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 GRAVITY Tutorial

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

Issue 0.2

Date 2017-04-26

Prepared: `C. Hummel` `2017-04-26`
Name Date Signature

Approved: –
Name Date Signature

Released: –
Name Date Signature

Change record

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
0.1	21/06/2016	All	Initial Release

This page was intentionally left blank

Contents

1 Introduction And Scope

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

Reflex and the data reduction workflows have been developed by ESO and instrument consortia and they are fully supported. If you have any issue, please contact *usd-help@eso.org* for further support.

This document is a tutorial designed to enable the user to employ the GRAVITY workflow to reduce his/her data in a user-friendly way, concentrating on high-level issues such as data reduction quality and signal-to-noise (S/N) optimisation.

A workflow accepts science and calibration data, as 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 GRAVITY single and dual mode observations only via the GRAVITY `Reflex` workflow. For more detail on the pipeline, the user is referred to the (preliminary) pipeline manual and the GRAVITY user manual ([2][2]) and to the ESO instrument web pages [3] for more information on the instrument itself as well as a summary of available documentation, recent news, and tools.

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 `usd-help@eso.org`.

---

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

[2] available at: http://www.eso.org/sci/facilities/paranal/instruments/gravity/doc

[3] http://www.eso.org/sci/facilities/paranal/instruments/gravity.html

2 Workflow Status

The GRAVITY Reflex workflow, in its current version, is preliminary but is capable, together with its underlying GRAVITY pipeline, of delivering uncalibrated data products. The Reflex workflows are built upon the GRAVITY pipeline delivered by LESIA, Observatoire de Paris.

The first step of the GRAVITY 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 GRAVITY Reflex workflow will correct the frames for their dark level and structure, flat-field the data, and compute a wavelength solution, A large number of data products are created and retained for the User to assess the quality of the pipeline processing.

Calibrating the science observations has to be done manually as the *gravity_viscal* workflow is not functional yet.

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 and the experience of GRAVITY use, the pipeline parameters have been set to default values that deliver the best results for the most cases. However, the User should make an effort to adjust and experiment with the parameters to achieve the best results.

3 Software Installation

Reflex 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 `macports` repositories support OS X, while the `rpm/yum` repositories support Fedora 20 to 25. For any other operating system it is recommended to use the `install_esoreflex` script.

3.1 Installing Reflex workflows via `macports`

This method is supported for the OS X operating system. It is assumed that macports (*http://www.macports.org*) and java are installed. If you have any problem with this installation method, please read the full documentation at
http://www.eso.org/sci/software/pipelines/installation/macports.html.

For a quick installation, the following steps will install the ESO pipeline macports repository, the GRAVITY pipeline, including the Reflex workflow support and Reflex itself:

- Set up the repository:

```
# curl ftp://ftp.eso.org/pub/dfs/pipelines/repositories/macports/setup/Portfile -o Portfile
# sudo port install
# sudo port sync
```

- Install the GRAVITY pipeline:
  `# sudo port install esopipe-gravity-all`

3.2 Installing Reflex workflows via `rpm/yum`

This method is supported for Fedora 20/21/22/23/24/25 operating systems. If you have any problem with this installation method, please read the full documentation at
http://www.eso.org/sci/software/pipelines/installation/rpm.html.

For a quick installation, the following steps will install the ESO pipeline rpm repository, the GRAVITY pipeline, including the Reflex workflow support and Reflex itself:

- Set up the repository for Fedora 20/21:

```
# sudo yum install yum-utils
# sudo yum-config-manager \
  --add-repo=ftp://ftp.eso.org/pub/dfs/pipelines/repositories/fedora/esorepo.repo
```

- Set up the repository for Fedora 22/23/24/25:

```
# sudo dnf install dnf-plugins-core
# sudo dnf config-manager \
  --add-repo=ftp://ftp.eso.org/pub/dfs/pipelines/repositories/fedora/esorepo.repo
```

- Install the GRAVITY pipeline (Fedora 20/21):
```
# sudo yum install esopipe-gravity-all
```
- Install the GRAVITY pipeline (Fedora 22/23/24/25):
```
# sudo dnf install esopipe-gravity-all
```

3.3 Installing Reflex workflows via `install_esoreflex`

The software pre-requisites for `Reflex 2.8.5` may be found at:
http://www.eso.org/sci/software/pipelines/reflex_workflows

To install the `Reflex 2.8.5` software and demo data, please follow these instructions:

1. From any directory, download the installation script:
```
wget ftp://ftp.eso.org/pub/dfs/reflex/install_esoreflex
```
2. Make the installation script executable:
```
chmod u+x install_esoreflex
```
3. Execute the installation script:
```
./install_esoreflex
```
   and the script will ask you to specify three directories: the download directory `<download_dir>`, the software installation directory `<install_dir>`, and the directory to be used to store the demo data `<data_dir>`. If you do not specify these directories, then the installation script will create them in the current directory with default names.
4. You will be asked whether you want to use your Internet connection. Unless you want to reuse already downloaded packages (only advanced users), use the default Yes.
5. You will be given a choice of pipelines (with the corresponding workflows) to install. Please specify the numbers for the pipelines you require, separated by a space, or type “A” for all pipelines.
6. For the pipelines to be installed you will be prompted for the demo data sets to be installed. Type “A” for all demo datasets. Take into account that if you are installing in a directory that already contains data, it won’t be removed.
7. The script will also detect whether previous versions of the workflows or Reflex were installed and in this case you have the option to update links or remove obsolete cache directories. It is advised to use the defaults.
8. If some of the prerequisite binaries for `Reflex` are not under one of the paths indicated by the command,

```
getconf PATH
```

then you will need to add the appropriate paths as a colon separated list to the `esoreflex.path` parameter in the configuration file `<install_dir>/etc/esoreflex.rc`. This will usually be necessary when the FITS viewer (`fv`) is installed outside of `/usr/bin`. As an example, assume `fv` is installed into the directory `/usr/local/fv5.4`, the file `esoreflex.rc` should then have the line setting `esoreflex.path` look similar to the following:

```
esoreflex.path=/usr/local/fv5.4
```

In the case of OS X `/Applications/fv.app/Contents/MacOS/` is the typically installation directory. Thus, this should be similar to the following line instead:

```
esoreflex.path=/opt/local/bin:/Applications/fv.app/Contents/MacOS
```

9. 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 allow you to run the `Reflex GRAVITY` 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 ~5 GB, ~9 GB and ~8 GB of free disk space for the directories `<download_dir>`, `<install_dir>` and `<data_dir>`, respectively.

