

EUROPEAN SOUTHERN OBSERVATORY

Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral

Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

VERY LARGE TELESCOPE

Reflex MATISSE Tutorial and Cookbook

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

Issue 3.0.0

Date 13.11.2026

Prepared:	C. Hummel	13.11.2026	
	Name	Date	Signature
Approved:			
	Name	Date	Signature
Released:			
	Name	Date	Signature

This page was intentionally left blank

Change record

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
1.0	25/09/2012	All	Beta release
1.4.1	20/05/2019	All	First release
1.5.1	20/05/2020	All	Updated release
1.6.0	20/05/2021	None	New release
1.7.5	20/05/2022	None	Release able to handle GRA4MAT data
1.7.6	22/08/2022	All	Tutorial updates
1.7.7	01/04/2023	None	Plotting scripts updated
1.7.8	02/04/2023	None	None
2.0.0	17/11/2023	None	None
2.0.2	15/05/2024	None	None
2.2.0	12/03/2025	None	None
3.0.0	13/11/2026	None	Workflow parameters aligned with pipeline 3.0.0

This page was intentionally left blank

Contents

1 Introduction to `Esoreflex` 7

2 Workflow Status 8

3 Software Installation 9

3.1 Installing `Esoreflex` workflows via `macports` . . . 9

3.2 Installing `Esoreflex` workflows via `rpm/yum/dnf` . . . 9

3.3 Installing `Esoreflex` workflows via `install_esoreflex` . . . 9

4 Demo Data 11

5 Quick Start: Reducing The Demo Data 13

6 About the main `esoreflex` canvas 17

6.1 Saving And Loading Workflows . . . 17

6.2 Buttons . . . 17

6.3 Workflow States . . . 17

7 The MATISSE Workflow 18

7.1 Workflow Canvas Parameters . . . 18

7.2 Workflow Actors . . . 19

7.2.1 Simple Actors . . . 19

7.2.2 Data Organisation And Selection . . . 19

7.2.3 DataSetChooser . . . 20

7.2.4 The ProductExplorer . . . 21

7.2.5 Lazy Mode . . . 23

8 Reducing and Calibrating Your Own Science Data with Gasgano 25

8.1 An introduction to Gasgano and EsoRex . . . 25

8.1.1 Using Gasgano . . . 25

8.1.2 Using EsoRex . . . 29

9 Reducing and Calibrating Your Own Science Data with Reflex **32**

9.1 Available Reflex workflows 32

9.2 Checking the status of your calibrator in the JSDC 33

9.3 Specifying data directories and selecting files 33

9.4 Interactive plots during workflow execution 33

9.5 Examining the workflow results 36

9.6 Calibrating your visibility data 36

10 Frequently Asked Questions **40**

1 Introduction to `Esoreflex`

This document is a tutorial designed to enable the user to to reduce his/her data with the ESO pipeline run under an user-friendly environment, called `EsoReflex`, concentrating on high-level issues such as data reduction quality and signal-to-noise (S/N) optimisation.

`EsoReflex` is the ESO Recipe Flexible Execution Workbench, an environment to run ESO VLT pipelines which employs a workflow engine to provide a real-time visual representation of a data reduction cascade, called a workflow, which can be easily understood by most astronomers. The basic philosophy and concepts of Reflex have been discussed by Freudling et al. (2013A&A...559A..96F). Please reference this article if you use Reflex in a scientific publication.

Reflex and the data reduction workflows have been developed by ESO and instrument consortia and they are fully supported. If you have any issue, please have a look to https://support.eso.org to see if this has been reported before or open a ticket for further support.

A workflow accepts science and calibration data, as downloaded from the archive using the CalSelector tool[1] (with associated raw calibrations) and organises them into DataSets, where each DataSet contains one science object observation (possibly consisting of several science files) and all associated raw and static calibrations required for a successful data reduction. The data organisation process is fully automatic, which is a major time-saving feature provided by the software. The DataSets selected by the user for reduction are fed to the workflow which executes the relevant pipeline recipes (or stages) in the correct order. Full control of the various recipe parameters is available within the workflow, and the workflow deals automatically with optional recipe inputs via built-in conditional branches. Additionally, the workflow stores the reduced final data products in a logically organised directory structure employing user-configurable file names.

This tutorial deals with the reduction of MATISSE observations only via the MATISSE `Reflex` workflow. For more detail on the pipeline, the user is referred to the pipeline manual and the MATISSE user manual [3] and to the ESO instrument web pages[2] for more information on the instrument itself as well as a summary of available documentation, recent news, and tools. The cookbook aspects derive from further guidance on MATISSE data reduction and analysis in general.

The quick start section (see Section 5) describes the minimum effort to get started, and it makes up only two pages of text in this tutorial. User support for this software is available by sending enquiries to `https://support.eso.org`.

[1] *https://www.eso.org/sci/archive/calselectorInfo.html*
[2] http://www.eso.org/sci/facilities/paranal/instruments/matisse.html

2 Workflow Status

The MATISSE Reflex workflow, in its current version, is capable, together with its underlying MATISSE pipeline, of delivering calibrated data products. The Reflex workflows are built upon the MATISSE pipeline.

The first task carried out by the MATISSE Reflex workflow is to organise the data of this instrument into an associated, organised, and classified structure including for each science or interferometric calibrator file the required instrument calibration files with matching spectral resolution and integration time. The user will be warned if any calibration frames are missing.

The MATISSE Reflex workflows, *matisse_wkf_LM* and *matisse_wkf_N*, will reduce the raw LM and N frames, respectively, and compute estimates for fluxes (total and correlated), and integrate the data incoherently and coherently. The transfer function needed to calibrate the science data is computed from the reduced calibrator data and its angular diameter. A number of data products are created and retained for the user to assess the quality of the pipeline processing.

Calibrating the reduced science observations is done with a separate workflow, *matisse_viscal*. This workflow will create an OIFITS file with the calibrated data.

During the processing within the Reflex workflow, the user has the ability to modify a number of pipeline parameters in order to optimise the data processing.

During the pipeline development, the pipeline parameters have been set to default values that deliver the best results for each band in most cases. However, the user should make an effort to adjust and experiment with the parameters to optimize the results.

3 Software Installation

`Esoreflex` and the workflows can be installed in different ways: via package repositories, via the `install_esoreflex` script or manually installing the software tar files.

The recommended way is to use the package repositories if your operating system is supported. The pipelines and Reflex can be installed from the ESO `macports` repositories that support macOS platforms, the and the `rpm/yum` repositories that support Fedora and CentOS platforms. For any other operating system it is recommended to use the `install_esoreflex` script.

The installation from package repository requires administrative privileges (typically granted via sudo), as it installs files in system-wide directories under the control of the package manager. If you want a local installation, or you do not have sudo privileges, or if you want to manage different installations on different directories, then use the `install_esoreflex` script. Note that the script installation requires that your system fulfill several software prerequisites, which might also need sudo privileges.

Please note that in case of major or minor (affecting the first two digit numbers) Reflex upgrades, the user should erase the `$HOME/KeplerData`, `$HOME/.kepler` directories if present, to prevent possible aborts (i.e. a hard crash) of the `esoreflex` process.

3.1 Installing `Esoreflex` workflows via `macports`

This method is supported for the macOS operating system. It is assumed that macports (*https://www.macports.org*) is installed. Please read the full documentation at *https://www.eso.org/sci/software/pipelines/installation/macports.html*, which also describes the versions of macOS that are currently supported.

3.2 Installing `Esoreflex` workflows via `rpm/yum/dnf`

This method is supported for Fedora and CentOS platforms and requires sudo rights. Please read the full documentation at *https://www.eso.org/sci/software/pipelines/installation/rpm.html*, which also describes the versions of Fedora and CentOS that are currently supported.

3.3 Installing `Esoreflex` workflows via `install_esoreflex`

This method is recommended for operating systems other than those indicated above, or if the user has no sudo rights. Software dependencies are not fulfilled by the installation script, therefore the user has to install all the prerequisites before running the installation script.

The software pre-requisites for `Reflex 2.10` may be found at:
https://www.eso.org/sci/software/pipelines/reflex_workflows

To install the `Reflex 2.10` software and demo data, please follow these instructions:

1. From any directory, download the installation script:

```
wget https://eso.org/sci/software/pipelines/install_esoreflex
```

2. Make the installation script executable:

```
chmod u+x install_esoreflex
```

3. Execute the installation script:

```
./install_esoreflex
```

and the script will ask you to specify three directories: the download directory <download_dir>, the software installation directory <install_dir>, and the directory to be used to store the demo data <data_dir>. If you do not specify these directories, then the installation script will create them in the current directory with default names.

4. Follow all the script instructions; you will be asked whether to use your Internet connection (recommended: yes), the pipelines and demo-datasets to install (note that the installation will remove all previously installed pipelines that are found in the same installation directory).

5. To start Reflex, issue the command:

```
<install_dir>/bin/esoreflex
```

It may also be desirable to set up an alias command for starting the Reflex software, using the shell command alias. Alternatively, the PATH variable can be updated to contain the <install_dir>/bin directory.

4 Demo Data

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

Note that you will need a minimum of ∼14 GB of free disk space for the directory `<data_dir>` for the raw science data and the processed calibrations. An additional space of 25 GB is also needed for the temporary and final pipeline products.

The raw input data (taken during Science Verification) consist of OBJ-SKY sequences of:

1. two calibrator OBs (ϵ Sco and γ 2 Nor), and
2. two science OBs (AH Sco).

The data set also includes the detector calibrations (BADPIX, NONLINEARITY, and SHIFT_MAP), as well as the KAPPA_MATRIX and OBS_FLATFIELD calibration files. If downloading the data from the archive (programme ID 60.A-9272(A), observing date 2019-05-10, files after UT 03:32:00), it is possible to select only the science frames, and in the next step select the option "Selected files + associated processed calibrations". This will download all calibrations with the raw science data plus the calibrator data processed by ESO QC (PRO.CATG: CALIB_RAW_INT), which includes the transfer function, thus allowing to skip the raw data processing of the calibrator data. If the observed calibrator (and hence its diameter) is not in the calibrator database, it is recommended to also download the raw data for the calibrator. The raw tutorial data sets are summarized in Table 4.1. The list of files as shown in the table can be obtained by executing the following command (part of the ESO SciSoft collection) in the demo data directory:

```
dfits *.fits | fitsort DATE-OBS DET.NAME OBJECT OBS.TARG.NAME PRO.CATG \
INS.BCD1.ID INS.BCD2.ID ISS.CHOP.ST OBS.CONTAINER.ID
```

Please note that if the data are downloaded from the archive, the successfully executed concatenations start after 03:30 UT. The data set consists of two concatenations, one with a calibrator for the L band, and another for the N band. The reason is that in this case, a calibrator bright in both band could not be found nearby the science target. The pipeline will automatically associate the corresponding data sets based on their container IDs.

