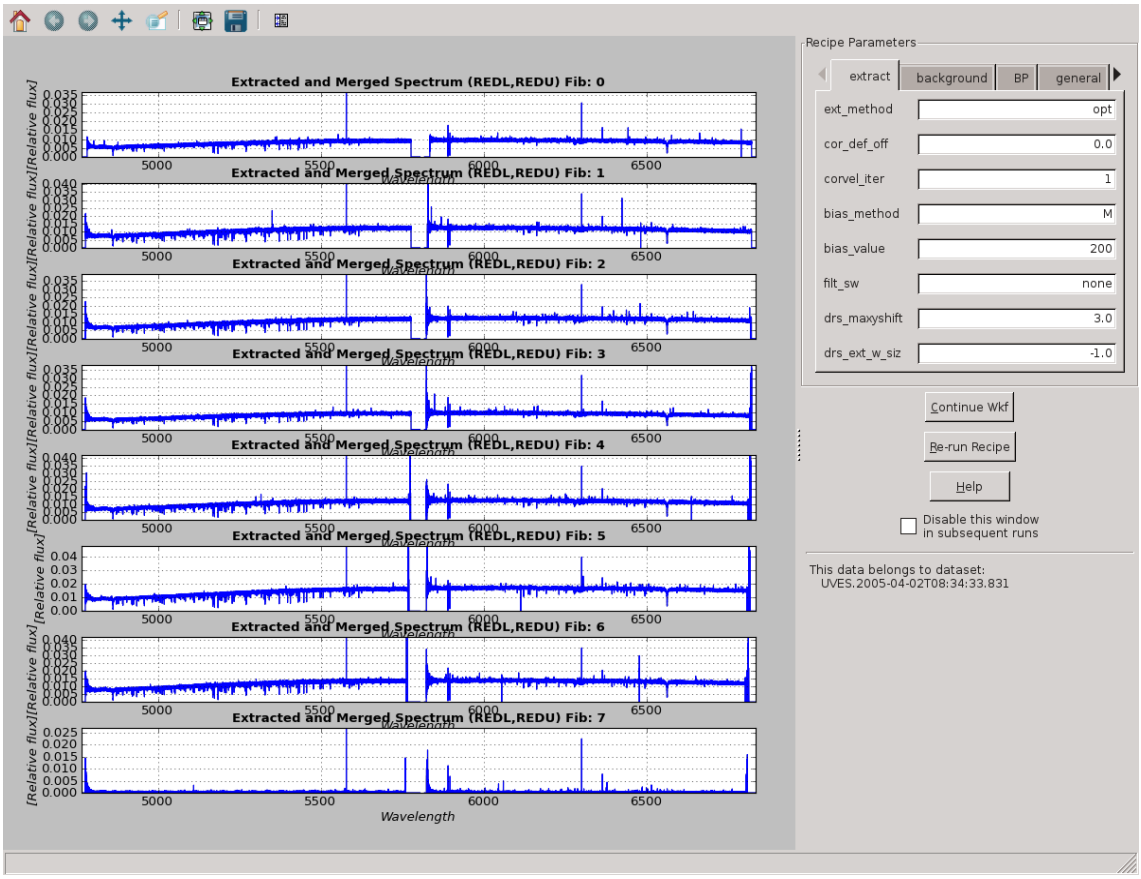


Figure 5.6: *The interactive pop-up window for the* `Spectrum Extraction` *actor and FLAMES-UVES pipeline recipe* `flames_obs_scired`*. The extracted and merged spectra for the second DataSet are displayed..*

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. When you have finished, click OK to save your changes.

5. Click the  button to start the workflow

6. 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.

7. The Data Set Chooser actor will be highlighted next and will display a "Select Datasets" window (see Figure 5.5) that lists the DataSets along with the values of a selection of useful header keywords[5]. The first column consists of a set of tick boxes which allow the user to select the DataSets to be processed, and by default all complete DataSets are selected.

8. Click the Continue button and watch the progress of the workflow by following the red highlighting of the actors. A window will show which DataSet is currently being processed.

9. When the reduction of the current DataSet finishes, a pop-up window will appear showing the directory were the final products have been saved.

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

11. 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.

5.2.3 Using Gasgano

To get familiar with the FLAMES-UVES pipeline recipes and their usage, it is advisable to begin with *Gasgano*, because it provides a complete graphic interface for data browsing, classification and association, and offers several other utilities such as easy access to recipe documentation and preferred data display tools.

Gasgano can be started from the system prompt in the following way:

```
gasgano $HOME/gasgano/config/UVES.prefs &
```

where we have passed as first optional argument explicitly the UVES preference file which defines proper defaults for UVES data reduction (for both ECHELLE and FIBER modes). The user may like to realias the gasgano command to the previous command. The *Gasgano* main window will appear. In Figure 5.7 (next

[5]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.

page), a view on a set of FLAMES-UVES data is shown as an example. *Gasgano* can be pointed to the directories where the data to be handled are located using the navigation panels accessible via the *Add/Remove Files* entry of the *File* menu (shown on the upper left of the figure).

The data are hierarchically organised as preferred by the user. After each file name are shown the classification and the values of the following FITS keywords (we omit the prefix HIERARCH.ESO):

Keyword name	Purpose
CLASSIFICATION	Data classification
OBS.TARG.NAME	Observation Block target name
EXPTIME	Exposure time
DATE	Observing date
INS.GRAT2.WLEN	Instrument setting central wavelength
DET.READ.SPEED	Readout speed
INS.OBSPLATE	Observation plate
OCS.SIMCAL	Simultaneous calibration

Additionally Gasgano uses other keywords neded to properly classify UVES data taken using the slit mode.

The CLASSIFICATION field provides either the value of the PRO.CATG, for pipeline products; or a user defined file classification, if provided, defined in the classification rule file, which can be accessed by Gasgano from the Tools –> Classification rules... tab; or the default value “UNDEFINED”. File classification rules are selection rules which assign to a FITS file a classification based on the value of a few FITS keywords, usually the DPR.TYPE, DPR.TECH, DPR.CATG values, which respectively define the file data type, acquisition technique and category.

Keyword name	Purpose
DET.CHIPS	# of chips in detector array
INS.MODE	Instrument mode used
INS.GRAT2.NAME	Instrument grating name
INS.SLIT3.WID	Instrument slit width
DET.WIN1.BINX	Binning factor along X
DET.WIN1.BINY	Binning factor along Y

Additional relevant keywords for FLAMES-UVES data are INS.GRATi.WLEN, INS.OBSPLATE, OCS.SIMCAL. In order to properly classify data taken in slit mode it is useful also to display the keyword values of DET.CHIPS, INS.SLITi.NAME, DET.WIN1.BINX/Y, INS.MODE, INS.GRATi.NAME. Those relevant keywords are indicated by Gasgano either in the file section, or by selecting each file, in the section which shows the FITS file header content.

More information about a single frame can be obtained by clicking on its name: the corresponding FITS file header will be displayed on the bottom panel, where specific keywords can be filtered and searched. Images and tables may be easily displayed using the viewers specified in the appropriate *Preferences* fields. Such a field also allows to set the file filter, which should point to the $HOME/gasgano/config/uves.oca. This rule file provides simple filtering rules to select FLAMES-UVES data corresponding to a given standard data reduction setting.

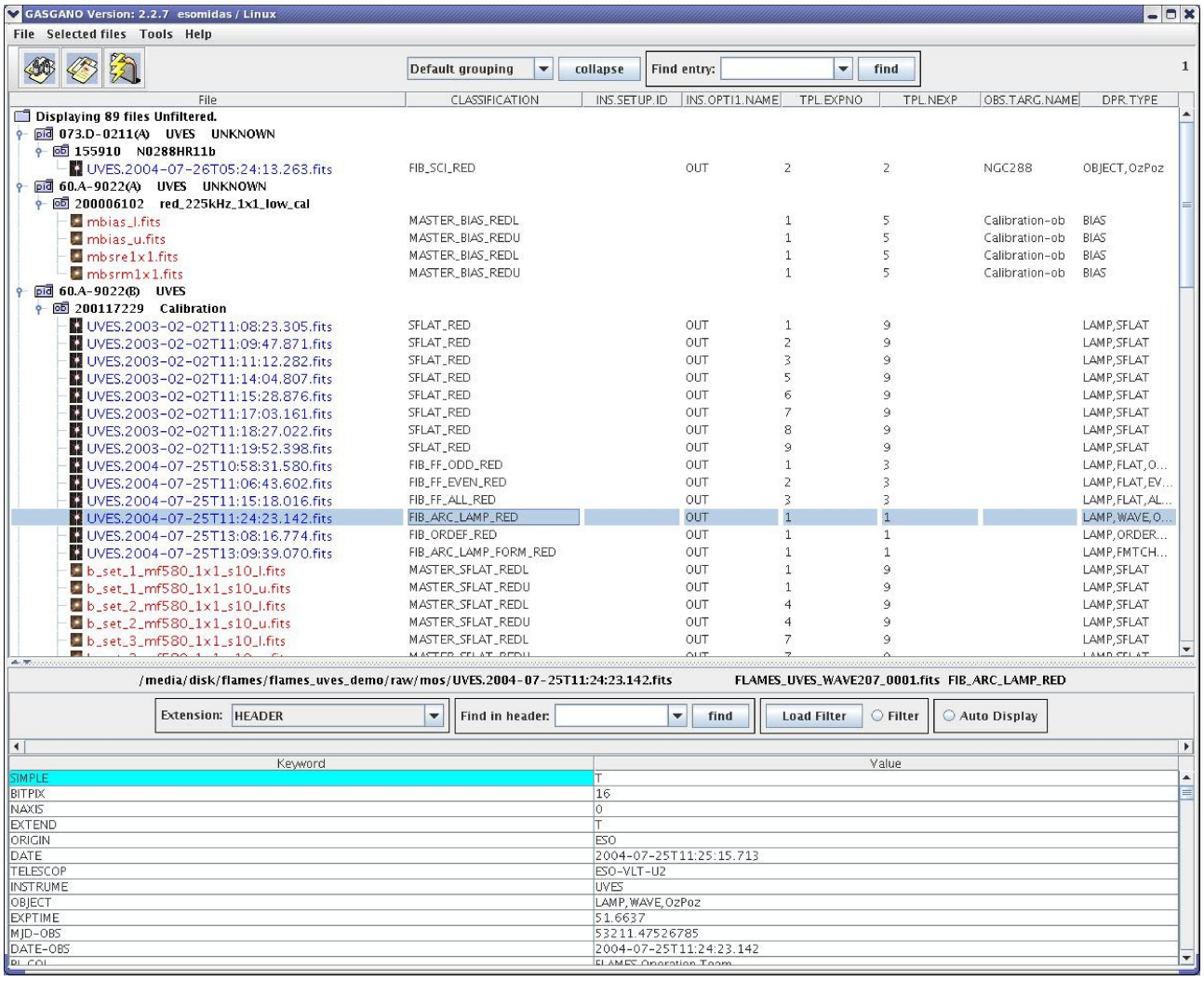


Figure 5.7: *The Gasgano main window.*

Frames can be selected from the main window for by the appropriate recipe: in Figure 5.8, the Wavelength calibration frame (CLASSIFICATION=FIB_ARC_LAMP_RED), previously produced master bias and master flat frames, together with line and order tables, a table with the reference standard star spectra and one with the atmospheric dispersion are all selected and sent to the *flames_cal_wavecal* recipe. This will open a *Gasgano* recipe execution window (see Figure 5.9), having all the specified files listed in its *Input Frames* panel.

Help for the recipe is available from the *Help* menu. Before launching the recipe, its configuration may be modified on the *Parameters* panel (on top). The window contents might be saved for later use by selecting the *Save Current Settings* entry from the *File* menu, as shown in figure.

At this point the recipe can be launched by pressing the *Execute* button. Messages from the running recipe will appear on the *Log Messages* panel at the bottom, and in case of successful completion the products will be listed on the *Output Frames* panel, where they can be easily viewed and located back on the Gasgano main

window. To produce useful plots the user needs to set to 'gnuplot -persist' the **plotter** recipe parameter value (and have a valid installation of gnuplot package, and the gnuplot command available in the PATH). This basic plotting functionality is supported by EsoRex but not by gasgano.

Please refer to the *Gasgano User's Manual* [13] for a more complete description of the *Gasgano* interface.

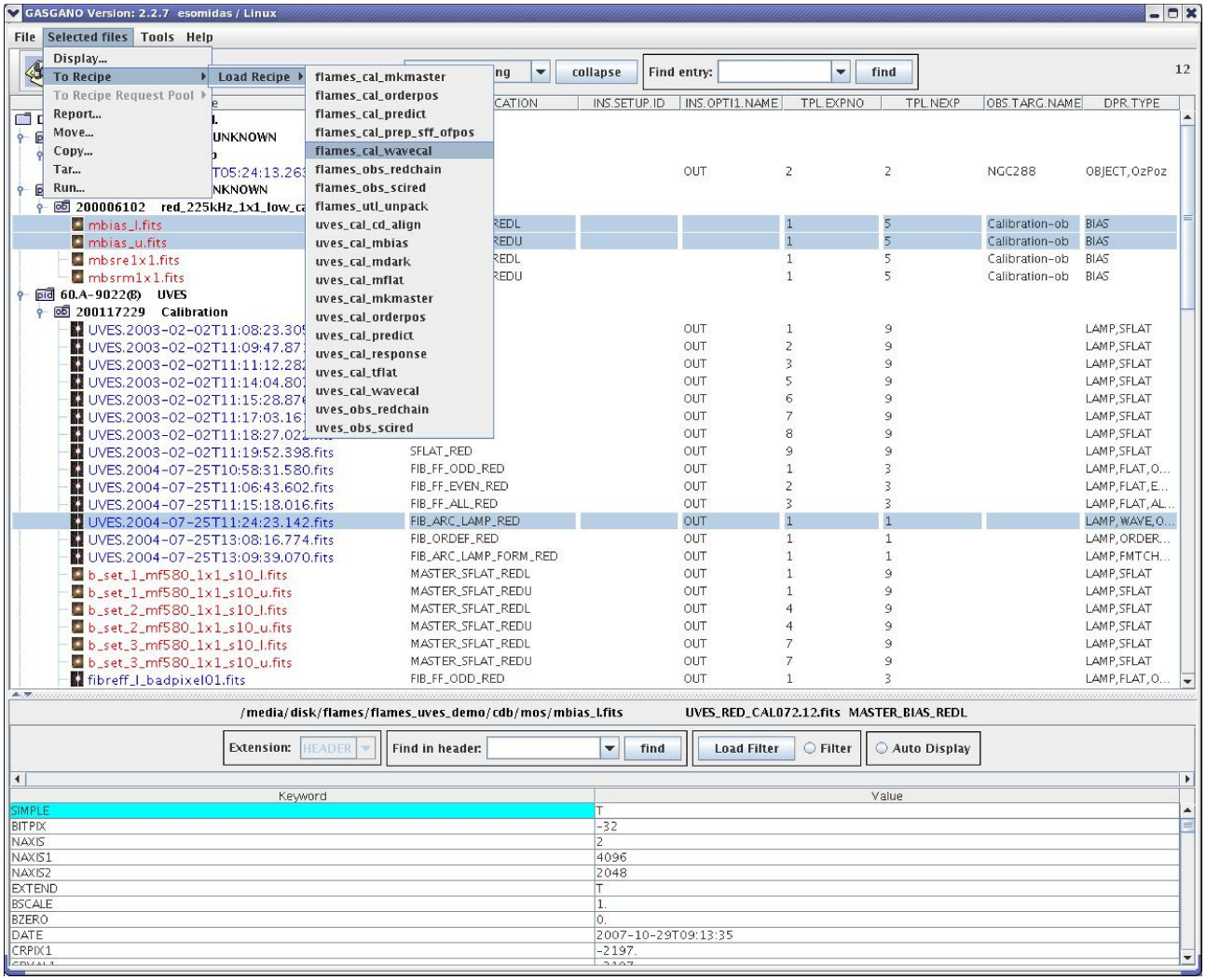


Figure 5.8: *Selecting files to be processed by a FLAMES-UVES pipeline recipe.*

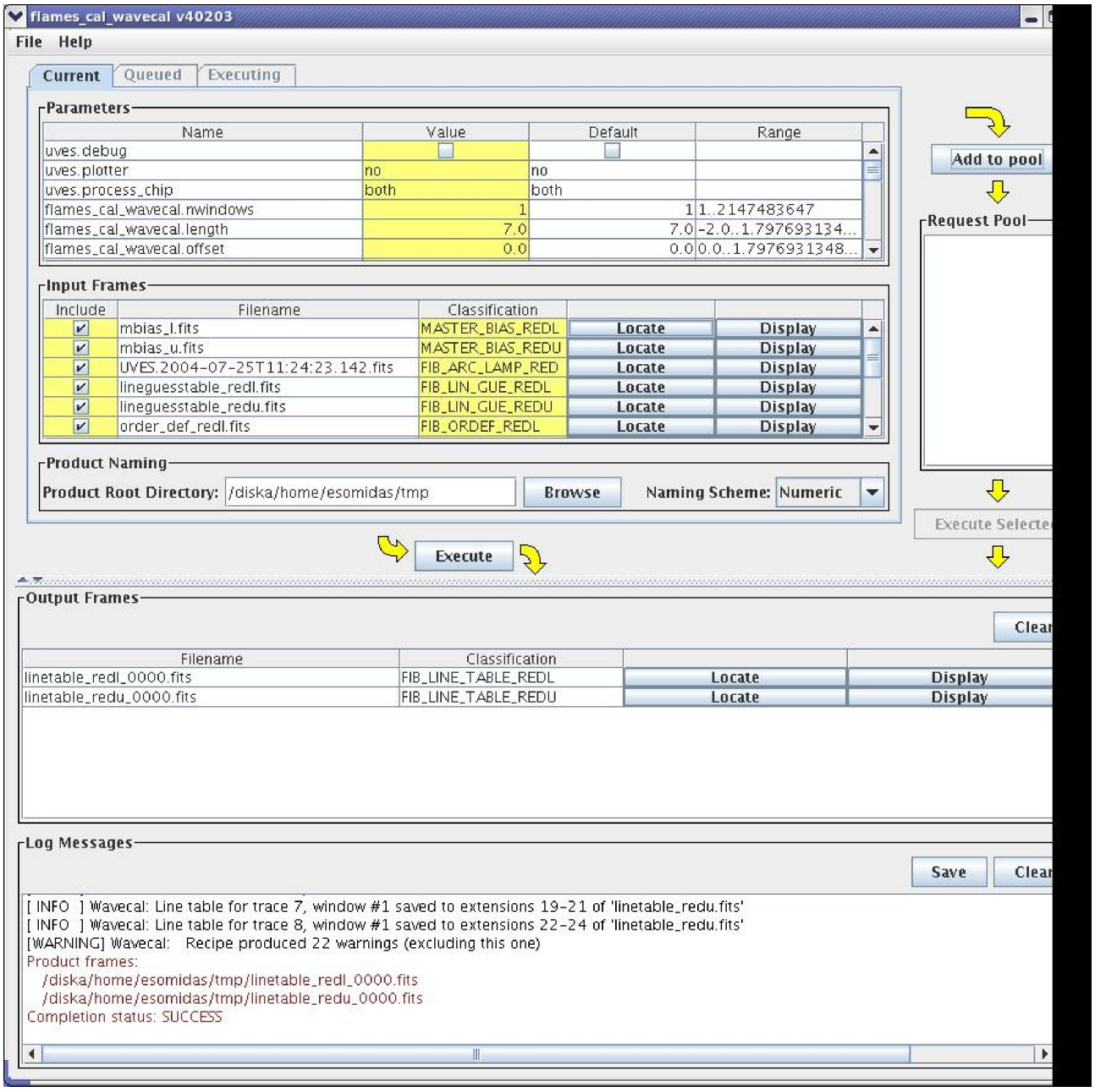


Figure 5.9: *The Gasgano recipe execution window.*

5.2.4 Using EsoRex

EsoRex is a command line utility for running pipeline recipes. It can be used in data reduction scripts for the automation of processing tasks. Unlike when using *Gasgano*, the user must classify and associate the data using the information contained in the FITS header keywords (see Section 7, page 44)[6]. The user should also take care of defining the input set-of-frames and the appropriate configuration parameters for each recipe run:

The set-of-frames: Each pipeline recipe is run on a set of input FITS data files. When using *EsoRex* the filenames must be listed together with their DO category[7] in an ASCII file, the *set-of-frames* (SOF), that is required when launching a recipe.[8] DO categories for the supported FLAMES-UVES input raw frames are indicated in section 7.8.

```
/path_raw/UVES.2004-08-14T10:20:56.497.fits FIB_ARC_LAMP_RED
/path_pro/orfl.fits                         FIB_ORDEF_TABLE_REDL
/path_pro/lineguess_redl.fits               FIB_LIN_GUE_REDL
/path_pro/orfu.fits                         FIB_ORDEF_TABLE_REDU
/path_pro/lineguess_redu.fits               FIB_LIN_GUE_REDU
/path_pro/thargood_3.fits                   LINE_REFER_TABLE
/path_pro/masterbias_redl.fits              MASTER_BIAS_REDL (optional)
/path_pro/masterbias_redu.fits              MASTER_BIAS_REDU (optional)
/path_pro/masterflat_redl.fits              MASTER_FLAT_REDL (optional-recommended)
/path_pro/masterflat_redu.fits              MASTER_FLAT_REDU (optional-recommended)
```

It contains for each input frame the full path file name and its DO category. The pipeline recipe will access the listed files when required by the reduction algorithm.

Note that the FLAMES-UVES pipeline recipes do not verify in any way the correctness of the classification tags specified by the user in the SOF. In the above example, the recipe *flames_cal_wavecal* will treat the frame `/path_raw/UVES.2004-08-14T10:20:56.497.fits` as an FIB_ARC_LAMP_RED, the frame `/path_pro/orfl.fits` as a FIB_ORDEF_TABLE_REDL, etc., even when they do not contain this type of data. The recipe will also assume that all frames are associated correctly, *i.e.*, that they all come from the same arm, dichroic and bin setting.

The reason for this lack of control is that the FLAMES-UVES recipes are just the DRS component of the complete pipeline running on Paranal, where the task of data classification and association is carried out by separate applications. Moreover, using *Gasgano* as an interface to the pipeline recipes will always ensure a correct classification of all the data frames, assigning the appropriate DO category to each one of them (see Section 5.2.3, page 25). Also this lack of control allows the user to reduce e.g. an arc lamp frame pretending it is a science frame.

A recipe handling an incorrect SOF may stop or display unclear error messages at best. In the worst cases, the recipe would apparently run without any problem, producing results that may look reasonable, but are actually flawed.

[6] Service Mode users could also use the association information contained in the Association Blocks (or ABs) included in their Service Mode Data Packages

[7] The indicated *DO category* is a label assigned to any data type after it has been classified, which is then used to identify the frames listed in the *set-of-frames*

[8] The set-of-frames corresponds to the *Input Frames* panel of the *Gasgano* recipe execution window (see Figure 5.9, page 29).

EsoRex syntax: The basic syntax to use EsoRex is the following:

esorex [esorex_options] recipe_name [recipe_options] set_of_frames

To get more information on how to customise EsoRex (see also [11]) run the command:

esorex - -help

To generate a configuration file esorex.rc in the directory $HOME/.esorex run the command:

esorex - -create-config

A list of all available recipes, each with a one-line description, can be obtained using the command:

esorex - -recipes

EsoRex searches for recipes in the directory specified by the option

esorex - -recipe-dir=installation_directory

All recipe parameters (aliases) and their default values can be displayed by the command

esorex - -params recipe_name

To get a brief description of each parameter meaning execute the command:

esorex - -help recipe_name

To get more details about the given recipe give the command at the shell prompt:

esorex - -man-page recipe_name

Recipe configuration: Each pipeline recipe may be assigned an *EsoRex* configuration file, containing the default values of the parameters related to that recipe.[9] The configuration files are normally generated in the directory `$HOME/.esorex`, and have the same name as the recipe to which they are related, with the filename extension `.rc`. For instance, the recipe *flames_cal_wavecal* has its *EsoRex* generated configuration file named `flames_cal_wavecal.rc`, and is generated with the command:

esorex - -create-config flames_cal_wavecal

The definition of one parameter of a recipe may look like this:

```
# --tolerance
# Tolerance of fit. If positive, 'tolerance' is in pixel units. If negative,
# abs('tolerance') is in wavelength units. Lines with residuals worse than
# the tolerance are excluded from the final fit. Unlike in previous versions,
# this parameter is not corrected for CCD binning.
flames_cal_wavecal.calibrate.tolerance=0.6
```

In this example, the parameter `flames_cal_wavecal.calibrate.tolerance` is set to the value `0.6`. In the configuration file generated by *EsoRex*, one or more comment lines are added containing information about the possible values of the parameter, and an alias that could be used as a command line option.

The command

esorex - -create-config recipe_name

[9] The *EsoRex* recipe configuration file corresponds to the *Parameters* panel of the *Gasgano* recipe execution window (see Figure 5.9, page 29).

generates a default configuration file **recipe_name.rc** in the directory **$HOME/.esorex**[10].

A recipe configuration file different from the default one can be specified on the command line:

esorex - -recipe-config=my_alternative_recipe_config

Recipe parameters are provided in section 10 and their role is described in Section 11.

More than one configuration file may be maintained for the same recipe but, in order to be used, a configuration file not located under `$HOME/.esorex`, or having a name different from the recipe name, should be explicitly specified when launching a recipe.

Recipe execution: A recipe can be run by giving its name to *EsoRex*, together with the name of a set-of-frames. For instance, the following command line would be used to run the recipe *flames_cal_wavecal* for processing the files specified in the set-of-frames `flames_cal_wavecal.sof`:

esorex flames_cal_wavecal flames_cal_wavecal.sof

The recipe parameters can be modified either by editing directly the used configuration file, or by specifying new parameter values on the command line using the command line options defined for this purpose. Such command line options should be inserted after the recipe name and before the SOF name, and they will supersede the system defaults and/or the configuration file settings. For instance, to set the *flames_cal_wavecal* recipe *tolerance* parameter to `0.07`, the following should be typed:

esorex flames_cal_wavecal - -tolerance=0.07 flames_cal_wavecal.sof

Every recipe provides a parameter *debug* which – when enabled – causes intermediate results to be saved to the local directory. This allows more detailed inspection of the recipe processing

esorex flames_cal_wavecal - -debug flames_cal_wavecal.sof

Basic plotting functionality can be enabled on systems which have the gnuplot tool:

esorex flames_cal_wavecal - -plotter='gnuplot -persist' flamses_cal_wavecal.sof

For more advanced visualisation a dedicated FITS viewer should be used with the FITS files produced by the recipe.

For more information on *EsoRex*, see `www.eso.org/cpl/esorex.html`.

5.3 Example of data reduction using EsoRex

Here we provide an example of data reduction for data obtained with the FLAMES-UVES.

In Table 5.3 we list the category Data Products (DPR) keywords for the currently supported data.

The plate No. is identified through the keyword ESO.INS.OBSPLATE. Plate No. 1 corresponds to a keyword value equal to 1. Plate No. 2 corresponds to a keyword value equal to 2.

The keyword ESO.OCS.SIMCAL will be used to associate the proper master calibration frame to the input raw frame separating the cases in which the simultaneous calibration fibre is involved (value equal to 1) or not (value equal to 0).

[10] If a number of recipe parameters are specified on the command line, the given values will be used in the created configuration file.