The raw input consists of OBJ-SKY-OBJ-SKY sequences of:

1. two calibrator OBs of HIP 64314, and
2. one science OB of a double star, HD 114529.

The data set also includes the DARK, P2VM, FLAT, WAVE, and WAVESC files. The latter is currently not supported by the pipeline. 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 obs.name dpr.type dpr.catg ins.filt1.name ins.filt2.name det2.seq1.dit
```

Table 4.1: The GRAVITY Reflex workflow tutorial data set, single field mode data with medium spectral resolution and no polarization.

File	OBS.NAME	DPR.TYPE	DPR.CATG	FILT1.NAME	FILT2.NAME	DET2.SEQ1.DIT
GRAVI.2016-06-22T16:26:14.536.fits	Calibration	DARK	CALIB	OUT	MED	0.300
GRAVI.2016-06-22T16:27:29.540.fits	Calibration	FLAT	CALIB	OUT	MED	0.300
GRAVI.2016-06-22T16:28:05.542.fits	Calibration	FLAT	CALIB	OUT	MED	0.300
GRAVI.2016-06-22T16:28:41.544.fits	Calibration	FLAT	CALIB	OUT	MED	0.300
GRAVI.2016-06-22T16:29:17.546.fits	Calibration	FLAT	CALIB	OUT	MED	0.300
GRAVI.2016-06-22T16:30:02.549.fits	Calibration	WAVESC	CALIB	OUT	MED	0.300
GRAVI.2016-06-22T16:35:38.568.fits	Calibration	WAVE	CALIB	OUT	MED	0.300
GRAVI.2016-06-22T16:40:23.583.fits	Calibration	P2VM	CALIB	OUT	MED	0.300
GRAVI.2016-06-22T16:42:59.592.fits	Calibration	P2VM	CALIB	OUT	MED	0.300
GRAVI.2016-06-22T16:45:35.601.fits	Calibration	P2VM	CALIB	OUT	MED	0.300
GRAVI.2016-06-22T16:48:11.610.fits	Calibration	P2VM	CALIB	OUT	MED	0.300
GRAVI.2016-06-22T16:50:47.619.fits	Calibration	P2VM	CALIB	OUT	MED	0.300
GRAVI.2016-06-22T16:53:23.628.fits	Calibration	P2VM	CALIB	OUT	MED	0.300
GRAVI.2016-06-22T17:11:47.690.fits	Calibration	DARK	CALIB	OUT	MED	10.000
GRAVI.2016-06-23T00:19:55.884.fits	CAL_HIP64314	OBJECT,SINGLE	CALIB	OUT	MED	10.000
GRAVI.2016-06-23T00:24:37.900.fits	CAL_HIP64314	SKY,SINGLE	CALIB	OUT	MED	10.000
GRAVI.2016-06-23T00:29:22.916.fits	CAL_HIP64314	OBJECT,SINGLE	CALIB	OUT	MED	10.000
GRAVI.2016-06-23T00:34:04.932.fits	CAL_HIP64314	SKY,SINGLE	CALIB	OUT	MED	10.000
GRAVI.2016-06-23T00:47:52.978.fits	SCI_HD114529	OBJECT,SINGLE	SCIENCE	OUT	MED	10.000
GRAVI.2016-06-23T00:52:34.994.fits	SCI_HD114529	SKY,SINGLE	SCIENCE	OUT	MED	10.000
GRAVI.2016-06-23T00:57:20.010.fits	SCI_HD114529	OBJECT,SINGLE	SCIENCE	OUT	MED	10.000
GRAVI.2016-06-23T01:02:02.026.fits	SCI_HD114529	SKY,SINGLE	SCIENCE	OUT	MED	10.000
GRAVI.2016-06-23T01:17:05.077.fits	CAL_HIP64314	OBJECT,SINGLE	CALIB	OUT	MED	10.000
GRAVI.2016-06-23T01:21:47.092.fits	CAL_HIP64314	SKY,SINGLE	CALIB	OUT	MED	10.000
GRAVI.2016-06-23T01:26:35.108.fits	CAL_HIP64314	OBJECT,SINGLE	CALIB	OUT	MED	10.000
GRAVI.2016-06-23T01:31:20.125.fits	CAL_HIP64314	SKY,SINGLE	CALIB	OUT	MED	10.000

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 GRAVITY demo data set supplied with the `Reflex 2.8.5` release. By following these steps, the user should have enough information to perform a reduction of his/her own data without any further reading:

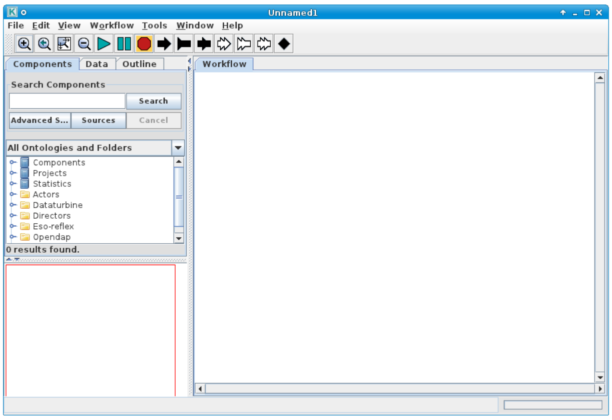


Figure 5.1: *The empty* `Reflex` *canvas.*

1. Start the `Reflex` application:

   ```
   esoreflex &
   ```

   The empty `Reflex` canvas as shown in Figure 5.1 will appear.

2. Now open the GRAVITY workflow by clicking on `File -> Open File`, selecting first `gravity-1.0.5` and then the file `gravity_wkf.xml` in the file browser. You will be presented with the workflow canvas shown in Figure 5.2. Note that the workflow will appear as a canvas in a new window.