Table 4.1: The MATISSE Reflex workflow tutorial data set, CalSelector "processed calibrations"

FILE	DATE-OBS	DET	OBJECT	NAME	PRO.CATG	BCD1	BCD2	CHOP	CONTAINER
MATIS.2019-05-11T03:32:05.932.fits	2019-05-11T03:32:05.9317	LM	SKY	AHSco		IN	IN	F	1808
MATIS.2019-05-11T03:32:07.071.fits	2019-05-11T03:32:07.0705	N	SKY	AHSco		IN	IN	F	1808
MATIS.2019-05-11T03:32:56.778.fits	2019-05-11T03:32:56.7774	LM	SKY	AHSco		OUT	OUT	F	1808
MATIS.2019-05-11T03:32:57.280.fits	2019-05-11T03:32:57.2794	N	SKY	AHSco		OUT	OUT	F	1808
MATIS.2019-05-11T03:34:28.526.fits	2019-05-11T03:34:28.5258	LM	AHSco	AHSco		IN	IN	F	1808
MATIS.2019-05-11T03:34:29.049.fits	2019-05-11T03:34:29.0491	N	AHSco	AHSco		IN	IN	F	1808
MATIS.2019-05-11T03:35:54.108.fits	2019-05-11T03:35:54.1072	LM	AHSco	AHSco		OUT	IN	F	1808
MATIS.2019-05-11T03:35:54.692.fits	2019-05-11T03:35:54.6924	N	AHSco	AHSco		OUT	IN	F	1808
MATIS.2019-05-11T03:37:16.277.fits	2019-05-11T03:37:16.2762	LM	AHSco	AHSco		IN	OUT	F	1808
MATIS.2019-05-11T03:37:16.782.fits	2019-05-11T03:37:16.7815	N	AHSco	AHSco		IN	OUT	F	1808
MATIS.2019-05-11T03:38:42.188.fits	2019-05-11T03:38:42.1881	LM	AHSco	AHSco		OUT	OUT	F	1808
MATIS.2019-05-11T03:38:42.692.fits	2019-05-11T03:38:42.6919	N	AHSco	AHSco		OUT	OUT	F	1808
MATIS.2019-05-11T03:40:53.993.fits	2019-05-11T03:40:53.9935	LM	AHSco	AHSco		IN	IN	T	1808
MATIS.2019-05-11T03:40:54.470.fits	2019-05-11T03:40:54.4702	N	AHSco	AHSco		IN	IN	T	1808
MATIS.2019-05-11T03:42:08.840.fits	2019-05-11T03:42:08.8401	LM	AHSco	AHSco		IN	IN	T	1808
MATIS.2019-05-11T03:42:09.247.fits	2019-05-11T03:42:09.2469	N	AHSco	AHSco		IN	IN	T	1808
MATIS.2019-05-11T03:43:24.079.fits	2019-05-11T03:43:24.0785	LM	AHSco	AHSco		IN	IN	T	1808
MATIS.2019-05-11T03:43:24.579.fits	2019-05-11T03:43:24.5792	N	AHSco	AHSco		IN	IN	T	1808
MATIS.2019-05-11T03:44:39.267.fits	2019-05-11T03:44:39.2676	LM	AHSco	AHSco		IN	IN	T	1808
MATIS.2019-05-11T03:44:39.737.fits	2019-05-11T03:44:39.7370	N	AHSco	AHSco		IN	IN	T	1808
MATIS.2019-05-11T03:46:03.278.fits	2019-05-11T03:46:03.2789	LM	AHSco	AHSco		OUT	OUT	T	1808
MATIS.2019-05-11T03:46:03.713.fits	2019-05-11T03:46:03.7135	N	AHSco	AHSco		OUT	OUT	T	1808
MATIS.2019-05-11T03:47:17.872.fits	2019-05-11T03:47:17.8722	LM	AHSco	AHSco		OUT	OUT	T	1808
MATIS.2019-05-11T03:47:18.378.fits	2019-05-11T03:47:18.3781	N	AHSco	AHSco		OUT	OUT	T	1808
MATIS.2019-05-11T03:48:33.152.fits	2019-05-11T03:48:33.1519	LM	AHSco	AHSco		OUT	OUT	T	1808
MATIS.2019-05-11T03:48:33.644.fits	2019-05-11T03:48:33.6438	N	AHSco	AHSco		OUT	OUT	T	1808
MATIS.2019-05-11T03:49:50.273.fits	2019-05-11T03:49:50.2728	LM	AHSco	AHSco		OUT	OUT	T	1808
MATIS.2019-05-11T03:49:50.746.fits	2019-05-11T03:49:50.7462	N	AHSco	AHSco		OUT	OUT	T	1808
MATIS.2019-05-11T04:23:55.573.fits	2019-05-11T04:23:55.5728	LM	SKY	AHSco		IN	IN	F	7225
MATIS.2019-05-11T04:23:56.657.fits	2019-05-11T04:23:56.6567	N	SKY	AHSco		IN	IN	F	7225
MATIS.2019-05-11T04:24:42.869.fits	2019-05-11T04:24:42.8687	LM	SKY	AHSco		OUT	OUT	F	7225
MATIS.2019-05-11T04:24:43.379.fits	2019-05-11T04:24:43.3796	N	SKY	AHSco		OUT	OUT	F	7225
MATIS.2019-05-11T04:26:17.450.fits	2019-05-11T04:26:17.4499	LM	AHSco	AHSco		IN	IN	F	7225
MATIS.2019-05-11T04:26:17.920.fits	2019-05-11T04:26:17.9200	N	AHSco	AHSco		IN	IN	F	7225
MATIS.2019-05-11T04:27:42.087.fits	2019-05-11T04:27:42.0868	LM	AHSco	AHSco		OUT	IN	F	7225
MATIS.2019-05-11T04:27:42.548.fits	2019-05-11T04:27:42.5477	N	AHSco	AHSco		OUT	IN	F	7225
MATIS.2019-05-11T04:29:03.200.fits	2019-05-11T04:29:03.1995	LM	AHSco	AHSco		IN	OUT	F	7225
MATIS.2019-05-11T04:29:03.693.fits	2019-05-11T04:29:03.6932	N	AHSco	AHSco		IN	OUT	F	7225
MATIS.2019-05-11T04:30:28.101.fits	2019-05-11T04:30:28.1008	LM	AHSco	AHSco		OUT	OUT	F	7225
MATIS.2019-05-11T04:30:28.574.fits	2019-05-11T04:30:28.5743	N	AHSco	AHSco		OUT	OUT	F	7225
MATIS.2019-05-11T04:32:35.636.fits	2019-05-11T04:32:35.6358	LM	AHSco	AHSco		IN	IN	T	7225
MATIS.2019-05-11T04:32:36.105.fits	2019-05-11T04:32:36.1048	N	AHSco	AHSco		IN	IN	T	7225
MATIS.2019-05-11T04:33:52.134.fits	2019-05-11T04:33:52.1344	LM	AHSco	AHSco		IN	IN	T	7225
MATIS.2019-05-11T04:33:52.611.fits	2019-05-11T04:33:52.6110	N	AHSco	AHSco		IN	IN	T	7225
MATIS.2019-05-11T04:35:06.183.fits	2019-05-11T04:35:06.1828	LM	AHSco	AHSco		IN	IN	T	7225
MATIS.2019-05-11T04:35:06.582.fits	2019-05-11T04:35:06.5816	N	AHSco	AHSco		IN	IN	T	7225
MATIS.2019-05-11T04:36:21.618.fits	2019-05-11T04:36:21.6186	LM	AHSco	AHSco		IN	IN	T	7225
MATIS.2019-05-11T04:36:22.131.fits	2019-05-11T04:36:22.1310	N	AHSco	AHSco		IN	IN	T	7225
MATIS.2019-05-11T04:37:44.373.fits	2019-05-11T04:37:44.3731	LM	AHSco	AHSco		OUT	OUT	T	7225
MATIS.2019-05-11T04:37:44.848.fits	2019-05-11T04:37:44.8483	N	AHSco	AHSco		OUT	OUT	T	7225
MATIS.2019-05-11T04:39:00.102.fits	2019-05-11T04:39:00.1016	LM	AHSco	AHSco		OUT	OUT	T	7225
MATIS.2019-05-11T04:39:00.608.fits	2019-05-11T04:39:00.6079	N	AHSco	AHSco		OUT	OUT	T	7225
MATIS.2019-05-11T04:40:15.221.fits	2019-05-11T04:40:15.2206	LM	AHSco	AHSco		OUT	OUT	T	7225
MATIS.2019-05-11T04:40:15.692.fits	2019-05-11T04:40:15.6921	N	AHSco	AHSco		OUT	OUT	T	7225
MATIS.2019-05-11T04:41:32.494.fits	2019-05-11T04:41:32.4947	LM	AHSco	AHSco		OUT	OUT	T	7225
MATIS.2019-05-11T04:41:32.963.fits	2019-05-11T04:41:32.9636	N	AHSco	AHSco		OUT	OUT	T	7225
M.MATISSE.2019-04-02T10:05:15.326.fits						JSDC_CAT			
M.MATISSE.2019-07-26T06:15:31.810.fits	2019-07-06T13:03:12.4442	LM	DARK		SHIFT_MAP	OUT	OUT		
M.MATISSE.2019-07-26T06:15:35.236.fits	2019-07-06T14:11:24.9258	N	DARK		SHIFT_MAP	OUT	OUT		
M.MATISSE.2019-08-08T05:40:24.570.fits	2019-07-05T17:17:08.8958	LM	FLAT		BADPIX	IN	IN		
M.MATISSE.2019-08-08T05:40:26.393.fits	2019-07-05T20:26:40.3898	N	DARK		BADPIX	IN	IN		
M.MATISSE.2019-08-08T05:40:46.110.fits	2019-07-05T17:17:08.8958	LM	FLAT		NONLINEARITY	IN	IN		
M.MATISSE.2019-08-08T05:40:50.463.fits	2019-07-05T20:26:40.3898	N	DARK		NONLINEARITY	IN	IN		
M.MATISSE.2019-08-08T09:00:20.016.fits	2019-05-06T10:29:46.4618	LM	KAPPA		KAPPA_MATRIX	OUT	OUT		
M.MATISSE.2019-08-08T09:23:13.620.fits	2019-05-11T03:57:52.8194	LM	STD	epsSco	CALIB_RAW_INT	IN	IN	F	1808
M.MATISSE.2019-08-08T09:23:17.863.fits	2019-05-11T03:59:18.5824	LM	STD	epsSco	CALIB_RAW_INT	OUT	IN	F	1808
M.MATISSE.2019-08-08T09:23:20.000.fits	2019-05-11T04:00:39.7335	LM	STD	epsSco	CALIB_RAW_INT	IN	OUT	F	1808
M.MATISSE.2019-08-08T09:23:22.013.fits	2019-05-11T04:02:01.6069	LM	STD	epsSco	CALIB_RAW_INT	OUT	OUT	F	1808
M.MATISSE.2019-08-08T09:23:34.916.fits	2019-05-11T03:57:53.3311	N	STD	epsSco	CALIB_RAW_INT	IN	IN	F	1808
M.MATISSE.2019-08-08T09:23:45.503.fits	2019-05-11T03:59:19.0644	N	STD	epsSco	CALIB_RAW_INT	OUT	IN	F	1808
M.MATISSE.2019-08-08T09:23:50.000.fits	2019-05-11T04:00:40.2132	N	STD	epsSco	CALIB_RAW_INT	IN	OUT	F	1808
M.MATISSE.2019-08-08T09:23:55.740.fits	2019-05-11T04:02:02.1030	N	STD	epsSco	CALIB_RAW_INT	OUT	OUT	F	1808
M.MATISSE.2019-08-08T09:24:21.600.fits	2019-05-11T04:49:33.8989	LM	STD	gam02Nor	CALIB_RAW_INT	IN	IN	F	7225
M.MATISSE.2019-08-08T09:24:23.643.fits	2019-05-11T04:50:55.5269	LM	STD	gam02Nor	CALIB_RAW_INT	OUT	IN	F	7225
M.MATISSE.2019-08-08T09:24:24.000.fits	2019-05-11T04:52:19.7118	LM	STD	gam02Nor	CALIB_RAW_INT	IN	OUT	F	7225
M.MATISSE.2019-08-08T09:24:25.386.fits	2019-05-11T04:53:46.6168	LM	STD	gam02Nor	CALIB_RAW_INT	OUT	OUT	F	7225
M.MATISSE.2019-08-08T09:24:29.550.fits	2019-05-11T04:49:34.3752	N	STD	gam02Nor	CALIB_RAW_INT	IN	IN	F	7225
M.MATISSE.2019-08-08T09:24:33.613.fits	2019-05-11T04:50:55.9861	N	STD	gam02Nor	CALIB_RAW_INT	OUT	IN	F	7225
M.MATISSE.2019-08-08T09:24:35.000.fits	2019-05-11T04:52:20.1765	N	STD	gam02Nor	CALIB_RAW_INT	IN	OUT	F	7225
M.MATISSE.2019-08-08T09:24:37.646.fits	2019-05-11T04:53:47.1030	N	STD	gam02Nor	CALIB_RAW_INT	OUT	OUT	F	7225
M.MATISSE.2019-08-08T09:27:58.993.fits	2019-05-11T10:12:20.8126	LM	FLAT,OFF		OBS_FLATFIELD	OUT	OUT		
M.MATISSE.2019-08-08T09:28:00.840.fits	2019-05-11T10:11:24.2940	N	FLAT,OFF		OBS_FLATFIELD	OUT	OUT		

5 Quick Start: Reducing The Demo Data

For the user who is keen on starting reductions without being distracted by detailed documentation, we describe the steps to be performed to reduce the science data provided in the MATISSE demo data set supplied with the `esoreflex 2.10` release. By following these steps, the user should have enough information to perform a reduction of his/her own data without any further reading:

1. First, type:

   ```
   esoreflex -l
   ```

   If the `esoreflex` executable is not in your path, then you have to provide the command with the executable full path `<install_dir>/bin/esoreflex -l` . For convenience, we will drop the reference to `<install_dir>`. A list with the available `esoreflex` workflows will appear, showing the workflow names and their full path.