Frame Type	TYPE	CATG	TECH	Note
Science frames				
Science in OzPoz mode	OBJECT,OzPoz	SCIENCE	MOS	8 OzPoz fibres
Science in SimCal mode	OBJECT,SimCal	SCIENCE	MOS	7 OzPoz, 1 SimCal
Calibration fibre fed frames				
Bias	BIAS	CALIB	IMAGE	Same as UVES
Dark	DARK	CALIB	IMAGE	Same as UVES
Nasmyth Screen FF	LAMP,FLAT,NASMYTH	CALIB	MOS	Nasmyth Screen
Wavelength calibration	LAMP,WAVE,OzPoz	CALIB	MOS	8 OzPoz fed by ThAr
Wavelength calibration	LAMP,WAVE,SimCal	CALIB	MOS	7 OzPoz, 1 SimCal fed by ThAr
Odd FF	LAMP,FLAT,ODD,OzPoz	CALIB	MOS	Odd Fiber FF, 4 OzPoz
Even FF	LAMP,FLAT,EVEN,OzPoz	CALIB	MOS	Even Fiber FF, 4 OzPoz
All FF	LAMP,FLAT,ALL,OzPoz	CALIB	MOS	All Fiber FF, 8 OzPoz
Odd FF	LAMP,FLAT,ODD,SimCal	CALIB	MOS	Odd fibre FF, 4 OzPoz
Even FF	LAMP,FLAT,EVEN,SimCal	CALIB	MOS	Even Fiber FF, 4 OzPoz
All FF	LAMP,FLAT,ALL,SimCal	CALIB	MOS	All Fiber FF, 7 OzPoz
Formatcheck	LAMP,FMTCHK,SimCal	CALIB	MOS	1 SimCal fed by ThAr
Order Trace	LAMP,ORDERDEF,SimCal	CALIB	MOS	1 SimCal fed by Flat
Special frames in echelle mode				
Slit Flat Fields	LAMP,SFLAT	CALIB	ECHELLE	3 sets of 3 slit FF each each set taken at diff Y pos

Table 5.3: Required DPR keywords

When an accuracy better than 2% is required for the absolute fluxes, one should take flat fields using the Nasmyth screen and “attached templates”.

In Table 5.4 we list the DO_CLASSIFICATION values for the currently supported data.

In the following a typical step-by-step data reduction procedure is described [11]. Figure 5.10 gives an overview.

`/path_ref` indicates the full path to the source tree directory containing reference ancillary data, `/path_pro` indicates the full path to the source tree directory containing product data.

Bias frames: these frames are characterized by DPR.TYPE=’BIAS’

```
/path_raw/uves_bias_red1.fits  BIAS_RED
/path_raw/uves_bias_red2.fits  BIAS_RED
/path_raw/uves_bias_red3.fits  BIAS_RED
/path_raw/uves_bias_red4.fits  BIAS_RED
/path_raw/uves_bias_red5.fits  BIAS_RED
```

Dark frames: these frames are characterized by DPR.TYPE=’DARK’

```
/path_raw/uves_dark_red1.fits  DARK_RED
/path_raw/uves_dark_red2.fits  DARK_RED
/path_raw/uves_dark_red3.fits  DARK_RED
```

[11] The procedure using *Gasgano* is conceptually identical

Slit Flat-field lamp frames: these frames are characterized by DPR.TYPE='LAMP,SFLAT'

```
/path_raw/uves_flat_set1_red1.fits SFLAT_RED
/path_raw/uves_flat_set1_red2.fits SFLAT_RED
/path_raw/uves_flat_set1_red3.fits SFLAT_RED
/path_raw/uves_flat_set2_red1.fits SFLAT_RED
/path_raw/uves_flat_set2_red2.fits SFLAT_RED
/path_raw/uves_flat_set2_red3.fits SFLAT_RED
/path_raw/uves_flat_set3_red1.fits SFLAT_RED
/path_raw/uves_flat_set3_red2.fits SFLAT_RED
/path_raw/uves_flat_set3_red3.fits SFLAT_RED
```

Formatcheck: these frames are characterized by DPR.TYPE='LAMP,FMTCHK',

```
/path_raw/flames_uves_arc_lamp_form_red.fits FIB_ARC_LAMP_FORM_RED
```

Single fibre order tracing flat-field frames: these frames are characterized by DPR.TYPE='LAMP,ORDERDEF,SimCal'

```
/path_raw/flames_uves_order_flat_red.fits FIB_ORDER_FLAT_RED
```

Arc lamp frames to compute the wavelength calibration: these frames have DPR.TYPE respectively equal to 'LAMP,WAVE,OzPoz'.

```
/path_raw/flames_uves_arc_lamp_red.fits FIB_ARC_LAMP_RED
```

science frames: these frames are characterized by DPR.TYPE='OBJECT'.

```
/path_raw/flames_uves_science_red.fits  FIB_SCI_RED
```

In the following examples we assume that pipeline product filenames are the original as set by the pipeline. This corresponds to have the parameter **esorex.caller.suppress-prefix** in the EsoRex configuration file (`$HOME/.esorex/esorex.rc`) set to TRUE. Otherwise EsoRex will rename the pipeline producs using a common prefix (set by the parameter **esorex.caller.output-prefix**), a four digits increasing number, and terminating the FITS file with the extention ".fits". We suggest to verify to have the flag *readonly* set to FALSE, if the user would like to run the same recipe several times with EsoRex having standard values for product files. This setting allows the pipeline to overwrite previously generated products [12].

1. Generated master bias frames. One selects the raw biases and lists them in an ASCII file uves_cal_mbias.sof:

[12] By default installation in the EsoRex configuration file ($HOME/.esorex/esorex.rc) the flag *suppress-prefix* is set to FALSE and the flag *readonly* is set to FALSE, a possible combination, in which case pipeline product filenames will have a prefix out_, an increasing four digit number, and extention .fits.

```
/path_raw/uves_bias_red1.fits  BIAS_RED
/path_raw/uves_bias_red2.fits  BIAS_RED
/path_raw/uves_bias_red3.fits  BIAS_RED
/path_raw/uves_bias_red4.fits  BIAS_RED
/path_raw/uves_bias_red5.fits  BIAS_RED
```

The command:

esorex uves_cal_mbias uves_cal_mbias.sof

will generate the following products:

default recipe filename	format	PRO.CATG	short description
masterbias_redl	2d image (pix-pix)	MASTER_BIAS_REDL	master bias
masterbias_redu	2d image (pix-pix)	MASTER_BIAS_REDU	master bias

mv *.fits *.paf /path_pro

2. Then one selects the raw darks and lists them in an ASCII file uves_cal_mdark.sof.

```
/path_raw/uves_dark_red1.fits  DARK_RED
/path_raw/uves_dark_red2.fits  DARK_RED
/path_raw/uves_dark_red3.fits  DARK_RED
/path_pro/masterbias_redl.fits MASTER_BIAS_REDL (optional-recommended)
/path_pro/masterbias_redu.fits MASTER_BIAS_REDU (optional-recommended)
```

The command:

esorex uves_cal_mdark uves_cal_mdark.sof

will generate the following products:

default recipe filename	format	PRO.CATG	short description
masterdark_redl	2d image (pix-pix)	MASTER_DARK_REDL	master dark
masterdark_redu	2d image (pix-pix)	MASTER_DARK_REDU	master dark

mv *.fits *.paf /path_pro

3. Generate order and line guesses. Formatcheck frames are listed together with the required calibration frames in an ASCII file, flames_cal_predict.sof. This file will look like as follows:

```
/path_raw/flames_uves_arc_lamp_form_red.fits FIB_ARC_LAMP_FORM_RED
/path_ref/thargood_3.fits                    LINE_REFER_TABLE
/path_pro/uves_masterbias_redl.fits          MASTER_BIAS_REDL (optional)
/path_pro/uves_masterbias_redu.fits          MASTER_BIAS_REDU (optional)
```

Then the user can generate the order and line guesses with the command

esorex flames_cal_predict flames_cal_predict.sof

This command will generate four files (in the following table FITS files have extention .fits):

default recipe filename	format	PRO.CATG	short description
lineguesstable_redl	table	FIB_LIN_GUE_REDL	guess line table
orderguesstable_redl	table	FIB_ORD_GUE_REDL	guess order table
lineguesstable_redu	table	FIB_LIN_GUE_REDU	guess line table
orderguesstable_redu	table	FIB_ORD_GUE_REDU	guess order table

mv *.fits *.paf /path_pro

4. Generate order table and order definition frames. Order definition frames are necessary later in the data reduction to have a full set of flat frames that cover also the simultaneous calibration fibre as the odd-even fibre frames cover only the other fibres.

   A set of narrow slit raw flat-field frames may be put in the ASCII file flames_cal_orderpos.sof.

```
/path_raw/flames_uves_order_flat_red.fits FIB_ORDEF_RED
/path_pro/orderguesstable_redl.fits FIB_ORD_GUE_REDL
/path_pro/orderguesstable_redu.fits FIB_ORD_GUE_REDU
/path_pro/masterbias_redl.fits      MASTER_BIAS_REDL  (optional)
/path_pro/masterbias_redu.fits      MASTER_BIAS_REDU  (optional)
```

   The user can generate order tables and reoriented single fibre order flats with the command:

   esorex flames_cal_orderpos flames_cal_orderpos.sof

   This command will generate the following products:

default recipe filename	format	PRO.CATG	short description
ordertable_redl	table	FIB_ORD_TAB_REDL	simultaneous calibration fibre order table
order_def_redl	image	FIB_ORDEF_REDL	simultaneous calibration fibre order frame
ordertable_redu	table	FIB_ORD_TAB_REDU	simultaneous calibration fibre order table
order_def_redu	image	FIB_ORDEF_REDU	simultaneous calibration fibre order frame

mv *.fits *.paf /path_pro

5. Then one selects the raw slit flat-field frames and lists them in an ASCII file flames_cal_mflat.sof together with some master calibrations and previously obtained products:

```
/path_raw/flames_uves_flat_set1_red1.fits SFLAT_RED
/path_raw/flames_uves_flat_set1_red2.fits SFLAT_RED
/path_raw/flames_uves_flat_set1_red3.fits SFLAT_RED
/path_raw/flames_uves_flat_set2_red1.fits SFLAT_RED
/path_raw/flames_uves_flat_set2_red2.fits SFLAT_RED
/path_raw/flames_uves_flat_set2_red3.fits SFLAT_RED
/path_raw/flames_uves_flat_set3_red1.fits SFLAT_RED
/path_raw/flames_uves_flat_set3_red2.fits SFLAT_RED
/path_raw/flames_uves_flat_set3_red3.fits SFLAT_RED
/path_pro/ordertable_redl.fits            FIB_ORD_TAB_REDL
/path_pro/ordertable_redu.fits            FIB_ORD_TAB_REDU
/path_pro/masterbias_redl.fits            MASTER_BIAS_REDL (optional-recommended)
/path_pro/masterbias_redu.fits            MASTER_BIAS_REDU (optional-recommended)
```

The command:

esorex flames_cal_mflat flames_cal_mflat.sof

will generate the following products:

default recipe filename	format	PRO.CATG	Note
masterflat_set1_redl	2d image (pix-pix)	MASTER_SFLAT_REDL	master flat
masterflat_set2_redl	2d image (pix-pix)	MASTER_SFLAT_REDL	master flat
masterflat_set3_redl	2d image (pix-pix)	MASTER_SFLAT_REDL	master flat
masterflat_set1_redu	2d image (pix-pix)	MASTER_SFLAT_REDU	master flat
masterflat_set2_redu	2d image (pix-pix)	MASTER_SFLAT_REDU	master flat
masterflat_set3_redu	2d image (pix-pix)	MASTER_SFLAT_REDU	master flat

mv *.fits *.paf /path_pro

6. Then the order table for all fibre traces is determined. A set of three raw fibre frames illuminating respectively the odd, even and all fibre frames are put in an the ASCII file flames_cal_prep_sff_ofpos.sof.

```
/path_raw/flames_uves_odd_red.fits   FIB_FF_ODD_RED
/path_raw/flames_uves_even_red.fits  FIB_FF_EVEN_RED
/path_raw/flames_uves_all_red.fits   FIB_FF_ALL_RED
/path_pro/masterbias_redl.fits       MASTER_BIAS_REDL (optional)
/path_pro/order_def_redl.fits        FIB_ORDEF_REDL
/path_pro/ordertable_redl.fits       FIB_ORD_TAB_REDL
/path_pro/masterflat_set1_redl.fits  MASTER_SFLAT_REDL
/path_pro/masterflat_set2_redl.fits  MASTER_SFLAT_REDL
/path_pro/masterflat_set2_redl.fits  MASTER_SFLAT_REDL
/path_pro/masterbias_redu.fits       MASTER_BIAS_REDU (optional)
/path_pro/order_def_redu.fits        FIB_ORDEF_REDU
/path_pro/ordertable_redu.fits       FIB_ORD_TAB_REDU
/path_pro/masterflat_set1_redu.fits  MASTER_SFLAT_REDU
/path_pro/masterflat_set2_redu.fits  MASTER_SFLAT_REDU
/path_pro/masterflat_set2_redu.fits  MASTER_SFLAT_REDU
```

Note that the input master bias frames are optional but recommended input.

The command

esorex flames_cal_prep_sff_ofpos flames_cal_prep_sff_ofpos.sof

will generate the following products:

default recipe filename	format	PRO.CATG	short description
xt_odd_1	table	FIB_FF_ODD_INFO_TAB	info table
xt_even_1	table	FIB_FF_EVEN_INFO_TAB	info table
xt_all_1	table	FIB_FF_ODD_INFO_TAB	info table
slitff_common_redl	imagelist	SLIT_FF_COM_REDL	slitff common frame

slitff_norm_redl	image	SLIT_FF_NOR_REDL	slitff common frame
slitff_dtc_redl	imagelist	SLIT_FF_DTC_REDL	slitff data frame
slitff_sgc_redl	imagelist	SLIT_FF_SGC_REDL	slitff sigma frame
slitff_bpc_redl	imagelist	SLIT_FF_BPC_REDL	slitff badpixel frame
slitff_bnc_redl	imagelist	SLIT_FF_BNC_REDL	slitff frames boundaries
orfl	table	FIB_ORDEF_TABLE_REDL	fibre order table
fibreff_common_redl	imagelist	FIB_FF_COM_REDL	fibff common frame
fibreff_norm_redl	imagelist	FIB_FF_NOR_REDL	fibff common frame
fibreff_nsigma_redl	imagelist	FIB_FF_NSG_REDL	Normalisation sigmas for fibre FF
fibreff_dtc_redl	imagelist	FIB_FF_DTC_REDL	fibff data frame
fibreff_sgc_redl	imagelist	FIB_FF_SGC_REDL	fibff sigma frame
fibreff_bpc_redl	imagelist	FIB_FF_BPC_REDL	fibff badpixel frame
slitff_common_redu	imagelist	SLIT_FF_COM_REDU	slitff common frame
slitff_norm_redu	image	SLIT_FF_NOR_REDU	slitff common frame
slitff_dtc_redu	imagelist	SLIT_FF_DTC_REDU	slitff data frame
slitff_sgc_redu	imagelist	SLIT_FF_SGC_REDU	slitff sigma frame
slitff_bpc_redu	imagelist	SLIT_FF_BPC_REDU	slitff badpixel frame
slitff_bnc_redu	imagelist	SLIT_FF_BNC_REDU	slitff frames boundaries
ordu	table	FIB_ORDEF_TABLE_REDU	fibre order table
fibreff_common_redu	imagelist	FIB_FF_COM_REDU	fibff common frame
fibreff_norm_redu	imagelist	FIB_FF_NOR_REDU	fibff common frame
fibreff_nsigma_redu	imagelist	FIB_FF_NSG_REDU	Normalisation sigmas for fibre FF
fibreff_dtc_redu	imagelist	FIB_FF_DTC_REDU	fibff data frame
fibreff_sgc_redu	imagelist	FIB_FF_SGC_REDU	fibff sigma frame
fibreff_bpc_redu	imagelist	FIB_FF_BPC_REDU	fibff badpixel frame

mv *.fits *.paf /path_pro

7. Then the wavelength calibration is performed. A raw frame illuminated by the long slit through the fibres by an arc lamp are put in the ASCII file flames_cal_wavecal.sof.

```
/path_raw/flames_uves_arc_lamp_red.fits   FIB_ARC_LAMP_RED
/path_pro/orfl.fits                       FIB_ORDEF_TABLE_REDL
/path_pro/lineguesstable_redl.fits        FIB_LIN_GUE_REDL
/path_pro/masterbias_redl.fits            MASTER_BIAS_REDL  (optional)
/path_pro/orfu.fits                       FIB_ORDEF_TABLE_REDU
/path_pro/lineguesstable_redu.fits        FIB_LIN_GUE_REDU
/path_pro/masterbias_redu.fits            MASTER_BIAS_REDU  (optional)
/path_ref/thargood_3.fits                 LINE_REFER_TABLE
```

Note that the input master bias and master flat frames are optional but recommended inputs.

The command

esorex flames_cal_wavecal flames_cal_wavecal.sof

will generate the following products:

default recipe filename	format	PRO.CATG	short description
linetable_redl	table	FIB_LINE_TABLE_REDL	line table
linetable_redu	table	FIB_LINE_TABLE_REDU	line table

This table contains the solutions for each extracted fibres in several extensions according to the following schema:

```
Line table for fibre 1, saved to extensions 1-3 of 'linetable_redl/redu'
Line table for fibre 2, saved to extensions 4-6 of 'linetable_redl/redu'
Line table for fibre 3, saved to extensions 7-9 of 'linetable_redl/redu'
Line table for fibre 4, saved to extensions 10-12 of 'linetable_redl/redu'
Line table for fibre 5, saved to extensions 13-15 of 'linetable_redl/redu'
Line table for fibre 6, saved to extensions 16-18 of 'linetable_redl/redu'
Line table for fibre 7, saved to extensions 19-21 of 'linetable_redl/redu'
Line table for fibre 8, saved to extensions 22-24 of 'linetable_redl/redu'
Line table for fibre 9, saved to extensions 25-27 of 'linetable_redl/redu'
```

mv *.fits *.paf /path_pro

8. Finally, the raw science frame is reduced.The raw science frame is listed together with master calibration products in the following ASCII file flames_obs_scired.sof:

```
/path_raw/flames_uves_science_red.fits FIB_SCI_RED
/path_pro/masterbias_redl.fits         MASTER_BIAS_REDL (optional)
/path_pro/slitff_com_redl.fits         SLIT_FF_COM_REDL
/path_pro/slitff_nor_redl.fits         SLIT_FF_NOR_REDL
/path_pro/slitff_dtc_redl.fits         SLIT_FF_DTC_REDL
/path_pro/slitff_bpc_redl.fits         SLIT_FF_BPC_REDL
/path_pro/slitff_sgc_redl.fits         SLIT_FF_SGC_REDL
/path_pro/fibff_com_redl.fits          FIB_FF_COM_REDL
/path_pro/fibff_nor_redl.fits          FIB_FF_NOR_REDL
/path_pro/fibff_dtc_redl.fits          FIB_FF_DTC_REDL
/path_pro/fibff_bpc_redl.fits          FIB_FF_BPC_REDL
/path_pro/fibff_sgc_redl.fits          FIB_FF_SGC_REDL
/path_pro/fibff_bnc_redl.fits          FIB_FF_BNC_REDL
/path_pro/orfl.fits                    FIB_ORDEF_TABLE_REDL
/path_pro/linetable_redl.fits          FIB_LINE_TABLE_REDL
/path_pro/masterbias_redu.fits         MASTER_BIAS_REDU (optional)
/path_pro/slitff_com_redu.fits         SLIT_FF_COM_REDU
/path_pro/slitff_nor_redu.fits         SLIT_FF_NOR_REDU
/path_pro/slitff_dtc_redu.fits         SLIT_FF_DTC_REDU
/path_pro/slitff_bpc_redu.fits         SLIT_FF_BPC_REDU
/path_pro/slitff_sgc_redu.fits         SLIT_FF_SGC_REDU
/path_pro/fibff_com_redu.fits          FIB_FF_COM_REDU
/path_pro/fibff_nor_redu.fits          FIB_FF_NOR_REDU
/path_pro/fibff_dtc_redu.fits          FIB_FF_DTC_REDU
/path_pro/fibff_bpc_redu.fits          FIB_FF_BPC_REDU
```

```
/path_pro/fibff_sgc_redu.fits          FIB_FF_SGC_REDU
/path_pro/fibff_bnc_redu.fits          FIB_FF_BNC_REDU
/path_pro/orfu.fits                    FIB_ORDEF_TABLE_REDU
/path_pro/linetable_redu.fits          FIB_LINE_TABLE_REDU
```

The command:

esorex flames_obs_scired flames_obs_scired.sof

will generate the following products:

default recipe file name	format	PRO.CATG	short description
bin_table_info_redl	2d table	FIB_SCI_INFO_TAB_REDL	table with fibre information
fxb_l_raw000i	2d (pix-ord) image	XB_SCI_RAW_REDL	extracted, flatfielded raw frame
fxb_l_rawsig000i	2d (pix-ord) image	ERR_XB_SCI_RAW_REDL	error frame
wfxb_l_raw000i	2d (wav-ord) image	WXB_SCI_RAW_REDL	rebinned, extracted, flatfielded raw frame
wfxb_l_rawsig000i	2d (wav-ord) image	ERR_WXB_SCI_RAW_REDL	error frame
mwfxb_l_rawsig000i	1d (wav) image	MWXB_SCI_RAW_REDL	merged, rebinned, extracted, flat-fielded raw frame
mwfxb_l_raw000i_sigma	1d (wav) image	ERR_MWXB_SCI_RAW_REDL	merged, rebinned, extracted, flat-fielded error frame
fxb_l_000i	2d (pix-ord) image	XB_SCI_REDL	extracted, flatfielded raw frame
fxb_l_sig000i	2d (pix-ord) image	ERR_XB_SCI_REDL	error frame
wfxb_l_000i	2d (wav-ord) image	WXB_SCI_REDL	rebinned, extracted, flatfielded raw frame
wfxb_l_sig000i	2d (wav-ord) image	ERR_WXB_SCI_REDL	error frame
mwfxb_l_000i	1d (wav) image	MWXB_SCI_REDL	merged, rebinned, extracted, flat-fielded raw frame
mwfxb_l_000i_sigma	1d (wav) image	ERR_MWXB_SCI_REDL	merged, rebinned, extracted, flat-fielded raw frame
bin_table_info_redu	2d table	FIB_SCI_INFO_TAB_REDU	table with fibre information
fxb_u_raw000i	2d (pix-ord) image	XB_SCI_RAW_REDU	extracted, flatfielded raw frame
fxb_u_rawsig000i	2d (pix-ord) image	ERR_XB_SCI_RAW_REDU	error frame
wfxb_u_raw000i	2d (wav-ord) image	WXB_SCI_RAW_REDU	rebinned, extracted, flatfielded raw frame
wfxb_u_rawsig000i	2d (wav-ord) image	ERR_WXB_SCI_RAW_REDU	error frame
mwfxb_u_rawsig000i	1d (wav) image	MWXB_SCI_RAW_REDU	merged, rebinned, extracted, flat-fielded raw frame
mwfxb_u_raw000i_sigma	1d (wav) image	ERR_MWXB_SCI_RAW_REDU	merged, rebinned, extracted, flat-fielded error frame
fxb_u_000i	2d (pix-ord) image	XB_SCI_REDU	extracted, flatfielded raw frame
fxb_u_sig000i	2d (pix-ord) image	ERR_XB_SCI_REDU	error frame
wfxb_u_000i	2d (wav-ord) image	WXB_SCI_REDU	rebinned, extracted, flatfielded raw frame
wfxb_u_sig000i	2d (wav-ord) image	ERR_WXB_SCI_REDU	error frame
mwfxb_u_sig000i	1d (wav) image	MWXB_SCI_REDU	merged, rebinned, extracted, flat-fielded raw frame
mwfxb_u_000i_sigma	1d (wav) image	ERR_MWXB_SCI_REDU	merged, rebinned, extracted, flat-fielded error frame

Frame Type	DO_CLASSIFICATION
Raw Frames	
ThAr line reference table	LINE_REFER_TABLE
Formatcheck frame	FMTCHK_<CHIPID>
Reference Formatcheck frame	ARC_LAMP_FORM_<CHIPID>
Single Order trace frame	FIB_ORDEF_<CHIPID>
Bias	BIAS_<CHIPID>
Master Bias	MASTER_<CHIPID>
Dark	DARK_<CHIPID>
Slit Flat Field	SFLAT_<CHIPID>
Nasmyth Screen Flat Field	FIB_FF_NASMYTH_<CHIPID>
Wavelength Calibration frame	FIB_ARC_LAMP_<CHIPID>
Odd Flat Field	FIB_FF_ODD_<CHIPID>
Even Flat Field	FIB_FF_EVEN_<CHIPID>
All Flat Field	FIB_FF_ALL_<CHIPID>
Science OzPoz	FIB_SCI_<CHIPID>
Science SimCal	FIB_SCI_SIM_<CHIPID>
Science COMBINED	FIB_SCI_COM_<CHIPID>
Data Products	
MASTER Slit Flat Field	MASTER_SFLAT_<CHIPID>
DRS Setup Table	FIB_DRS_<CHIPID>
Guess Order Table	FIB_ORD_GUE_<CHIPID>
Guess Line Table	FIB_LIN_GUE_<CHIPID>
Background Table	FIB_BKG_<CHIPID>
Order-Fibre Table	FIB_ORDEF_TABLE_<CHIPID>
Line-Fibre Table	FIB_LINE_TABLE_<CHIPID>
Fibre data* frames	FIB_FF_D0n_<CHIPID>
Fibre common frames	FIB_FF_COM_<CHIPID>
Fibre norm frames	FIB_FF_NOR_<CHIPID>
Fibre nsigma frames	FIB_FF_NSIGMA_<CHIPID>
Fibre sigma frames	FIB_FF_S0n_<CHIPID>
Fibre badpixel frames	FIB_FF_BPn_<CHIPID>
Slit FF data frames	SLIT_FF_D0n_<CHIPID>
Slit FF common frames	SLIT_FF_COM_<CHIPID>
Slit FF norm frames	SLIT_FF_NOR_<CHIPID>
Slit FF sigma frames	SLIT_FF_S0n_<CHIPID>
Slit FF badpixel frames	SLIT_FF_BPn_<CHIPID>
Slit FF bound frames	SLIT_FF_BNDn_<CHIPID>

Table 5.4: List of frame types and corresponding tag values (see Section 1.4 for a description of the _<CHIPID> suffix). *Note: n=1,2,..nflats where in case of slit flat fields NFLATS is a FITS keyword indicating the (automatically determined) minimum number of equalized flat fields which are necessary to cover the total Y span, while in case of the fibre flat fields it indicates the number of basic (uncontaminated) fibre flat fields, typically (but not necessarily) 2 (odd/even).