3. To aid in the visual tracking of the reduction cascade, it is advisable to use component (or actor) highlighting. Click on `Tools -> Animate at Runtime`, enter the number of milliseconds representing the animation interval (100 ms is recommended), and click `OK`.

4. Under "Setup Directories" in the workflow canvas there are seven parameters that specify important directories (green dots). Changing the value of `ROOT_DATA_DIR` and/or `RAW_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` 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.

---

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

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 `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 5.3) 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. By default all complete DataSets which have not yet been reduced will be 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 called *Product Explorer* will appear showing the datasets which have been so far reduced together with the list of final products. This actor allows the user to inspect the final data products, as well as to search and inspect the input data used to create any of the products of the workflow. Figure 5.4 shows the Product Explorer window.

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.

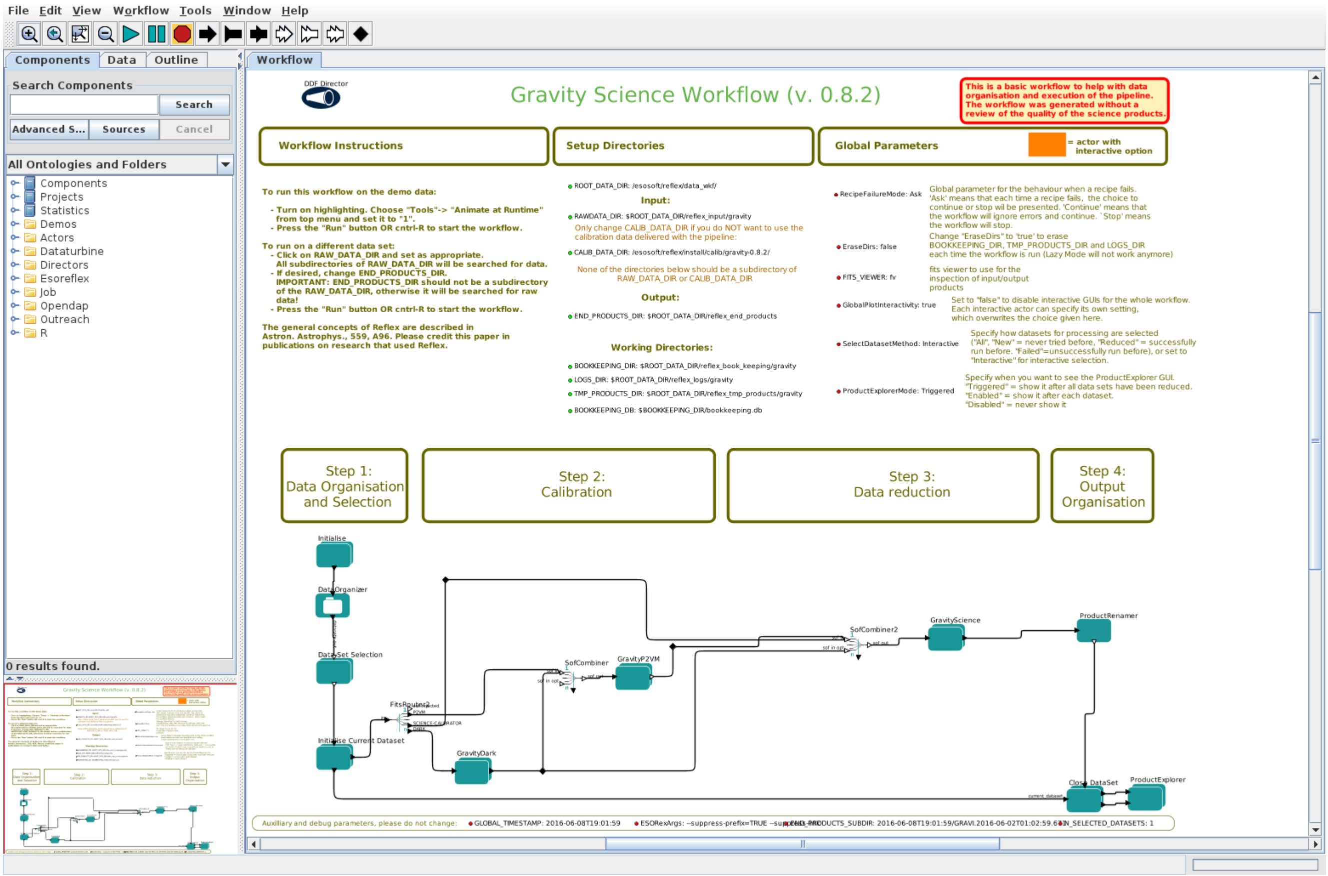
Well done! You have successfully completed the quick start section and you should be able to use this knowledge to reduce your own data. However, there are many interesting features of `Reflex` and the GRAVITY workflow that merit a look at the rest of this tutorial.

5.1 Reducing Your Own Science Data

To reduce your own science data, simply change the path to the raw data directory. This is defined at the top of the workflow window in the area labeled `Setup Directories`. Simply double click on the `RAW_DATA_DIR`, enter the path to your raw science directory and then re-run the workflow in the same way as was done for the tutorial demo data. **Please note that the GRAVITY pipeline requires at least 24 GB of RAM installed for mode HIGH-COMBINED and even more for HIGH-SPLIT!**.

---

[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.



This is a basic workflow to help with data organisation and execution of the pipeline. The workflow was generated without a review of the quality of the science products.

To run this workflow on the demo data:

- Turn on highlighting. Choose "Tools"-> "Animate at Runtime" from top menu and set it to "1".
- Press the "Run" button OR cntrl-R to start the workflow.

To run on a different data set:
- Click on RAW_DATA_DIR and set as appropriate. All subdirectories of RAW_DATA_DIR will be searched for data.
- If desired, change END_PRODUCTS_DIR. IMPORTANT: END_PRODUCTS_DIR should not be a subdirectory of the RAW_DATA_DIR, otherwise it will be searched for raw data!
- Press the "Run" button OR cntrl-R to start the workflow.

The general concepts of Reflex are described in Astron. Astrophys., 559, A96. Please credit this paper in publications on research that used Reflex.

- ROOT_DATA_DIR: /esosoft/reflex/data_wkf/

Input:

- RAWDATA_DIR: $ROOT_DATA_DIR/reflex_input/gravity

Only change CALIB_DATA_DIR if you do NOT want to use the calibration data delivered with the pipeline:

- CALIB_DATA_DIR: /esosoft/reflex/install/calib/gravity-0.8.2/

None of the directories below should be a subdirectory of RAW_DATA_DIR or CALIB_DATA_DIR

Output:

- END_PRODUCTS_DIR: $ROOT_DATA_DIR/reflex_end_products