2. Open the `matisse` by typing:

   ```
   esoreflex matisse&
   ```

   Alternatively, you can type only the command `esoreflex` the empty canvas will appear (Figure 5.1) and you can select the workflow to open by clicking on `File -> Open File`. Note that the loaded workflow will appear in a new window. The `matisse` workflow is shown in Figure 5.2.

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. Change directories set-up. Under "Setup Directories" in the workflow canvas there are seven parameters that specify important directories (green dots).

   By default, the `ROOT_DATA_DIR`, which specifies the working directory within which the other directories are organised. is set to your `$HOME/reflex_data` directory. All the temporary and final products of the reduction will be organized under sub-directories of `ROOT_DATA_DIR`, therefore make sure this parameter points to a location where there is enough disk space. To change `ROOT_DATA_DIR`, double click on it 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.

   Changing the value of `RAW_DATA_DIR` is the only necessary modification if you want to process data other than the demo data

5. Click the  button to start the workflow

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

in `RAW_DATA_DIR` or in `CALIB_DATA_DIR`, otherwise datasets may be incomplete and cannot be processed. However, if the same reference file was downloaded twice to different places this creates a problem as `esoreflex` 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 5.3) that lists the datasets along with the values of a selection of useful header keywords[3]. The first column consists of a set of tick boxes which allow the user to select the datasets to be processed. By default all complete datasets which have not yet been reduced will be selected. A full description of the options offered by the `Data Set Chooser` will be presented in Section 7.2.3.

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. Once the reduction of all datasets has finished, a pop-up window called *Product Explorer* will appear, showing the datasets which have been reduced together with the list of final products. This actor allows the user to inspect the final data products, as well as to search and inspect the input data used to create any of the products of the workflow. Figure 5.4 shows the Product Explorer window. A full description of the *Product Explorer* will be presented in Section 7.2.4.

10. After the workflow has finished, all the products from all the datasets can be found in a directory under `END_PRODUCTS_DIR` named after the workflow start timestamp. Further subdirectories will be found with the name of each dataset.

Well done! You have successfully completed the quick start section and you should be able to use this knowledge to reduce your own data. However, there are many interesting features of `Reflex` and the MATISSE workflow that merit a look at the rest of this tutorial.

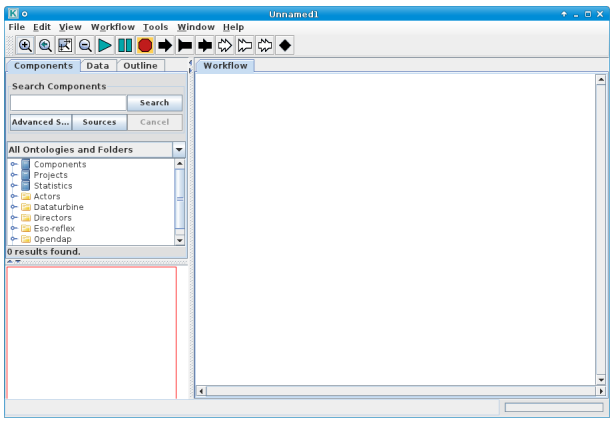


Figure 5.1: *The empty* `Reflex` *canvas.*

[3]The keywords listed can be changed by double clicking on the `DataOrganiser` Actor and editing the list of keywords in the second line of the pop-up window. Alternatively, instead of double-clicking, you can press the right mouse button on the `DataOrganiser` Actor and select `Configure Actor` to visualize the pop-up window.

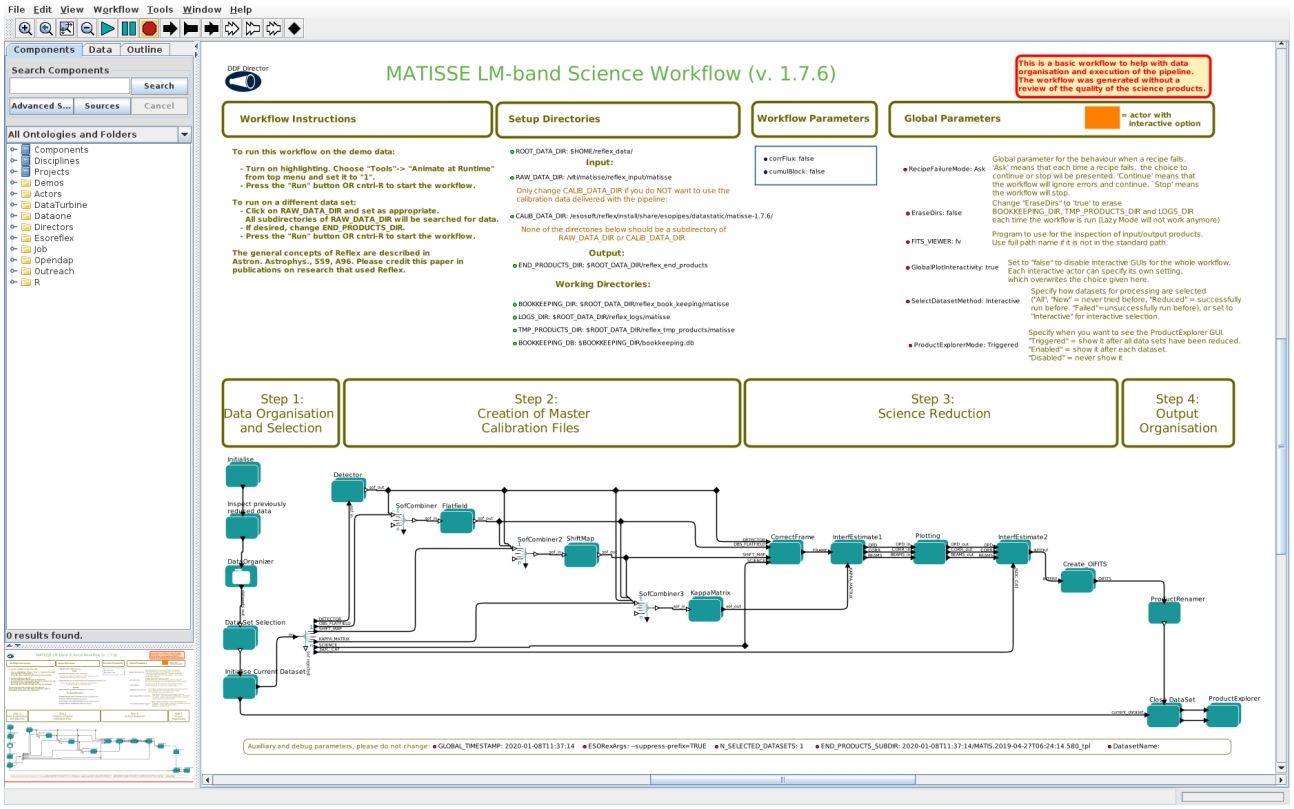


Figure 5.2: *matisse_wkf_LM workflow layout.*

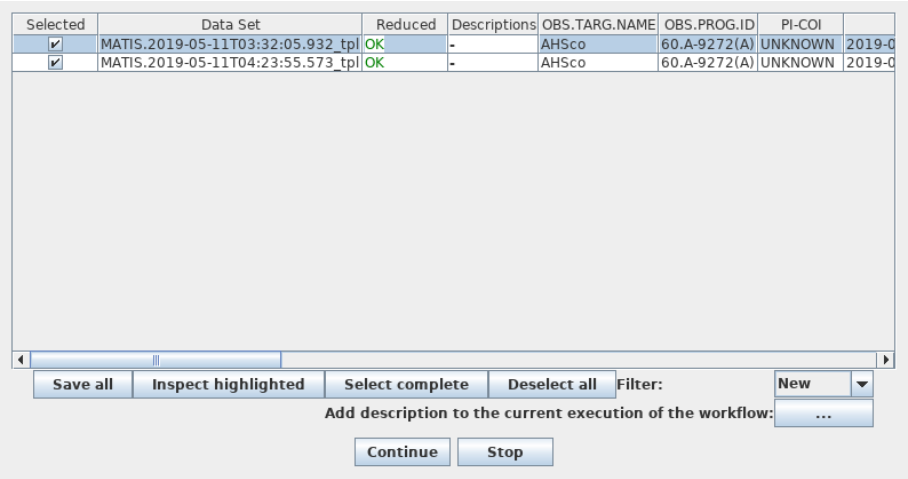


Figure 5.3: *The "Select Datasets" pop-up window.*

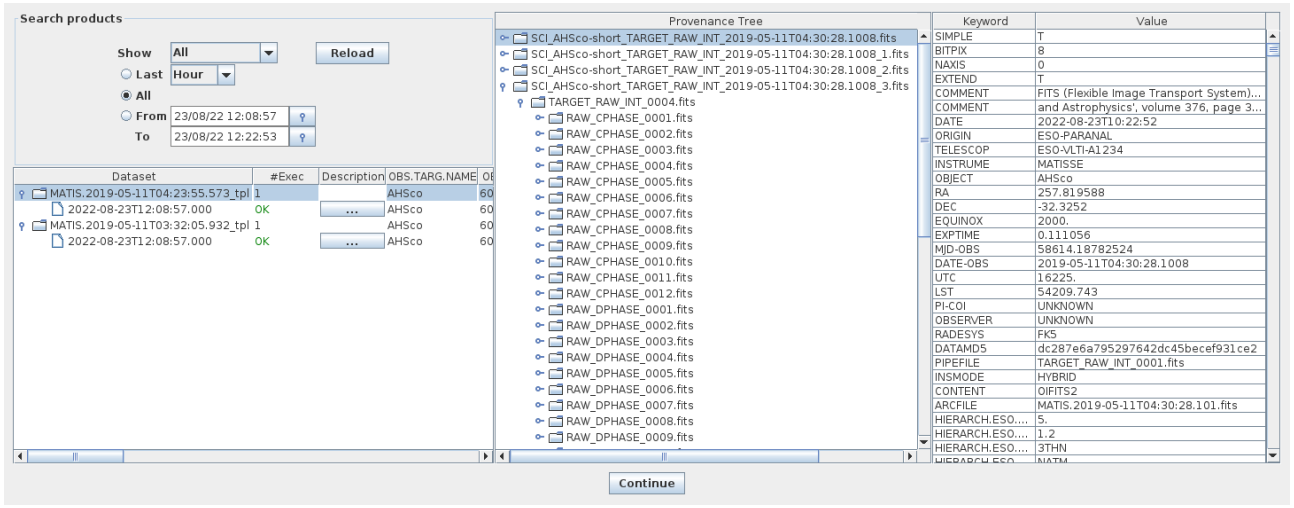


Figure 5.4: *The Product Explorer shows all datasets reduced in previous executions together with the full reduction chain for all the pipeline products.*

6 About the main `esoreflex` canvas

6.1 Saving And Loading Workflows

In the course of your data reductions, it is likely that you will customise the workflow for various data sets, even if this simply consists of editing the `ROOT_DATA_DIR` to a different value for each data set. Whenever you modify a workflow in any way, you have the option of saving the modified version to an `XML` file using `File -> Export As` (which will also open a new workflow canvas corresponding to the saved file). The saved workflow may be opened in subsequent `esoreflex` sessions using `File -> Open`. Saving the workflow in the default Kepler format (.kar) is only advised if you do not plan to use the workflow with another computer.