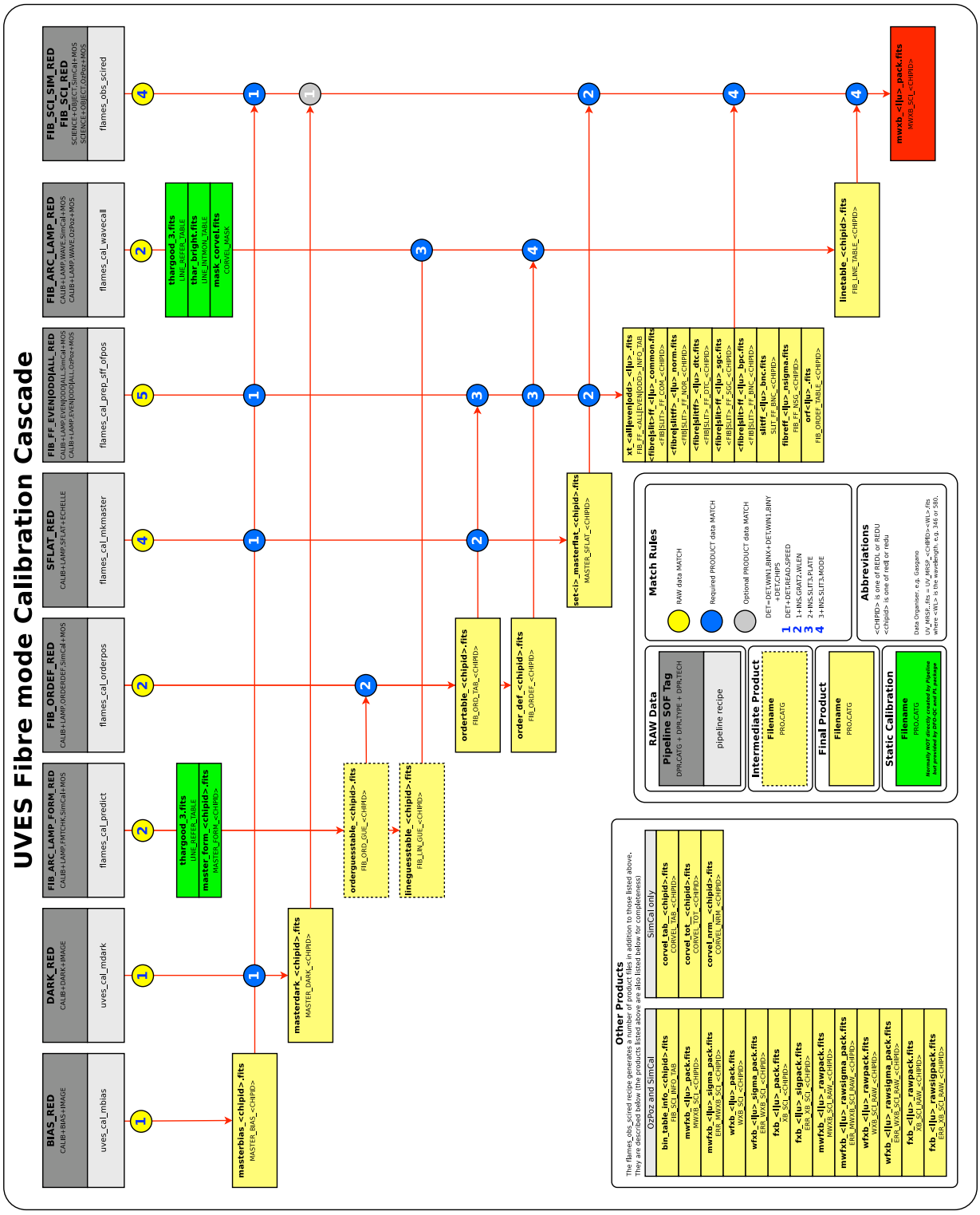


Figure 5.10: *The FLAMES-UVES calibration cascade.*

6 Known problems

We suggest the user to execute the data reduction recipes using parameter defaults and all the reference and master calibrations indicated in this manual. The following is a list of currently-known problems with FLAMES-UVES recipes, and workarounds, if available:

- The radial velocity correlation has not yet been verified to give the same results as the ones provided by FLAMES-UVES MIDAS based pipeline. In particular the zero points, are not yet computed during wavelength calibration. The user interested on those results is invited to run the MIDAS based pipeline release.
- Gnuplots do not work with Gasgano.
- From time to time the reduction of science frames may fail indicating that "Keyword ESO TEL GEOLAT does not exist". This keyword is required to measure heliocentric velocity correction. The user should make sure the input data have proper FITS header, in particular contains the keywords "RA", "DEC", "ESO TEL GEOLAT", "ESO TEL GEOLAT","UTC", "MJD-OBS".
- From time to time the reduction of science frames obtained with the 860 instrument setting may fail with error: "Input data do not match: Spectrum contains 12 orders, but line table absolute order numbering is 70 - 60" This problem is due to the fact that the wavelength calibration has switched off the generation of a solution for some fibre beacuse has not found the minimum number of searched liunes (minimum set to 1000). In this case we recommend the user to set the parameter **minlines** of flames_cal_wavecal to 600 (or less), and repeat the flames_cal_wavecal and flames_obs_scired steps.
- We are aware that the science reduction on platforms where the compiler is clang (for example MacOSX) may be not good. We recommend the user to install and compile the kit using the GNU compiler (gcc).
- Seldomly the recipe `flames_cal_prep_ofpos` fails reporting in the log errors like:

```
[ INFO  ] OFPos:           doptimal: Extracting orders from nr. 8 to nr. 8...
[ INFO  ] OFPos:           doptimal: Extracting orders from nr. 9 to nr. 12..
[ ERROR ] OFPos:         GAUSSJ: Singular Matrix-2
[ ERROR ] OFPos:         GAUSSJ: Singular Matrix-2
[ ERROR ] OFPos:         GAUSSJ: Singular Matrix-2
```

  We have not found yet the exact source and corresponding solution to this numerical problem. If the user finds it we recommend to use calibration data from the day before or near or close-by.

7 Instrument Data Description

To reduce a science frame, the following calibration frames are needed:

- Bias(es)
- Dark(s)
- Fibre formatcheck
- Single Fibre Order definition
- Slit Flat-field(s)
- Fibre flat-field frames having illuminated: odd, even, all fibres
- Fibre wavelength calibration

It is also necessary to have handy a reference (ThAr) line table. The table thargood_3.tfits is included in the pipeline installation kit. The table thargood_$< n >$.tfits (where n is 2 or 3) was also included in Pre P82 Service mode packages. The file UV_GLRE_<YYMMDD>A_line_refer_table.fits is included in Service mode packages for P82 onward.

7.1 Bias frames

Bias frames give the read out of the CCD detector of zero integration time with the shutter closed. Usually they are taken as a set of five exposures from which, through stacking, a Master Bias is created thus reducing the read out noise. This needs to be subtracted for example from the science frame to get the signal contribution from the source only.

7.2 Dark frames

Dark frames are measured occasionally, with the shutter closed. They are used to measure the dark current. They are measured for 1x1 binning with typical exposure times of 1h. There are also open-shutter DARKs (since December 2001). They include, in addition to the CCD dark current, contributions from the camera enclosure. Typical values are reported on the ESO Website under *https://www.eso.org/observing/dfo/quality/UVES/qc/pdark_qc1.html*. As the contribution of UVES dark exposures may be considered, in first approximation, negligible, they can be excluded from the data reduction chain as we will assume here in the following section.

7.3 Slit flat-field frames

Slit flat-field frames are long slit exposures taken with a continuum lamp. They give information on the response of the detector, allowing to measure variations in efficiency at small (pixel-to-pixel), intermediate (fringing, in the far red) and large (the blaze function) scale. Thus they are usually taken as three sets of three frames each at a

given cross disperser setting, which after bias subtraction are stacked in a master to reject statistical outliers like cosmic ray events. The final Master flat-field is also background subtracted to eliminate diffused light from the orders in the inter-order regions. Science frames need to be corrected for pixel-to-pixel variations, interference fringes and the blaze function through division by the master flat-field frame.

To properly evaluate the background level and avoid overlapping of orders, each slit flat-field frame has to be taken with a limited slit aperture. To cover all the fibres (allowing also for some variability in their placement) several slit flat-field frames are thus necessary, taken with different offsets with respect to a central position and overlapping between each other.

7.4 Formatcheck frames

It is necessary to have a special formatcheck frame taken by illuminating the simultaneous calibration fibre with a ThAr lamp. This is used in combination with a physical model of UVES and the information contained in the FITS header and in a ThAr reference line table, and the known offset of the simultaneous calibration fibre from the UVES slit's center, to find a "guess" solution of the spectral format (order locations and wavelength calibration). This allows the user later on to obtain robust and automatic spectral format solutions.

7.5 Single fibre order definition frames

A single fibre order definition frame is a calibration exposure obtained with the calibration fibre illuminated by a continuum lamp. It is a very high signal-to-noise ratio fibre-echelle frame describing accurately the simultaneous calibration fibre order locations.

7.6 Fibre Flat-Field (odd-even-all) frames

The fibre flat-fields serve to locate the fibre positions, both relative to one another and absolute, within some tolerance; to know the cross-sectional profile of each fibre at each wavelength; and to determine the relative throughput of different fibres. These fibre flat-fields are only usable if the fibre traces fall in regions illuminated in at least one of the slit flat-fields; fibres whose light falls, even in part, in dark regions of the slit flat-fields are discarded. The data reduction software considers as really flat only a subregion of each slit flat-field frame discarding a few pixels at the flat-field border.

The fibres' images on the detector are so closely packed that the cross-sectional profiles of adjacent fibres overlap to some non-negligible extent. Therefore, a single flat-field frame, would not yield enough information on individual fibres to perform a correct data reduction. Instead, the pipeline requires fibre flat-fields on which the fibres can be clearly separated, such as one containing only odd-numbered fibres and another containing only even-numbered fibres.

The order definition frame, if taken with the simultaneous fibre, could also be used as an additional flat-field frame covering the simultaneous calibration fibre. Such inclusion is necessary to be able to extract the simultaneous calibration fibre. In such a case one would add it into the pool of odd-even fibre flat fields (see below), and this set of frames would thus contain completely separable images of all fibres.

In addition to the above, an all-fibre flat-field is needed to correct for relative throughput differences between odd and even fibres. This can be replaced by a Nasmyth screen all-fibre flat-field for higher accuracy.

If it is planned to observe using the simultaneous calibration fibre, it is also necessary to have in the calibration data a ThAr frame with the ThAr lamp feeding all the same fibres used during the night (7OzPoz+1SimCal) as well as an all fibre flat-field with the flat-field lamp feeding all the same fibres used during the night (7OzPoz+1SimCal). This is the minimum fibre flat-field set usable by the pipeline.

7.7 Wavelength calibration frames

Wavelength calibration frames are fibre exposures taken (for FLAMES-UVES) illuminating all fibres with a ThAr arc lamp. They are used to find the wavelength calibration solution.

The wavelength calibration frames must be taken through the same set of fibres as the science frame.

We would like to emphasize the importance of having "correct" calibration data to be able to reduce the science observations with best quality. For example, if because of thermal shifts the available slit flat-fields do not cover the position where a (science-frame) fibre is located, this fibre will be discarded and the pipeline will be unable to extract its signal. The same is true if fibre positions in the science-frame are shifted too much with respect to the corresponding positions of the fibre flat-field frames used to do the data reduction. This may occur in case of automatic online pipeline data reduction on Paranal where usually the calibration data base is updated with several months time scale. Moreover, the cross-sectional profiles of the fibres may change e. g. due to slightly different focusing. Since this, in turn, changes the contamination between adjacent fibres, a significantly different profile between calibrations and science data will lead to an incorrect deconvolution of the fibres. The pipeline searches for possible instrument shifts along Y (i. e. perpendicular to dispersion) in the interval [-6,6] pixel. It satisfactorily reduced commissioning data which, due a not completely stable instrument setup at the time, were characterized by shifts between the science observation and the calibration in the range [-3.5,3.5]. It must be said that the source of such Y shift was discovered and fixed during an instrument software upgrade conducted before the beginning of FLAMES operations and so we do not expect the user will receive data affected by shifts between science and calibrations of more than one pixel. The pipeline allows to interactively recover extraction also in cases of shifts wider than 6 pixels. This can be done by executing the science recipe once with parameter defaults; then by plotting the information available in the product cor_table_chip.fits (chip='l' or 'u') the user can asses the proper value of the correlation shift, and set correspondly the correlation offset in order to reduce, within a narrower correlation range centered on the given offset, the given science data in a following data reduction iteration.

The FLAMES positioner uses two plates and two distinct sets of fibres, with two distinct entrance positions in UVES.

A simultaneous calibration fibre for precise radial velocity measurements is also available. Using this fibre the spectral format changes, including (at most) 7 OzPoz fibres and 1 SimCal fibre. Therefore, only calibration data need to be taken with the same FLAMES plate and ($\leq$ 8 OzPoz fibres)/($\leq$ 7 OzPoz fibres+1 SimCal fibre) configuration as the actual science frames to be reduced should be used in the reduction.

Although we introduced means to detect and appropriately treat saturated pixels in the data reduction, it might happen that a saturated frame affects the data reduction. More specifically, while isolated saturated pixels can be easily detected by the pipeline and excluded from the subsequent data reduction, strongly saturated emission lines can produce a very irregular distribution of scattered light on the frame, causing a poor background subtrac-

tion and, as a consequence, wrong results in the determination of shifts to be applied to each fibre (correlation step). In particularly pathological cases, this may lead to a failure of the automatic data reduction procedure, and require manual intervention to properly extract the data.

7.8 Supported raw frames (keyword identifiers)

In this section we describe all possible types of raw frames for the different observing modes. More information on those data may be found on *https://www.eso.org/observing/dfo/quality/UVES/reports/RAWDISP/REFERENCE/rawfiles_START_ALL.html*. The different frame types can be identified by the values of the DPR keywords of their FITS headers (see [8]). These keywords are generated by the FLAMES-UVES templates (for a description of the FLAMES-UVES templates see [6]). A given frame type can be processed by one or several different dedicated pipeline recipes. The individual pipeline recipes are described in section 11. In most cases, calibration data frames are needed to reduce a given frame. These calibration data have to match the input frame in a number of instrument parameters (e.g. to apply a flat-field correction to a science frame only a flat-field frame taken in the same central wavelength, same slit length, etc. should be used for the correction). These parameters are listed under *relevant instrument parameters*.

The following raw frame types are possible:

- Relevant instrument parameters group 1 common to all raw frames:

  Conversion e→ADU: `ESO DET OUT1 CONAD`

- Relevant instrument parameters group 2 common to some raw frames:

  Grating used: `ESO INS GRAT2 ID`
  Central wavelength used: `ESO INS GRAT2 WLEN`
  Observation Plate used: `INS OBSPLATE`
  Simultaneous calibration: `OCS SIMCAL`

Bias frames

- Template signature:
  `UVES_x_cal_bias`
  `(x: red, dic1, dic2)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = BIAS ESO DPR TECH = IMAGE`
- DO category:
  `BIAS_RED`
- Pipeline recipe: `uves_cal_mbias`

- Relevant instrument parameters: group 1.
- Reference: Figure 7.1 (a).

Dark frames

- Template signature:
  `UVES_x_cal_dark`
  `(x: red, dic1, dic2)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = DARK`
  `ESO DPR TECH = IMAGE`
- DO category:
  `DARK_RED`
- Pipeline recipe: `uves_cal_mdark`
- Relevant instrument parameters: group 1.

Order definition flat-field frames

- Template signature:
  `UVES_x_tec_orderdef`
  `(x: red, dic1, dic2)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,ORDERDEF,SimCal`
  `ESO DPR TECH = MOS`
- DO category:
  `ORDER_FLAT_RED`
- Pipeline recipe: `flames_cal_orderpos`
- Relevant instrument parameters groups 1, 2.
- Reference: Figure 7.1 (c).

Spectroscopic Slit flat-field frames

- Template signature:
  `UVES_x_cal_y`
  `(x: red, dic1, dic2)`
  `(y: flatatt, flatfree)`

- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,SFLAT`
  `ESO DPR TECH = ECHELLE`
- DO category:
  `FLAT_RED`
- Pipeline recipe: `flames_cal_mflat`
- Relevant instrument parameters groups 1, 2.
- Reference: Figure 7.1 (e).

Format check spectra

- Template signature:
  `UVES_x_tec_fmtchk`
  `(x: red, dic1, dic2)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,FMTCHK,SimCal`
  `ESO DPR TECH = MOS`
- DO category:
  `FIB_ARC_LAMP_FORM_RED`
- Pipeline recipe: `flames_cal_predict`
- Relevant instrument parameters: groups 1, 2.
- Reference: Figure 7.1 (b).

Odd Fibre flat calibration spectra

- Template signature:
  `UVES_x_cal_y`
  `(x: red, dic1, dic2)`
  `(y: waveatt, wavefree)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,FLAT,ODD`
  `ESO DPR TECH = MOS`
- DO category:
  `FIB_FF_ODD_RED`

- Pipeline recipe: `flames_cal_prep_sff_ofpos`
- Relevant instrument parameters: groups 1, 2.
- Reference: Figure 7.1 (d).

Even Fibre flat calibration spectra

- Template signature:
  ```
  UVES_x_cal_y
  (x:  red, dic1, dic2)
  (y:  waveatt, wavefree)
  ```
- DPR keywords:
  ```
  ESO DPR CATG = CALIB
  ESO DPR TYPE = LAMP,FLAT,EVEN
  ESO DPR TECH = MOS
  ```
- DO category:
  `FIB_FF_EVEN_RED`
- Pipeline recipe: `flames_cal_prep_sff_ofpos`
- Relevant instrument parameters: groups 1, 2.
- Reference: Figure 7.1 (d).

All Fibre flat calibration spectra

- Template signature:
  ```
  UVES_x_cal_y
  (x:  red, dic1, dic2)
  (y:  waveatt, wavefree)
  ```
- DPR keywords:
  ```
  ESO DPR CATG = CALIB
  ESO DPR TYPE = LAMP,FLAT,ALL
  ESO DPR TECH = MOS
  ```
- DO category:
  `FIB_FF_ALL_RED`
- Pipeline recipe: `flames_cal_prep_sff_ofpos`
- Relevant instrument parameters: groups 1, 2.
- Reference: Figure 7.1 (d).

OzPoz-fibre Wavelength calibration spectra

- Template signature:
  `UVES_x_cal_y`
  `(x: red, dic1, dic2)`
  `(y: waveatt, wavefree)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,WAVE,OzPoz`
  `ESO DPR TECH = MOS`
- DO category:
  `FIB_ARC_LAMP_RED`
- Pipeline recipe: `flames_cal_wavecal`
- Relevant instrument parameters: groups 1, 2.
- Reference: Figure 7.1 (d).

Simcal-fibre Wavelength calibration spectra

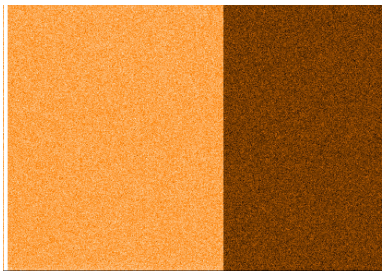
- Template signature:
  `UVES_x_cal_y`
  `(x: red, dic1, dic2)`
  `(y: waveatt, wavefree)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,WAVE,SimCal`
  `ESO DPR TECH = MOS`
- DO category:
  `FIB_ARC_LAMP_RED`
- Pipeline recipe: `flames_cal_wavecal`
- Relevant instrument parameters: groups 1, 2.
- Reference: Figure 7.1 (d).

Nasmyth screen Fibre flat calibration spectra

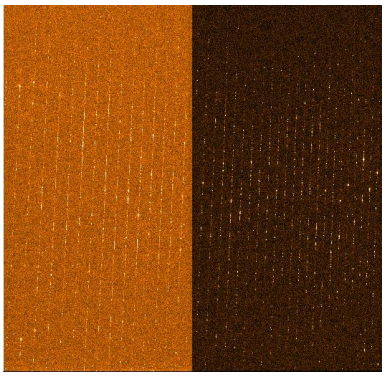
- Template signature:
  `UVES_x_cal_y`
  `(x: red, dic1, dic2)`
  `(y: waveatt, wavefree)`
- DPR keywords:
  `ESO DPR CATG = CALIB`
  `ESO DPR TYPE = LAMP,FLAT,NASMYTH`
  `ESO DPR TECH = MOS`
- DO category:
  `FIB_FF_NASMYTH_RED`
- Pipeline recipe: `flames_cal_prep_sff_ofpos`
- Relevant instrument parameters: groups 1, 2.
- Reference: Figure 7.1 (d).

Science spectra

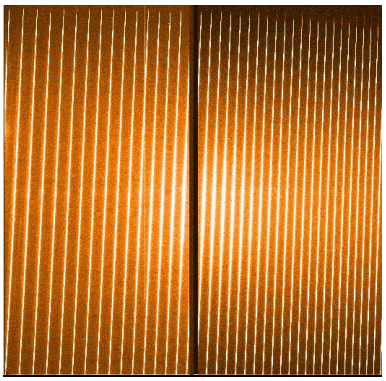
- Template signatures:
  `UVES_x_obs_y`
  `(x: blue, red, dic1, dic2)`
  `(y: exp, expfree)`
- DPR keywords:
  `ESO DPR CATG = SCIENCE`
  `ESO DPR TECH = MOS`
- DO category:
  `FIB_SCI_RED (ESO DPR TYPE = OBJECT,OzPoz)`
  `FIB_SCI_SIM_RED (ESO DPR TYPE = OBJECT,SimCal )`
  `FIB_SCI_COM_RED (ESO DPR TYPE = OBJECT,OzPoz )`
- Pipeline recipe: `uves_obs_scired`
- Relevant instrument parameters: groups 1, 2.



(a raw bias frame)



(a raw formatcheck simcal fibre frame)



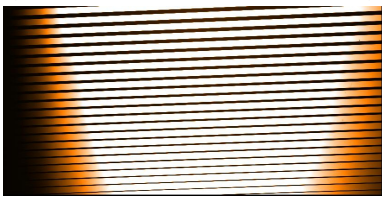
(a raw order simcal fibre frame)



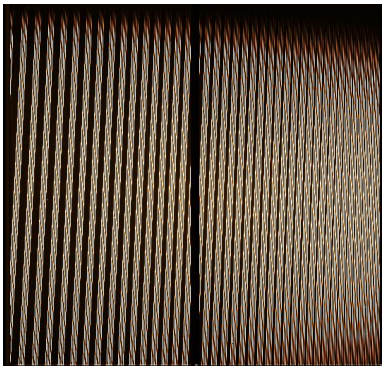
(a master sflat frame (set1))



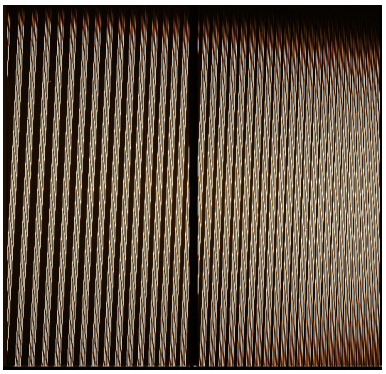
(a master sflat frame (set2))



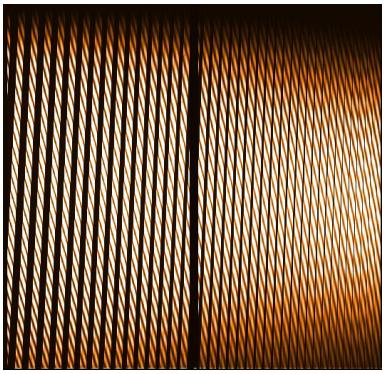
(a master sflat frame (set3))



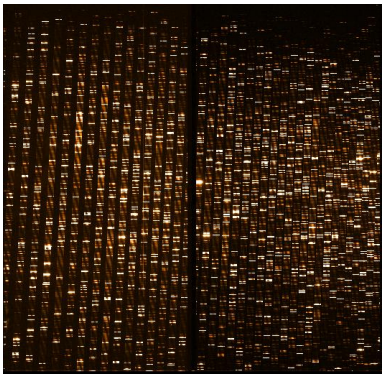
(a raw odd fibres frame)



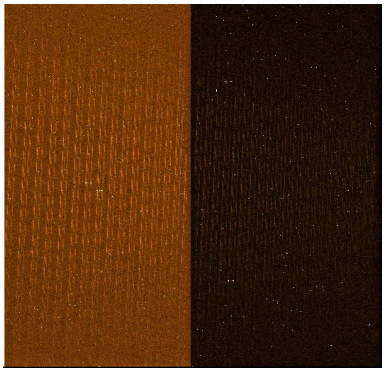
(a raw even fibres frame)



(a raw all fibres frame)



(a raw arc lamp fibre frame)



(a raw science frame)

Figure 7.1: Several FLAMES-UVES raw frames.

8 Static Calibration Data

In the following section ancillary data required for FLAMES-UVES data reduction are listed. For each of them we indicate the corresponding value of the HIERARCH ESO PRO CATG, in short PRO.CATG, FITS keyword. This has to be used to identify the frames listed in the *Set of Frames* (see Section 5.2.4, page 30). More information on those data may be found on *https://www.eso.org/sci/observing/phase3/data_types.html*

8.1 Line reference table

A reference list of arc lines is necessary to perform the wavelength calibration. Its PRO.CATG is LINE_REFER_TABLE. The wavelength values in the reference table refer to air. This frame is an input of the recipes uves_cal_predict and uves_cal_wavecal.

This release of the FLAMES-UVES pipeline provides a new reference line catalog (thargood_3.fits)[13] more accurate than the previous one (thargood_2.fits). Both catalogs are included but the user is recommended to use the new one.

8.2 Table to monitor line intensity

For quality control purposes, a table listing reference lines of intermediate intensity uniformly distributed on the blue/red detectors can be provided by DFO and has PRO.CATG equal to LINE_INTMON_TABLE.

[13]These line reference tables are also included in Service Mode data packages. As of P82 thargood_3.fits has been renamed to UV_GLRE_070222A_line_refer_table.fits

9 Data Reduction

In this section, after an overview of the main problems the data reduction needs to solve, we list the required data and the recipes which allow to solve them, giving the data reduction sequence necessary to reduce calibration and science data.

9.1 Data reduction overview

In order to fully reduce a set of FLAMES-UVES data, including calibration and science frames, the following data reduction problems need to be solved:

- The detector bias and dark current levels should be measured and subtracted.
- The echelle fiber order traces need to be determined in a robust manner.
- Compute and correct for detector pixel to pixel gain variations and the blaze function.
- For each fibre perform the wavelength calibration in a robust and automatic manner.
- Reduce science fiber data recovering the cross order fibres contamination.

9.2 Required input data

To be able to reduce science data one needs to use raw, product and static calibration data together with the pipeline recipes in a given sequence so as to generate all the necessary input to each pipeline recipe at each step along the way. We call this sequence "**data reduction cascade**". The FLAMES-UVES pipeline involves the following input data:

- Raw frames:
  - Bias frames to determine a master bias.
  - Dark(s) (optional) frames to determine a master dark.
  - Slit flat frames to determine a set of master slit flats covering all the nine fibres.
  - Fibre formatcheck frames to determine guess order and line tables.
  - Calibration fibre flat frames to determine guess order tables and have a flat-field for that fibre.
  - Odd-even-all fibre order definition frames to determine the fibre order traces tables and construct a reference base of non contaminated fibre frames.
  - Fibre arc lamp frames to determine the line table for all fibres.
- calibration data products[14]:
  - A master bias and a master dark to subtract them from master flat, and science frames.

[14]These are either created by processing the above data, or they might be provided by another source, e.g. as part of a Service mode data package, in which case the corresponding RAW data listed above is not needed

  - Master slit flat-field frames to correct for different detector pixel
  - Calibration fibre guess order tables to have a stable order tracing.
  - Calibration fibre order flat frames to have a flat frame to flat the calibration fibre and completed the base of fibre flats used to estimate the contamination of each fibre to the others.
  - Calibration line and order guess tables to have an automatic, accurate and stable wavelength calibration. efficiencies, the blaze function, the detector fringing at longer wavelengths.
  - Line tables to calibrate the object and sky spectra in wavelength.

- Reference files:

  - Line table to produce guess and final line tables.

9.3 Reduction cascade

Here we outline the logical sequence of steps which are needed to perform a complete spectral extraction of FLAMES-UVES data. All these operations can be carried out using the provided recipes. Examples of data reduction are in chapter 5.

1. Master bias and dark frames are generated.
2. For most frames (fibre and slit flat-field frames, science frames) a variance frame and a bad-pixel mask must be created.
3. A so-called format-check frame is examined. This frame must have been taken with a single fibre illuminated by a (Th-Ar) wavelength-calibration lamp, and all other fibres dark. This frame is compared with a physical model of the CCD illumination, with the help of an appropriate table of lines emitted by the calibration lamp, to derive a first guess of the order positions and of the wavelength-calibration solution.
4. The simultaneous calibration fibre flat frame is used to refine the guess order table and add an otherwise missing flat fibre trace to the pool of odd-even fibres to be used in the science frame extraction.
5. The whole set of fibres is considered (or at least those used in the science exposure), by processing both an odd-numbered- and an even-numbered fibre flat-field frames. In this way a complete (raw) fibre-order position table is obtained.
6. From the latter, the inter-order (background) table is computed, for later computation and subtraction of scattered-light contamination.
7. Slit flat-field frames are processed. At least two half-slit flat-field frames are needed to cover the fibre-illuminated region without overlapping, and these are combined and normalized. They will be used to remove pixel-to-pixel effects later. Associated variance frames and bad-pixel masks are also created.
8. odd/even fibre flat-field frames are processed. These frames contain the required information on the fibre cross-dispersion profile, and need to be input as separate odd/even fibre frames since adjacent fibres have partially overlapping profiles. Fibres for which there is no corresponding slit flat-field information are ignored. The frames are corrected for background light, and associated variance frames and bad-pixel masks are created.

9. The algorithm requires also an all-fibre flat-field frame, to compute the relative throughput among the odd and even fibres (that may have different intensity levels). It is treated as if it were a science frame, with the same extraction procedure, to obtain for each fibre and order a wavelength-dependent normalization (instead of a spectrum). Since the FLAMES-UVES data reduction includes two extraction methods (standard and optimal), the same method should be used to reduce the flat field and science frames if possible. Moreover, the all-fibre flat-field frame is also used as a reference for the positions of all fibres, since they are simultaneously lit here. For this purpose, the cross-dispersion shifts between the fibre positions in the odd and even fibre flat-field frames are computed with this common reference, and this additional information is stored in the order-fibre table.

10. Wavelength calibration is done, using a fibre frame with each fibre illuminated by a ThAr lamp, the guess line and fibre-order tables determined from previous data reduction steps. This step extracts each fibre spectrum, looks for ThAr lines and identifies them with help of the guess solution.