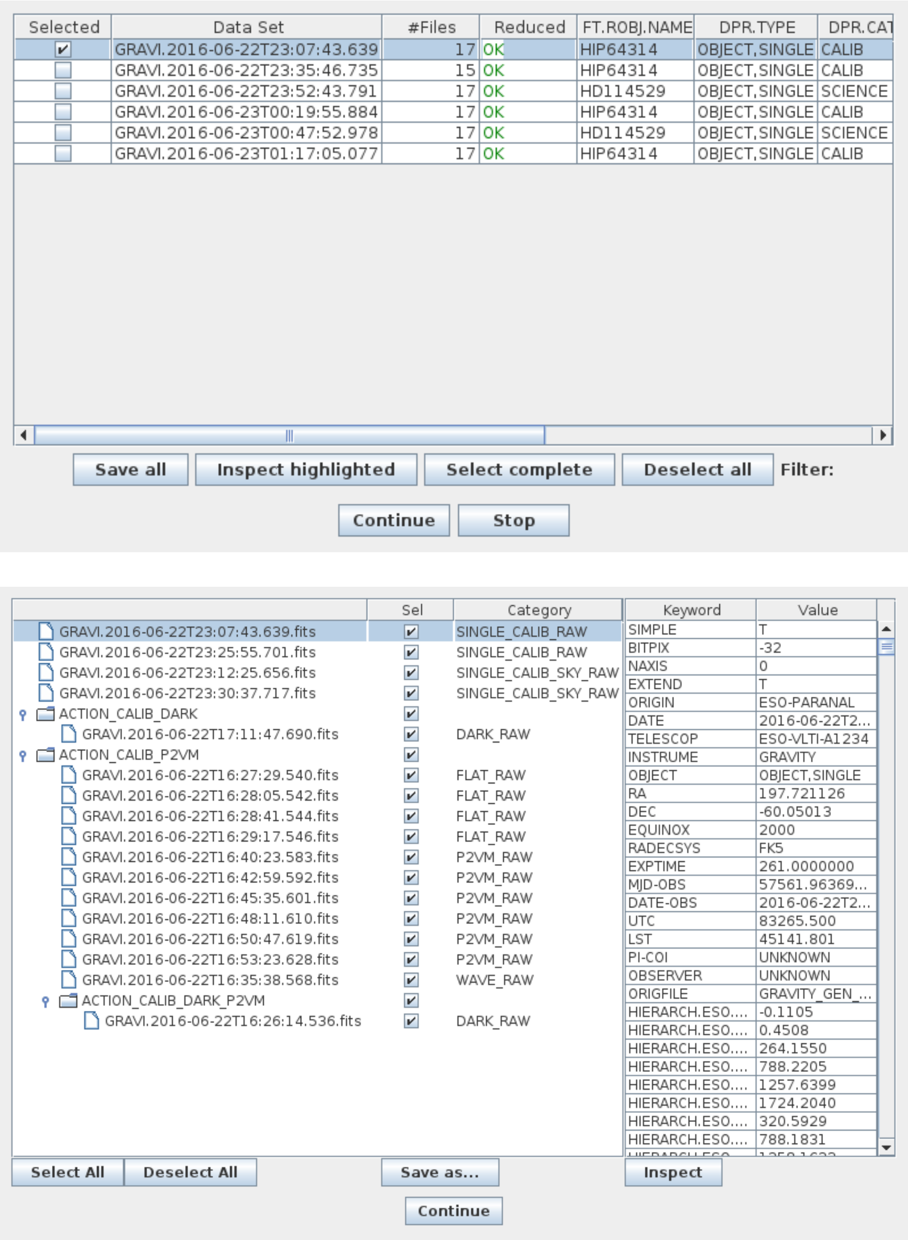
Working Directories:

- BOOKKEEPING_DIR: $ROOT_DATA_DIR/reflex_book_keeping/gravity
- LOGS_DIR: $ROOT_DATA_DIR/reflex_logs/gravity
- TMP_PRODUCTS_DIR: $ROOT_DATA_DIR/reflex_tmp_products/gravity
- BOOKKEEPING_DB: $BOOKKEEPING_DIR/bookkeeping.db

- RecipeFailureMode: Ask — Global parameter for the behaviour when a recipe fails. 'Ask' means that each time a recipe fails, the choice to continue or stop wil be presented. 'Continue' means that the workflow will ignore errors and continue. `Stop' means the workflow will stop.
- EraseDirs: false — Change "EraseDirs" to 'true' to erase BOOKKEEPING_DIR, TMP_PRODUCTS_DIR and LOGS_DIR each time the workflow is run (Lazy Mode will not work anymore)
- FITS_VIEWER: fv — fits viewer to use for the inspection of input/output products
- GlobalPlotInteractivity: true — Set to "false" to disable interactive GUIs for the whole workflow. Each interactive actor can specify its own setting, which overwrites the choice given here.
- SelectDatasetMethod: Interactive — Specify how datasets for processing are selected ("All", "New" = never tried before, "Reduced" = successfully run before. "Failed"=unsuccessfully run before), or set to "Interactive" for interactive selection.
- ProductExplorerMode: Triggered — Specify when you want to see the ProductExplorer GUI. "Triggered" = show it after all data sets have been reduced. "Enabled" = show it after each dataset. "Disabled" = never show it

Auxilliary and debug parameters, please do not change: ● GLOBAL_TIMESTAMP: 2016-06-08T19:01:59 ● ESORexArgs: --suppress-prefix=TRUE --sup ● END_PRODUCTS_SUBDIR: 2016-06-08T19:01:59/GRAVI.2016-06-02T01:02:59.6 ● N_SELECTED_DATASETS: 1

Figure 5.2: *GRAVITY workflow general layout.*

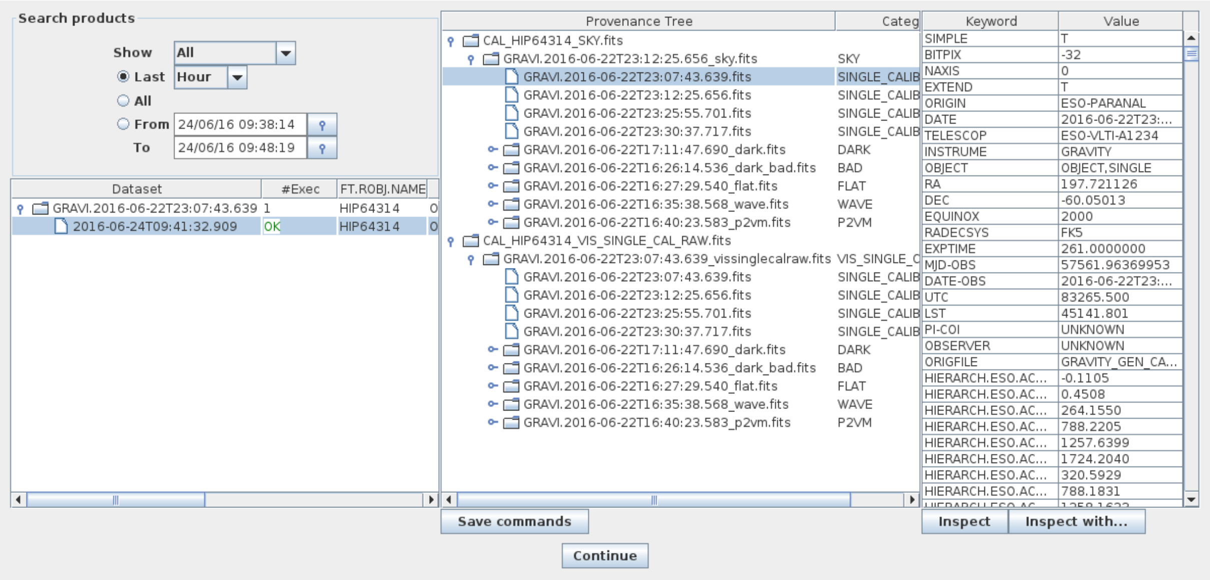


Selected	Data Set	#Files	Reduced	FT.ROBJ.NAME	DPR.TYPE	DPR.CAT
☑	GRAVI.2016-06-22T23:07:43.639	17	OK	HIP64314	OBJECT,SINGLE	CALIB
☐	GRAVI.2016-06-22T23:35:46.735	15	OK	HIP64314	OBJECT,SINGLE	CALIB
☐	GRAVI.2016-06-22T23:52:43.791	17	OK	HD114529	OBJECT,SINGLE	SCIENCE
☐	GRAVI.2016-06-23T00:19:55.884	17	OK	HIP64314	OBJECT,SINGLE	CALIB
☐	GRAVI.2016-06-23T00:47:52.978	17	OK	HD114529	OBJECT,SINGLE	SCIENCE
☐	GRAVI.2016-06-23T01:17:05.077	17	OK	HIP64314	OBJECT,SINGLE	CALIB

	Sel	Category
GRAVI.2016-06-22T23:07:43.639.fits	☑	SINGLE_CALIB_RAW
GRAVI.2016-06-22T23:25:55.701.fits	☑	SINGLE_CALIB_RAW
GRAVI.2016-06-22T23:12:25.656.fits	☑	SINGLE_CALIB_SKY_RAW
GRAVI.2016-06-22T23:30:37.717.fits	☑	SINGLE_CALIB_SKY_RAW
ACTION_CALIB_DARK	☑	
GRAVI.2016-06-22T17:11:47.690.fits	☑	DARK_RAW
ACTION_CALIB_P2VM	☑	
GRAVI.2016-06-22T16:27:29.540.fits	☑	FLAT_RAW
GRAVI.2016-06-22T16:28:05.542.fits	☑	FLAT_RAW
GRAVI.2016-06-22T16:28:41.544.fits	☑	FLAT_RAW
GRAVI.2016-06-22T16:29:17.546.fits	☑	FLAT_RAW
GRAVI.2016-06-22T16:40:23.583.fits	☑	P2VM_RAW
GRAVI.2016-06-22T16:42:59.592.fits	☑	P2VM_RAW
GRAVI.2016-06-22T16:45:35.601.fits	☑	P2VM_RAW
GRAVI.2016-06-22T16:48:11.610.fits	☑	P2VM_RAW
GRAVI.2016-06-22T16:50:47.619.fits	☑	P2VM_RAW
GRAVI.2016-06-22T16:53:23.628.fits	☑	P2VM_RAW
GRAVI.2016-06-22T16:35:38.568.fits	☑	WAVE_RAW
ACTION_CALIB_DARK_P2VM	☑	
GRAVI.2016-06-22T16:26:14.536.fits	☑	DARK_RAW

Keyword	Value
SIMPLE	T
BITPIX	-32
NAXIS	0
EXTEND	T
ORIGIN	ESO-PARANAL
DATE	2016-06-22T2...
TELESCOP	ESO-VLTI-A1234
INSTRUME	GRAVITY
OBJECT	OBJECT,SINGLE
RA	197.721126
DEC	-60.05013
EQUINOX	2000
RADECSYS	FK5
EXPTIME	261.0000000
MJD-OBS	57561.96369...