6.2 Buttons

At the top of the `esoreflex` canvas are a set of buttons which have the following functions:

-  - Zoom in.
-  - Reset the zoom to 100%.
-  - Zoom the workflow to fit the current window size (Recommended).
-  - Zoom out.
-  - Run (or resume) the workflow.
-  - Pause the workflow execution.
-  - Stop the workflow execution.

The remainder of the buttons (not shown here) are not relevant to the workflow execution.

6.3 Workflow States

A workflow may only be in one of three states: executing, paused, or stopped. These states are indicated by the yellow highlighting of the , , and  buttons, respectively. A workflow is executed by clicking the  button. Subsequently the workflow and any running pipeline recipe may be stopped immediately by clicking the  button, or the workflow may be paused by clicking the  button which will allow the current actor/recipe to finish execution before the workflow is actually paused. After pausing, the workflow may be resumed by clicking the  button again.

7 The MATISSE Workflow

The MATISSE workflow canvas is organised into a number of areas. From top-left to top-right you will find general workflow instructions, directory parameters, and global parameters. In the middle row you will find five boxes describing the workflow general processing steps in order from left to right, and below this the workflow actors themselves are organised following the workflow general steps.

7.1 Workflow Canvas Parameters

The workflow canvas displays a number of parameters that may be set by the user. Under "Setup Directories" the user is only required to set the `RAW_DATA_DIR` to the working directory for the dataset(s) to be reduced, which, by default, is set to the directory containing the demo data. The `RAW_DATA_DIR` is recursively scanned by the `Data Organiser` actor for input raw data. The directory `CALIB_DATA_DIR`, which is by default within the pipeline installation directory, is also scanned by the `Data Organiser` actor to find any static calibrations that may be missing in your dataset(s). If required, the user may edit the directories `BOOKKEEPING_DIR`, `LOGS_DIR`, `TMP_PRODUCTS_DIR`, and `END_PRODUCTS_DIR`, which correspond to the directories where book-keeping files, logs, temporary products and end products are stored, respectively (see the Reflex User Manual for further details; [4]).

There is a mode of the `Data Organiser` that skips the built-in data organisation and uses instead the data organisation provided by the CalSelector tool. To use this mode, click on `Use CalSelector associations` in the `Data Organiser` properties and make sure that the input data directory contains the XML file downloaded with the CalSelector archive request (note that this does not work for all instrument workflows).

Under the "Global Parameters" area of the workflow canvas, the user may set the `FITS_VIEWER` parameter to the command used for running his/her favourite application for inspecting FITS files. Currently this is set by default to `fv`, but other applications, such as `ds9`, `skycat` and `gaia` for example, may be useful for inspecting image data. Note that it is recommended to specify the full path to the visualization application (an alias will not work).

By default the `EraseDirs` parameter is set to `false`, which means that no directories are cleaned before executing the workflow, and the recipe actors will work in Lazy Mode (see Section 7.2.5), reusing the previous pipeline recipe outputs if input files and parameters are the same as for the previous execution, which saves considerable processing time. Sometimes it is desirable to set the `EraseDirs` parameter to `true`, which forces the workflow to recursively delete the contents of the directories specified by `BOOKKEEPING_DIR`, `LOGS_DIR`, and `TMP_PRODUCTS_DIR`. This is useful for keeping disk space usage to a minimum and will force the workflow to fully re-reduce the data each time the workflow is run.

The parameter `RecipeFailureMode` controls the behaviour in case that a recipe fails. If set to `Continue`, the workflow will trigger the next recipes as usual, but without the output of the failing recipe, which in most of the cases will lead to further failures of other recipes without the user actually being aware of it. This mode might be useful for unattended processing of large number of datasets. If set to `Ask`, a pop-up window will ask whether the workflow should stop or continue. This is the default. Alternatively, the `Stop` mode will stop the workflow execution immediately.

The parameter `ProductExplorerMode` controls whether the `ProductExplorer` actor will show its window or not. The possible values are `Enabled`, `Triggered`, and `Disabled`. `Enabled` opens the Product-

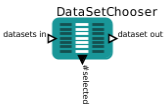
Explorer GUI at the end of the reduction of each individual dataset. `Triggered` (default and recommended) opens the ProductExplorer GUI when all the selected datasets have been reduced. `Disabled` does not display the ProductExplorer GUI.

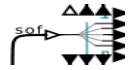
7.2 Workflow Actors

7.2.1 Simple Actors

Simple actors have workflow symbols that consist of a single (rather than multiple) green-blue rectangle. They may also have an icon within the rectangle to aid in their identification. The following actors are simple actors:

- 

- The `DataOrganiser` actor.
- 

- The `DataSetChooser` actor (inside a composite actor).
- 

- The `FitsRouter` actor Redirects files according to their categories.
- 

- The `ProductRenamer` actor.
- 

- The `ProductExplorer` actor (inside a composite actor).

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

7.2.2 Data Organisation And Selection

The `DataOrganiser` (DO) is the first crucial component of a Reflex workflow. The DO takes as input `RAW_DATA_DIR` and `CALIB_DATA_DIR` and it detects, classifies, and organises the files in these directories and any subdirectories. The output of the DO is a list of "DataSets". A DataSet is a special Set of Files (SoF). A DataSet contains one or several science (or calibration) files that should be processed together, and all files needed to process these data. This includes any calibration files, and in turn files that are needed to process these calibrations. Note that different DataSets might overlap, i.e. some files might be included in more than one DataSet (e.g., common calibration files).

A DataSet lists three different pieces of information for each of its files, namely 1) the file name (including the path), 2) the file category, and 3) a string that is called the "purpose" of the file. The DO uses the

OCA[4] rules to find the files to include in a DataSet, as well as their categories and purposes. The file category identifies different types of files, and it is derived by information in the header of the file itself. A category could for example be RAW_CALIBRATION_1, RAW_CALIBRATION_2 or RAW_SCIENCE, depending on the instrument. The purpose of a file identifies the reason why a file is included in a DataSet. The syntax is action_1/action_2/action_3/ ... /action_n, where each action_i describes an intended processing step for this file (for example, creation of a MASTER_CALIBRATION_1 or a MASTER_CALIBRATION_2). The actions are defined in the OCA rules and contain the recipe together with all file categories required to execute it (and predicted products in case of calibration data). For example, a workflow might include two actions action_1 and action_2. The former creates MASTER_CALIBRATION_1 from RAW_CALIBRATION_1, and the later creates a MASTER_CALIBRATION_2 from RAW_CALIBRATION_2. The action_2 action needs RAW_CALIBRATION_2 frames and the MASTER_CALIBRATION_1 as input. In this case, these RAW_CALIBRATION_1 files will have the purpose action_1/action_2. The same DataSet might also include RAW_CALIBRATION_1 with a different purpose; irrespective of their purpose the file category for all these biases will be RAW_CALIBRATION_1.

The Datasets created via the DataOrganiser will be displayed in the DataSet Chooser. Here the users have the possibility to inspect the various datasets and decide which one to reduce. By default, DataSets that have not been reduced before are highlighted for reduction. Click either Continue in order to continue with the workflow reduction, or Stop in order to stop the workflow. A full description of the DataSet Chooser is presented in Section 7.2.3.

Once the Continue is pressed, the workflow starts to reduce the first selected DataSet. Files are broadcasted according to their purpose to the relevant actors for processing.

The categories and purposes of raw files are set by the DO, whereas the categories and purpose of products generated by recipes are set by the RecipeExecuter. The file categories are used by the FitsRouter to send files to particular processing steps or branches of the workflow (see below). The purpose is used by the SofSplitter and SofAccumulator to generate input SoFs for the RecipeExecuter. The SofSplitter and SofAccumulator accept several SoFs as simultaneous input. The SofAccumulator creates a single output SoF from the inputs, whereas the SofSplitter creates a separate output SoF for each purpose.

7.2.3 DataSetChooser

The DataSetChooser displays the DataSets available in the “Select Data Sets” window, activating vertical and horizontal scroll bars if necessary (Fig. 5.3).

Some properties of the DataSets are displayed: the name, the number of files, a flag indicating if it has been successfully reduced (a green OK), if the reduction attempts have failed or were aborted (a red FAILED), or if it is a new dataset (a black "-"). The column "Descriptions" lists user-provided descriptions (see below), other

[4]OCA stands for OrganisationClassificationAssociation and refers to rules, which allow to classify the raw data according to the contents of the header keywords, organise them in appropriate groups for processing, and associate the required calibration data for processing. They can be found in the directory <install_dir>/share/esopipes/<pipeline-version>/reflex/, carrying the extension .oca. The variable <install_dir> depends on the operative system and installation procedure. For installation through rpm: <install_dir>=/usr; for installation through macport <install_dir>=/opt/local; for installation through the installation script install_esoreflex it depends on the path specified during installation, e.g. <install_dir>=<specified_path>/install

columns indicate the instrument set-up and a link to the night log.

Sometimes you will want to reduce a subset of these DataSets rather than all DataSets, and for this you may individually select (or de-select) DataSets for processing using the tick boxes in the first column, and the buttons `Deselect All` and `Select Complete` at the bottom, or configure the "Filter" field at the bottom left. Available filter options are: "New" (datasets not previously reduced will be selected), "Reduced" (datasets previously reduced will be selected), "All" (all datasets will be selected), and "Failed" (dataset with a failed or aborted reduction will be selected).

You may also highlight a single DataSet in blue by clicking on the relevant line. If you subsequently click on `Inspect Highlighted`, then a "Select Frames" window will appear that lists the set of files that make up the highlighted DataSet including the full filename[5], the file category (derived from the FITS header), and a selection tick box in the right column. The tick boxes allow you to edit the set of files in the DataSet which is useful if it is known that a certain calibration frame is of poor quality (e.g: a poor raw flat-field frame). The list of files in the DataSet may also be saved to disk as an `ASCII` file by clicking on `Save As` and using the file browser that appears.

By clicking on the line corresponding to a particular file in the "Select Frames" window, the file will be highlighted in blue, and the file FITS header will be displayed in the text box on the right, allowing a quick inspection of useful header keywords. If you then click on `Inspect`, the workflow will open the file in the selected FITS viewer application defined by the workflow parameter `FITS_VIEWER`.

To exit from the "Select Frames" window, click `Continue`.

To add a description of the reduction, press the button `...` associated with the field "Add description to the current execution of the workflow" at the bottom right of the Select Dataset Window; a pop-up window will appear. Enter the desired description (e.g. "My first reduction attempt") and then press `OK`. In this way, all the datasets reduced in this execution, will be flagged with the input description. Description flags can be visualized in the `SelectFrames` window and in the `ProductExplorer`, and they can be used to identify different reduction strategies.

To exit from the "Select DataSets" window, click either `Continue` in order to continue with the workflow reduction, or `Stop` in order to stop the workflow.

7.2.4 The ProductExplorer

The ProductExplorer is an interactive component in the `esoreflex` workflow whose main purpose is to list the final products with the associated reduction tree for each dataset and for each reduction attempt (see Fig. 5.4).

Configuring the ProductExplorer

You can configure the ProductExplorer GUI to appear after or before the data reduction. In the latter case you can inspect products as reduction goes on.

1. To display the ProductExplorer GUI at the end of the datareduction:

   - Click on the global parameter "ProductExplorerMode" before starting the data reduction. A configuration window will appear allowing you to set the execution mode of the Product Explorer. Valid options are:

[5] keep the mouse pointer on the file name to visualize the full path name.

  - "Triggered" (default). This option opens the ProductExplorer GUI when all the selected datasets have been reduced.
  - "Enabled". This option opens the ProductExplorer GUI at the end of the reduction of each individual dataset.
  - “Disable”. This option does not display the ProductExplorer GUI.

- Press the  button to start the workflow.

2. To display the ProductExplorer GUI “before” starting the data reduction:

- double click on the composite Actor "Inspect previously reduced data". A configuration window will appear. Set to "Yes" the field "Inspect previously reduced data (Yes/No)". Modify the field "Continue reduction after having inspected the previously reduced data? (Continue/Stop/Ask)". "Continue" will continue the workflow and trigger the DataOrganizer. "Stop" will stop the workflow; "Ask" will prompt another window deferring the decision whether continuing or not the reduction after having closed the Product Explorer.