11. Eventually, the science spectra are extracted, either using standard or optimal extraction. Because of UVES stability limitations, there may be shifts between the fibre images in the flat-field frames and in the science frames. These shifts are expected to occur in the cross-dispersion direction and to be at most ± 1 pixel. Since especially optimal extraction is extremely sensitive to such shifts (spectra with large S/N cannot be extracted satisfactorily for shifts larger than 0.1 pix), the extraction routine computes the value of the shift (using a correlation-function method), and applies the opposite shift to the fibre flat-field frames (that is why pixel-to-pixel effects need to be computed separately using slit flat-field frames). After doing that, the optimal extraction is performed using the fibre flat-field frames as a model to fit the science frame at every wavelength (the fitted amplitudes are the extracted spectra at each wavelength), using at the same time the fibre profile information to deconvolve the partial fibre overlap. Standard extraction is somewhat simpler, doing only a sum of the science frame fibre spectra over a pre-defined window across dispersion, without fitting, but including also a deconvolution of fibre cross-contamination.

12. Next, extracted spectra are wavelength-calibrated.

13. Last, wavelength-calibrated, merged spectra are created.

9.3.1 Data reduction peculiarities

The data reduction software design to reduce FLAMES-UVES fibre-echelle mode data (described in [12], [1] and [19]) has some peculiarities when compared for example to the standard echelle data reduction, or with respect to the extraction of fibre spectra like those of FEROS.

In constrast to the standard echelle package:

- The fibre flat-field cross dispersion profiles (with pixel resolution) are used as "true" physical model of the fibre light distribution at each pixel on the detector in the optimal extraction of science targets, instead of some analytic approximation (In UVES-echelle mode, the corr-dispersion profile is measured using the science frame itself).

- A very good solution of the adjacent fibre contamination (which is a peculiar problem of fibre-fed multi-object spectrographs with close packed fibres and thus does not occur in echelle mode) is achieved.

- Spectra corresponding to different fibres, having a different fibre throughput, are corrected giving reliable relative fibre throughputs.
- Thanks to the use of three different kind of flat-field spectra (single fibre, odd-even-all fibres, slit flat-field frames) usually the final merged spectra have a very high quality, in which any residual oscillation artifacts are well below the noise level.

With respect to FEROS we have the same differences as before with an additional one:

- In FEROS the flat-fielding is done on the extracted spectra, first extracting the science and the flat-field and then dividing the first by the latter. Although this procedure has empirically been shown to lead to negligible differences with respect to the "correct" one, we still use the "correct" one (first correct science by flat-field, next extract the spectra).

The adopted data reduction together with the mentioned pros implies a number of cons. Essentially we have two:

- We need a lot of calibrations, compared for example to FEROS.
- The achieved high quality extraction quality is paid for with a quite extensive computational cost.

10 Pipeline Recipes Interfaces

In this section, for each recipe example, the required input data (and their classification tags) are provided. The column labeled "nr" indicates the number of required input frames with the following convention:

- 1 for a single frame (in case of pipeline calibration products, one for each detector chip of the corresponding arm),
- + for one or more input frames,
- ? for input frames which are optional (in case of pipeline calibration products one for the redl and one for the redu detector chips).
- ! recommended input frame

In the following it is assumed that the input files (in our examples raw data are located in the directory `/path_raw`, reference data are in the directory `/path_ref` and pipeline products are in the directory `/path_pro`) are existing FITS files (e.g. `/data1/uves/com2/UVES.2004-08-16T02:54:04.353.fits`, and `/cal/uves/ech/cal/thargood_3.fits`, `/cal/uves/ech/cal/linetable_redl.fits`, `/cal/uves/ech/cal/linetable_redu.fits`).

A full list of pipeline products is also provided for each recipe, indicating their default recipe name (optionally replaced by EsoRex to a given standard), the value of the FITS keyword HIERARCH ESO PRO CATG (in short PRO.CATG) and a short description. The keyword PRO.CATG is used to classify each frame. The DPR and other relevant keywords are used to associate to each raw frame the proper calibration frame.

The data may be recognized and organized according to the values of the following FITS keywords:

Association keyword	Information
HIERARCH ESO DPR TYPE	raw data type
HIERARCH ESO DPR CATG	raw data category
HIERARCH ESO DPR TECH	raw data technique
HIERARCH ESO INS GRAT2 WLEN	Instrument setup central wavelength
HIERARCH ESO DET DIT	Integration time
HIERARCH ESO INS OBSPLATE	Observing plate
HIERARCH ESO OCS SIMCAL	Simultaneous calibration observation mode

The pipeline is able to also process pipeline products generated by the MIDAS based pipeline. The viceversa is not true.

For each recipe the input parameters (as they appear in the recipe configuration file), the corresponding parameter aliases (to be set on the command line) and their default values are listed. Each recipe has the following common parameters:

```
--debug              : Whether or not to save intermediate results to local
                       directory. [FALSE]
--plotter            : Any plots produced by the recipe are redirected to the
```

```
                    command specified by this parameter. The plotting
                    command must contain the substring 'gnuplot' and must
                    be able to parse gnuplot syntax on its standard
                    input. Valid examples of such a command may include
                    'gnuplot -persist' and 'cat > mygnuplot$$.gp'. A
                    finer control of the plotting options can be obtained
                    by writing an executable script, e.g. my_gnuplot.pl,
                    that executes gnuplot after setting the desired
                    gnuplot options (e.g. set terminal pslatex color). To
                    turn off plotting, set this parameter to 'no'. [no]
--process_chip      : For RED arm data proces the redl,
                    redu, or both chip(s). <both | redl | redu | REDL |
                    REDU> [both]
```

A full description of each parameter is obtained by running the command **esorex - -parameters**, or **esorex - -help** or **esorex - -man-page**, or by looking at the Recipe Input Parameters section of the dedicated Gasgano window. Also, the role of the most important parameters is described in section 11.

Also quality control parameters are computed. They are written to the header of corresponding pipeline products. More and up-to-date information on instrument quality control can be found on www.eso.org/qc.

10.1 uves_cal_mbias

The recipe uves_cal_mbias creates a master bias frame.

10.1.1 Input

frame tag/category	nr	filename example
BIAS_<ARM>	+	/path_raw/uves_bias_<arm>.fits

10.1.2 Output

default recipe file name	format	PRO.CATG	short description
masterbias_<chipid>.fits	image	MASTER_BIAS_<CHIPID>	master bias frame

10.1.3 Quality control

The recipe computes the following QC parameters:

QC.DUTYCYCL	Time to store a frame [days]
QC.DUTY.CYCL	Time to complete an exposure, minus actual exposure time
QC.OUTi.RON.RAW	Read noise on raw images [ADU]
QC.OUTi.RON.MASTER	Read noise on master bias frame [ADU]
QC.OUTi.STRUCTY	Structure noise in Y [ADU]

QC.OUTi.STRUCTX	Structure noise in X [ADU]
QC.MEAN.RATIO.REF	Ratio to reference frame
QC.STD.MEAN.RATIO.REF	Sigma of MEAN.RATIO.REF

Additional QC parameters are logged in the power spectrum product:

QC.PD.STD	Raw file: std
QC.PD.STDMAD	Raw file: stdmad
QC.RATIO.PDi	Raw filei: ratio std/stdmad
QC.RATIO.PD.AVG	Average ratio std/stdmad
QC.RATIO.PD.MAX	Maximum ratio std/stdmad
QC.INDEX.PD.MAX	Index of the maximum ratio

10.1.4 Parameters

parameter	alias	default	min	max
uves_cal_mbias.clean_traps	clean_traps	FALSE		
uves_cal_mbias.stack_method	stack_method	median		
uves_cal_mbias.klow	klow	5.	0.	100.
uves_cal_mbias.khigh	khigh	5.	0.	100.
uves_cal_mbias.niter	niter	5	0	100

Note that the default parameters are robust.

10.2 uves_cal_mdark

The recipe uves_cal_mdark creates a master dark frame.

10.2.1 Input

frame tag/category	nr	filename example
DARK_<ARM>	+	/path_raw/uves_dark_<arm>.fits
MASTER_BIAS_<CHIPID>	?	/path_raw/masterbias_<chipid>.fits

10.2.2 Output

default recipe file name	format	PRO.CATG	short description
masterdark_<chipid>.fits	image	MASTER_DARK_<CHIPID>	master dark frame

10.2.3 Quality control

The recipe computes the following QC parameters:

QC.DATANCOM	Number of coadded frames
PRO.DATAMED	Median frame level [ADU]
QC.REGij.MIN	Min of region i j of size **box_sx** × **box_sy** [ADU]
QC.REGij.MAX	Max of region i j of size **box_sx** × **box_sy** [ADU]
QC.REGij.AVG	Mean of region i j of size **box_sx** × **box_sy** [ADU]
QC.REGij.MED	Median of region i j of size **box_sx** × **box_sy** [ADU]
QC.REGij.RMS	Rms of region i j of size **box_sx** × **box_sy** [ADU]
QC.REG.MIN.MIN	Min of all region Mins [ADU]
QC.REG.MIN.MAX	Max of all region Mins [ADU]
QC.REG.MIN.AVG	Mean of all region Mins [ADU]
QC.REG.MIN.MED	Median of all region Mins [ADU]
QC.REG.MIN.RMS	Rms of all region Mins [ADU]
QC.REG.MAX.MIN	Min of all region Maxs [ADU]
QC.REG.MAX.MAX	Max of all region Maxs [ADU]
QC.REG.MAX.AVG	Mean of all region Maxs [ADU]
QC.REG.MAX.MED	Median of all region Maxs [ADU]
QC.REG.MAX.RMS	Rms of all region Maxs [ADU]
QC.REG.AVG.MIN	Min of all region Means [ADU]
QC.REG.AVG.MAX	Max of all region Means [ADU]
QC.REG.AVG.AVG	Mean of all region Means [ADU]
QC.REG.AVG.MED	Median of all region Means [ADU]
QC.REG.AVG.RMS	Rms of all region Means [ADU]
QC.REG.MED.MIN	Min of all region Medians [ADU]
QC.REG.MED.MAX	Max of all region Medians [ADU]
QC.REG.MED.AVG	Mean of all region Medians [ADU]
QC.REG.MED.MED	Median of all region Medians [ADU]
QC.REG.MED.RMS	Rms of all region Medians [ADU]
QC.REG.RMS.MIN	Min of all region Rms [ADU]
QC.REG.RMS.MAX	Max of all region Rms [ADU]
QC.REG.RMS.AVG	Mean of all region Rms [ADU]
QC.REG.RMS.MED	Median of all region Rms [ADU]
QC.REG.RMS.RMS	Rms of all region Rms [ADU]
QC.DARK.MEAN.RAW	Mean value of raw frame
QC.DARK.MEAN	Mean value across master dark
QC.DARK.CURRENT	DARK.MEAN, normalised to 3600 secs exposure time

10.2.4 Parameters

parameter	alias	default	min	max
uves_cal_mdark.stack_method	stack_method	median		

uves_cal_mdark.klow	klow	5.	0.	100.
uves_cal_mdark.khigh	khigh	5.	0.	100.
uves_cal_mdark.niter	niter	5	0	100
uves_cal_mdark.qc_dark.reg.num_x	qc_dark.reg.num_x	4	0	
uves_cal_mdark.qc_dark.reg.num_y	qc_dark.reg.num_y	4	0	
uves_cal_mdark.qc_dark.reg.box_sx	qc_dark.reg.box_sx	100	0	
uves_cal_mdark.qc_dark.reg.box_sy	qc_dark.reg.box_sy	100	0	
uves_cal_mdark.qc_dark.reg.border_x	qc_dark.reg.border_x	100	0	
uves_cal_mdark.qc_dark.reg.border_y	qc_dark.reg.border_y	100	0	
uves_cal_mdark.qc_dark.reg.when	qc_dark.reg.when	0	0	1

Note that the default parameters are robust.

10.3 flames_cal_mkmaster

The recipe flames_cal_mkmaster computes the master slit flat frames.

10.3.1 Input

frame tag/category	nr	filename example
SFLAT_RED	+	/path_raw/uves_flat_set1_red.fits
SFLAT_RED	+	/path_raw/uves_flat_set2_red.fits
SFLAT_RED	+	/path_raw/uves_flat_set3_red.fits
FIB_ORD_TAB_REDL		/path_pro/ordertable_redl.fits
FIB_ORD_TAB_REDU		/path_pro/ordertable_redu.fits
MASTER_BIAS_REDL	?!	/path_pro/masterbias_redl.fits
MASTER_BIAS_REDU	?!	/path_pro/masterbias_redu.fits

10.3.2 Output

default recipe file name	format	PRO.CATG	short description
set1_masterflat_redl.fits	2d (pix-pix) image	MASTER_SFLAT_REDL	master flat frame
set1_masterflat_redu.fits	2d (pix-pix) image	MASTER_SFLAT_REDU	master flat frame
set2_masterflat_redl.fits	2d (pix-pix) image	MASTER_SFLAT_REDL	master flat frame
set2_masterflat_redu.fits	2d (pix-pix) image	MASTER_SFLAT_REDU	master flat frame
set3_masterflat_redl.fits	2d (pix-pix) image	MASTER_SFLAT_REDL	master flat frame
set3_masterflat_redu.fits	2d (pix-pix) image	MASTER_SFLAT_REDU	master flat frame

10.3.3 Quality control

The recipe computes the following QC parameters:

QC.DELTA.TEMP4	Difference INS.TEMP4.MEAN to reference
QC.DELTA.YSHIFT	Difference Y-shift to reference
QC.SIGMA.PH	Photon noise
QC.SIGMA.DX	Noise from DX derivative
QC.SIGMA.DY	Noise from DY derivative
QC.SIGNAL	Mean value across subwindow

The pipeline monitors the number of coadded frames (PRO.DATANCOM). Additional quality control information is monitored by DFO and can be found at www.eso.org/qc.

10.3.4 Parameters

```
--backsub.mmethod     : Background measuring method. If equal to 'median' the
                        background is sampled using the median of a
                        subwindow. If 'minimum', the subwindow minimum value
                        is used. If 'no', no background subtraction is done.
                        <median | minimum | no> [median]
--backsub.npoints     : This is the number of columns in interorder space used
                        to sample the background. [82]
--backsub.radiusy     : The height (in pixels) of the background sampling
                        window is (2*radiusy + 1). This parameter is not
                        corrected for binning. [2]
--backsub.sdegree     : Degree of interpolating splines. Currently only degree
                        = 1 is supported. [1]
--backsub.smoothx     : If spline interpolation is used to measure the
                        background, the x-radius of the post-smoothing window
                        is (smoothx * image_width). Here, 'image_width' is
                        the image width after binning. If negative, the
                        default values are used: (25.0/4096) for blue
                        flat-field frames, (50.0/4096) for red flat-field
                        frames, (300.0/4096) for blue science frames and
                        (300.0/4096) for red science frames. [-1.0]
--backsub.smoothy     : If spline interpolation is used to measure the
                        background, the y-radius of the post-smoothing window
                        is (smoothy * image_height). Here, 'image_height' is
                        the image height after binning. If negative, the
                        default values are used: (100.0/2048) for blue
                        flat-field frames, (300.0/2048) for red flat-field
                        frames, (200.0/2048) for blue science frames and
                        (500.0/2048) for red science frames. [-1.0]
```

10.4 flames_cal_predict

The recipe flames_cal_predict computes the line and order guess tables using a model of UVES and the atmospheric pressure information written in the header of the frame, the known offset between the UVES slit's center position and the one of the simultaneous calibration fibre, temperature and the instrument setting provided by the FITS header of the raw formatcheck frame which is acquired by illuminating the simultaneus fibre with a line emission lamp.

10.4.1 Input

frame tag/category	nr	filename example
FIB_ARC_LAMP_FORM_RED	1	/path_raw/flames_uves_arc_lamp_form_red.fits
LINE_REFER_TABLE	1	/path_ref/thargood_3.fits
MASTER_BIAS_REDL	?	/path_pro/masterbias_redl.fits
MASTER_BIAS_REDU	?	/path_pro/masterbias_redu.fits

10.4.2 Output

default recipe file name	format	PRO.CATG	short description
lineguesstable_redl.fits	table	FIB_LIN_GUE_REDL	Guess line table
lineguesstable_redu.fits	table	FIB_LIN_GUE_REDU	Guess line table
orderguesstable_redl.fits	table	FIB_ORD_GUE_REDL	Guess order table
orderguesstable_redu.fits	table	FIB_ORD_GUE_REDU	Guess order table

The guess line table contains the following columns:

X	Position along x
Y	Position along y
PEAK	line peak
Ident	line catalog wavelength
YNEW	Computed predicted line y position
Order	Relative order number
WAVEC	Predicted line wavelength of line peak
Aux	Product of wavelength and order number
XREG	Result of the polynomial regression XREG=Aux(X)
Pixel	Local dispersion
RORD	Order loaction (ouble precision to use it in a fit)
XPRED	Predicted X line position
YPRED	Predicted Y line position
XDIF	Difference between measured and predicted X line position
YDIF	Difference between measured and predicted Y line position
SELPLOT	selection column

The guess order table contains the following columns:

ABS_ORDER	Absolute order number
ORDER	Relative order number
X	Position along x
Y	Position along y
YFIT	Computed predicted order y position
RESIDUAL	Residual (Y-YFIT)

10.4.3 Quality control

The pipeline generates the following QC parameters:

QC.MODEL.NLINALL	Total number of detected lines
QC.MODEL.NLINSEL	Number of selected lines
QC.MODEL.DIFFXRMS	RMS difference of predicted and measured line x positions
QC.MODEL.DIFFXAVG	Mean difference of predicted and measured line x positions
QC.MODEL.DIFFXMED	Median difference of predicted and measured line x positions
QC.MODEL.DIFFYRMS	RMS difference of predicted and measured line y positions
QC.MODEL.DIFFYAVG	Mean difference of predicted and measured line y positions
QC.MODEL.DIFFYMED	Median difference of predicted and measured line y positions
QC.MODEL.RESXRMS	Std dev of X difference to physical model
QC.MODEL.RESXAVG	Average of X difference to physical model
QC.MODEL.RESXMED	Median of X difference to physical model
QC.MODEL.RESYRMS	Std dev of Y difference to physical model
QC.MODEL.RESYAVG	Average of Y difference to physical model
QC.MODEL.RESYMED	Median of Y difference to physical model
QC.MODEL.WLENMIN	Minimum predicted lines wavelength
QC.MODEL.WLENMAX	Maximum predicted lines wavelength
QC.MODEL.ORDMIN	Minimum predicted absolute order
QC.MODEL.ORDMAX	Maximum predicted absolute order
QC.WLENMIN	Minimum wavelength of spectral format
QC.WLENMAX	Maximum wavelength of spectral format
QC.ORDMIN	Minimum relative order
QC.ORDMAX	Maximum relative order
QC.FIB1.ABSTRANS	Average transmission countrate measured on the fibre
QC.FIB1.NHOTPIX	Number of found hot pixels and hot columns.
QC.FIB1.PLATENO	This is the Id of the plate to which the fibres were connected

10.4.4 Parameters

```
--mbox_x              : Match box X size. [40]
--mbox_y              : Match box Y size. [40]
--trans_x             : Detector translation along X. [0.0]
--trans_y             : Detector translation along Y. [0.0]
--ech_angle_off       : Offset on echelle angle. [0.0]
--cd_angle_off        : Offset on cross disperser angle. [0.0]
--ccd_rot_angle_off   : Offset on CCD rotation angle. [0.0]
--compute_regression_sw  : Compute regression?. [TRUE]
--x_axis_scale        : Scale X axis. [0.0]
--y_axis_scale        : Scale Y axis. [0.0]
--def_pol1            : Polynomial X deg. [4]
--def_pol2            : Polynomial Y deg. [5]
--kappa               : Kappa value in kappa sigma clipping on RESIDUAL
                        between YFIT and Y columns. [4.5]
--tol                 : Tolerance in kappa sigma clipping on RESIDUAL between
                        YFIT and Y columns. [2.0]
```

Pls note that to have proper wavelength accuracy the parameter **tolerance** should be kept small, we recommend

a value of 0.6.

10.5 flames_cal_orderpos

The recipe flames_cal_orderpos generates an order table from a raw frame taken with the calibration fibre illuminated by a continuum lamp. This order table is more accurate than the order guess table previously generated. Additional product of this step is an order fibre flat frame, that covers the simultaneous calibration fibre, to be added to the pool of odd-even order flat frame to build a complete base of uncontaminate fibres frame to extract science data.

10.5.1 Input

frame tag/category	nr	filename example
FIB_ORDEF_RED	1	/path_raw/flames_uves_order_flat_red.fits
FIB_ORD_GUE_REDL	?!	/path_pro/orderguesstable_redl.fits
FIB_ORD_GUE_REDU	?!	/path_pro/orderguesstable_redu.fits
MASTER_BIAS_REDL	?	/path_pro/masterbias_redl.fits
MASTER_BIAS_REDU	?	/path_pro/masterbias_redu.fits

10.5.2 Output

default recipe file name	format	PRO.CATG	short description
ordertable_redl.fits	table	FIB_ORD_TAB_REDL	order table
ordertable_redu.fits	table	FIB_ORD_TAB_REDU	order table
order_def_redl.fits	table	FIB_ORDEF_REDL	fiber order def rotated frame
order_def_redu.fits	table	FIB_ORDEF_REDU	fiber order def rotated frame

The output table contains the columns

Order	Relative order number
X	Position along x
Y	Order line centroid location
dY	Uncertainty of Y
Residual_Square	Squared residual
OrderRMS	Root mean squared residual of initial one-dimensional linear fit of order
OrderSlope	Slope of order
Yfit	The fitted order location
dYfit_Square	Variance of Yfit
Residual	(Y - Yfit)

10.5.3 Quality control

QC.ORD.RESIDMIN	Min residuals in order table
QC.ORD.RESIDMAX	Max residuals in order table
QC.ORD.RESIDAVG	Mean residuals in order table
QC.ORD.RESIDRMS	RMS residuals in order table
QC.ORD.NPRED	Predicted number of orders
QC.ORD.NDET	Detected number of orders
QC.ORD.NPOSALL	Number of positions found
QC.ORD.NPOSSEL	Number of positions selected
QC.ORDMIN	Minimum (relative) order value
QC.ORDMAX	Maximum (relative) order value

Where the residuals measure the difference between the order solution obtained by applying the polynomial model and the corresponding order location measurements on the frame.

10.5.4 Parameters

```
--use_guess_tab       : If a Guess order table is provided this parameter
                        defines how it is used:0: No usage, 1: use it to set
                        lower/upper Y rows where order are searched 2: the
                        order table tries to fully match the guess. [1]
--radx                : X-radius of median filtering window. [2]
--rady                : Y-radius of median filtering window. [1]
--mmethod             : Background subtraction method. If equal to 'median'
                        the background is sampled using the median of a
                        sub-window. If 'minimum', the minimum sub-window
                        value is used. If 'no', no background subtraction is
                        done. <median | minimum | no> [median]
--backsubgrid         : Number of grid points (in x- and y-direction) used to
                        estimate the background (mode=poly). [50]
--backsubradiusy      : The height (in pixels) of the background sampling
                        window is (2*radiusy + 1). This parameter is not
                        corrected for binning. [2]
--backsubkappa        : The value of kappa in the one-sided kappa-sigma
                        clipping used to estimate the background (mode=poly).
                        [4.0]
--backsubdegx         : Degree (in x) of polynomial used to estimate the
                        background (mode=poly). [2]
--backsubdegy         : Degree (in y) of polynomial used to estimate the
                        background (mode=poly). [2]
--samplewidth         : Separation of sample traces (used by Hough transform)
                        in input image. [50]
--minslope            : Minimum possible line slope. This should be the
                        'physical' slope on the chip, i.e. not taking binning
                        factors into account, which is handled by the recipe.
                        [0.0]
--maxslope            : Maximum possible line slope. [0.2]
--sloperes            : Resolution (width in pixels) of Hough space. [120]
--pthres              : In automatic mode, or if the number of orders to
                        detect is read from a guess table, the detection of
                        new lines stops when the intensity of a candidate
                        line drops to less than 'pthres' times the intensity
                        of the previous detection. [0.2]
--tracestep           : The step size used when tracing the orders. [10]
```

```
--minthresh          : The minimum threshold value is (min + minthres*(max -
                       min)). Here 'min' and 'max' are the lowest and
                       highest pixel values in the central bin of the order.
                       [0.01]
--maxgap             : If the order line drops below detection threshold, the
                       order tracing algorithm will try to jump a gap of
                       maximum size 'maxgap' multiplied by the image width.
                       [0.2]
--maxrms             : When fitting the orders with straight lines, this is
                       the maximum allowed RMS relative to the median RMS of
                       all orders. [100.0]
--defpol1            : The degree of the bivarite fit (cross dispersion
                       direction). If negative, the degree is optimized to
                       give the best fit. [-1]
--defpol2            : The degree of the bivarite fit (order number). If
                       negative, the degree is optimized to give the best
                       fit. [-1]
--kappa              : Used for kappa-sigma clipping of the final polynomial
                       fit. If negative, no clipping is done. [6.0]
```

10.6 flames_cal_prep_sff_ofpos

The recipe uves_cal_prep_sff_ofpos is used to determine the fibre order table and construct several frames needed to extract a science fibre frame using tree input fibre frames obtained by illumination with a continuum source first the odd fibres, then the even fibres and, finally all the fibres.

10.6.1 Input

frame tag/category	nr	filename example
FIB_FF_ODD_RED	1	/path_raw/flames_uves_odd_red.fits
FIB_FF_EVEN_RED	1	/path_raw/flames_uves_even_red.fits
FIB_FF_ALL_RED	1	/path_raw/flames_uves_all_red.fits
MASTER_BIAS_REDL	?	/path_pro/masterbias_redl.fits
MASTER_BIAS_REDU	?	/path_pro/masterbias_redu.fits
FIB_ORD_TAB_REDL	1	/path_pro/ordertable_redl.fits
FIB_ORD_TAB_REDU	1	/path_pro/ordertable_redu.fits
MASTER_SFLAT_REDL	1	/path_pro/masterflat_set1_redl.fits
MASTER_SFLAT_REDL	1	/path_pro/masterflat_set2_redl.fits
MASTER_SFLAT_REDL	1	/path_pro/masterflat_set3_redl.fits
MASTER_SFLAT_REDU	1	/path_pro/masterflat_set1_redu.fits
MASTER_SFLAT_REDU	1	/path_pro/masterflat_set2_redu.fits
MASTER_SFLAT_REDU	1	/path_pro/masterflat_set3_redu.fits

Note: Simultaneous fibre order table frames used to be classified as FIB_ORD_GUE_REDL/REDU. This tag is the same adopted for guess order tables generated by the uves_cal_predict and flames_cal_predict recipes. In order to remove this product classification name clash, and for backward compatibility (in order to support data reduction of input data generated by the MIDAS based pipeline) both tags are supported. The user is advised to use the new tag (FIB_ORD_TAB_REDL/REDU), in case of data reduced with the CPL based pipeline.