DATE-OBS	2016-06-22T2...
UTC	83265.500
LST	45141.801
PI-COI	UNKNOWN
OBSERVER	UNKNOWN
ORIGFILE	GRAVITY_GEN_...
HIERARCH.ESO....	-0.1105
HIERARCH.ESO....	0.4508
HIERARCH.ESO....	264.1550
HIERARCH.ESO....	788.2205
HIERARCH.ESO....	1257.6399
HIERARCH.ESO....	1724.2040
HIERARCH.ESO....	320.5929
HIERARCH.ESO....	788.1831

Figure 5.3: Upper panel: the "Select Datasets" pop-up window; Lower panel: the "Selected Frame" pop-up window, obtained after pressing Inspect highlighted in the Select Datasets window.



Search products

Show: All

● Last: Hour

○ All

○ From: 24/06/16 09:38:14

To: 24/06/16 09:48:19

Dataset	#Exec	FT.ROBJ.NAME	
GRAVI.2016-06-22T23:07:43.639	1	HIP64314	O
2016-06-24T09:41:32.909	OK	HIP64314	O

Provenance Tree	Categ
CAL_HIP64314_SKY.fits	
GRAVI.2016-06-22T23:12:25.656_sky.fits	SKY
GRAVI.2016-06-22T23:07:43.639.fits	SINGLE_CALIB
GRAVI.2016-06-22T23:12:25.656.fits	SINGLE_CALIB
GRAVI.2016-06-22T23:25:55.701.fits	SINGLE_CALIB
GRAVI.2016-06-22T23:30:37.717.fits	SINGLE_CALIB
GRAVI.2016-06-22T17:11:47.690_dark.fits	DARK
GRAVI.2016-06-22T16:26:14.536_dark_bad.fits	BAD
GRAVI.2016-06-22T16:27:29.540_flat.fits	FLAT
GRAVI.2016-06-22T16:35:38.568_wave.fits	WAVE
GRAVI.2016-06-22T16:40:23.583_p2vm.fits	P2VM
CAL_HIP64314_VIS_SINGLE_CAL_RAW.fits	
GRAVI.2016-06-22T23:07:43.639_vissinglecalraw.fits	VIS_SINGLE_C
GRAVI.2016-06-22T23:07:43.639.fits	SINGLE_CALIB
GRAVI.2016-06-22T23:12:25.656.fits	SINGLE_CALIB
GRAVI.2016-06-22T23:25:55.701.fits	SINGLE_CALIB
GRAVI.2016-06-22T23:30:37.717.fits	SINGLE_CALIB
GRAVI.2016-06-22T17:11:47.690_dark.fits	DARK
GRAVI.2016-06-22T16:26:14.536_dark_bad.fits	BAD
GRAVI.2016-06-22T16:27:29.540_flat.fits	FLAT
GRAVI.2016-06-22T16:35:38.568_wave.fits	WAVE
GRAVI.2016-06-22T16:40:23.583_p2vm.fits	P2VM

Keyword	Value
SIMPLE	T
BITPIX	-32
NAXIS	0
EXTEND	T
ORIGIN	ESO-PARANAL
DATE	2016-06-22T23:...
TELESCOP	ESO-VLTI-A1234
INSTRUME	GRAVITY
OBJECT	OBJECT,SINGLE
RA	197.721126
DEC	-60.05013
EQUINOX	2000
RADECSYS	FK5
EXPTIME	261.0000000
MJD-OBS	57561.96369953
DATE-OBS	2016-06-22T23:...
UTC	83265.500
LST	45141.801
PI-COI	UNKNOWN
OBSERVER	UNKNOWN
ORIGFILE	GRAVITY_GEN_CA...
HIERARCH.ESO.AC...	-0.1105
HIERARCH.ESO.AC...	0.4508
HIERARCH.ESO.AC...	264.1550
HIERARCH.ESO.AC...	788.2205
HIERARCH.ESO.AC...	1257.6399
HIERARCH.ESO.AC...	1724.2040
HIERARCH.ESO.AC...	320.5929
HIERARCH.ESO.AC...	788.1831

Figure 5.4: *The GRAVITY product explorer.*

6 GRAVITY Reduced Data Description

A number of intermediate pipeline products from the tutorial data set can be found in subdirectories of the `TMP_PRODUCT_DIR`. These master calibration files are summarized in Table 6.1.

Table 6.1: The GRAVITY Reflex workflow tutorial data set: calibration products

File	PRO.CATG	Description
gravity_p2vm:		
GRAVI.2016-06-22T16:26:14.536_dark_bad.fits	BAD	bad pixel map
GRAVI.2016-06-22T16:27:29.540_flat.fits	FLAT	flat field
GRAVI.2016-06-22T16:35:38.568_wave.fits	WAVE	wavelength calibration
GRAVI.2016-06-22T16:40:23.583_p2vm.fits	P2VM	p2vm
gravity_dark:		
GRAVI.2016-06-22T16:26:14.536_dark.fits	DARK	dark frame

The final products of the reduction pipeline can be found in the directory `END_PRODUCT_DIR` that is defined in the “Setup Directories” section at the top of the workflow.

The science data products from the two tutorial data sets are summarized in tables 6.2. Their description will be done in Section 6.1.

Table 6.2: The GRAVITY Reflex workflow science products from the tutorial data set nod-to-sky mode (all K-band).