- Press the  button to start the workflow. Now the ProductExplorer GUI will appear before starting the data organization and reduction.

Exploring the data reduction products

The left window of the ProductExplorer GUI shows the executions for all the datasets (see Fig. 5.4). Once you click on a dataset, you get the list of reduction attemps. Green and red flags identify successfull or unsucessfull reductions. Each reduction is linked to the “Description” tag assigned in the “Select Dataset” window.

1. To identify the desired reduction run via the “Description” tag, proceed as follows:

- Click on the symbol at the left of the dataset name. The full list of reduction attempts for that dataset will be listed. The column Exec indicates if the reduction was succesful (green flag: "OK") or not (red flag: "Failed").

- Click on the entries in the field "Description" to visualize the description you have entered associated to that dataset on the Select Dataset window when reducing the data.

- Identify the desired reduction run. All the products are listed in the central window, and they are organized following the data reduction cascade.

You can narrow down the range of datasets to search by configuring the field "Show" at the top-left side of the ProductExplorer (options are: "All", "Successful", "Unsuccessful"), and specifying the time range (Last, all, From-to).

2. To inspect the desired file, proceed as follows:

- Navigate through the data reduction cascade in the ProductExplorer by clicking on the files.
- Select the file to be inspected and click with the mouse right-hand button. The available options are:
  - Options available always:
    - Copy full path. It copies the full name of the file onto the clipboard. Shift+Ctr+v to past it into a terminal.
    - Inspect Generic. It opens the file with the FITS viewer selected in the main workflow canvas.
    - Inspect with. It opens the file with an executable that can be specified (you have to provide the full path to the executable).
  - Options available for files in the `TMP_PRODUCTS_DIR` directory only:
    - command line. Copy of the environment configuration and recipe call used to generate that file.
    - Xterm. It opens an Xterm at the directory containing the file.
  - Options available for products associated to interactive windows only:
    - Display pipeline results. It opens the interactive windows associated to the recipe call that generated the file. Note that this is for visualization purposes only; the recipe parameters cannot be changed and the recipe cannot be re-run from this window.

7.2.5 Lazy Mode

By default, all `RecipeExecuter` actors in a pipeline workflow are "Lazy Mode" enabled. This means that when the workflow attempts to execute such an actor, the actor will check whether the relevant pipeline recipe has already been executed with the same input files and with the same recipe parameters. If this is the case, then the actor will not execute the pipeline recipe, and instead it will simply broadcast the previously generated products to the output port. The purpose of the Lazy Mode is therefore to minimise any reprocessing of data by avoiding data re-reduction where it is not necessary.

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

The re-reduction of data at each execution may sometimes be desirable. To force a re-reduction of data for any single `RecipeExecuter` actor in the workflow, right-click the actor, select `Configure Actor`, and uncheck the Lazy mode parameter tick-box in the "Edit parameters" window that is displayed. For many workflows the `RecipeExecuter` actors are actually found inside the composite actors in the top level workflow. To access such embedded `RecipeExecuter` actors you will first need to open the sub-workflow by right-clicking on the composite actor and then selecting `Open Actor`.

To force the re-reduction of all data in a workflow (i.e. to disable Lazy mode for the whole workflow), you must uncheck the Lazy mode for every single `RecipeExecuter` actor in the entire workflow. It is also possible to change the name of the bookkeeping directory, instead of modifying any of the Lazy mode parameters. This will also force a re-reduction of the given dataset(s). A new reduction will start (with the lazy mode still enabled), but the results of previous reduction will not be reused. Alternatively, if there is no need to keep any of the previously reduced data, one can simply set the `EraseDirs` parameter under the "Global Parameters" area of

the workflow canvas to `true`. This will then remove all previous results that are stored in the bookkeeping, temporary, and log directories before processing the input data, in effect, starting a new clean data reduction and re-processing every input dataset. *Note: The option* `EraseDirs = true` *does not work in* `esoreflex` *version 2.9.x and causes the workflow to crash.*

8 Reducing and Calibrating Your Own Science Data with Gasgano

There are three ways to run the ESO pipelines, in all cases the executed recipes are, however, the same. The differences are in the user interfaces. Below, we list the first two ways; the third and **recommended way (Reflex)** is discussed in Sect. 9.

1. Gasgano is a data management tool that simplifies the data organisation process, offering automatic data classification and making the data association easier (even if automatic association of frames is not provided). Gasgano determines the classification of a file by applying instrument specific rules, while users must provide this information to the recipes when they are executed manually using Esorex from the command line. In addition, Gasgano allows the user to execute directly the pipeline recipes on a set of selected files. Gasgano is automatically installed when installing a stand-alone pipeline kit available from `http://www.eso.org/sci/software/pipelines/`, but not as part of the Reflex installation.
2. Esorex, a command-line utility for running pipeline recipes is also available. Both Gasgano and Reflex use Esorex to run the pipeline recipes. Esorex may be embedded by users at their home institute into data reduction scripts for the automation of processing tasks. See `http://www.eso.org/sci/software/cpl/esorex.html` for more information.

8.1 An introduction to Gasgano and EsoRex

Before being able to call pipeline recipes on a set of data, the data must be opportunely 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 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. As all the required information is stored in the FITS headers, data association is based on a set of keywords (called "association keywords") and is specific to each type of calibration.

The process of data classification and association is known as data organisation. The *DO Category* is the label assigned to a data type as a result of data classification.

An instrument pipeline consists of a set of data processing modules that can be called from different host applications, either from the command line with *Esorex*, from the automatic data management tools available at Paranal, or from the graphical *Gasgano* tool.

8.1.1 Using Gasgano

To get familiar with the pipeline recipes and their usage, it is useful 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 recipes documentation and preferred data display tools.

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

```
gasgano &
```

The *Gasgano* main window will appear. To adapt the file and keyword displays to MATISSE data, use the *Preferences...* editor under the *File* menu to add MATISSE as instrument with keyword DET.NAME in the table at the bottom of the *File Display & Grouping* tab. Then click *OK* and select the *Add/Remove Files* entry of the menu to specify the directory containing your data files. These preferences will be store in the file *.gasganorc* in the home directory.

On Figure 8.1 (next page), a view on a set of MATISSE data is shown as an example. Here, the files belonging to the two MATISSE detectors are displayed with separate handles due to the specification of DET.NAME for the file display. The files selected here (highlighted by clicking the files) include the raw files, processed calibrations (*M.** files), and the JSDC calibrator catalog necessary to run the main data reduction recipe *mat_raw_estimates*. The figure also shows how these files can be sent to the recipe for processing.

The data are hierarchically organised as preferred by the user. After each file name are shown the classification, template ID, original file name, the template exposure number and the number of exposures in the template.

The file names corresponding to raw data (*i.e.*, those produced by the instrument) appear in blue, while the names of the processed data and of the static calibration tables appear in red.

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 opportunely filtered and searched. Images and tables may be easily displayed using the viewers specified in the appropriate *Preferences* fields.

Frames can be selected from the main window for being processed by the appropriate recipe: Figure 8.2, shows our selection along with the pipeline parameter values to be used for the reduction Pressing the Execute button will run the recipe.

Help about the recipe may be obtained from the *Help* menu. Before launching the recipe, its configuration may be opportunely 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 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.

Please refer to the *Gasgano User's Manual* [2] for a more complete description of the *Gasgano* interface.