10.6.2 Output

default recipe filename	format	PRO.CATG	short description
xt_odd_l	table	FIB_FF_ODD_INFO_TAB	info table
xt_even_l	table	FIB_FF_EVEN_INFO_TAB	info table
xt_all_l	table	FIB_FF_ODD_INFO_TAB	info table
slitff_common_redl	imagelist	SLIT_FF_COM_REDL	slitff common frame
slitff_norm_redl	image	SLIT_FF_NOR_REDL	slitff common frame
slitff_dtc_redl	imagelist	SLIT_FF_DTC_REDL	slitff data frame
slitff_sgc_redl	imagelist	SLIT_FF_SGC_REDL	slitff sigma frame
slitff_bpc_redl	imagelist	SLIT_FF_BPC_REDL	slitff badpixel frame
slitff_bnc_redl	imagelist	SLIT_FF_BNC_REDL	slitff frames boundaries
orfl	table	FIB_ORDEF_TABLE_REDL	fibre order table
fibreff_common_redl	imagelist	FIB_FF_COM_REDL	fibff common frame
fibreff_norm_redl	imagelist	FIB_FF_NOR_REDL	fibff common frame
fibreff_nsigma_redl	imagelist	FIB_FF_NSG_REDL	Normalisation sigmas for fibre FF
fibreff_dtc_redl	imagelist	FIB_FF_DTC_REDL	fibff data frame
fibreff_sgc_redl	imagelist	FIB_FF_SGC_REDL	fibff sigma frame
fibreff_bpc_redl	imagelist	FIB_FF_BPC_REDL	fibff badpixel frame
slitff_common_redu	imagelist	SLIT_FF_COM_REDU	slitff common frame
slitff_norm_redu	image	SLIT_FF_NOR_REDU	slitff common frame
slitff_dtc_redu	imagelist	SLIT_FF_DTC_REDU	slitff data frame
slitff_sgc_redu	imagelist	SLIT_FF_SGC_REDU	slitff sigma frame
slitff_bpc_redu	imagelist	SLIT_FF_BPC_REDU	slitff badpixel frame
slitff_bnc_redu	imagelist	SLIT_FF_BNC_REDU	slitff frames boundaries
orfu	table	FIB_ORDEF_TABLE_REDU	fibre order table
fibreff_common_redu	imagelist	FIB_FF_COM_REDU	fibff common frame
fibreff_norm_redu	imagelist	FIB_FF_NOR_REDU	fibff common frame
fibreff_nsigma_redu	imagelist	FIB_FF_NSG_REDU	Normalisation sigmas for fibre FF
fibreff_dtc_redu	imagelist	FIB_FF_DTC_REDU	fibff data frame
fibreff_sgc_redu	imagelist	FIB_FF_SGC_REDU	fibff sigma frame
fibreff_bpc_redu	imagelist	FIB_FF_BPC_REDU	fibff badpixel frame

10.6.3 Quality control

The recipe computes the following QC parameters:

QC.MIN.FIB	Minimum fibre ID found
QC.MAX.FIB	Maximum fibre ID found
QC.N.FIB	Number of fibers found
QC.N.FIB.TOT	Total number of fibers
QC.N.FIB.MISS	Number of fibers missing

10.6.4 Parameters

```
--ext_method          : Extraction method. <std | opt | fst | fop> [opt]
--bias_method         : Bias subtraction method, M for master bias frame, N
                        for no bias subtraction, V to subtract a constant
                        bias level defined by the parameter bias_value. <M |
                        V | N> [M]
--bias_value          : Bias value (only if bias_method = V). [200]
--filter_switch       : Filter switch. <none | median> [none]
--sat_thr             : Saturation threshold. [55000]
--fileprep            : Slitff* and Fibreff* file preparation. If fast
                        extraction method is used it should be set to FALSE.
                        [TRUE]
--cubify              : Cubify switch. [TRUE]
--save_flat_size      : To be sure to use the flat part of a slit flats one may
                        need to subtract this bit. The default value -1, is used
                        for automatic setting: if WCEN=520 save_flat_size=0, else
                        save_flat_size=2. Values explicitly set by user overwrite
                        this rule. [-1]
```

10.7 flames_cal_wavecal

The recipe flames_cal_wavecal is used to determine the wavelength dispersion coefficients and construct a wavelength calibration table from a frame were all the fibres are illuminated from a arc line calibration lamp.

10.7.1 Input

frame tag/category	nr	filename example
FIB_ARC_LAMP_RED	1	/path_raw/flames_uves_arc_lamp_red.fits
MASTER_BIAS_REDL	?	/path_pro/masterbias_redl.fits
MASTER_BIAS_REDU	?	/path_pro/masterbias_redu.fits
MASTER_FLAT_REDL	?!	/path_pro/masterflat_redl.fits
MASTER_FLAT_REDU	?!	/path_pro/masterflat_redu.fits
FIB_ORDEF_TABLE_REDL	1	/path_pro/orfl.fits
FIB_ORDEF_TABLE_REDU	1	/path_pro/orfu.fits
FIB_LIN_GUE_REDL	1	/path_pro/lineguesstable_redl.fits
FIB_LIN_GUE_REDU	1	/path_pro/lineguesstable_redu.fits
LINE_REFER_TABLE	1	/path_ref/thargood_3.fits

10.7.2 Output

default recipe file name	format	PRO.CATG	short description
linetable_redl.fits	table	FIB_LINE_TABLE_REDL	line table
linetable_redu.fits	table	FIB_LINE_TABLE_REDU	line table

The output line table(s), LINE_TABLE_REDL, LINE_TABLE_REDU contains three extensions per fibre (i.e. 6x3=18 or 8x3=24 extensions). The first extensions contain a table with the columns described below

X	Horizontal position (from Gaussian fit) of detected line
dX	Uncertainty (one sigma) of X
XWidth	Width (in pixels) of detected line (from Gaussian fit)
Y	Relative order number of detected line
Peak	Intensity of detected line
Background	Fitted background (ADU) of detected line
Slope	Linear background slope (ADU/pixel) of detected line
Order	Relative order number of detected line
AbsOrder	Absolute order number of detected line
Ynew	Vertical position of detected line
WaveC	Wavelength of this line (computed using the resulting dispersion relation)
dLambdaC	Uncertainty (one sigma) of 'WaveC'.
Pixel	The local dispersion in A/pixel.
Residual	Residual (in A) of this line
Residual_pix	Residual (in pixels) of this line
Lambda_candidate	Nearest line in catalogue
dLambda_cat_sq	Squared distance to nearest catalogue line
dLambda_nn_sq	Squared distance to nearest neighbour multiplied by ALPHA
Ident	The wavelength associated with this emission line, or NULL if this line was not identified.
dIdent	Uncertainty of catalogue wavelength
Select	1 if the line was identified, 0 otherwise
NLinSol	1 if the line was identified and accepted for the polynomial fit, 0 otherwise
Intensity	Intensity of detected line scaled to unit exposure time. (This column is present only if a LINE_INTMON_TABLE is provided.)

The 2nd table extension contains the dispersion relation on the form $p(x, m) = \lambda \cdot m$, where m is the order number. The 3rd table extension contains the map from (pixel, pixel)-space to physical order numbers (used internally by the calibration recipe, a 2d polynomial on the form $p(x, y) = m$).

If there is more than one extraction window (default is 3), the results of each calibration is stored in subsequent table extensions of the same FITS file. For example, extensions 4, 5 and 6 would contain the resulting line table (and its two associated polynomials) for the second extraction window. The results for the calibration of the n'th extraction window is stored in extensions (3*n - 2) to 3*n.

The offset of the extraction window is stored in the FITS keyword like "HISTORY OFFSET -7.5". The corresponding window number (counting from 1) is stored in keywords like "HISTORY WINDOW 3"

The polynomials are stored in table extensions as in the example

Order1	Order2	Coeff
-1	-1	a0
-1	-1	a1

-1	-1	a2
-1	-1	b0
-1	-1	b1
-1	-1	b2
0	0	c00
0	1	c01
0	2	c02
1	0	c10
1	1	c11
1	2	c12
2	0	c20
2	1	c21
2	2	c22

The third column contains the polynomial coefficients corresponding to the degree defined by the two first columns. The six first table rows defines a linear transformation of the dependent and independent variables. For example the table shown represents the polynomial p defined by

$$(p(x,y) - a_0)/b_0 = q((x - a_1)/b_1, (y - a_2)/b_2)$$

and

$$q(x,y) = \sum_{i=0}^{2} \sum_{j=0}^{2} c_{ij} x^i y^j$$

The linear transformation of the three variables was introduced in order to ensure numerical stability in the polynomial fitting routine.

10.7.3 Quality control

The recipe computes the following quality control parameters:

QC.FWHMAVG	Average FWHM of lines selected
QC.FWHMRMS	Standard deviation of FWHM of selected lines
QC.FWHMMED	Median FWHM of selected lines
QC.RESOLAVG	Average resolving power of selected lines
QC.RESOLRMS	Standard deviation of the resolving power of selected lines
QC.RESOLMED	Median resolving power of selected lines
QC.LINE.RESIDAVG	Mean of residuals of line positions to fit
QC.LINE.RESIDRMS	Sigma of residuals of line positions to fit
QC.WLENMIN	Minimum wavelength of detected order
QC.WLENMAX	Maximum wavelength of detected order
QC.ORDMIN	Minimum order number detected

QC.ORDMAX	Maximum order number detected
QC.NLINTOT	Total number of lines found on the frame
QC.NLINSEL	Number of selected lines
QC.NLINRES1	Number of lines with residuals < 0.1 nm
INS.SLIT3.WID	Slit width
INS.GRAT2.WLEN	Grating central wavelength [nm].
INS.TEMP2.MEAN	Average temperature [C].
QC.NHOTPIX	Number of pixelsoutside the range [-20,55000]
QC.PLATENO	Plate setting value
QC.FIB8.DRSNO	Sequential number of detected fibre
QC.FIB8.SEQ	Sequential number of detected fibre
QC.FIB8.POS	Detected fibre Y position relative to order slit center
QC.FIB8.MSK	Detected fibre mask value (1 if the fibre was recognised as lit)
QC.FIB8.FWHMAVG	Average FWHM of lines detected for the given fibre
QC.FIB8.FWHMRMS	RMS of lines detected for the given fibre
QC.FIB8.FWHMMED	Median FWHM of lines detected for the given fibre
QC.FIB8.RESOLAVG	Average instrumental resolution for the given fibre
QC.FIB8.RESOLRMS	RMS of instrumental resolution for the given fibre
QC.FIB8.RESOLMED	Median instrumental resolution for the given fibre
QC.FIB8.LINE.RESIDAVG	Average line residuals to reference values for the given fibre
QC.FIB8.LINE.RESIDRMS	RMS of line residuals to reference values for the given fibre
QC.FIB8.WLENMIN	Minimum wavelength [nm] of detected lines for the given fibre
QC.FIB8.WLENMAX	Maximum wavelength [nm] of detected lines for the given fibre
QC.FIB8.ORDMIN	Minimum order value for the given fibre
QC.FIB8.ORDMAX	Maximum order value for the given fibre
QC.FIB8.NLINTOT	Total number of detected lines for the given fibre
QC.FIB8.NLINSEL	Number of selected lines for the given fibre
QC.FIB8.NLINSOL	Number of lines used for the final wavelength solution for the given fibre
QC.FIB8.NLINRES1	Number of lines with residuals less than 1 Åfor the given fibre

The line FWHMs and the corresponding resolving power are measured along the dispersion direction (see also Figure 11.2). The residuals measures the differences between the solution from the polynomial fit model and the corresponding line positions found on the arc lamp frame (see also Figure 11.1).

10.7.4 Parameters

```
--nwindows          : Number of extraction windows per trace. The windows
                      will be aligned (i.e. no overlap and no spacing
                      between adjacent windows). Unless an offset is
                      specified, the middle window(s) is centered on the
                      trace. [1]
--length            : Length (in pixels) of each extraction window. This
                      parameter is also equal to the seperation of adjacent
                      window centers, causing the extraction windows to
                      always be aligned. The parameter is automatically
                      adjusted according to the binning of the input raw
                      frame. If negative, the extraction window length is
                      determined automatically to cover the full slit.
```

```
                        [7.0]
--offset              : A global offset (in pixels) of all extraction windows.
                        [0.0]
--extract.method      : Extraction method. <average | linear | 2d | weighted |
                        optimal> [average]
--extract.kappa       : In optimal extraction mode, this is the threshold for
                        bad (i.e. hot/cold) pixel rejection. If a pixel
                        deviates more than kappa*sigma (where sigma is the
                        uncertainty of the pixel flux) from the inferred
                        spatial profile, its weight is set to zero. If this
                        parameter is negative, no rejection is performed.
                        [10.0]
--extract.chunk       : In optimal extraction mode, the chunk size (in pixels)
                        used for fitting the analytical profile (a fit of the
                        analytical profile to single bins would suffer from
                        low statistics). [32]
--extract.profile     : In optimal extraction mode, the kind of profile to
                        use. 'gauss' gives a Gaussian profile, 'moffat' gives
                        a Moffat profile with beta=4 and a possible linear
                        sky contribution. 'virtual' uses a virtual resampling
                        algorithm (i.e. measures and uses the actual object
                        profile). 'constant' assumes a constant spatial
                        profile and allows optimal extraction of wavelength
                        calibration frames. 'auto' will automatically select
                        the best method based on the estimated S/N of the
                        object. For low S/N, 'moffat' or 'gauss' are
                        recommended (for robustness). For high S/N, 'virtual'
                        is recommended (for accuracy). In the case of virtual
                        resampling, a precise determination of the order
                        positions is required; therefore the order-definition
                        is repeated using the (assumed non-low S/N) science
                        frame. <constant | gauss | moffat | virtual | auto>
                        [auto]
--extract.skymethod   : In optimal extraction mode, the sky subtraction method
                        to use. 'median' estimates the sky as the median of
                        pixels along the slit (ignoring pixels close to the
                        object), whereas 'optimal' does a chi square
                        minimization along the slit to obtain the best
                        combined object and sky levels. The optimal method
                        gives the most accurate sky determination but is also
                        a bit slower than the median method. <median |
                        optimal> [optimal]
--extract.oversample  : The oversampling factor used for the virtual
                        resampling algorithm. If negative, the value 5 is
                        used for S/N <=200, and the value 10 is used if the
                        estimated S/N is > 200. [-1]
--extract.best        : (optimal extraction only) If false (fastest), the
                        spectrum is extracted only once. If true (best), the
                        spectrum is extracted twice, the second time using
                        improved variance estimates based on the first
                        iteration. Better variance estimates slightly improve
                        the obtained signal to noise but at the cost of
                        increased execution time. [TRUE]
--range               : Width (in pixels) of search window is 2*range + 1.
                        This parameter is automatically adjusted according to
                        binning. [8]
--minlines            : Minimum number of lines to detect. If zero, the
                        default value (2000 for BLUE/REDL chips; 1000 for
                        REDU chip) is used. [0]
--maxlines            : Maximum number of lines to detect. If zero, the
                        default value (2400 for BLUE/REDL chip; 1400 for REDU
                        chip) is used. [0]
--shiftmax            : The maximum shift (in pixels) compared to guess
                        solution. This parameter is automatically corrected
                        for binning. [10.0]
```

```
--shiftstep          : The step size (in pixels) used when searching for the
                       optimum shift. This parameter is automatically
                       corrected for binning. [0.1]
--shifttoler         : Tolerance (in pixels) when matching shifted lines.
                       This parameter is not adjusted according to binning.
                       [0.05]
--alpha              : The parameter that controls the distance to the
                       nearest neighbours. [0.1]
--maxerror           : This parameter controls the graceful exit of the
                       identification loop. If the RMS of the global fit
                       exceeds this value (in pixels) the iteration stops.
                       [20.0]
--degree             : Degrees of the global 2d dispersion polynomial. If a
                       negative number is specified, the polynomial degrees
                       are automatically selected by starting from (1, 1)
                       and inreasing the degrees as long as the RMS residual
                       decreases significantly. [5]
--tolerance          : Tolerance of fit. If positive, the tolerance is in
                       pixel units. If negative, abs(tolerance) is in
                       wavelength units. Lines with residuals worse than the
                       tolerance are excluded from the final fit. Unlike in
                       previous versions, this parameter is not corrected
                       for CCD binning. This rejection based on the absolute
                       residual in pixel can be effectively disabled by
                       setting the tolerance to a very large number (e.g.
                       9999). In that case outliers will be rejected using
                       only kappa sigma clipping. [0.6]
--kappa              : Lines with residuals more then kappa stdev are
                       rejected from the final fit. [4.0]
--rebin.wavestep     : The bin size used for BLUE/REDL data (in w.l.u.) in
                       wavelength space. If negative, a step size of 2/3 * (
                       average pixel size ) is used. [-1.0]
--rebin.wavestep_redu : The bin size used for REDU data (in w.l.u.) in
                       wavelength space. If negative, a step size of 2/3 * (
                       average pixel size ) is used. [-1.0]
--rebin.scale        : Whether or not to multiply by the factor dx/dlambda
                       (pixels per wavelength) during the rebinning. This
                       option is disabled as default in concordance with the
                       method used in the MIDAS pipeline. This option should
                       be set to true to convert the observed flux (in
                       pixel-space) to a flux per wavelength (in
                       wavelength-space). [FALSE]
```

10.8 flames_obs_scired

This recipe reduces a science fiber frame.

10.8.1 Input

frame tag/category	nr	filename example
FIB_SCI_RED	1	/path_raw/flames_uves_science_red.fits
SLIT_FF_COM_REDL	1	/path_pro/slitff_com_redl.fits
SLIT_FF_COM_REDU	1	/path_pro/slitff_com_redu.fits
SLIT_FF_NOR_REDL	1	/path_pro/slitff_nor_redl.fits
SLIT_FF_NOR_REDU	1	/path_pro/slitff_nor_redu.fits
SLIT_FF_DTC_REDL	1	/path_pro/slitff_dtc_redl.fits

SLIT_FF_DTC_REDU	1	/path_pro/slitff_dtc_redu.fits
SLIT_FF_BPC_REDL	1	/path_pro/slitff_bpc_redl.fits
SLIT_FF_BPC_REDU	1	/path_pro/slitff_bpc_redu.fits
SLIT_FF_SGC_REDL	1	/path_pro/slitff_sgc_redl.fits
SLIT_FF_SGC_REDU	1	/path_pro/slitff_sgc_redu.fits
SLIT_FF_BNC_REDL	1	/path_pro/slitff_bnc_redl.fits
SLIT_FF_BNC_REDU	1	/path_pro/slitff_bnc_redu.fits
FIB_FF_COM_REDL	1	/path_pro/fibreff_com_redl.fits
FIB_FF_COM_REDU	1	/path_pro/fibreff_com_redu.fits
FIB_FF_NOR_REDL	1	/path_pro/fibreff_nor_redl.fits
FIB_FF_NOR_REDU	1	/path_pro/fibreff_nor_redu.fits
FIB_FF_DTC_REDL	1	/path_pro/fibreff_dtc_redl.fits
FIB_FF_DTC_REDU	1	/path_pro/fibreff_dtc_redu.fits
FIB_FF_BPC_REDL	1	/path_pro/fibreff_bpc_redl.fits
FIB_FF_BPC_REDU	1	/path_pro/fibreff_bpc_redu.fits
FIB_FF_SGC_REDL	1	/path_pro/fibreff_sgc_redl.fits
FIB_FF_SGC_REDU	1	/path_pro/fibreff_sgc_redu.fits
FIB_FF_NSG_REDL	1	/path_pro/fibreff_l_nsigma.fits
FIB_FF_NSG_REDU	1	/path_pro/fibreff_u_nsigma.fits
FIB_FF_BNC_REDL	1	/path_pro/fibreff_bnc_redl.fits
FIB_FF_BNC_REDU	1	/path_pro/fibreff_bnc_redu.fits
FIB_ORDEF_TABLE_REDL	1	/path_pro/orfl.fits
FIB_ORDEF_TABLE_REDU	1	/path_pro/orfu.fits
FIB_LINE_TABLE_REDL	1	/path_pro/linetable_redl.fits
FIB_LINE_TABLE_REDU	1	/path_pro/linetable_redu.fits
MASTER_BIAS_REDL	?	/path_pro/masterbias_redl.fits
MASTER_BIAS_REDU	?	/path_pro/masterbias_redu.fits

10.8.2 Output

default recipe file name	format	PRO.CATG	short description
bin_table_info_redl	table	FIB_SCI_INFO_TAB_REDL	table with information on fibres setup
fxb_l_raw000i	2d (pix-ord) image	XB_SCI_RAW_REDL	extracted, flatfielded raw frame
wfxb_l_raw000i	2d (wav-ord) image	WXB_SCI_RAW_REDL	rebinned, extracted, flatfielded raw frame
mwfxb_l_raw000i	1d (wav) image	MWXB_SCI_RAW_REDL	merged,rebinned, extracted, flatfielded raw frame, uncorrected for fibre throughput
fxb_l_rawsig000i	2d (pix-ord) image	ERR_XB_SCI_RAW_REDL	error frame for XB_SCI_RAW_REDL
wfxb_l_rawsig000i	2d (wav-ord) image	ERR_WXB_SCI_RAW_REDL	error frame for WXB_SCI_RAW_REDL
mwfxb_l_raw000i_sigma	1d (wav) image	ERR_MWXB_SCI_RAW_REDL	error for MWXB_SCI_RAW_REDL
fxb_l_000i	2d (pix-ord) image	XB_SCI_REDL	extracted, flatfielded raw frame
wfxb_l_000i	2d (wav-ord) image	WXB_SCI_REDL	rebinned, extracted, flatfielded, fibre throughput corrected raw frame
mwfxb_l_000i	1d (wav) image	MWXB_SCI_REDL	merged, rebinned, extracted, flatfielded, fibre throughput corrected raw frame
fxb_l_sig000i	2d (pix-ord) image	ERR_XB_SCI_REDL	error frame for XB_SCI_REDL
wfxb_l_sig000i	2d (wav-ord) image	ERR_WXB_SCI_REDL	error frame for WXB_SCI_REDL
mwfxb_l_000i_sigma	1d (wav) image	ERR_MWXB_SCI_REDL	error frame for MWXB_SCI_REDL
fxb_u_raw000i	2d (pix-ord) image	XB_SCI_RAW_REDU	extracted, flatfielded raw frame
wfxb_u_raw000i	2d (wav-ord) image	WXB_SCI_RAW_REDU	rebinned, extracted, flatfielded raw frame
mwfxb_u_raw000i	1d (wav) image	MWXB_SCI_RAW_REDU	merged,rebinned, extracted, flatfielded raw frame uncorrected for fibre throughput

fxb_u_rawsig000i	2d (pix-ord) image	ERR_XB_SCI_RAW_REDU	error frame for XB_SCI_RAW_REDU
wfxb_u_rawsig000i	2d (wav-ord) image	ERR_WXB_SCI_RAW_REDU	error frame for WXB_SCI_RAW_REDU
mwfxb_u_raw000i_sigma	1d (wav) image	ERR_MWXB_SCI_RAW_REDU	error frame for MWXB_SCI_RAW_REDU
fxb_u_000i	2d (pix-ord) image	XB_SCI_REDU	extracted, flatfielded raw frame
wfxb_u_000i	2d (wav-ord) image	WXB_SCI_REDU	rebinned, extracted, flatfielded raw frame
mwfxb_u_000i	1d (wav) image	MWXB_SCI_REDU	merged,rebinned, extracted, flatfielded raw frame
fxb_u_sig000i	2d (pix-ord) image	ERR_XB_SCI_REDU	error frame for XB_SCI_REDU
wfxb_u_sig000i	2d (wav-ord) image	ERR_WXB_SCI_REDU	error frame for WXB_SCI_REDU
mwfxb_u_000i_sigma	1d (wav) image	ERR_MWXB_SCI_REDU	error frame for MWXB_SCI_REDU

Please note that the recipes extract only the fibers that have been allocated to an object (or to the sky).

10.8.3 Quality control

The recipe computes the folowing quality control parameters in the FITS header of the reduced frame.

QC.FWHMAVG	Average FWHM of lines selected

10.8.4 Parameters

```
--ext_method          : Extraction method. <std | opt | fst | fop> [opt]
--cor_max_fnd         : Find correlation maximum?. <N | Y> [Y]
--cor_def_rng         : Correlation range size?. [6.0]
--cor_def_pnt         : Correlation sampling points?. [25]
--cor_def_off         : Correlation center offset?. [0.0]
--corvel_iter         : Velocity correlation iteration number (SimCal). [1]
--bias_method         : Bias subtraction method. <M | V | N> [M]
--bias_value          : Bias value (only if bias_method = V). [200]
--cubify_sw           : Cubify switch. <Y | N> [N]
--filt_sw             : Filter switch. <none | median> [none]
--bkg_max_io_win      : Background window number in each full inter order. [500]
--bkg_xy_win_sz_x     : x maximum size of each background window:. [6]
--bkg_xy_win_sz_y     : y maximum size of each background window:. [2]
--pixel_thresh_max    : Pixel saturation threshold max. [55000]
--pixel_thresh_min    : Pixel saturation threshold min. [-20]
--drs_k_s_thre        : Kappa sigma threshold. [10.0]
--drs_base_name       : Base name for science products. [fxb]
--drs_maxyshift       : Half width of the interval to scan for correlation, when determining y
                        shift. [3.0]
--drs_ext_w_siz       : Integration window size good: 10 (if fibre deconvolution works fine).
                        [10.0]
--rebin.wavestep      : The bin size used for BLUE/REDL data (in w.l.u.) in wavelength space. If
                        negative, a step size of 2/3 * ( average pixel size ) is used.  [-1.0]
--rebin.wavestep_redu : The bin size used for REDU data (in w.l.u.) in wavelength space. If
                        negative, a step size of 2/3 * ( average pixel size ) is used.  [-1.0]
--rebin.scale         : Whether or not to multiply by the factor dx/dlambda (pixels per wavelength)
                        during the rebinning. This option is disabled as default in concordance
                        with the method used in the MIDAS pipeline. This option should be set to
                        true to convert the observed flux (in pixel-space) to a flux per wavelength
                        (in wavelength-space). [FALSE]
--merge               : Order merging method. If 'optimal', the flux in the overlapping region is
                        set to the (optimally computed, using the uncertainties) average of single
                        order spectra. If 'sum', the flux in the overlapping region is computed as
                        the sum of the single order spectra. If flat-fielding is done, method
                        'optimal' is recommended, otherwise 'sum'. <optimal | sum> [optimal]
--merge_delt1         : Order merging left hand (short wavelength) cut. To reduce the amount of
                        order overlapping regions we allow to cut short and long wavelength ranges.
```

```
                      This may reduce the ripple possibly introduced by the order merging.
                      Suggested values are: 10 (W<=390), 12 (390<W<=437, 520<W<=564), 14
                      (437<W<=520, 564<W<860), 4 (W>=860). [-1.0]
--merge_delt2         : Order merging right hand (long wavelength) cut. To reduce the amount of
                      order overlapping regions we allow to cut short and long wavelength ranges.
                      This may reduce the ripple possibly introduced by the order merging.
                      Suggested values is 4 for W<860, else 0. [-1.0]
--clean_tmp_products  : Input data format. [FALSE]
--velocity_correction : Whether to apply velocity correction or not, and which if so <none |
                      barycentric | heliocentric> [none]
```

10.9 flames_obs_redchain

10.9.1 Input

frame tag/category	nr	filename example
BIAS_RED	+	/path_raw/uves_bias_red.fits
DARK_RED	+	/path_raw/uves_dark_red.fits
MASTER_SFLAT_<CHIPID>	1	/path_pro/set1_master_flat_<chipid>.fits
MASTER_SFLAT_<CHIPID>	1	/path_pro/set2_master_flat_<chipid>.fits
MASTER_SFLAT_<CHIPID>	1	/path_pro/set3_master_flat_<chipid>.fits
FIB_ARC_FORM_RED	1	/path_raw/flames_uves_arc_form_red.fits
FIB_ORDER_FLAT_RED	1	/path_raw/flames_uves_oflat_red.fits
FIB_FF_ODD_RED	1	/path_raw/flames_uves_odd_red.fits
FIB_FF_EVEN_RED	1	/path_raw/flames_uves_even_red.fits
FIB_FF_ALL_RED	1	/path_raw/flames_uves_all_red.fits
FIB_ARC_LAMP_RED	1	/path_raw/flames_uves_arc_lamp_red.fits
FIB_SCI_RED	1	/path_raw/flames_uves_science_red.fits
LINE_REFER_TABLE	1	/path_ref/thargood_3.fits

Note: in present release flames_obs_redchain cannot directly handle raw SFLAT frames.

10.9.2 Output

This recipe generates all the products described for the previous recipes.

10.9.3 Quality control

This recipe generates all the quality control parameters described for the previous recipes.