File	Description	
File	PRO.CATG	Description
GRAVI.2016-06-22T23:07:43.639/CAL_HIP64314_SKY.fits	SKY	sky frame
GRAVI.2016-06-22T23:07:43.639/CAL_HIP64314_VIS_SINGLE_CAL_RAW.fits	VIS_SINGLE_CAL_RAW	uncalibrated averaged visibility data
GRAVI.2016-06-22T23:35:46.735/CAL_HIP64314_SKY.fits	SKY	sky frame
GRAVI.2016-06-22T23:35:46.735/CAL_HIP64314_VIS_SINGLE_CAL_RAW.fits	VIS_SINGLE_CAL_RAW	uncalibrated averaged visibility data
GRAVI.2016-06-22T23:52:43.791/SCI_HD114529_SKY.fits	SKY	sky frame
GRAVI.2016-06-22T23:52:43.791/SCI_HD114529_VIS_SINGLE_SCI_RAW.fits	VIS_SINGLE_SCI_RAW	uncalibrated averaged visibility data
GRAVI.2016-06-23T00:19:55.884/CAL_HIP64314_SKY.fits	SKY	sky frame
GRAVI.2016-06-23T00:19:55.884/CAL_HIP64314_VIS_SINGLE_CAL_RAW.fits	VIS_SINGLE_CAL_RAW	uncalibrated averaged visibility data
GRAVI.2016-06-23T00:47:52.978/SCI_HD114529_SKY.fits	SKY	sky frame
GRAVI.2016-06-23T00:47:52.978/SCI_HD114529_VIS_SINGLE_SCI_RAW.fits	VIS_SINGLE_SCI_RAW	uncalibrated averaged visibility data
GRAVI.2016-06-23T01:17:05.077/CAL_HIP64314_SKY.fits	SKY	sky frame
GRAVI.2016-06-23T01:17:05.077/CAL_HIP64314_VIS_SINGLE_CAL_RAW.fits	VIS_SINGLE_CAL_RAW	uncalibrated averaged visibility data

6.1 Description of final products

In this section we provide a short description of the workflow science products that are stored in the `reflex_end_produ` directory. They are produced by the pipeline recipes *gravity_vis*. For further information, please consult the GRAVITY pipeline manual. In the list, the products are identified by their PRO.CATG keyword.

- `VIS_SINGLE_CAL_RAW`. Product of the recipe *gravity_vis* for a CAL observation. This file conforms to the OIFITS file format used for interferometric data [1]. If the visibilities of the calibrator are corrected for the non-zero diameter of the calibrator star, the resulting visibility amplitudes are called the *transfer*

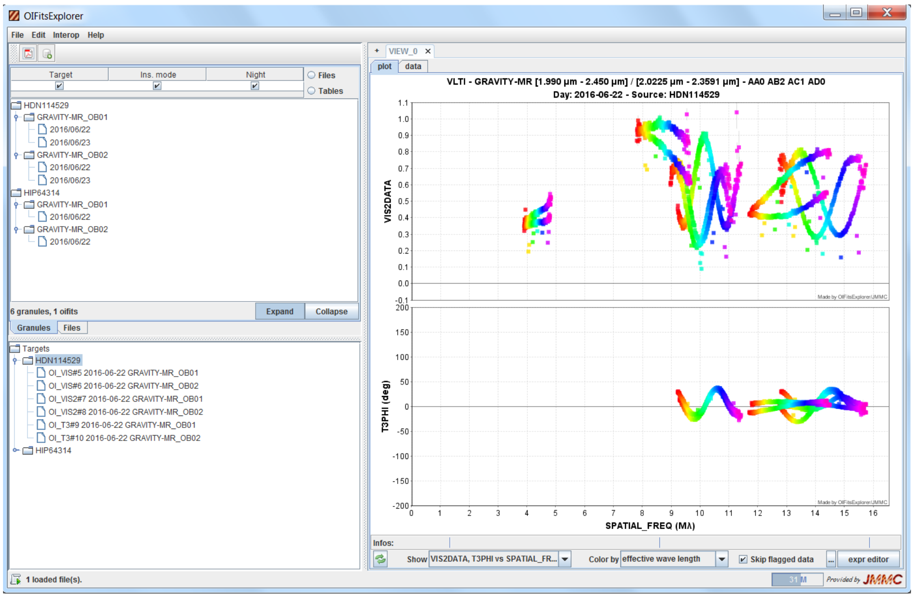


Figure 6.1: OIFitsExplorer view of the demo data (all science files combined).

function.

- `VIS_SINGLE_SCI_RAW`. Product of the recipe *gravity_vis* for a SCI observation. This file conforms to the OIFITS file format used for interferometric data [1]. The visibility data are averaged, but not calibrated. Currently, calibration has to be performed manually by dividing the science visibility by the calibrator transfer function. One can use the OIFitsExplorer (`http://www.jmmc.fr/oifitsexplorer_page.htm`) to examine the data and LitPro (`http://www.jmmc.fr/litpro_page.htm`) to fit simple models.

7 About The Reflex Canvas

7.1 Saving And Loading Workflows

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

7.2 Buttons

At the top of the `Reflex` 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.

7.3 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.

8 The GRAVITY Workflow

The GRAVITY 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.

8.1 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; [3]).

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.

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

The parameter `RecipeFailureMode` controls the behaviour in case that a recipe fails. If set to `Continue`, the workflow will trigger the next recipes as usual, but without the output of the failing recipe, which in most of the cases will lead to further 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 `GlobalPlotInteractivity` controls whether the interactive windows will appear for those windows which are *enabled* by default. The possible values are `true, false`. Take into account that some windows are disabled in the default configuration and therefore are not affected by this parameter.