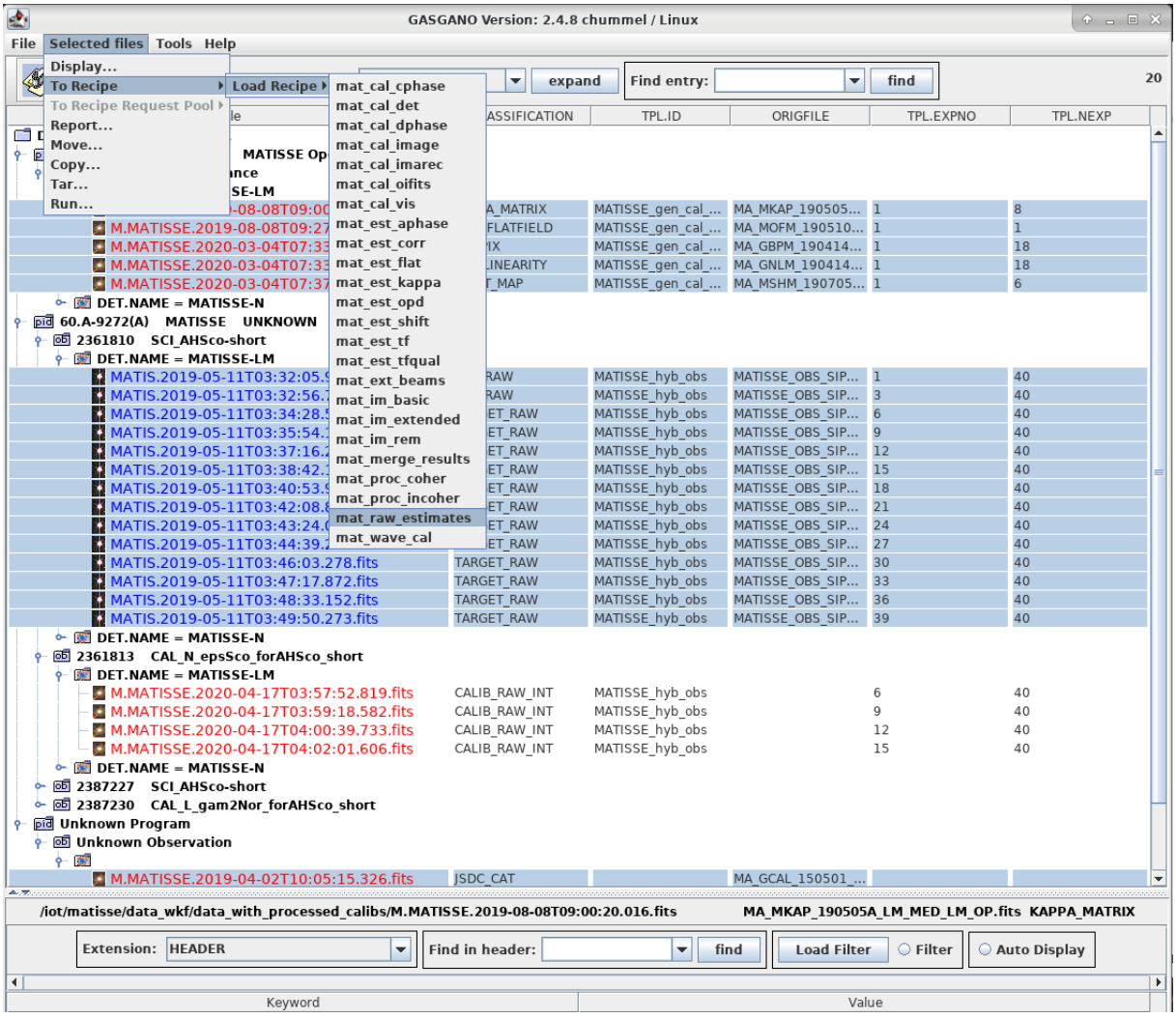


Figure 8.1: *The Gasgano main window.*

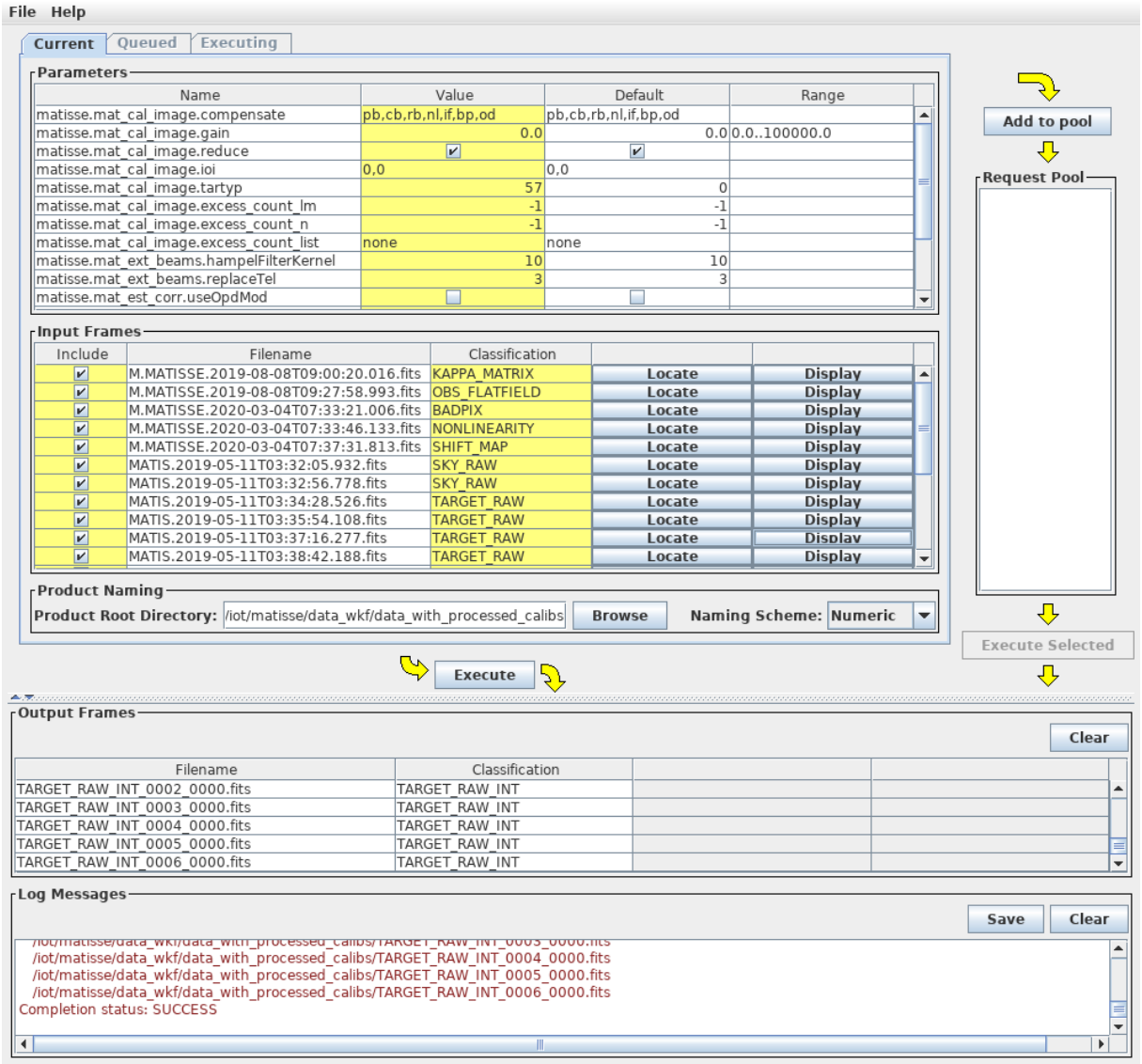


Figure 8.2: *Selecting files to be processed by a pipeline recipe and the log messages after executing the recipe.*

8.1.2 Using EsoRex

EsoRex is a command line utility for running pipeline recipes. It may be embedded by users into data reduction scripts for the automation of processing tasks. On the other side, *EsoRex* doesn't offer all the facilities available with *Gasgano*, and the user must classify and associate the data using the information contained in the FITS header keywords. 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 Data Organizer (DO) category in an ASCII file, the *set-of-frames* (SOF), that is required when launching a recipe. [6]

Here is an example of SOF, valid for the *mat_raw_estimates* recipe

```
MATIS.2019-05-11T03:32:05.932.fits      SKY_RAW
MATIS.2019-05-11T03:32:56.778.fits      SKY_RAW
MATIS.2019-05-11T03:34:28.526.fits      TARGET_RAW
MATIS.2019-05-11T03:35:54.108.fits      TARGET_RAW
MATIS.2019-05-11T03:37:16.277.fits      TARGET_RAW
MATIS.2019-05-11T03:38:42.188.fits      TARGET_RAW
MATIS.2019-05-11T03:40:53.993.fits      TARGET_RAW
MATIS.2019-05-11T03:42:08.840.fits      TARGET_RAW
MATIS.2019-05-11T03:43:24.079.fits      TARGET_RAW
MATIS.2019-05-11T03:44:39.267.fits      TARGET_RAW
MATIS.2019-05-11T03:46:03.278.fits      TARGET_RAW
MATIS.2019-05-11T03:47:17.872.fits      TARGET_RAW
MATIS.2019-05-11T03:48:33.152.fits      TARGET_RAW
MATIS.2019-05-11T03:49:50.273.fits      TARGET_RAW
M.MATISSE.2019-04-02T10:05:15.326.fits  JSDC_CAT
M.MATISSE.2019-07-26T06:15:31.810.fits  SHIFT_MAP
M.MATISSE.2019-08-08T05:40:24.570.fits  BADPIX
M.MATISSE.2019-08-08T05:40:46.110.fits  NONLINEARITY
M.MATISSE.2019-08-08T09:00:20.016.fits  KAPPA_MATRIX
M.MATISSE.2019-08-08T09:27:58.993.fits  OBS_FLATFIELD
```

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 MATISSE pipeline recipes do not verify in any way the correctness of the *DO Category* specified by the user in the SOF. The reason of this lack of control is that the MATISSE recipes are just the DRS (data reduction system) 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 8.1.1, page 25).

[6] The set-of-frames corresponds to the *Input Frames* panel of the *Gasgano* recipe execution window (see Figure 8.2, page 28).

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.

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 [7]) 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

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.[7] 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 *mat_raw_estimates* has its *EsoRex* generated configuration file named `mat_raw_estimates.rc`, and is generated with the command:

esorex - -create-config mat_raw_estimates

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

```
# --tartyp
# TARTYP estimation (0 = none, 1 = N*S+U+N*T+U, 2 = show intensity, 4 = show
# correlation, 8 = estimate TARTYP, 16 = change TIME, TARTYP, LOCALOPD and
# STEPPING_PHASE, 32 = exchange U with S or T).
matisse.mat_cal_image.tartyp=0
```

In this example, the parameter `matisse.mat_raw_estimates.tartyp` is set to the value `0`. 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 recipes provided by the MATISSE pipeline are designed to implement a cascade of macro data reduction steps, each controlled by its own parameters. For this reason and to prevent parameter name clashes we specify as parameter prefix not only the instrument name but also the name of the step they refer to. Shorter parameter aliases are made available for use on the command line.

[7] The *EsoRex* recipe configuration file corresponds to the *Parameters* panel of the *Gasgano* recipe execution window (see Figure 8.2, page 28).

The command

esorex - -create-config recipe_name

generates a default configuration file **recipe_name.rc** in the directory **$HOME/.esorex**[8].

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 and their role are provided in the MATISSE pipeline manual.

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 specifying 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 *mat_estimatesestimates* for processing the files specified in the set-of-frames `mat_estimatesestimates.sof`:

esorex mat_estimatesestimates mat_raw_estimates.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 *mat_raw_estimates* recipe *tartyp* parameter to `1`, the following should be typed:

esorex mat_raw_estimates - -tartyp=1 mat_raw_estimates.sof

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

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

9 Reducing and Calibrating Your Own Science Data with Reflex

1. Reflex is the **recommended** environment to reduce ESO data. It automatically organizes input files according to their category and runs the entire reduction chain at the push of a button. It supports break points in the reduction sequence in order to inspect and interact with intermediate and final products and rerun the corresponding step if necessary. A description on how to use Reflex in more detail than presented in Sect. 5 is provided in the following. Start Reflex (`esoreflex`) and select a workflow by clicking on `File -> Open`

9.1 Available Reflex workflows

The first two workflows (**matisse_wkf_LM** and **matisse_wkf_N**, Fig. 5.2) reduce the raw interferometric observations into uncalibrated interferometric measurements (visibility, close phase, differential phase). They have initial parameter settings appropriate for the respective detector (band). They processes all exposures of an Observation Block simultaneously. The second workflow (**matisse_viscal**, Fig. 9.1) calibrates the science data of both bands.

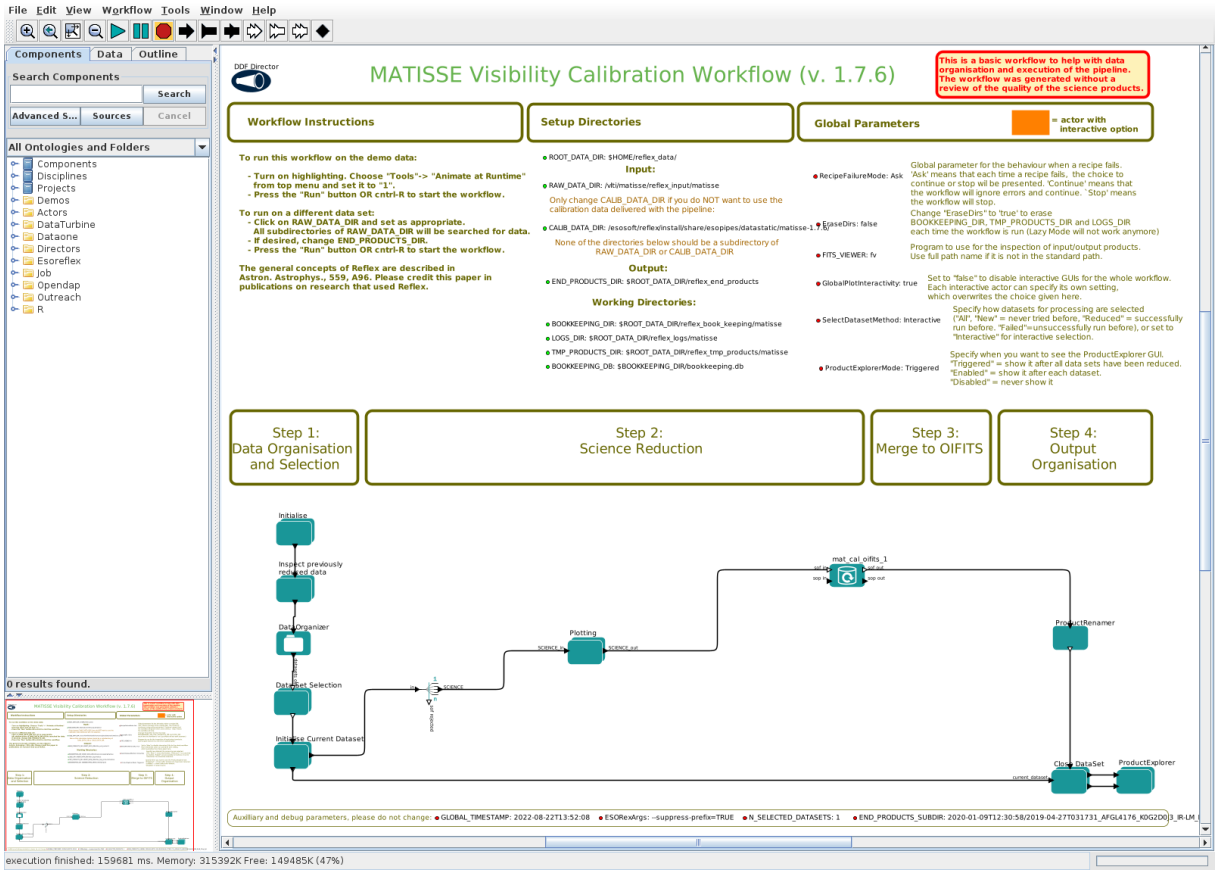


Figure 9.1: *MATISSE calibration workflow.*

9.2 Checking the status of your calibrator in the JSDC

Before reducing the calibrator files, it is important to check if the observed calibrators are listed in the JSDC (The JMMC Stellar Diameters Catalog[9] (JSDC) [1]). If not, the workflow would not calculate the transfer function (TF) and the science data could not be calibrated. If the calibrator is not listed in the JSDC, you can use the tool `catalog.py` available at https://gitlab.oca.eu/MATISSE/tools/-/tree/master/mat_tools/catalog.py to produce a FITS file from a text file containing the required information like the angular diameter of the calibrator. The text file has this format (target name and coordinates abbreviated for clarity):

```
#NAME      , RAJ2000 , DEJ2000   , LMAG, MMAG, NMAG, UDDK, UDDL, UDDM, UDDN, E_LDD
#-------------------------------------------------------------------------------
IRAS 08534, 08 55 41, -24 17 32, NULL, NULL, NULL, 3.6 , 3.6 , 3.6 , 3.6 , 0.35
```

This file should be put in the `CALIB_DATA_DIR` and you should make sure this file is included (and marked) in the SOF and not the JSDC when reducing this particular data set.