10.9.4 Parameters

This recipe accepts all the same parameters as the individual recipes except with the recipe name prefixing the parameter command line option, e.g. - uves_cal_mbias.clean_traps to set the clean_traps parameter for the

mbias part of the recipe. The additional parameter **scired** (default value is TRUE) switches on/off the execution of the last step (science data reduction).

```
--scired              : Whether or not to do science reduction. If false, only master calibration
                        frames are created. If false, either zero or all necessary calibration
                        frames must be provided for each arm. [TRUE]
--debug               : Whether or not to save intermediate results to local directory. [FALSE]
--plotter             : Any plots produced by the recipe are redirected to the command specified
                        by this parameter. The plotting command must contain the substring
                        'gnuplot' and must be able to parse gnuplot syntax on its standard input.
                        Valid examples of such a command may include 'gnuplot -persist' and
                        'cat > mygnuplot$$.gp'. A finer control of the plotting options can be
                        obtained by writing an executable script, e.g. my_gnuplot.pl, that executes
                        gnuplot after setting the desired gnuplot options (e.g. set terminal
                        pslatex color). To turn off plotting, set this parameter to 'no'. [no]
--process_chip        : For RED arm data proces the redl, redu, or both chip(s).
                          <both | redl | redu | REDL | REDU> [both]
--uves_cal_mbias.clean_traps  : Clean detector traps. If TRUE detector traps are interpolated.The
                                bad pixels are replaced by the average of nearest good pixels in
                                the same column, or simply marked as bad. The positions of bad
                                pixels are i hard-coded (as function of UVES chip). [TRUE]
--uves_cal_mbias.stack_method  : Method used to build master frame. <median | mean> [median]
--uves_cal_mbias.klow : Kappa used to clip low level values, when method is set to 'mean'. [5.0]
--uves_cal_mbias.khigh: Kappa used to clip high level values, when method is set to 'mean'. [5.0]
--uves_cal_mbias.niter: Number of kappa sigma iterations, when method is set to 'mean'. [5]
--uves_cal_mdark.stack_method  : Method used to build master frame. <median | mean> [median]
--uves_cal_mdark.klow : Kappa used to clip low level values, when method is set to 'mean'. [5.0]
--uves_cal_mdark.khigh: Kappa used to clip high level values, when method is set to 'mean'. [5.0]
--uves_cal_mdark.niter: Number of kappa sigma iterations, when method is set to 'mean'. [5]
--uves_cal_mdark.stack_method  : Method used to build master frame. <median | mean> [median]
--uves_cal_mdark.klow : Kappa used to clip low level values, when method is set to 'mean'. [5.0]
--uves_cal_mdark.khigh: Kappa used to clip high level values, when method is set to 'mean'. [5.0]
--uves_cal_mdark.niter: Number of kappa sigma iterations, when method is set to 'mean'. [5]
--uves_cal_mdark.qc_dark.reg.num_x  : Number of regions along the X axis (where mean/med/rms are
                                      computed). [4]
--uves_cal_mdark.qc_dark.reg.num_y  : Number of regions along the Y axis(where mean/med/rms are
                                      computed). [4]
--uves_cal_mdark.qc_dark.reg.box_sx  : Region X size [pix]. [100]
--uves_cal_mdark.qc_dark.reg.box_sy  : Region Y size [pix]. [100]
--uves_cal_mdark.qc_dark.reg.border_x  : X distance between the left hand side of the detector
                                         and the left hand side of the region [pix]. [100]
--uves_cal_mdark.qc_dark.reg.border_y  : X distance between the left hand side of the detector
                                         and the left hand side of the region [pix]. [100]
--uves_cal_mdark.qc_dark.reg.when  : When QC analysis is performed. 0: on each raw frame or 1: on
                                     the master frame. [0]
--flames_cal_predict.mbox_x  : Match box X size. [40]
--flames_cal_predict.mbox_y  : Match box Y size. [40]
--flames_cal_predict.trans_x  : Detector translation along X. [0.0]
--flames_cal_predict.trans_y  : Detector translation along Y. [0.0]
--flames_cal_predict.ech_angle_off  : Offset on echelle angle. [0.0]
--flames_cal_predict.cd_angle_off  : Offset on cross disperser angle. [0.0]
--flames_cal_predict.ccd_rot_angle_off  : Offset on CCD rotation angle. [0.0]
--flames_cal_predict.compute_regression_sw  : Compute regression?. [TRUE]
--flames_cal_predict.def_pol1  : Polynomial X deg. [4]
--flames_cal_predict.def_pol2  : Polynomial Y deg. [5]
--flames_cal_predict.kappa  : Kappa value in kappa sigma clipping on RESIDUAL between YFIT and Y
                              columns. [4.5]
--flames_cal_predict.tol  : Tolerance in kappa sigma clipping on RESIDUAL between YFIT and Y
                            columns. [2.0]
--flames_cal_orderpos.use_guess_tab  : If a Guess order table is provided this parameter set how
                                       it is used:0: No usage, 1: use it to set lower/upper Y raws
                                       where order are searched 2: the order table try to fully
                                       match the guess. [1]
--flames_cal_orderpos.radx  : Half X size of median filtering window. [2]
```

```
--flames_cal_orderpos.rady  : Half Y size of median filtering window. [1]
--flames_cal_orderpos.mmethod  : Background subtraction method. If equal to 'median' the background
                                 is sampled using the median of a sub-window. If 'minimum', the
                                 minimum sub-window value is used. If 'no', no background
                                 subtraction is done. <median | minimum | no> [median]
--flames_cal_orderpos.backsubgrid  : Number of grid points (in x- and y-direction) used to estimate
                                     the background (mode=poly). [50]
--flames_cal_orderpos.backsubradiusy  : The height (in pixels) of the background sampling window is
                                        (2*radiusy + 1). This parameter is not corrected for
                                       binning. [2]
--flames_cal_orderpos.backsubkappa  : The value of kappa in the one-sided kappa-sigma clipping used
                                      to estimate the background (mode=poly). [4.0]
--flames_cal_orderpos.backsubdegx  : Degree (in x) of polynomial used to estimate the background
                                     (mode=poly). [2]
--flames_cal_orderpos.backsubdegy  : Degree (in y) of polynomial used to estimate the background
                                     (mode=poly). [2]
--flames_cal_orderpos.samplewidth  : Separation of sample traces (used by Hough transform) in input
                                     image. [50]
--flames_cal_orderpos.minslope  : Minimum possible line slope. This should be the 'physical' slope
                                  on the chip, i.e. not taking binning factors into account, which
                                  is handled by the recipe. [0.0]
--flames_cal_orderpos.maxslope  : Maximum possible line slope. [0.2]
--flames_cal_orderpos.sloperes  : Resolution (width in pixels) of Hough space. [120]
--flames_cal_orderpos.pthres  : In automatic mode, or if the number of orders to detect is read
                                from a guess table, the detection of new lines stops when the
                                intensity of a candidate line drops to less than 'pthres' times the
                                 intensity of the previous detection. [0.2]
--flames_cal_orderpos.tracestep  : The step size used when tracing the orders. [10]
--flames_cal_orderpos.minthresh  : The minimum threshold value is (min + minthres*(max - min)).
                                   Here 'min' and 'max' are the lowest and highest pixel values in
                                   the central bin of the order. [0.2]
--flames_cal_orderpos.maxgap  : If the order line drops below detection threshold, the order
                                tracing algorithm will try to jump a gap of maximum size 'maxgap'
                                multiplied by the image width. [0.2]
--flames_cal_orderpos.maxrms  : When fitting the orders with straight lines, this is the maximum
                                allowed RMS relative to the median RMS of all orders. [100.0]
--flames_cal_orderpos.defpol1  : The degree of the bivarite fit (cross dispersion direction). If
                                 negative, the degree is optimized to give the best fit. [-1]
--flames_cal_orderpos.defpol2  : The degree of the bivarite fit (order number). If negative, the
                                 degree is optimized to give the best fit. [-1]
--flames_cal_orderpos.kappa  : Used for kappa-sigma clipping of the final polynomial fit. If
                               negative, no clipping is done. [4.0]
--uves_cal_mflat.norm_method  : Method used to build master frame. <exptime | explevel> [explevel]
--uves_cal_mflat.backsub.mmethod  : Background measuring method. If equal to 'median' the
                                    background is sampled using the median of a subwindow. If
                                    'minimum', the subwindow minimum value is used. If 'no', no
                                    background subtraction is done.
                                    <median | minimum | no> [median]
--uves_cal_mflat.backsub.npoints  : This is the number of columns in interorder space used to
                                    sample the background. [82]
--uves_cal_mflat.backsub.radiusy  : The height (in pixels) of the background sampling window is
                                    (2*radiusy + 1). This parameter is not corrected for binning.
                                    [2]
--uves_cal_mflat.backsub.sdegree  : Degree of interpolating splines. Currently only degree = 1 is
                                    supported. [1]
--uves_cal_mflat.backsub.smoothx  : If spline interpolation is used to measure the background, the
                                    x-radius of the post-smoothing window is
                                    (smoothx * image_width).
                                    Here, 'image_width' is the image width after binning. If
                                    negative, the default values are used: (25.0/4096) for blue
                                    'flat-field frames, (50.0/4096) for red flat-field frames,
                                    (300.0/4096) for blue science frames and (300.0/4096) for red
                                    science frames.  [-1.0]
--uves_cal_mflat.backsub.smoothy  : If spline interpolation is used to measure the background, the
                                    y-radius of the post-smoothing window is
```

```
                                (smoothy * image_height). Here, 'image_height' is the image
                                height after binning. If negative, the default values are used:
                                (100.0/2048) for blue flat-field frames, (300.0/2048) for
                                red flat-field frames, (200.0/2048) for blue science frames and
                                (500.0/2048) for red science frames. [-1.0]
--flames_cal_wavecal.nwindows  : Number of extraction windows per trace. The windows will be
                                 aligned (i.e. no overlap and no spacing between adjacent windows).
                                 Unless an offset is specified, the middle window(s) is centered
                                 on the trace. [1]
--flames_cal_wavecal.length  : Length (in pixels) of each extraction window. This parameter is also
                               equal to the seperation of adjacent window centers, causing the
                               extraction windows to always be aligned. The parameter is
                               automatically adjusted according to the binning of the input raw
                               frame. If negative, the extraction window length is determined
                               automatically to cover the full slit. [7.0]
--flames_cal_wavecal.offset  : A global offset (in pixels) of all extraction windows. [0.0]
--flames_cal_wavecal.extract.method  : Extraction method. (2d/optimal not supported by
                                       uves_cal_wavecal, weighted supported only by
                                       uves_cal_wavecal, 2d not supported by uves_cal_response).
                                       <average | linear | 2d | weighted | optimal> [average]
--flames_cal_wavecal.extract.kappa  : In optimal extraction mode, this is the threshold for bad
                                      (i.e. hot/cold) pixel rejection. If a pixel deviates more
                                      than kappa*sigma (where sigma is the uncertainty of the
                                      pixel flux) from the inferred spatial profile, its weight is
                                      set to zero. Range: [-1,100]. If this parameter is negative,
                                      no rejection is performed. [10.0]
--flames_cal_wavecal.extract.chunk  : In optimal extraction mode, the chunk size (in pixels) used
                                      for fitting the analytical profile (a fit of the analytical
                                      profile to single bins would suffer from low statistics).
                                      [32]
--flames_cal_wavecal.extract.profile  : In optimal extraction mode, the kind of profile to use.
                                        'gauss' gives a Gaussian profile, 'moffat' gives a Moffat
                                        profile with beta=4 and a possible linear sky contribution.
                                        'virtual' uses a virtual resampling algorithm (i.e.
                                        measures and uses the actual object profile).  'constant'
                                        assumes a constant spatial profile and allows optimal
                                        extraction of wavelength calibration frames. 'auto' will
                                        automatically select the best method based on the estimated
                                        S/N of the object. For low S/N, 'moffat' or 'gauss' are
                                        recommended (for robustness). For high S/N, 'virtual' is
                                        recommended (for accuracy). In the case of virtual
                                        resampling, a precise determination of the order positions
                                        is required; therefore the order-definition is repeated
                                        using the (assumed non-low S/N) science frame.
                                        <constant | gauss | moffat | virtual | auto> [auto]
--flames_cal_wavecal.extract.skymethod  : In optimal extraction mode, the sky subtraction method to
                                          use. 'median' estimates the sky as the median of pixels
                                          along the slit (ignoring pixels close to the object),
                                          whereas 'optimal' does a chi square minimization along
                                          the slit to obtain the best combined object and sky i
                                          levels. The optimal method gives the most accurate sky
                                          determination but is also a bit slower than the median
                                          method. <median | optimal> [optimal]
--flames_cal_wavecal.extract.oversample  : The oversampling factor used for the virtual resampling
                                           algorithm. If negative, the value 5 is used for
                                           S/N <=200, and the value 10 is used if the estimated
                                           S/N is > 200. [-1]
--flames_cal_wavecal.extract.best  : (optimal extraction only) If false (fastest), the spectrum is
                                     extracted only once. If true (best), the spectrum is extracted
                                     twice, the second time using improved variance estimates based
                                     on the first iteration. Better variance estimates slightly
                                     improve the obtained signal to noise but at the cost of
                                     increased execution time. [TRUE]
--flames_cal_wavecal.range  : Width (pix) of search window is 2*range + 1. This parameter is
                              automatically adjusted according to binning. [8]
```

```
--flames_cal_wavecal.minlines  : Minimum number of lines to detect. If zero, the default value
                                 (1100 for BLUE/REDL chips; 1000 for REDU chip) is used. [0]
--flames_cal_wavecal.maxlines  : Maximum number of lines to detect. If zero, the default value
                                 (1600 for BLUE/REDL chip; 1400 for REDU chip) is used. [0]
--flames_cal_wavecal.shiftmax  : The maximum shift (pix) in either direction compared to guess
                                 solution. This parameter is automatically corrected for binning.
                                 [10.0]
--flames_cal_wavecal.shiftstep  : The step size (pix) used when searching for the optimum shift.
                                  This parameter is automatically corrected for binning. [0.1]
--flames_cal_wavecal.shifttoler  : Tolerance (pix) when matching shifted lines. This parameter is
                                   not adjusted according to binning.  [0.05]
--flames_cal_wavecal.alpha  : The parameter that controls the distance to the nearest neighbours.
                              [0.1]
--flames_cal_wavecal.maxerror  : This parameter controls the graceful exit of the identification
                                 loop. If the RMS of the global fit exceeds this value (pix) the
                                 iteration stops. [20.0]
--flames_cal_wavecal.degree  : Degrees of the global 2d dispersion polynomial. If a negative number
                               is specified, the polynomial degrees are automatically selected by
                               starting from (1, 1) and inreasing the degrees as long as the RMS
                               residual decreases significantly. [4]
--flames_cal_wavecal.tolerance  : Tolerance of fit. If positive, the tolerance is in pixel units.
                                  If negative, abs(tolerance) is in wavelength units. Lines with
                                  residuals worse than the tolerance are excluded from the final
                                  fit. Unlike in previous versions, this parameter is not corrected
                                  for CCD binning. This rejection based on the absolute residual in
                                  pixel can be effectively disabled by setting the tolerance to a
                                  very large number (e.g. 9999). In that case outliers will be
                                  rejected using only kappa sigma clipping.  [0.6]
--flames_cal_wavecal.kappa  : Lines with residuals more then kappa stdev are rejected from the
                              final fit. [4.0]
--flames_cal_wavecal.rebin.wavestep  : The bin size used for BLUE/REDL data (in w.l.u.) in
                                       wavelength space. If negative, a step size of 2/3 * (
                                       average pixel size ) is used. [-1.0]
--flames_cal_wavecal.rebin.wavestep_redu  : The bin size used for REDU data (in w.l.u.) in
                                            wavelength space. If negative, a step size of 2/3 * (
                                            average pixel size ) is used. [-1.0]
--flames_cal_wavecal.rebin.scale  : Whether or not to multiply by the factor dx/dlambda (pixels per
                                    wavelength) during the rebinning. This option is disabled as
                                    default in concordance with the method used in the MIDAS
                                    pipeline. This option should be set to true to convert the
                                    observed flux (in pixel-space) to a flux per wavelength (in
                                    wavelength-space). [FALSE]
--flames_cal_prep_sff_ofpos.ext_method  : Extraction method. <std | opt | fst | fop> [opt]
--flames_cal_prep_sff_ofpos.bias_method  : Bias subtraction method, M for master bias frame, N for
                                           no bias subtraction, V to subtract a constant bias level
                                           defined by the parameter bias_value. <M | V | N> [M]
--flames_cal_prep_sff_ofpos.bias_value  : Bias value (only if bias_method = V). [200]
--flames_cal_prep_sff_ofpos.filter_switch  : Filter switch. <none | median> [none]
--flames_cal_prep_sff_ofpos.sat_thr  : Saturation threshold. [55000]
--flames_cal_prep_sff_ofpos.fileprep  : Slitff* and Fibreff* file preparation. If fast extraction
                                        method is used it should be set to FALSE. [TRUE]
--flames_cal_prep_sff_ofpos.cubify  : Cubify switch. [TRUE]
--flames_cal_prep_sff_ofpos.save_flat_size  : To be sure to use the flat part of a slit flatsone
                                              may need to subtract this bit. [2]
--flames_cal_prep_sff_ofpos.clean_tmp_products  : Input data format. [FALSE]
--flames_obs_scired.ext_method  : Extraction method. <std | opt | fst | fop> [opt]
--flames_obs_scired.cor_max_fnd  : Find correlation maximum?. <N | Y> [Y]
--flames_obs_scired.cor_def_rng  : Correlation range size?. [6.0]
--flames_obs_scired.cor_def_pnt  : Correlation sampling points?. [25]
--flames_obs_scired.cor_def_off  : Correlation center offset?. [0.0]
--flames_obs_scired.corvel_iter  : Velocity correlation iteration number (SimCal). [1]
--flames_obs_scired.bias_method  : Bias subtraction method. <M | V | N> [M]
--flames_obs_scired.bias_value  : Bias value (only if bias_method = V). [200]
--flames_obs_scired.cubify_sw  : Cubify switch. <Y | N> [N]
--flames_obs_scired.filt_sw  : Filter switch. <none | median> [none]
```

```
--flames_obs_scired.bkg_max_io_win  : Background window number in each full inter order. [500]
--flames_obs_scired.bkg_xy_win_sz_x  : x maximum size of each background window:. [6]
--flames_obs_scired.bkg_xy_win_sz_y  : y maximum size of each background window:. [2]
--flames_obs_scired.pixel_thresh_max  : Pixel saturation threshold max. [55000]
--flames_obs_scired.pixel_thresh_min  : Pixel saturation threshold min. [-20]
--flames_obs_scired.drs_k_s_thre  : Kappa sigma threshold. [10.0]
--flames_obs_scired.drs_base_name  : Base name for science products. [fxb]
--flames_obs_scired.drs_maxyshift  : Half width of the interval to scan for correlation, when
                                     determining y shift. [3.0]
--flames_obs_scired.drs_ext_w_siz  : Integration window size good: 10 (if fibre deconvolution
                                     works fine). [10.0]
--flames_obs_scired.rebin.wavestep  : The bin size used for BLUE/REDL data (in w.l.u.) in wavelength
                                      space. If negative, a step size of 2/3 * ( average pixel size
                                      ) is used. [-1.0]
--flames_obs_scired.rebin.wavestep_redu  : The bin size used for REDU data (in w.l.u.) in wavelength
                                           space. If negative, a step size of 2/3 * ( average pixel
                                           size ) is used. [-1.0]
--flames_obs_scired.rebin.scale  : Whether or not to multiply by the factor dx/dlambda (pixels per
                                   wavelength) during the rebinning. This option is disabled as
                                   default in concordance with the method used in the MIDAS
                                   pipeline. This option should be set to true to convert the
                                   observed flux (in pixel-space) to a flux per wavelength
                                   (in wavelength-space). [FALSE]
--flames_obs_scired.merge  : Order merging method. If 'optimal', the flux in the overlapping
                             region is set to the (optimally computed, using the uncertainties)
                             average of single order spectra. If 'sum', the flux in the overlapping
                             region is computed as the sum of the single order spectra. If
                             flat-fielding is done, method 'optimal' is recommended, otherwise
                             'sum'. <optimal | sum> [optimal]
--flames_obs_scired.merge_delt1  : Order merging left hand (short wavelength) cut. To reduce the
                                   amount of order overlapping regions we allow to cut short and
                                   long wavelength ranges. This may reduce the ripple possibly
                                   introduced by the order merging. Suggested values are:
                                   10 (W<=390), 12 (390<W<=437, 520<W<=564), 14 (437<W<=520,
                                   564<W<860), 4 (W>=860). [-1.0]
--flames_obs_scired.merge_delt2  : Order merging right hand (long wavelength) cut. To reduce the
                                   amount of order overlapping regions we allow to cut short and
                                   long wavelength ranges. This may reduce the ripple possibly
                                   introduced by the order merging. Suggested values is 4
                                   for W<860, else 0. [-1.0]
--flames_obs_scired.clean_tmp_products  : Input data format. [FALSE]
--flames_obs_scired.velocity_correction : Whether to apply velocity correction or not, and which if
                                          so <none | barycentric | heliocentric> [none]
```

11 Algorithms and recipe details

In this section we describe the main algorithms implemented in the FLAMES-UVES pipeline recipes. Relevant data reduction parameters are typed in **bold** face. Parameters that are meant to be critical for data reduction quality and recommended not to be changed unless the user really knows what is the correspondent effect, are in upper case. They are also written in the FITS header of relevant products as FITS keywords. In this release they are hardcoded. This was also true in the MIDAS based release. The chosen values demonstrated in years of operations to provide a good accuracy and robustness.They are defined in flames_def_drs_par.c module and the user may change them at own risk. Those are listed in Appendix C.

11.1 Algorithms

11.1.1 Error model

The pipeline does full error propagation using the error propagation formula and making the usual assumption about Gaussian error bars. The error of the i'th pixel of any input raw frame is defined as

$$\sigma_i = \sqrt{g(C_{i,ADU} - b_i) + ron^2 + (g^2 - 1)/12}, \quad (1)$$

where $C_{i,ADU}$ is the flux in analog-to-digital units, b_i is the master bias level and g and ron are the detector gain and read-out noise as defined in the FITS header.

11.1.2 Physical model offset setting

On the basis of the plate number, wavelength setting, CCD chip, appropriate values are set to describe the physical model offsets between the simultaneous calibration fibre and the UVES slit center. This allows to use the same physical model implementation as the one describing the echelle mode spectral format.

11.1.3 Fibre frames preparation

To each input fibre frame are associated an error and a bad pixel frame. The descriptor indicating the maximum number of fibres and which fibre is on/off are set.

This function takes as input the same odd/even fibre flat-field frames which are used for the fibre-order tracing (see 11.2.6), normalizes them (each fibre is normalized to 1 integrated flux at each x position), and selects only fibres falling within the region illuminated in the slit flat-field frames, to create a new set of odd/even fibre flat-field frames plus other auxilliary frames (normalizzation, fibre boundaries) used by subsequent DRS commands. Any bad pixels in the input fibre flat-fields are filled with interpolated values, if possible. Each fibre is then correlated with a Gaussian centered at the position traced by the order-fibre table, with a variable offset.

The variable y offset is allowed to span the intervall between -**MAXYSHIFT** and +**MAXYSHIFT**. The half width at half maximum of the Gaussian pseudo-fibres is read from the **GUASSFIBRESIGMA** keyword, and their total half width is read from the **GAUSSHALFWIDTH** keyword. The offset corresponding to the maximum correlation is saved in a keyword (**YOFFSET**), to be used as a "zero-point" correction for fibre asymmetry

when measuring actual fibre offsets on the frames. For this correction to be effective, subsequent correlations are forced to use the same Gaussian parameters (i.e. **GAUSSFIBRESIGMA** and **GAUSSHALFWIDTH**) which were used here.

11.1.4 Slit frames preparation

To each input slit frame are associated an error and a bad pixel map frames.

This function also orders the set of input slit flats by Y position and rescale them in intensity so that each frame intensity is the same as the one of the previous adjacent one where the two overlap, and creates a new, minimal set of slit flat-field frames plus other auxiliary frames (normalization, boundary data) used by subsequent data reduction commands. On each input slit flat frame, the **HALFWIDTH** and **YSHIFT** FITS keywords are set, defining respectively the half width size of the flat part of the orders and the offset between the order centres in this frame and the order centres as traced by the order-fibre table. See also 11.1.13. Of course, for relative equalisation to be possible, each frame *must* have some overlap with the neighbouring ones in the set.

11.1.5 Lit fibres flag setting

This algorithm fills the **MAXFIBRES**, **FIBREMASK** descriptors in the input frame, describing how many fibres are illuminated, and their fibre numbers. This step is executed as soon as the input fibre frames are re-oriented. The descriptors values are assigned according to which fibre is actually used as indicated by the OzPoz extention table. Only illuminated fibres are used for data reduction.

11.1.6 Inter-order background determination

This command takes as input the order table to create a table of positions where the scattered-light background may be computed, in regions not illuminated by any fibre. The number of windows in each inter-order space is set by keyword **BKG_MAX_IO_WIND** (set by default to 500). The x,y widths of each window are set by keyword **BKG_XY_WIN_SZ** (6 pixels in x, and 2 pixels in y).

- First, the order centres in the input frame are computed as the middle point between the first and last lit fibres (as read from the **FIBREMASK** FITS keyword). If this is offset with respect to the order positions mapped by the order table, the positions of the windows in the background table are corrected accordingly. Then, all the windows in the background table are trimmed to exclude any position in the input frame where light from the fibres is expected to be present. Windows which happen to be trimmed to zero size are obviously discarded at this stage.
- Step two: if the **BKGBADSCAN** keyword is set to “fraction” or “absolute”, a neighbourhood of size determined by the **BKGBADWIN** keyword of each pixel in each window left from the previous step is checked for bad pixels. Bad pixels in each neighbourhood are counted, and if they exceed the fraction set in **BKGBADMAXFRAC** (if **BKGBADSCAN**=“fraction”) or the total number set in **BKGBADMAXTOT** (if **BKGBADSCAN**=“absolute”), that pixel is removed from the window. Windows which turn out to be empty after this additional trimming are discarded. This second trimming step is skipped if **BKGBADSCAN** is set to “none”.

- Step three: a list of points to be fit is built from the input frame and the remaining windows from the steps above, in one out of four possible ways, selected by the value of the **BKGFITMETHOD** keyword. If **BKGFITMETHOD=**"all", the list consists simply of all the pixels contained in the windows. If **BKGFITMETHOD=**"median", for each window the pixel with the median value is found and added to the list. If **BKGFITMETHOD=**"minimum", for each window the pixel with the minimum value is found and added to the list. Finally, if **BKGFITMETHOD=**"average", for each window the median value is found first, all pixels in the window differing from it more than kappa_sigma_threshold times their standard deviation are discarded, and the weighted average of the remaining pixels in the window is added to the list.

  This data reduction step generates the fitted background image.

- As a fourth step, a bivariate polynomial is actually fit to the list of pixels.

- Then, the values in the list are compared to the fitted values: if any pixel in the list deviates from the fitted values more than kappa_sigma_threshold (**DRS_K_S_THRE**) times the standard deviation, at most **MAXDISCARDFRACT** times the number of pixels in the list are discarded in one pass, in order of worst fit, and the fit repeated on the remaining ones. This loop is repeated until no more pixels are rejected from the list, or until it has been repeated **MAXBACKITERS** times, whichever comes first.

  The scattered light background is estimated in a FLAMES-UVES frame, fitting a bivariate polynomial on the regions of the input frame listed in the background table. The degrees of the polynomial are read from the **BKGPOL** keyword.

- The resulting bivariate polynomial is then evaluated over the full input frame, and saved.

11.1.7 Fibre shift determination

This algorithm measures the shift of each fibre of an input frame with respect to the corresponding fibre of the odd-even base, using a set of normalised fibre flat-field frames, as produced by the fibre frame preparation (11.1.3) and updated by fibre frames normalization, (11.1.16), and using a set of equalised slit flat-field frames, as produced by the slit frames preparation (11.1.4) and an order-fibre table, as produced by flames_cal_prep_ofpos (11.2.6) and updated by the fibre frame normalization 11.1.16). The **MAXYSHIFT** keyword sets the value of the maximum y shift (in world coordinates) to be searched between the input frame and the flat-field frames. The default value of the maximum allowed yshift is set by the keyword **MAXYSHIFT**.

The measured **YSHIFT** is written into the input frame's FITS header, and measures the y shift values of each fibre of the input frame with respect to the fibres in each normalised flat-field frame.

The descriptor **NFLATS** written into the input frame containing the size of the **YSHIFT** descriptor.

The y-axis shift of the orders/fibres in the input frame is found by correlating the frame with a synthetic one composed of Gaussian-shaped pseudo-fibres centred on the positions traced by the order-fibre table, with a variable offset in the direction perpendicular to dispersion.

The size of the Gaussian shapes is forced to be the same as the one which was used to measure the "zero-point" shift measured at the fibre frames preparation (11.1.3) stage.

The correlation is computed using one pixel every **CORRELXSTEP**; thus **CORRELXSTEP=**1 means that all pixels are used, **CORRELXSTEP=**2 means that every other pixels is used and so on. The safest bet (and

the default) is **CORRELXSTEP**=1. Increasing **CORRELXSTEP** cuts the correlation times significantly, but **CORRELXSTEP** values larger than the size of the resolution element are risky: on a line emission spectrum it may happen that only the dark parts of the frame are used for the correlation, resulting in numerical instability and useless results.