The parameter `ProductExplorerMode` controls whether the `ProductExplorer` actor will show its window or not. The possible values are `Enabled`, `Disabled` and `Triggered`. The latter, recommended, means that the `ProductExplorer` actor will be shown only at the end of the workflow execution.

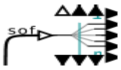
8.2 Workflow Actors

8.2.1 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 `Data Organiser` actor.
- 

 - The `Data Set Chooser` actor (inside a composite actor).
- 

 - The `Fits Router` actor
- 

 - The `Product Renamer` actor.
- 

 - The `Product Explorer` 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).

8.2.2 GRAVITY-specific actors: the workflow data-reduction cascade

The present GRAVITY workflow is designed to process the datasets according to a specific data reduction cascade. This cascade triggers a series of pipeline recipes, which are associated to the following composite actors:

- 

 `GravityDark`: it executes the recipe *gravity_dark* . It processes the dark frames both for the recipe *gravity_p2vm* and *gravity_vis*.

- `GravityP2VM`: it executes the recipe *gravity_p2vm*. It requires products of *gravity_dark* as input, as well as raw P2VM, FLAT, and WAVE files.

We refer the user to the GRAVITY pipeline manual for a complete description of the recipes and their parameters.

As noted in the Quick Start Section 5, the workflow will then proceed through its remaining processing steps and write out all pipeline products to the end products directory (specified by the parameter `END_PRODUCTS_DIR` under "Setup Directories" in the workflow canvas). The science data products from the tutorial data set are summarized in section 6 in table 6.2. The intermediate pipeline calibration products can be found in subdirectories of the `TMP_PRODUCT_DIR` and are summarized in section 6 and in table 6.1.

8.2.3 Lazy Mode

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

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

The forced rereduction of data at each execution may sometimes be desirable. To force a rereduction of all data for all `RecipeExecuter` actors in the workflow (i.e. to disable Lazy Mode for the whole workflow), set the `EraseDirs` parameter under the "Global Parameters" area of the workflow canvas to `true`. This will then remove all previous results as well. To force a rereduction of data for any single `RecipeExecuter` actor in the workflow (which will be inside the relevant composite actor), right-click the `RecipeExecuter` actor, select `Configure Actor`, and uncheck the Lazy Mode parameter tick-box in the "Edit parameters" window that is displayed. If the Lazy Mode is switched off for an actor, all subsequent actors that use products from that one will also reprocess the data, as they see new products.

9 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.

- **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 ([3]). 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 ([3]) 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 ([3]) 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.