9.3 Specifying data directories and selecting files

To reduce your own science data, simply edit the paths to the root (optional) and data directories. Under the former, Reflex will create sub-directories which will contain temporary and end products, as well as book keeping and log files. The latter directory should contain your raw files downloaded from the ESO archive. If you downloaded processed calibrations, just move the folder *data_with_processed_calibrations* into the data folder. Reflex searches for files in this directory recursively. The paths are defined at the top of the workflow window in the area labeled `Setup Directories`. Simply double click on `RAWDATA_DIR`, enter the path to your raw science directory and then re-run the workflow in the same way as you did for the tutorial demo data. In case the data sets listed in the first window created by the work flow (Fig. 5.3) are greyed out, calibration files are missing (hovering with the mouse over the grey file entry will give more details). You can click the entry and a GUI opens up showing the dependency tree of the science (or calibrator) file on calibrations (Fig. 5.3).

9.4 Interactive plots during workflow execution

During the workflow execution, interactive plot windows are displayed (`GlobalPlotInteractivity` is `true`). Examples from the reduction of the demo data, one for the LM band and one for the N band, are shown below. The workflow pauses here to allow the inspection of various plots (spectra, fringe motion, etc.), clicking `Continue` (at the bottom) resumes the workflow. For convenience, it is recommended to skip plotting during the first reduction of the data, and run the reduction again with plotting, as the availability of the intermediate data products will cause the workflow to proceed faster.

[9] The catalog is online available at https://vizier.u-strasbg.fr/viz-bin/VizieR?-source=II/346&-to=3.

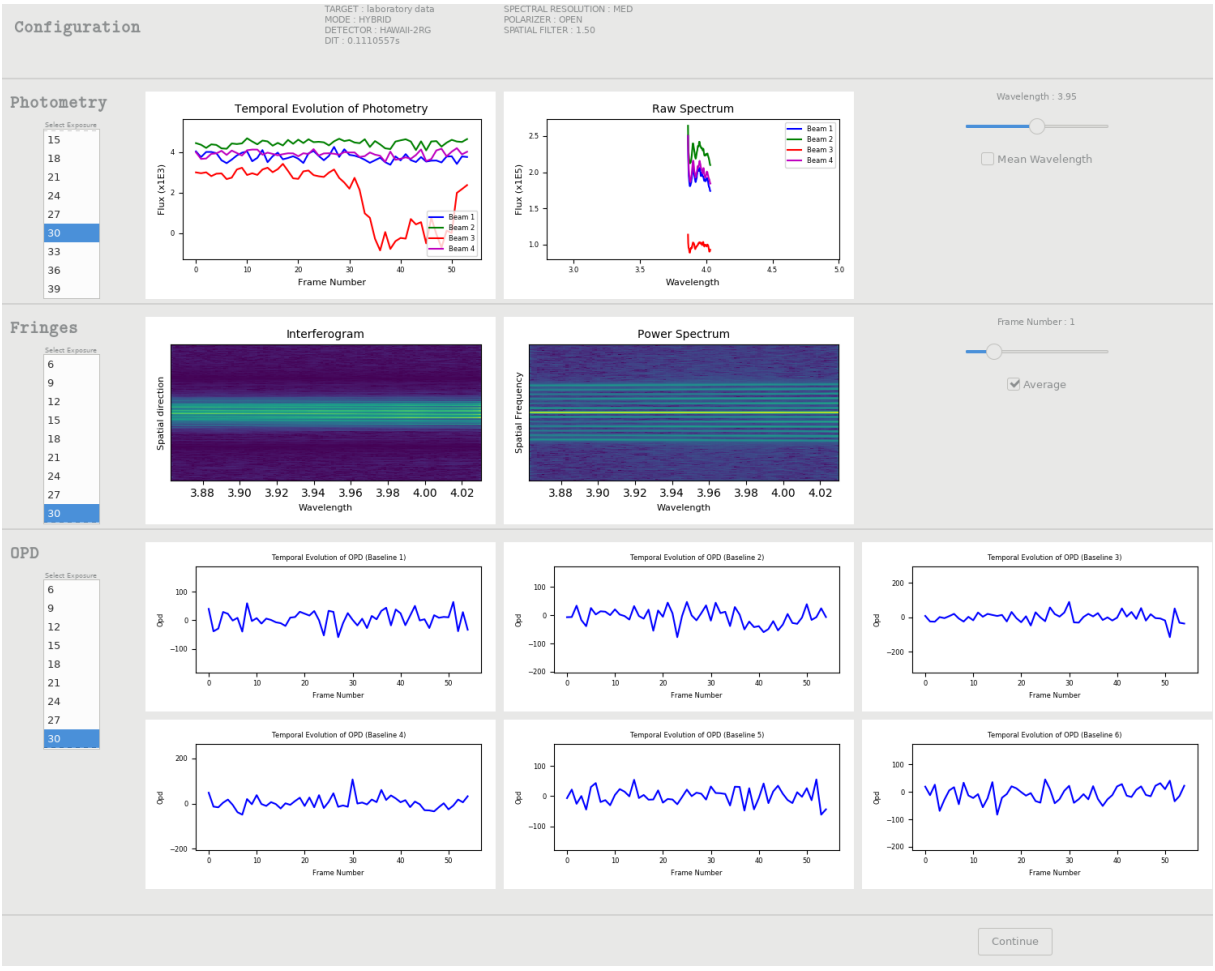


Figure 9.2: *The MATISSE interactive plot window for reduced LM-data. Click in the boxes on the left side to select specific exposures (TPL.EXPNO) Note that here the medium spectral resolution was specified in the OB, so only a rather small wavelength range is displayed around the specified central wavelength. Despite a brief drop-out in flux for beam three in the displayed exposure, not effect can be seen in the fringe and OPD panels. The OPD traces appear flat because fringe tracking was performed in the L-band.*



Figure 9.3: *The MATISSE interactive plot window for reduced N-band data. Note that here, in low spectral resolution, the entire N-band wavelength range is displayed. The photometry plot is empty as no simultaneous photometry is recorded during fringe tracking in the N band.*

9.5 Examining the workflow results

When the workflow has finished, the Product Explorer window opens (Fig. 5.4). Select a data file and unfold the file tree in the "Provenance Tree" window. This provides information on the dependency of product files on the calibration files and other files from recipes executed before. You can inspect a data file by clicking the "Inspect with..." button, and entering the path to your favourite FITS file viewer (e.g., *fv*).

9.6 Calibrating your visibility data

The workflow *matisse_viscal* allows you to select the science files to calibrate (an example is shown in Fig. 9.4), and it automatically associates the corresponding calibrator files. If you downloaded the demo data set with the pipeline, you can run the workflow as is. Otherwise, you need to enter the fully qualified path to the folder *reflex_end_products*, where the reduction workflow saved the reduced data files. To display the calibrator data files associated with the science data, highlight the science file and click "Inspect highlighted". A window like the one shown in Fig. 9.5 will appear.

The association of calibrator files with science files is performed using the so-called "OCA" rules, contained in a file whose location can be displayed by right-clicking the icon of the DataOrganisor and selecting the option "Configure Actor". The association rules do not consider the container ID, but rather the time difference between the observation of the science target and any suitable calibrator taken with the same setup. If you want to enforce the use only of the data for the calibrators you selected for observation in the same concatenation as the science target (and you happen to have calibrator data in the same folder as your science data, you would need to add the following in the OCA file at the end of the line (before the semi-colon) which begins with "select file as CALIB_RAW_INT":

and inputFile.OBS.CONTAINER.ID==OBS.CONTAINER.ID

After clicking "Continue", plots like the ones shown in Figs. 9.6 and 9.7 will appear (click "Continue" near the bottom of each to continue the workflow). They provide a simple overview of the data quality to be expected.

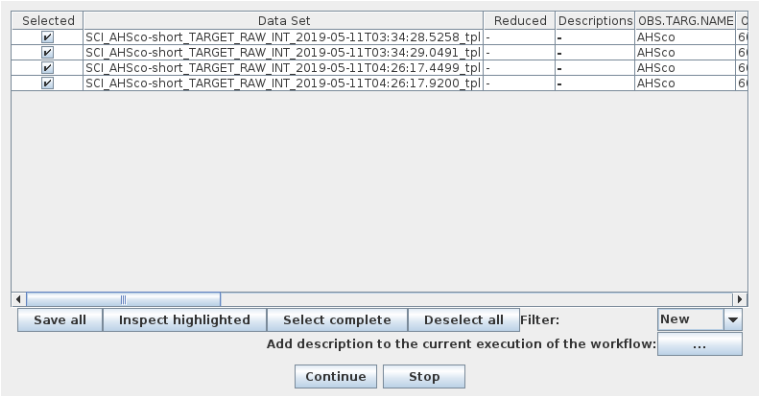


Figure 9.4: *Selection of data to be calibrated. Please note that files with (almost) the same time stamps are for the LM and N-band detectors of MATISSE, respectively.*

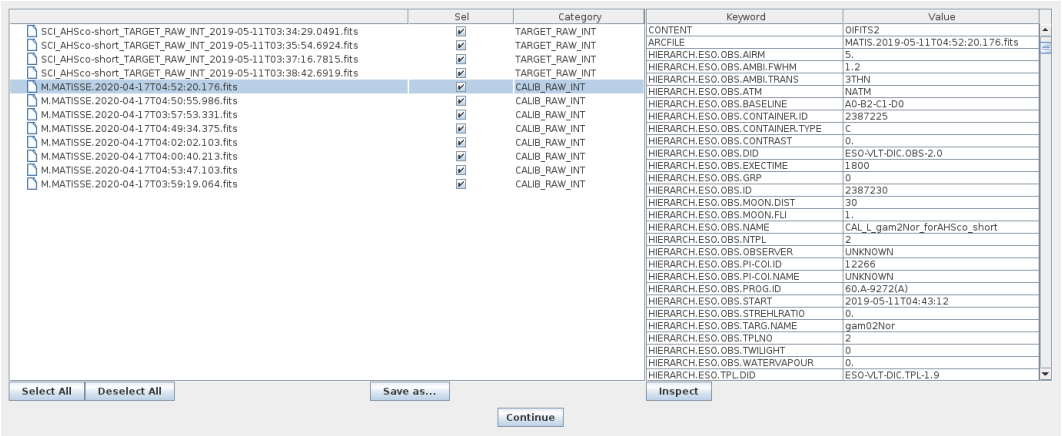


Figure 9.5: *Inspection of calibrator data sets associated with a science file. In this example, we show the processed calibrator file downloaded from the archive (M.MATISSE....) and the reduced calibrator file when raw calibrator data were downloaded from the archive. Of course, we only need one of the two calibrator files.*