The variable y offset is allowed to span the interval between -**MAXYSHIFT** and +**MAXYSHIFT**, and is referred to the fibre positions as measured on an all fibres flat-field frame by the Fiber frames normalization (see 11.1.16 command), and stored in the order table.

Note that during the science data reduction, due to the fibre illumination pattern peculiar to the given observation, the cross correlation step may not lead to a single maximum. At this purpose we allow the user to control manually that step, through parameters **cor_max_fnd cor_def_rng**, **cor_def_pnt**, **cor_def_off**. The user should check the cross correlation profile in the file cor_shape_suffix.fits (where suffix='l' or 'u'), and then set proper values of **cor_def_off** (centered to the expected correlation maximum) and **cor_def_rng** (allowing to scan correlation values near the expected maximum) and **cor_def_pnt** (this set the accuracy in the determination of the correlation maximum position).

The maximum of the correlation is determined using a Brent search with accuracy better than **CORRELTOL** and the corresponding y offset corrected for the "zero-point" shift measured at the fibre frames preparation (see 11.1.3) stage.

This final y offset is saved as a descriptor called **YSHIFT** in the input frame. Since it is referred to a unique reference position, it is duplicated as many times as the number of normalised fibre flat-field frames.

11.1.8 Generation of UVES related fibre information

This algorithm extracts from the table extension of an input raw frame the fibre information corresponding to the UVES fibre link.

This auxiliary command is used to extract important information relative to FLAMES-UVES observations contained in the binary table extensions of the input raw data.

11.1.9 Frames preparation

This data reduction step prepares frames read from disk for usage by the following steps. Its input may be a single frame or a set of frames. For each indicated frame, two new frames are created, a variance frame and a bad-pixel mask frame, respectively with suffixes _sigma.fits and _mask.fits. Their names are written into the FITS keywords **SIGMAFRAME** and **BADPXFRAME** of the input frame.

The variance and bad pixel frames are updated during the following data reduction steps.

Moreover, two additional methods to detect and flag bad pixels are provided: recursive median filtering and clipping of values out of a given validity interval.

Recursive median filtering If recursive median filtering is enabled setting the parameter **filter_switch** (of the recipe flames_cal_prep_sff_ofpos) to median, for each pixel the median of the good pixels in a neighbourhood defined by the **DRS_FILT_HW_X** and **DRS_FILT_HW_Y** keywords is computed and if the pixel value differs

by more than **DRS_K_S_THRE** times the computed sigma, the pixel is flagged as bad. This procedure is iterated until no new bad pixels are detected. While this procedure is very effective at pinpointing bad pixels, it is also computationally intensive, and thus very time consuming. For this reason it is disabled by default (**filter_switch**=none).

Threshold filtering If the **DRS_FILT_SAT_SW** is set to YES, each pixel is compared with the validity interval provided by the **DRS_PTHRE_MIN** and **DRS_PTHRE_MAX** parameters and any pixels outside this interval are flagged as bad. This is meant to quickly catch saturated pixels and/or bad values due to electronics. The cuts must be chosen keeping in mind whether the input frame was already bias-subtracted or not, and allows for normal noise, in order not to cause a spurious mass rejection of pixels.

The final bad pixel mask will thus be the merged set of all bad pixels detected by any means, or previously marked as such in a user-defined bad pixel frame whose name(s) was (were) set in the **BADPXFRAME** keyword (in a FITS file called like '*_mask.fits', for example b_sci_prefix_mask.fits, or b_odd_prefix_mask.fits, or b_even_prefix_mask.fits, or b_all_prefix_mask.fits etc..., where prefix='l' or 'u' for the lower or upper detector chip).

11.1.10 Bad pixel map preparation

The location of the traps and dead columns on the CCD image are mapped according to the instrument setting (wavelength, bin, CCD chip), as determined by examining some reference image. Those positions are hardwired in the code. In addition, all pixels above the saturation threshold set by the keyword **SAT_THRESHOLD** are marked as bad. The user may change this value modifying the parameter **sat_thr** of flames_cal_prep_sff_ofpos, and **pixel_thresh_max** in flames_obs_scired. With this information a bad-pixel image is created, that will be used in the following data reduction to mask appropriately the bad pixels.

11.1.11 Fibre order tracing

This data reduction step finds all "traces" in a set of fibre flat-field frames, in which lit fibres do not overlap, typically an even and an odd fibres frame. The **NBTRACES** keyword, if set to a value different from 0, defines the number of fibres to be detected.

The traces are found using the Hough transform technique. To improve robustness of the wavelength calibration and of the extraction the user may set with the FITS keywords **DRS_SCAN_MIN** and **DRS_SCAN_MAX**. Those keywords are defined by three values, **DRS_SCAN_MIN**=55,73,73 and **DRS_SCAN_MAX**=1993,1975, 1975, respectively for the 520, 580 and 860 wavelength setting, and are used to specify the y extension of detector region where the fibre traces are searched.

These traces are then correlated with the order positions from the first guess order table, and labelled according to order and fibre number as deduced from the **FIBREMASK** keyword in the fibre flat field frames.

The orders are then fit with a polynomial of degrees as defined in the **DEFPOL** keyword, whose component are defined, during the single fibre order tracing, assuming constant relative positions of the fibres. The polynomial coefficients are stored in the **COEFFD** FITS keyword of the output order-fibre table, while the relative position of the fibres are stored in each component of the **FIBREPOS** FITS keyword of the same table.

11.1.12 Iter order background determination

To determine the interorder background the information about the fibre location contained in the input order table is used to create a table of positions where the scattered-light background may be computed, in regions not illuminated by any fibre.

11.1.13 Measure of flat part size of Slit frames

The pipeline finds the **FWHM** across y of the illuminated part of a slit flat-field frame near frame center, finds the offset (y shift) of the order centres in this frame with respect to order positions as traced by the order table, and writes these data as keywords **YSHIFT** and **HALFWIDTH** of the input slit flat-field frame.

A cross-order trace cut is determined at the image X center. This profile is smoothed (radius=5pix) and normalized to its maximum. Then the average master flat size is determined. To this value is subtracted the value **save_flat_size**. This value is critical to the final accuracy. When the parameter has its default value, "-1", the recipe sets internally a value -2 for 580 and 860 settings (thus we actually slightly increase the size of the flat as computed on the frame) and 0 for 520 setting. The user may set different values, in the range [-3,+1] in case of not satysfying quality, in which case the recipe will use the value set by the user. The value of **HALFWIDTH** is equal to half of the computed flat width. Then is determined the flat offset with respect to ther order table trace. To obtain this value we build a synthetic flat frame of flat size equal to the one previously computed and centered on the order trace. Then cross correlating the synthetic frame and each slit flat frame we determine the corresponding shift to the order trace and store it in the **YSHIFT** keyword.

11.1.14 Slit frames preparation

This algorithm uses a set of calibration slit flat-field frames (at least 2 frames), orders them by Y shift and rescales them in intensity so that each frame intensity is the same as the one of the previous adjacent frame where the two overlap, and creates a new, minimal set of slit flat-field frames plus other auxiliary frames (containing information related to frame normalization, and slit flats boundaries) that is later on used during data reduction. On each input slit flat field frame, the **HALFWIDTH** and **YSHIFT** FITS keyword should be set as described in 11.1.13. Those define respectively the half width of the flat part of the orders and the offset between the order centres of the corresponding frame and the order centres as traced by the order-fibre table (see 11.1.13).

Of course, for relative equalisation to be possible, each frame *must* have some overlap with the neighbouring ones in the set.

11.1.15 Fiber frames preparation

The input odd/even fibre flat-field frames are normalized (each fibre is normalised to 1 integrated flux at each x position), the fibres falling within the region illuminated in the slit flat-field frames are selected, and a new set of odd/even fibre flat-field frames and other auxiliary frames (normalization, fibre boundaries) used by subsequent data reduction commands are created. Any bad pixel in the input fibre flat-fields is filled with interpolated values, if possible. Each fibre is then correlated with a syntetic Gaussian profile which peaks at the position traced by the order-fibre table plus a variable offset.

The variable y offset is allowed to span the interval between **-MAXYSHIFT** and **+MAXYSHIFT**.

The half width at half maximum of the syntetic Gaussian pseudo-fibres is read from the **GAUSSFIBRESIGMA** keyword, and their total half width is read from the **GAUSSHALFWIDTH** keyword.

The offset corresponding to the maximum correlation is saved in FITS keywords, to be used as a "zero-point" correction for fibre asymmetry when measuring actual fibre offsets on other frames. For this correction to be effective, subsequent correlations are forced to use the same Gaussian parameters (i. e. **GAUSSFIBRESIGMA** and **GAUSSHALFWIDTH**) which were used here.

11.1.16 Fiber frames normalization

The all-fibre flat-field is (according to the value chosen for the parameter **method**) optimally or standard extracted.

Background determination First, if the **BKGFITINLINE** keyword is set to "yes" (as it is by default), a polynomial background is fit to the positions in the background table bkg_prefix.fits (prefix='l' or 'u'), and subtracted from the all-fibre flat frame (see 11.1.6 for details).

Setting **BKGFITINLINE** to "no" disables inline background fit and subtraction.

Y shift determination Then the y-axis shift of the orders/fibres in the all fibre flat-field frame is found by correlating the frame with a synthetic one composed of Gaussian-shaped pseudo-fibres centred on the positions traced by the order-fibre table, with a variable offset in the direction perpendicular to dispersion. The maximum of this correlation is found, and the corresponding offset is corrected for the "zero-point" shift computed by the the fibre preparation (see 11.1.3) algorithm.

For robustness, this step perform a preliminar correlation function shape determination and a rough search for its maximum. This feature can be controlled by the keyword(s) **DRS_COR_MAX_FND**, **DRS_COR_DEF_RNG**, **DRS_COR_DEF_PNT** that are set by the corresponding flames_obs_scired parameters cor_max_fnd, cor_def_rng, and cor_def_pnt.

The y offsets of each fibre are saved as a FITS keyword called **YSHIFT** in the science frame. The correlation step can be disabled by setting the **max_shift** parameter to zero or, equivalently, setting to zero the **MAXYSHIFT** keyword and leaving the **max_shift** parameter blank. In this case, the values already contained in the **YSHIFT** descriptor are used.

Y shift correction The normalised fibre flat-field frames are then shifted to bring them to coincidence with the above offset and are multiplied by the slit flat-field frames to approximate fibre flat-field frames exactly matching the all-fibre flat-field frame. These frames are used to perform an optimal or standard extraction on the science frame, including a deconvolution of adjacent fibres (the deconvolution coefficients are computed directly from the odd/even fibre flat-fields).

Fibre spectra saving The extracted "spectra" are saved to be later used as relative normalisation factors between fibres in the subsequent extraction of science frames.

11.1.17 Fiber frames extraction

This data reduction step performs an optimal or a standard spectrum extraction. There are two supported modes: "normal" (**ext_method**=opt/std), the one suggested to the user; "fast" (**ext_method**=fop/fst), which uses input slit flat fields to clean the fibre PSFs but skips the steps of Y-shift determination and compensation (see below), a bit faster than "normal".

Then are determined the backgound and the Y shift of each fibre of the science frame to the odd-even base, and each fibre position is corrected as described in the fibre frames normalization algorithm (see 11.1.16) for the all flat frame.

Extraction. The frames determined as explained above are then used to perform an optimal extraction on the science frame, including a deconvolution of adjacent fibres (the deconvolution coefficients are computed directly from the odd/even fibre flat-fields).

Notes:

- Should the correlation step fail to detect a reasonable y shift, or if speed is of utmost importance, it is advisable to use standard extraction, disabling the correlation and reducing the integration window to the core of the fibres only, rather than optimal extraction, to minimise fibre to fibre contamination, at the price of losing a considerable fraction of the signal.
- In cases of well-behaved set of frames, however, optimal extraction gives the best signal/noise ratio, especially for faint objects. The calculated spectra are finally written on the disk on one set of files for each illuminated fibre.
- The standard extraction includes the deconvolution of adjacent fibres, being them neighbouring fibres of the same order or the first and last fibres of adjacent orders. The deconvolution coefficients are computed directly from the shifted fibre flat-field frames.
- Since in the case of fast optimal extraction no attempt at all is made to compensate for any y offset of the science frame with respect to the normalised fibre flat-field frames, this method should be selected with some caution.
- In cases in which the y offset determination is a problem, the fast extraction should be used instead, limiting the integration window to the cores of the fibres to minimize ill effects at the cost of losing a considerable fraction of the signal.

11.1.18 Fiber frames merging

First the extracted spectrum, its variance and its bad-pixel mask are rebinned.

Then the orders are optimally merged, computing every pixel in the merged spectrum as a weighted average of all good pixels available covering its wavelength bin, with weights equal to the inverse of the respective variances.

Important information is contained in the binary table extensions.

11.1.19 Interorder background subtraction

Two different methods are used to subtract the scattered light background:

- Spline method. The input image is sampled at half-integer order locations at **backsub.npoints** equally spaced sample points. According to the user defined value of the parameter **backsub.mmethod**, the median or the minimum values computed in a subwindow of height (2 * **backsub.radiusy**/biny_size + 1). The window width is given by the distance between the sample positions.

  After the spline interpolation, the interorder background image is filtered using an average filter with radius (**backsub.smoothx, backsub.smoothy**).

  The spline degree is set by using **backsub.sdegree**. Currently, only splines of degree 1 are supported (i.e. linear interpolation). If the **backsub.mmethod** parameter is set to 'no', no background subtraction is done.

- Polynomial method. The polynomial method is used in the order definition recipe because the order locations, required for the spline method, are not known at this initial stage.

  A low degree 2d polynomial is fit to a subset of the image pixels and outlier points (such as the orders themselves) that have large positive residuals are continuously rejected (one-sided kappa-sigma clipping).

  The input image is sampled on a regular grid with mask size (**image_width / backsub.npoints , image_height / backsub.npoints)**.

  After the initial order line detection, the interorder background is sampled (at locations separated by **image_width / backsub.npoints**) between the order lines.

11.1.20 Hough transform

The Hough transform is computed by sampling the input image at every column seperated by **samplewidth**.

Each echelle order maps to a peak in the Hough image. After detecting a peak, the peak itself and the area around it are deleted to avoid redetecting the same feature.

The accurate peak locations are calculated as the centroid of the area around the local maximum in the Hough space.

The function detects orders until the intensity of the next candidate drops to below a fraction **pthres** of the dimmest line.

An important parameter for the peak removal to work is the (approximate) interorder spacing.

It is estimated as the first minimum of the auto-correlation function along the column in the Hough image which contains the global maximum. This fully automatic way of detecting the orders assumes only that the interorder spacing does not vary too much as function of order number.

Possible order line slopes range from **minslope** to **maxslope** The resolution of the slope-axis in Hough space is defined by **sloperes**.

11.1.21 Order tracing

This algorithm, shared with the echelle-slit mode order tracing, is applied on fibre mode to trace the simultaneous calibration fibre order flat.

The parameter **use_guess_tab** value, defaulted to 1, allow the user to benefit of the information contained in the guess order table in a different way:

- 0: No usage.
- 1: Use the guess order table to set the lower/upper Y raws where order are searched.
- 2: The order table try to fully match the guess one.

The order tracing is performed as follows:

- First, all orders are traced in both directions starting from the center which is inferred from the solution of the Hough detection (if **use_guess_tab** is set to 0), or from the "guess" order table (if **use_guess_tab** is set to 1 or 2).

  A Gaussian is fit to the order line at x-positions seperated by the parameter **tracestep**. The trace stops if the intensity of the order line drops below the threshold defined by the **minthresh** value in an x-range determined by the parameter **maxgap**.
- Then each order is fit with a straight line, and the entire order is rejected if the RMS is large compared with the average RMS.
- A global polynomial of automatic degree is fit to all orders, and individual points are rejected using kappa-sigma clipping. Alternatively, the user can define polynomial degrees using the parameters (**defpol1, defpol2**).

11.1.22 Line Search

In fiber mode is used the same algorithm as in echelle-slit mode.

A number of emission lines defined by the range **minlines**-**maxlines** (both inclusive) is searched for in the extracted arc lamp spectrum. This is achieved by adjusting the detection threshold level until an appropriate number of lines is detected.

A 5-parameters Gaussian fit, including the continuum slope, is made if the line peak is above the treshold with respect to the local interorder background level, which is defined as the median of a window of width (2 * **range** + 1) centered on the current position. Finally, doublets (defined as lines with positions within 2.0 pixels) are removed from the set of detected lines.

11.1.23 Wavelength calibration first solution determination

In fiber mode is used the same algorithm as in echelle-slit mode.

An initial dispersion relation is obtained by fitting the relation

$$\lambda * m = f(x, m),$$

to a guess line table containing associations from (x, m) = (pixel, order) to wavelengths. Here f is a 2d polynomial.

A systematic x-shift up to **shiftmax** pixels is recovered by finding the maximum position of the cross-correlation function and applying this shift to the inital dispersion solution. The resolution of the cross-correlation function is defined by **shiftstep**. The parameter **shifttoler** defines the tolerance in pixels for the line match. The default polynomial **degree** is 5.

11.1.24 Line identification

In fiber mode is used the same algorithm as in echelle-slit mode.

The wavelength calibration starts from a first guess dispersion solution.

Lines are iteratively identified and a dispersion solution is fit, until no new identifications can be made. After the first convergence, all identifications are cleared (to remove possible false identifications), and the loop repeats, but this time ignoring lines with residuals worse than **tolerance** and lines with residuals worse than **kappa** sigma. If set to a negative value, the polynomial degree is automatically adjusted based on the line residuals.

Identifications are made based on a match between the detected line predicted wavelength, λ_{com} and a catalogue wavelength, λ_{cat}. An identification is made if

- The nearest catalogue wavelength is within two linewidths of the predicted wavelength:

  $|\lambda\_cat - \lambda\_com| < 2 * \sigma$, where σ is the detected line width,

- The distance to the 2nd nearest neighbours (in the spectrum as well as in the catalogue) is much larger than the residual of the match $|\lambda\_cat - \lambda\_nn| * ALPHA > |\lambda\_cat - \lambda\_com|$, (where **ALPHA** is a "safety parameter" less than one.)

- The nearest neighbour (in the spectrum and in the catalogue) is farther away than the average tolerance distance, which measures the precision of the identifications: $\mathbf{tolerance} < \mathbf{ALPHA} * |\lambda\_\mathbf{cat} - \lambda\_\mathbf{nn}|$. Refer to the source code for the exact definition of the $\mathbf{tolerance}$.

The purpose of the first criterion is to make the correct identifications. The purpose of the latter two criterions is to avoid making incorrect identifications.

11.2 Recipes

In the following sections we are going to describe the recipes provided by the FLAMES-UVES pipeline. Recipe common parameters are: **debug**, to activate the debug mode which saves intermediate results to disk and **plotter** to activate the plotting facility (gnuplot). Additionally each recipe can be customized by modifying specific data reduction parameters which are described in the following section or in the referred algorithm description.

11.2.1 Master bias generation

We refer the reader to the UVES context documentation for the recipe uves_cal_mbias. In fact FLAMES data reduction uses that command (see [14]).

11.2.2 Master dark generation

We refer the reader to the UVES context documentation for the recipe uves_cal_mdark. In fact FLAMES data reduction uses that command (see [14]).

11.2.3 Spectral format guess

The recipe flames_cal_predict expects as input a fibre formatcheck frame which is obtained by illuminating the calibration fibre with a ThAr lamp, and a reference ThAr lines table. Optional input is a reference formatcheck frame.

This recipe runs the UVES physical model as for ECHELLE data. For details we remind the user to [14]. The only difference is that in case of FIBER mode data, to take into account of the offset between the calibration fiber and the central position of the order, X and Y offsets are set depending on the given instrument setting (wavelength, plate number).

11.2.4 Single fibre order table generation

To get an order table from a simultaneous calibration (single) fibre flat field one can use the recipe flames_cal_orderpos providing in input a guess order table the one generated by flames_cal_predict.

11.2.5 Master slit flat field generation

The recipe flames_cal_mkmaster generates a master flat frame from a set of slit flats.

To have full coverage of all 9 fibres and wide enough windows to determine the inter order background are taken three sets of three slit flats each, each set taken at different Y positions. From each set it is computed a master flat field as for echelle mode data. The recipe divides the raw slit flat fields in groups each having the same Y position (having values of the FITS keyword ESO.INS.SLIT3.X1ENC and ESO.INS.SLIT3.X2ENC within 5 encoder steps) and next on each group applies a standard master flat field generation (without inter-order background subtyraction as this operation is performed later in the data reduction). The output are a set of (usually three) master slit flat fields each corresponding to a given Y position.

11.2.6 Slit flats preparation, odd-even-all fibre flats preparation, normalization and fibre-order table determination

The recipe flames_cal_prep_sff_ofpos prepares the input slit flat-field and fibre flat-field frames for inclusion in the calibration database. First are processed the lower-chip data and next the upper-chip data. The slit FF frames

are normalised to be equal where they overlap, then a minimal set of frames covering the largest y interval is produced.

This recipe executes the following algorithm in sequence:

- A bad pixel map is generated (see 11.1.10),
- A variance and a bad pixel map are associated to the input frames (see 11.1.9),
- All fibres are located and an order position table is determined (see 11.2.6),
- Suitable positions for background-light extraction are determined (see 11.1.12),
- Input slit flats are prepared (see 11.1.4),
- Input fibre flats are prepared (see 11.1.3),
- Fibre to fibre normalizations are computed (see 11.1.16).
- All output products are classified.

To improve robustness (of this command and flames_cal_wavecal, flames_obs_scired commands) this step checks that for each detected order all the fibres are traced and if not a fibre tracing is forced. Moreover the recipe flames_cal_prep_sff_ofpos, to prevent problems in the wavelength calibrations, excludes from the order tracing the bottom right and upper left hand side corners. This is done setting appropriate values of keywords **DRS_SCAN_MIN** and **DRS_SCAN_MAX**.

11.2.7 Wavelength Calibration

The recipe flames_cal_wavecal performs the wavelength calibration using previously determined solutions for the first guess line table and the order table. This recipe is a modification of the **uves_cal_wavecal** recipe to reduce UVES echelle data, in the sense that instead of determining the solution corresponding to tree order traces, one at the center and the other two at 15 pixels up and down, now are determined the solutions correspoinding to each of the fiber traces as indicated by the **FIBREMASK** FITS keyword. For details we remind the user to [14].

11.2.8 Science data reduction

The flames_obs_scired recipe performs the actual spectrum extraction, either optimal or standard. The normalized odd/even fibre flat-field frames output of previous commands are compared to the science frame, any y-axis shift between them is computed and compensated by shifting the odd/even fibre flat-fields; an optimal/standard extraction is performed on the science frame, and the resulting spectra are created, one set of files for each illuminated fibre. Next, using the line table created by **flames_cal_wavecal**, the extracted spectra are rebinned and merged, and the products are then classified. Finally, the products are velocity corrected and the `SPECSYS` keyword added. (The correction factor used depends on the **velocity_correction** parameter: "barycentric" uses the value from the `HIERARCH ESO DRS BARYCORR` keyword while "heliocentric" uses

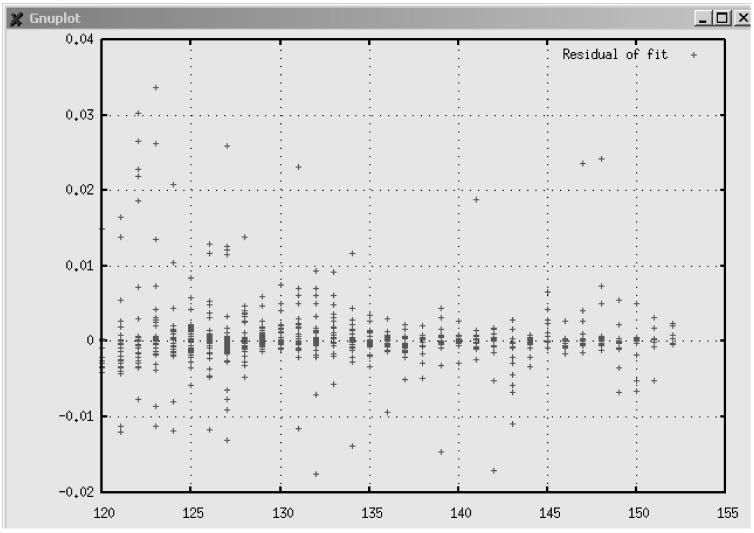


Figure 11.1: Wavelength calibration residuals (A) as function of order number.

the value from `HIERARCH ESO DRS HELICORR`. The parameter defaults to "none".) Note that the application of the correction implies an adjustment of the wavelength values. Therefore if you wish to combine the spectra of the same target, you must interpolate the spectra to a common wavelength range.

This recipe supports different extraction modes. The normal extraction mode (the one just described, using optimal or standard extraction, setting the parameter **ext_method** to “opt” or “std”) is the one we suggest to the user. Fast extraction mode (**ext_method** set to “fop” of “fst”) is faster as it skips the correlation step (but as such it may have also lower quality).

In case the correlation step fails (usually due to a large Y shift between the observation and the calibrations) we suggest the user to do the following. The shape of the correlation function is contained in tables cor_shape_x.fits (x=l or x=u respectively for EEV or MIT chips). From this table one can get the Y offset at which the correlation function has a maximum.

Next repeat the extraction appropriately setting the correlation function parameters (through parameter **ext_method**). In this case usually **cor_def_rng** can be decreased to 2 or even 1.

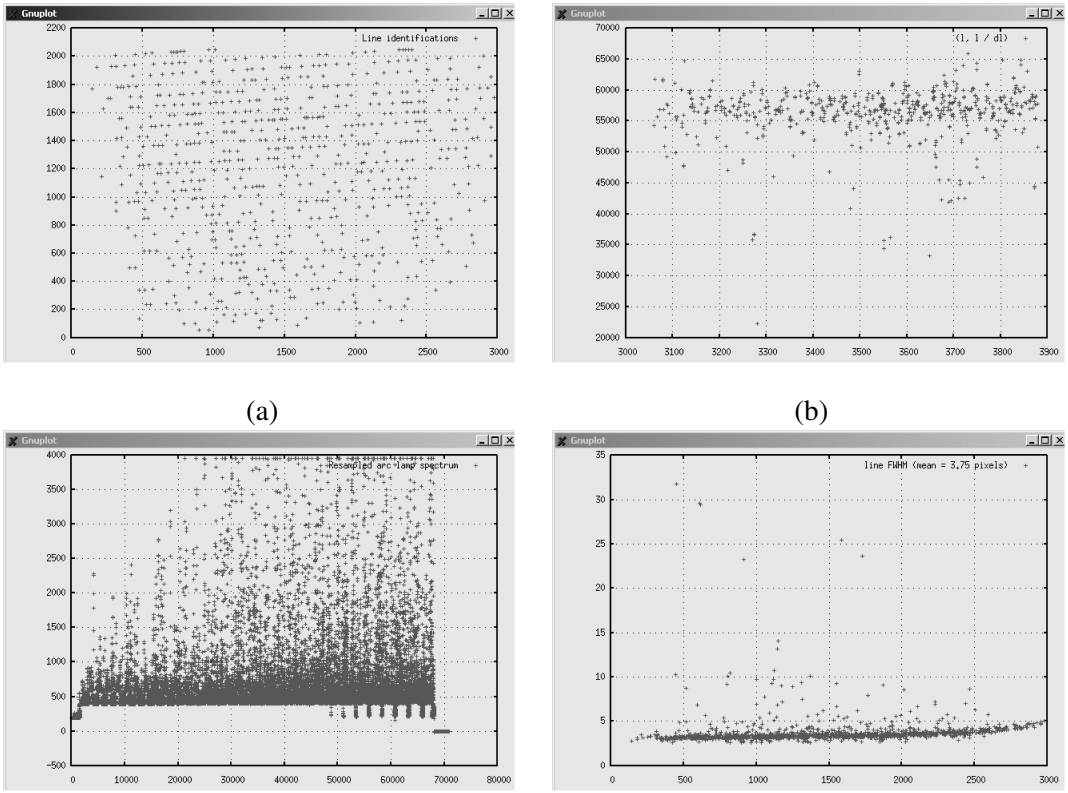
For example to do optimal extraction with correlation (OPT) function search (Y) in the range [-2,2], using 2*25+1 pixels and having measured a -4.5 pixel shift, one may give the command:

```
esorex flames_obs_scired --cor_def_pnt=25 --cor_def_rng=2 --cor_def_off=4.5 flames_obs_scired.sof
```

11.2.9 Full UVES-FIBRE data reduction

The recipe flames_obs_redchain runs the full FLAMES-UVES reduction chain. It runs in a cascade the following recipes:

- uves_cal_mbias (if no master bias is provided)



(a) (b)

(c) (d)

Figure 11.2: These plots display: (a) the distribution of identified lines across the detector, (b) the resolving power as a function of wavelength, (c) the resampled ThAr spectrum, (d) the line FWHM as a function of wavelength.