- **If I enter "-" into an empty integer parameter of an interactive window it is automaticaly completed to "-1". Why?**

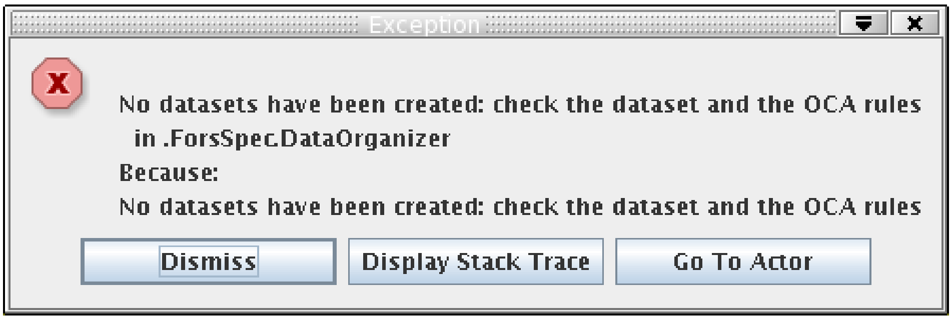
  The parameters are validated for correctness according to their type (e.g. string, integer, float). In the case of an integer or float parameter "-" alone is considered an invalid input and is therefore automatically completed to "-1". This is part of the validation of input done by the WxPython library.
- **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.
- **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 at the end of the reduction will list all the previous reductions of a particular data set. 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.

10 Troubleshooting



No datasets have been created: check the dataset and the OCA rules in .ForsSpec.DataOrganizer
Because:
No datasets have been created: check the dataset and the OCA rules

Figure 10.1: *TheDataOrganizer interactive window reports an error ":No DataSets have been created, check the data set and the OCA rules.".*

1. **I downloaded the data from the ESO archive, put them into a new directory, tried to run `Reflex` on them, but**
   (a) **it crashs**
   This may happen if one of the files was downloaded only partially (check for a file with the extension `fits.Z.part`. You will have to download that file again in order to have an uncorrrupted file (and remove the partial one).
   (b) **it fails with error message ":No DataSets have been created, check the data set and the OCA rules."(see Figure 10.1.)**
   This error may be due to the fact that the data provided by the ESO archive are compressed (`<filename>.fits.Z`). Please remember to uncompress the data before executing `Reflex`.

   (c) **all DataSets are greyed out in the DataSets interactive window.**
   The ESO archive used with CalSelector does not always supply all static calibration files. As a consequence some/all DataSets are greyed out because they were missing such required data.
   Missing static calibration should be found by `reflex` in
   `<install_directory>/calib/<pipeline_version>/cal`.

2. **The "Select DataSets" window displays my DataSets, but some/all of them are greyed out. What is going on?**
   If a DataSet in the "Select DataSets" window is greyed out, then it means that the DataSet that was constructed is missing some key calibration(s) (i.e. the DataSet is incomplete). To find out what calibration(s) are missing from a greyed out DataSet, click on the DataSet in question to highlight it in blue, and then click on the button `Inspect Highlighted`. The "Select Frames" window that appears will report the category of the calibration products that are missing (e.g. `DARK`). From this the user has then to determine the missing raw data (in this case dark frames). If static calibrations are missing the mechanism unfortunately does not work, but should be found by `reflex` in
   `<install_directory>/calib/<pipeline_version>/cal`

Acknowledgements

This tutorial has contributions from the GRAVITY IOT.

[1] G. Duvert, J. Young, and C.A. Hummel. OIFITS 2: the 2nd version of the Data Exchange Standard for Optical (Visible/IR) Interferometry. *Astronomy & Astrophysics*, 597:A8, 2015. 18, 19

[2] GRAVITY IOT GRAVITY Consortium. *GRAVITY 2016 User Manual*, 2016. VLT-MAN-ESO-XXXXX-XXX. 7

[3] Forchì V. *Reflex User's Manual*. ESO/SDD/DFS, *http://www.eso.org/gasgano/*, 0.7 edition, 2012. VLT-MAN-ESO-19000-5037. 21, 24, 25