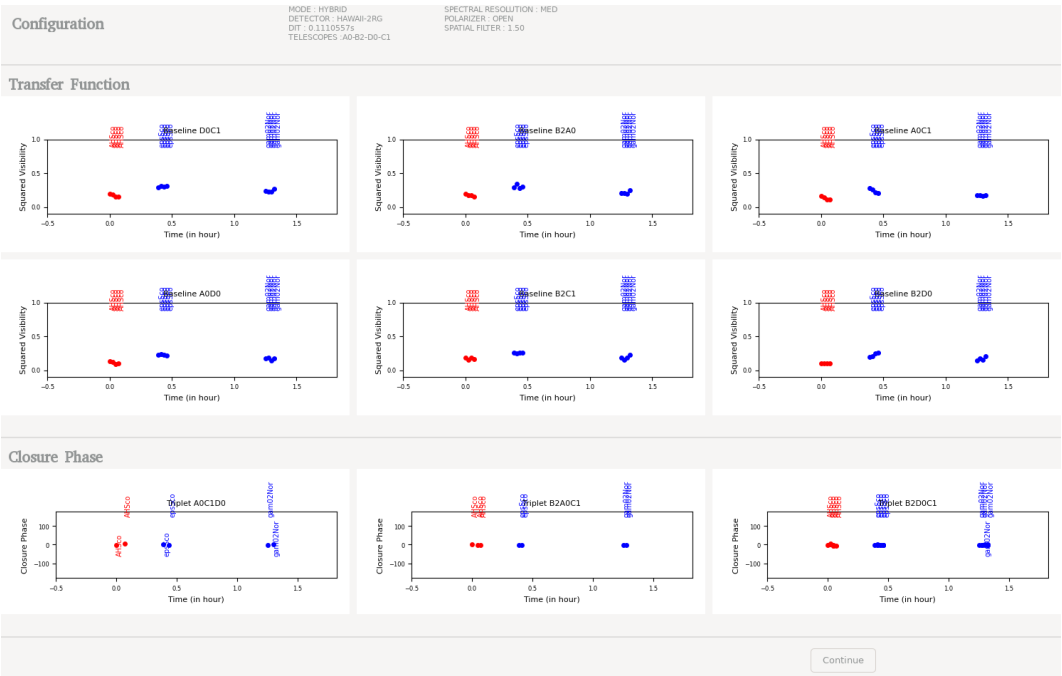


Figure 9.6: *Transfer function (visibilities and phases uncorrected for the object structure/diameter) in the L band (first exposure). Note that the science target is moderately resolved.*



Figure 9.7: *Transfer function (visibilities and phases uncorrected for the object structure/diameter) in the N band (first exposure). Note that the calibrator phases are near zero, but the amplitudes of both CAL and SCI are low due to instrmental loss of coherence. Here, the calibrator has a flux of 50 Jy in the N band, and was intended to calibrate the N band data only.*

The products of the workflow are then found also in *reflex_end_products*: the directory tree begins with one named with the time stamp of the execution of the recipe, where the next level down is named after the science target, under which one can find the related calibrated files named after the calibrator used, and the string "CAL_INT" in the name of the file.

Once the workflow finishes, it displays the product explorer (Fig. 9.8), where you select, in the left pane, a file starting with “SCI”, then expand the tree by clicking the handle. The top entry in the list corresponds to the most recent execution of the recipe. Select it and expand the provenance tree in the middle pane. The top entry is your calibrated science file, while the entries in the tree below it display the input data (for closure phase, differential phase, and visibility) for the output file. You can right-click a file starting with “SCI_” to enter a FITS viewer for display.

The reason for seeing only one science file displayed in the Product Explorer for every four raw science files on the same target is that the pipeline has automatically combined the input files taken for the different BCD settings to remove biases in the closure phases introduced by internal path differences.

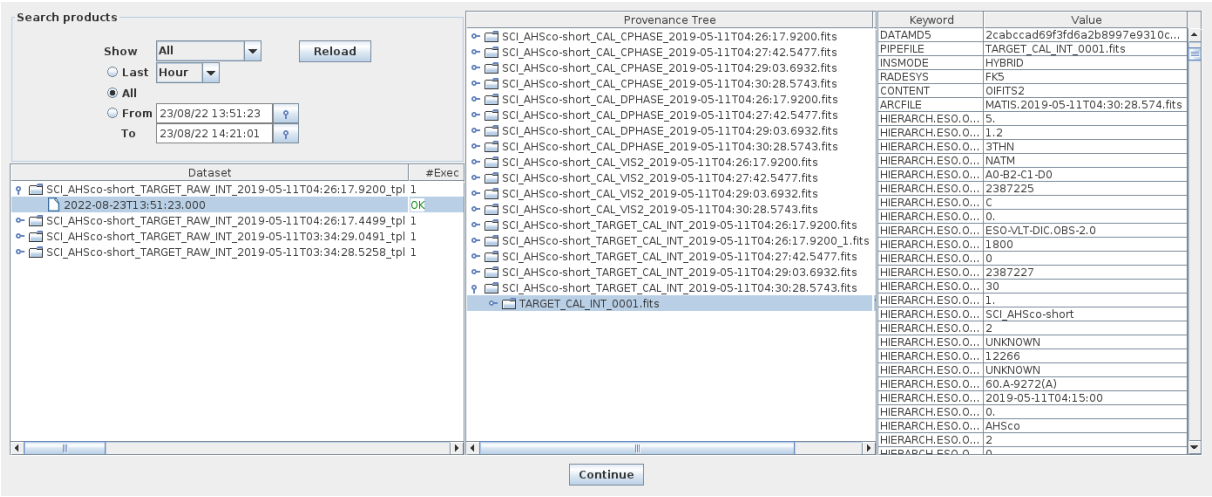


Figure 9.8: *Product explorer for calibrated science file.*

10 Frequently Asked Questions

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

  Press the Alt key together with your left mouse button to move the window upwards and to the left. At the bottom the `Continue`/`Stop` buttons will be visible. This bug is known but could not yet be fixed.

- **I tried to `Open` (or `Configure`) an `Actor` while the workflow is running and now it does not react any more. What should I do?**

  This is a limitation of the underlying Kepler engine. The only way out is to kill the workflow externally. If you want to change anything while a workflow is running you first need to pause it.

- **After a successful reduction of a data set, I changed this data set in some way (e.g. modified or removed some files, or changed the rules of the Data Organizer). When I restart Reflex, the Data Set Chooser correctly displays my new data set, but marks it as "reduced ok", even though it was never reduced before. What does this mean?**

  The labels in the column "Reduced" of the Data Set Chooser mark each dataset with "OK", "Failed" or "-". These labels indicate whether a data set has previously successfully been reduced at least once, all previous reductions failed, or a reduction has never been tried respectively. Data sets are identified by their name, which is derived from the first science file within the data set. As long as the data set name is preserved (i.e. the first science file in a data set has not changed), the Data Organizer will consider it to be the same data set. The Data Organizer recognizes any previous reductions of data sets it considers to be the same as the current one, and labels the current data set with "OK" if any of them was successful, even if the previously reduced data set differs from the current one.

  Note that the Product Explorer will list all the previous reductions of a particular data set only at the end of the reduction. This list might include successful and/or unsuccessful reduction runs with different parameters, or in your case with different input files. The important fact is that these are all reductions of data sets with the same first raw science file. By browsing through all reductions of a particular raw science file, the users can choose the one they want to use.

- **Where are my intermediate pipeline products?** Intermediate pipeline products are stored in the directory `<TMP_PRODUCTS_DIR>` (defined on the workflow canvas, under Setup Directories) and organised further in directories by pipeline recipe.

- **Can I use different sets of bias frames to calibrate my flat frames and science data?** Yes. In fact this is what is currently implemented in the workflow(s). Each file in a DataSet has a purpose attached to it ([4]). It is this purpose that is used by the workflow to send the correct set of bias frames to the recipes for flat frame combination and science frame reduction, which may or may not be the same set of bias frames in each case.

- **Can I run `Reflex` from the command line?** Yes, use the command:

  ```
  esoreflex -n <workflow_path>/<workflow>.xml
  ```

  The -n option will set all the different options for Kepler and the workflows to avoid opening any GUI elements (including pipeline interactive windows).

  It is possible to specify workflow variables (those that appear in the workflow canvas) in the command line. For instance, the raw data directory can be set with this command:

```
esoreflex -n -RAW_DATA_DIR <raw_data_path> \
             <workflow_path>/<workflow>.xml
```

You can see all the command line options with the command `esoreflex -h`.

Note that this mode is not fully supported, and the user should be aware that the path to the workflow must be absolute and even if no GUI elements are shown, it still requires a connection to the window manager.

- **How can I add new actors to an existing workflow?** You can drag and drop the actors in the menu on the left of the `Reflex` canvas. Under `Eso-reflex -> Workflow` you may find all the actors relevant for pipeline workflows, with the exception of the recipe executer. This actor must be manually instantiated using `Tools -> Instantiate Component`. Fill in the "Class name" field with `org.eso.RecipeExecuter` and in the pop-up window choose the required recipe from the pull-down menu. To connect the ports of the actor, click on the source port, holding down the left mouse button, and release the mouse button over the destination port. Please consult the Reflex User Manual ([4]) for more information.

- **How can I broadcast a result to different subsequent actors?** If the output port is a multi-port (filled in white), then you may have several relations from the port. However, if the port is a single port (filled in black), then you may use the black diamond from the toolbar. Make a relation from the output port to the diamond. Then make relations from the input ports to the diamond. Please note that you cannot click to start a relation from the diamond itself. Please consult the Reflex User Manual ([4]) for more information.

- **How can I manually run the recipes executed by Reflex?** If a user wants to re-run a recipe on the command line he/she has to go to the appropriate reflex_book_keeping directory, which is generally reflex_book_keeping/<workflow>/<recipe_name>_<number> There, subdirectories exist with the time stamp of the recipe execution (e.g. 2013-01-25T12:33:53.926/). If the user wants to re-execute the most recent processing he/she should go to the `latest` directory and then execute the script `cmdline.sh`. Alternatively, to use a customized `esorex` command the user can execute

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

  where INSTALL_DIR is the directory where Reflex and the pipelines were installed.

  If a user wants to re-execute on the command line a recipe that used a specific raw frame, the way to find the proper data.sof in the bookkeeping directory is via `grep <raw_file> */data.sof`. Afterwards the procedure is the same as before.

  If a recipe is re-executed with the command explained above, the products will appear in the directory from which the recipe is called, and not in the reflex_tmp_products or reflex_end_products directory, and they will not be renamed. This does not happen if you use the `cmdline.sh` script.

- **Can I reuse the bookkeeping directory created by previous versions of the pipeline?**

  In general no. In principle, it could be reused if no major changes were made to the pipeline. However there are situations in which a previously created bookkeeping directory will cause problems due to pipeline versions incompatibility. This is especially true if the parameters of the pipeline recipes have changed. In that case, please remove the bookkeeping directory completely.

- **How to insert negative values into a textbox?**

  Due to a bug in wxPython, the GUI might appear to freeze when attempting to enter a negative number in a parameter's value textbox. This can be worked around by navigating away to a different control in the GUI with a mouse click, and then navigating back to the original textbox. Once focus is back on the original textbox the contents should be selected and it should be possible to replace it with a valid value, by typing it in and pressing the enter key.

- **I've updated my Reflex installation and when I run esoreflex the process aborts. How can I fix this problem?**

  As indicated in Section 3, in case of major or minor (affecting the first two digit numbers) Reflex upgrades, the user should erase the `$HOME/KeplerData`, `$HOME/.kepler` directories if present, to prevent possible aborts (i.e. a hard crash) of the esoreflex process.

- **How can include my analysis scripts and algorithms into the workflow?**

  EsoReflex is capable of executing any user-provided script, if properly interfaced. The most convenient way to do it is through the Python actor. Please consult the tutorial on how to insert Python scripts into a workflow available here: `www.eso.org/sci/data-processing/Python_and_esoreflex.pdf`

[1] L. Bourgés, S. Lafrasse, G. Mella, O. Chesneau, J. L. Bouquin, G. Duvert, A. Chelli, and X. Delfosse. The JMMC Stellar Diameters Catalog v2 (JSDC): A New Release Based on SearchCal Improvements. In N. Manset and P. Forshay, editors, *Astronomical Data Analysis Software and Systems XXIII*, volume 485 of *Astronomical Society of the Pacific Conference Series*, page 223, May 2014. 33

[2] ESO, *http://www.eso.org/gasgano/*. *Gasgano User's Manual*. VLT-PRO-ESO-19000-1932. 26

[3] ESO, *http://www.eso.org/sci/facilities/paranal/instruments/matisse/doc.html*. *Matisse User Manual*. 7

[4] ESO, *http://www.eso.org/sci/software/reflex/*. *Reflex User Manual*. VLT-SPE-XXX-XXX. 18, 40, 41