- uves_cal_mdark (if darks are provided)
- flames_cal_predict (if no guess order and line tables are provided)
- flames_cal_orderpos (if no order table is provided)
- flames_cal_mflat (if no master slit flat is provided)
- flames_cal_prep_sff_ofpos (if no slitff* and fibreff* are provided)
- flames_cal_wavecal (if no dispersion solution line table is provided)
- flames_obs_scired (unless the option **scired** is set to FALSE)

A Workflow technicalities

This section will present some detailed explanations of the FLAMES-UVES Reflex-based workflow.

A.1 Data requirements and optional inputs

In order to run the workflow with a given data set, a number of calibrations and science data must be present. The workflow functions with the concept of science data sets: a set of files that contain a coherent set of science and calibration data.

The minimum coherent fiber mode data set is composed of:

- Object(s) source(s) science frame.
- A set of bias frames that can be applied to the science, flat and arc lamp frames.
- A frame with the calibration fibre illuminated by an arc lamp to obtain a guess solution for the wavelength calibration and the order trace location of the calibration fibre.
- A frame with the calibration fibre illuminated by a flat lamp to refine the order trace location of the calibration fibre.
- A set of three frames where respectively odd or even numbered fibres or all the fibres are illuminated by a flat calibration lamp.
- A set of slit flat frames that can be applied to the fibre flat frames.
- An arc lamp frame for wavelength calibration of all illuminated fibres.
- Static calibrations: Line reference table and extinction coefficient table

The DataSetChooser shows a nice tree diagram with all the associations.

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[15] 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.

For a detailed explanation about how these calibrations are organised and associated, please refer to Section A.3.

A.2 Overall layout

The workflow structure has several parts:

[15] http://www.eso.org/sci/archive/calselectorInfo.html

- Top area. Contains annotations about the workflow and the main parameters to setup the workflow. Note that not *all* the parameters of the workflow are set here: each actor has its own parameters that are set individually.
- Bottom area. This contains all the actors that actually execute or perform a workflow action. It is also composed of several parts:
  - Initialisation actors on the left side. These actors prepare the rest of the workflow to start with the data reduction. It includes setup of required intermediate variables, the data organisation, the data selection and the routing of the data. Please note that the data organisation and selection is performed only once, while the routing is performed once for each data set that has been selected.
  - Recipe execution. The middle part of the bottom area contains the actors that execute pipeline recipes. This is basically where the logic of the data reduction chain is implemented.
  - Closing of the reduction for this data set. The right side of the workflow contains actors that perform the last actions needed after the reduction of a data set. This includes renaming of the final science data, an interactive data display (disabled by default) and some housekeeping of variables.
- Composite actors. Some of the actors are composite of other actors. That means that they contain another workflow inside. To inspect a composite actor, right-click on it, and select `Open Actor`.

A.3 OCA rules

The OCA rules are the mechanism used to Organise, Classify and Associate the data. These rules are stored in a human-readable file which contains several sections, one for each of the required tasks.

The DataOrganiser is the component that makes use of these rules. The parameter `Oca File` specifies the proper OCA rule file to use. It is advised to check that it points to the right location (although the installation procedure should take care of that). Figure A.1 shows the parameter to inspect to check that the OCA rules are ok.

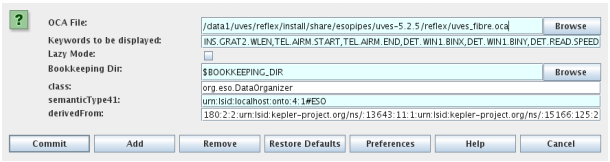


Figure A.1: *Parameter to change for the OCA file, used for classification, grouping and association of data.*

Here we present a brief summary of the meaning of the current OCA rules provided with the workflow:

- The raw data is classified according to the DPR keywords.
- The products of the recipes are classified according to the PRO.CATG keyword.
- The raw biases are grouped by detector binning and observation template.
- The raw slit flats are grouped by detector binning and observation template.

- The raw flats and wavecal frames are grouped by OzPoz or SimCal mode, observation plate id, wavelength grating, and observation template.
- The raw calibration fibre format check data and order reference data are grouped by wavelength grating.
- The science data is grouped by OzPoz or SimCal mode, observation plate id, wavelength grating, and observation template.
- All the required calibrations are directly associated to the science data. If several groups of calibration data match the rules, then the closest in time is chosen.
- The raw biases are associated if the detector binning match.
- The raw slit flats are associated if the detector binning, wavelength grating.
- The format check data and the order reference data are associated if the detector binning and the wavelength grating match.

A.4 Workflow fine tuning and hints

We have collected several hints to fine tune and exploit all the capabilities of the workflow.

- Check disk space before starting to reduce the data. The reduction of all the demo data sets will need at least 7 GB. However, at least 10 GB are recommended. This disk space requirement applies to directories pointed to by the `TMP_PRODUCTS` and `END_PRODUCTS_DIR` variables.
- Should the user find data reduction problems, the recipe parameter **debug** allows to increase recipe verbosity. Change this parameter for the recipe of interest.
- The ProductRenamer can be setup to create the desired filenames. Use the `Rename keywords` parameter to change the renaming scheme. It is possible to use keywords from the the header or literal strings (quoted by ”). Currently, there is only one ocurrence in the workflow of the ProductRenamer, just after the science recipe. However, this actor can be placed in several places in the workflow (for instance, to store in the final directory the master flat).
- All the intermediate products created by the workflow are stored in `TMP_PRODUCTS_DIR` directory. The subdirectory structure is shown in Figure A.2. For each recipe instance there is a subdirectory, and inside this there is another subdirectory with the timestamp of the execution time.

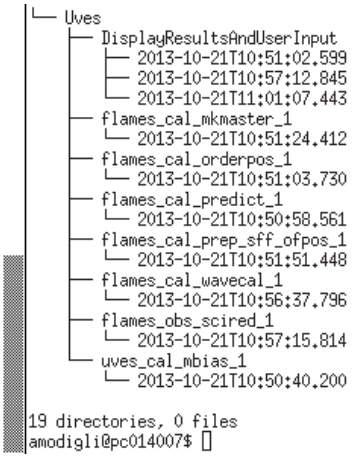


Figure A.2: *Structure of the* `REFLEX_PRODUCTS` *directory.*

B Installation

This chapter gives instructions on how to obtain, build and install the UVES pipeline (ECHELLE and FIBRE modes). Even if this chapter is kept as up-to-date as possible, it may not be fully applicable to a particular release. This might especially happen for patch releases. One is therefore advised to read the installation instructions delivered with the UVES pipeline distribution kit. These release-specific instructions can be found in the file `README` located in the top-level directory of the unpacked UVES pipeline source tree. The supported platforms are listed in Section B.1. It is recommended reading through Section B.3 before starting the installation.

ESO pipelines can be installed via several methods, depending on your OS, most of which facilitate easy installation, upgrade and removal. Please see the *ESO Data Reduction Pipelines and Workflow Systems* page (*https://www.eso.org/pipelines*).

B.1 Supported platforms

The UVES pipeline has been verified to install and execute correctly with EsoRex on the VLT target platforms:

- Linux (tested Fedora core 11 and Scientific Linux 5.5)

and on

- Mac Darwin 12.2.0 (compiling with CC=gcc)

using the GNU C compiler (version 4.4 or newer). Correct execution using Gasgano has been verified on

- Linux Fedora core 11 (with gcc4.4.1),15 (with gcc 4.6.3), 16 (with gcc4.6.3), 17 (with gcc4.7.2),
- Scientific Linux 5.5,
- Ubuntu 10.04.4
- Mac Darwin 12.2.0 (compiling with CC=gcc)

B.2 Requirements

To compile and install the UVES pipeline one needs:

- the GNU C compiler (version 4.4 or later),
- the GNU `gzip` data compression program,
- a version of the `tar` file-archiving program and
- the GNU `make` utility.

For Gasgano support one needs in addition

- the Java Development Kit (version 1.6)

B.3 Building the UVES pipeline

The UVES pipeline distribution kit contains:

uves-manual-6.6.2.pdf	The UVES pipeline manual
install_pipeline	Install script
cfitsio-3490.tar.gz	CFITSIO 3490
cpl-7.3.tar.gz	CPL 7.3
esorex-3.13.6.tar.gz	esorex 3.13.6
gasgano-2.4.8.tar.gz	GASGANO 2.4.8
uves-6.6.2.tar.gz	UVES 6.6.2
uves-calib-6.6.2.tar.gz	UVES static calibration files 6.6.2

Here is a description of the installation procedure:

1. Change directory to where you want to retrieve the UVES pipeline 6.6.2 package. It can be any directory of your choice but not:

   ```
   $HOME/gasgano
   $HOME/.esorex
   ```

2. Download from the ESO ftp server, www.eso.org/pipelines, the latest release of the UVES pipeline distribution.

3. Verify the checksum value of the tar file with the cksum command. **cksum uves-kit-6.6.2.tar.gz**

4. Unpack using the following commands:
   gunzip uves-kit-6.6.2.tar.gz tar -xvf uves-kit-6.6.2.tar

   Note that the size of the installed software (including *Gasgano*) together with the static calibration data is about 27Mb.

5. Set the environment variable JAVA_HOME to the directory where you have the JDK 1.8 installed. If this value is not set, the installation script will try to guess it, but if no JDK is found, the gasgano distribution will not be installed; CPL, EsoRex and the pipeline will be installed anyway.

6. Install: after moving to the top installation directory,
   cd uves-kit-6.6.2

   it is possible to perform a simple installation using the available installer script (*recommended*):
   ./install_pipeline

   Note: on recent Mac OS in order to properly install the kit it may be useful to set the following environment variable:

   ```
   export JAVA_HOME=/System/Library/Frameworks/JavaVM.framework/
   ```

7. Check the installation log: probably this will suggest you to set the environment variable CPLDIR and to extend your PATH.

   By default the script will install the UVES recipes, *Gasgano*, *EsoRex*, all the necessary libraries, and the static calibration tables, into a directory tree rooted at `$HOME`. A different path may be specified as soon as the script is run.

   The only exception to all this is the *Gasgano* tool, that, if you have the required proper installation of the JDK (version 1.8), will always be installed under the directory `$HOME/gasgano`. Note that the installer will move an existing `$HOME/gasgano` directory to `$HOME/gasgano.old` before the new *Gasgano* version is installed.

   Important: the installation script would ensure that any existing *Gasgano* and *EsoRex* setup would be inherited into the newly installed configuration files (avoiding in this way any conflict with other installed instrument pipelines).

C FLAMES-UVES pipeline data reduction parameters

The FLAMES-UVES pipeline gives the user the possibility to customize the data reduction. This can be done modifying the settings in the file ${PIPE_HOME}/flames/pipe/proc/flames_def_drs_par.prg

We suggest to do it with care, being sure of what one does and at his own risk. Here we describe the tuning parameters, and suggest also whether each MIDAS keyword (data reduction parameter) should or should not be changed.

- NICE_CREA: this keyword affect the display (Y) or not (N) of some frames. Default is N. Can be modified. If set to Y the data reduction is slightly slowed.
- DECENTSNR: How large should the SNR on a fibre be in a calibration frame, at a given order and x, for that slice to be considered "good"? Default: 10. Suggested to not change it. Since it is used while preprocessing the flat field calibration data, for changes to take effect one ought to rerun some or all of the relevant steps (e. g. PREPSLIT/FLAMES, PREPFIBRE/FLAMES, PRENOR/FLAMES. . .).
- MATCHTHRESH: How close should a detected fibre be to the position predicted by the zero order approximation, so that it is recognized and labeled by the matchorder step. Default 4. Suggested to not change it. Setting it too low will cause orders/fibres to be skipped, setting it too large will cause traces to be assigned to the wrong order/fibre. Only used in OFPOS/FLAMES.
- HALFIBREWIDTH: Half size of each fibre on the detector measured so that the entire size contains 100% of the flux. Default: 7.5. This parameter is only read in OFPOS/FLAMES, and its value hence stored in a descriptor and used throughout data reduction, for consistency. If one wants to change it, he should probably repeat the whole data reduction procedure for the new value to take effect (don't do it).
- MAXFIBRES: Maximum number of fibres. Default: 9. Keep it. If decreased the DRS will probably crash.
- SAV_BORD_SZ: Amount of pixels from the 50% intensity threshold of the SlitFF size which is discarded on both sides of the slitFF to be sure to be in its flat part. A large value ($\sim$5) makes you sure that only the flat part of the slitFF is used, and probably the final quality will be better; on the other hand, this will shrink the effective usable size of the used slitFF so that you need to have enough overlapping between them and enough slitFF to cover all the fibres you like to extract. A small value (2) may introduce some quality defect if the flat part of the slitFFs is not really flat. Default: 3. It may be changed if really necessary with the given advices. Only used in MSFFSZ/FLAMES. If you *really* want to tune the YSHIFT and HALFWIDTH descriptors of the slitFFs, measure them with GET/GCURSOR on a cross-dispersion cut of the frames and set the desired values manually.
- X_WIND_SIZE. Half X window size for median filter used in determining the half size of each slitFF and its offset. Default: 3. Leave it as it is. Only used in MSFFSZ/FLAMES. See the comments for SAV_BORD_SZ.
- Y_WIND_SIZE. Half Y window size for median filter used in determining the half size of each slitFF and its offset. Default: 5. Leave it as it is. Only used in MSFFSZ/FLAMES. See the comments for SAV_BORD_SZ.

- Y_SEARCH_WIND. Search window size to filter image used in determining the half size of each slitFF and its offset. Default: 100. Leave it as it is. Only used in MSFFSZ/FLAMES. See the comments for SAV_BORD_SZ.
- ORD_TRESH. Number of orders cut either on max and min detected orders. Used in determining the half size of each slitFF and its offset. We cut some orders to be sure to get Y intercepts on the cross-order cut at the detector center within the detector. Default: 2. It may be slightly increased, but it is preferable to leave it as it is. Only used in MSFFSZ/FLAMES. See the comments for SAV_BORD_SZ.
- N_CLIP_MED. K-S clipping iterations over median. Used in determining the half size of each slitFF and its offset. Default: 4. Leave it as it is. Only used in MSFFSZ/FLAMES. See the comments for SAV_BORD_SZ.
- N_CLIP_AVG. K-S clipping iterations over average. Used in determining the half size of each slitFF and its offset. Default: 2. Leave it as it is. Only used in MSFFSZ/FLAMES. See the comments for SAV_BORD_SZ.
- INT_TRESH. Signal fraction of the slitFF top value used to define the borders of the flat part of the slitFF. A value as small as 0.5 is good to be sure to detect slitFF borders. Default: 0.5. Leave it as it is. Only used in MSFFSZ/FLAMES. See the comments for SAV_BORD_SZ.
- MAXYSHIFT. This is the halfwidth of the interval of y shifts which are sampled when searching the local maximum of the correlation. Default: 3. Making it smaller may make the search for the maximum slightly faster, but it *will* cause it to fail if the actual y shift is outside this interval. Making it larger may bracket more than one maximum and cause a spurious y shift determination. Don't touch this if you don't need to.
- NBTRACES. 0. Number of traces (order times fibres). Default 0.
- MINFIBREFRAC. Minimum fibre fraction coverage for extraction. Default: 0.3. Leave it as it is.
- BKGFITNLINE. Inline background fitting. Default yes. Keep it.
- BKGFITMETHOD. Background fitting method. Possible values are ALL, MEDIAN, MINIMUM, AVERAGE. Default: AVERAGE.
- BKGBADSCAN. Background table bad pixel frame scanning switch. Possible values are FRACTION, ABSOLUTE and NONE. Default: NONE.
- BKGBADWIN. Background table bad pixel frame scanning window size. Default: 50,50
- BKGBADMAXFRAC. Background table bad pixel frame scanning threshold fraction. Default: 0.02
- BKGBADMAXTOT. Background table bad pixel scanning threshold number. Default: 200.
- BKG_MAX_IO_WIN. Number of background windows in each full inter order. Default: 500.
- DRS_BKG_FIT_POL. Polynomial degree used for BKG fit order. Default 2,2
- BKG_XY_WIN_SZ. x,y maximum size of each background window. Default 6,2.
- DRS_FILT_HW_XY. Size of X half window and Y half window. Default 2,1.

- DRS_FILT_IMAX. Maximum filtering iterations in frame preparation. Default 300.
- DRS_FILT_KS. Threshold above which pixel are flagged as bad in filtering in frame preparation. Default 10.
- DRS_FILT_SAT_SW. Do you want mask saturated pixels in frame preparation ([yes]/No)?. Default is YES.
- DRS_FILT_MASK. Do you want a filter/generated bad pixel mask: ([NONE]/MEDIAN)?. Default is NONE.
- MAXYSHIFT. Half width of the interval to scan for correlation, when determining y shift. Deafult is 3.
- GAUSSFIBRESIGMA. Gaussian pseudofibre total halfwidth for correlation. Default: 1.5. Only read in PREPFIBRE/FLAMES, then the value is stored in a table descriptor and used throughout, for consistency. Don't make it much smaller or it will make the correlation numerically fragile due to increased pixelisation effects. Don't make it too large or signals outside the fibres will spuriously affect the correlation. As a rule of the thumb, 3·GAUSSFIBRESIGMA $\lesssim$ GAUSSHALFWIDTH $\lesssim$ HALFIBREWIDTH.
- GAUSSHALFWIDTH Gaussian pseudofibre total halfwidth for correlation. Default: 6. Only read in PREPFIBRE/FLAMES, then the value is stored in a table descriptor and used throughout, for consistency. See comments above for GAUSSFIBRESIGMA.
- MAXBACKITERS This is the maximum number of kappa-sigma clipping iterations which we are willing to perform in background fitting. Default 20.
- MAXDISCARDFRACT This is the maximum fraction of windows/pixels which we are willing to discard by kappa-sigma clipping in each iteration of the background fitting loop. Default: 0.1. Making it larger will cause the background fitting to converge in less iterations, but can make it more fragile. Making it smaller will render the kappa-sigma clipping more cautious, hence it will require more iterations for all bad pixels to be discarded and convergence reached. The default chosen works well in most situations, leave it as it is.
- MAXCORRITERS. This is the maximum number of iterations which we are willing to perform in correlation. Default: 30.
- CORRELTOL. This is the absolute accuracy with which the maximum in the correlation function must be determined. Default 0.005. Leave it as it is.
- CORRELXSTEP. This is the X step to use while computing the correlation: it must be a positive integer, 1 means "use all pixels" 2 means "use every other pixel:, 3 means "use one every three" etc.. Default: 1. A larger value can make the correlation much faster, but it will also make it more fragile in case of spectra dominated by strong and narrow emission lines.
- GAUSSCORRELSCL. Obsolete (to be removed in future releases).
- GAUSSCORRELWND. Obsolete (to be removed in future releases).
- MAXCLEANITERS This is the maximum number of cleaning iterations to be tried on a given slice in flames_prep_fibre, before giving up on its normalisability and falling back to the second cleaning strategy. Default 10. Only used in PREPFIBRE/FLAMES.

- MAXSINGLEPXFRC This is the maximum acceptable fraction of the flux in a given slice in one single pixel; a higher fraction than this means that there was some numerical instability and/or unmasked bad pixel, and that it must be discarded. Default: .3. Only used in PREPFIBRE/FLAMES. Leave it as it is.
- MAXOPTITERSINT This is the maximum number of iterations which we are willing to perform in optimal extraction. Default: 25.
- MINOPTITERSINT This is the minimum number of iterations we are willing to perform in optimal extraction. Default: 2.
- XKILLSIZE
- YKILLSIZE When performing sigma-clipping in the optimal extraction, how many other adjacent pixels in the x and/or y direction(s) should be discarded along with the one exceeding threshold? A cosmic or cosmetic problem is likely to affect a spot larger than 1 pixel. Default: XKILLSIZE=1, YKILLSIZE=1. Larger values will cause a slightly faster convergence of optimal extractions, as clusters of bad pixels are removed in one go. However, some good pixels may get thrown away along with the bad ones. The default is the safe (and slow) choice.
- MAXORDER. In flames_tracing, a few arrays are statically defined, and need to be large enough to contain all the orders which could be possibly found. We defined it to a hefty one thousend. Default 1000. You will hardly ever want to touch this.
- DYRANGE Width of the y half window to be used when performing order/fibre labeling. Default: 300. It works well, leave it as it is.
- DYSTEP Step of the scan in y while performing order/fibre labelling. Default 0.1. It works well, leave it as it is.
- MASKTHRES Defines the minimum value a rebinned mask must have in order to consider the corresponding pixel of the rebinned spectrum “good”. Since a pixel in the rebinned frame may be computed using also bad pixels, we want to throw out pixels which contain even a very small fraction of “badness”. Default: 0.99. It must be a number close enough to 1 to ensure that only good pixels were used to derive a rebinned value, but not too close, to avoid throwing away good pixels just due to numerical rounding errors. The default works well, leave it as it is or, at most, tune it slightly. Used only in MERGE/FLAMES
- FRACSLICESTHRES Defines the minimum fraction of slices that must be good, for a fibre to be considered covered enough at the end of fillholes, in flames_prep_fibreff, to avoid numerical instabilities in gaussselfcorrel if a fibre is only very barely covered by the slitFF frames. Default 0.3. Leave it as it is.
- MAXROWS The maximum permitted sizes for the order tables created by the DRS: Default 300000. You will hardly ever want to touch this.
- MAXCOLS The maximum permitted sizes for the order tables created by the DRS: Default 10. You will hardly ever want to touch this.
- DRS_PTHRE_MAX Maximum threshold for pixel saturation detection: Default $6*10^4$. Good $5\text{-}6*10^4$
- DRS_PTHRE_MIN Minimum threshold for pixel saturation detection: Default -20. Good -20

- DRS_EXT_MTD Extraction method. Default opt. Possible values opt: optimal; std: standard; fop: fast-optimal; fst: fast-standard.
- DRS_K_S_THRE Kappa-sigma Threshold used in extraction. Default 10.
- DRS_KSIGMA_THRE Kappa sigma Threshold. Default 10.
- DRS_EXT_W_SIZ Integration window size. Default 10. 10 is a good value if fibre deconvolution works fine.
- DRS_MIDAS2FITS MIDAS to FITS conversion switch. At the end of a procedure it allow MIDAS to FITS format conversion. Default N. Possible value Y/N.
- DRS_BIAS_MTD Bias subtraction Method: M: Master Bias subtraction; <num> constant value subtraction; N: no bias subtraction.
- DRS_FILT_SW Filter switch. Default none. Possible values: none/average
- DRS_SFF_FIBFF Slitff*-Fibreff frames preparation. Y: yes, slower, for DFO N: no, faster, for PSO once CDB is ready. Default Y
- DRS_CREA_BP_TAB Switch to create (Y) or not (N) a bad pixels table. Default N
- DRS_PHYSMOD_REC Physical model auto recover switch: Y: yes the physical model autorecovers N: No the physical model does not autorecover. Default Y.
- DRS_USE_ORDEF Use (Y) or not (N) of the ORDERDEF as part of the odd/even fibre FF Keep it at Y to support SimCal mode. Default is N.
- DRS_MER_MTD What merging method are we using: FLAMES or ECHELLE. If ECHELLE method REDUCE/FLAMES uses REBIN/ECHELLE and MERGE/ECHELLE comands. If FLAMES method REDUCE/FLAMES uses MERGE/FLAMES. ECHELLE method is more robust.

  Default is ECHELLE.
- DRS_SCI_RAW Produce or not raw science data. Default is Y.
- DRS_VERBOSITY DRS MIDAS C Library verbosity level. Default LOW.
- DRS_MES_LEV MIDAS procedure main messaging level. Default is 4
- DRS_CUBIFY Switch to activate creation and usage of slitff* and fibreff* cubes. Default is N.
- DRS_COR_MAX_FND DRS strengthened correlation shape definition and pre search of maximum switch: Y do pre search of correlation's maximum. N don't do it (do only search of max with modified Brent method starting from points -3,0,+3). Default is Y.
- `DRS_COR_DEF_RNG` DRS Correlation function's range:

  [−`DRS_COR_DEF_RNG`,+`DRS_COR_DEF_RNG`] pix 6.
- DRS_COR_DEF_PNT DRS Correlation function's number of definition points. Effective number of points is 2*DRS_COR_DEF_PNT+1. Default is 25.

- DRS_COR_DEF_OFF DRS Correlation function's center of definition. Default is 0.
- DRS_WCAL_MODE. Wavecal mode. Default is Auto.
- DRS_WCAL_RPLT. Default for wavecal resolution plots creations. Y.
- DRS_WCAL_FITS. Defaults for FITS table production in wavecal. Default N.
- DRS_BASE_NAME. Default base name for extracted science file prefix. Default is science.
- DRS_SCI_CUBE. For packing science products on planes and cubes. Default is N.
- DRS_WCAL_DC. Default for wavecal solution polynomial degree. Originally was 5, more precise, next to have better robustness we switched to 4.
- DRS_WCAL_TOL. Default for wavelength calibration polynomial solution parameter TOL. This sets a threshold. Points which discard by the found solution more than TOL pix are outlayers. Current defaults is 0.6. A user willing to achieve a very accurate wavelength calibration solution, using the improved ThAr table (thargood_3.tfits), may decide (at the risk of a decreased wavecal recipe robustness), to decrease this value down to 0.07.
- DRS_WCAL_PAR. DRS_WCAL_PAR=DRS_WCAL_MODE, DRS_WCAL_RPLT, DRS_WCAL_FITS, DRS_WCAL_DC, DRS_WCAL_TOL
- DRS_SCI_PAR1=DRS_EXT_MTD, DRS_COR_MAX_FND, DRS_COR_DEF_RNG, DRS_COR_DEF_PNT, DRS_COR_DEF_OFF
- DRS_SCI_PAR2=DRS_BIAS_MTD,DRS_FILT_SW,DRS_PTHRE_MAX
- DRS_P8_OFPOS Parameter P8 of hogh/echelle. Default ".10,-.1,.1,.005,1.". Reasonable balues are:
  - 520 ".05,-.1,.1,.005,1."
  - 580 ".10,-.1,.1,.005,1."
  - 860 ".10,-.1,.1,.005,1."
  - DRS_P8_OFPOS_S1 First component of parameter P8 used in hogh/echelle. Default 0.05,.10,.10.
  - DRS_SCAN_MIN
  - DRS_SCAN_MAX This MIDAS keyword set the parameter SCAN(1) and SCAN(2) respectively used by the command hough/echelle. The chosen values, allow to skip a CCD portion approximatively equal to the Y span covered by eight fibres, in the prliminary determination of fibre traces done by the command OFPOS/FLAMES. This to prevent detection of very small portion of orders in the lower right hand side and upper left hand side part of a splitted frame. This help to make wavelength calibration and science extraction more robust for 860 setting. For other settings possible values are also 1,2048 which allows to cover the entire detector. We suggest the user to keep the chosen defaults: DRS_SCAL_MIN= 55,73,73 and DRS_SCAL_MAX= 1993,1975,1975 respectively for settings 520, 580, 860.
  - DRS_MER_DELTA Cutting values using in merging frames. Default 5,5.
  - DRS_DEL_SW Choice switch to set DRS_MER_DELTA values. Used in UVES echelle extraction. Possible values are D (Default: no setting), A (Automatic setting), and U (User defined setting). Default A.

- DRS_SCI_SIM Switch to activate use of SimCal setting.
- DRS_SFF_HW_MIN Threshold to minimum value accepted in Slit Flat Width Half Width. Default is 10.

D Abbreviations and acronyms

ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
CalibDB	Calibration Database
CPL	Common Pipeline Library
DFO	Data Flow Operations department
DFS	Data Flow System department
DMD	Data Management and Operations Division
DRS	Data Reduction System
ESO	European Southern Observatory
EsoRex	ESO-Recipe Execution tool
FITS	Flexible Image Transport System
FOV	Field Of View
FPN	Fixed Patter Noise
GUI	Graphical User Interface
OB	Observation Block
PSO	Paranal Science Operations
QC	Quality Control
RON	Read Out Noise
UVES	Ultraviolet Visual Echelle Spectrograph
SOF	Set Of Frames
UT	Unit Telescope
VLT	Very Large Telescope