



Workflow Instructions

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 cntri-R to start the workflow.

To run on a different data set:

- Click on ROOT_DATA_DIR and set as appropriate. All subdirectories of this directory will be used as data.
- If desired, change ENO_PRODUCT_SUFFIX.
- Press the "Run" button OR cntri-R to start the workflow.

To monitor the progress of the workflow in more detail:

- Open "Window" -> "Runtime Window" in top menu before starting the workflow.

Setup Directories

Input:

• ROOT_DATA_DIR: /home/wfreudl/reflexdata/

• RAWDATA_DIR: \${ROOT_DATA_DIR}/reflex_inputs_blue

Working Directories:

• BOOKKEEPING_DIR: \${ROOT_DATA_DIR}/reflex_toolkeeping/blue

• LOGS_DIR: \${ROOT_DATA_DIR}/reflex_logs

• ETC_PRODUCTS_DIR: \${ROOT_DATA_DIR}/reflex/etc_products

Global Parameters

• ETS_VIEWER: tv

ETS viewer to use for the inspection of input/output products

• EsoReflexArgs --suppress-prefix=TRUE esoreflex arguments

• ESO_DB: /var/www/eso.org/sci/software/pipelines/reflex_workflows/DB/work.db

• ESO_DB: /var/www/eso.org/sci/software/pipelines/reflex_workflows/DB/work.db

This is set automatically

This is set automatically

The ESO Recipe Flexible Workbench

EsoReflex

II. Advanced Features

Wolfram Freudling

http://www.eso.org/sci/software/pipelines/reflex_workflows



Workflow driven Data Reduction

A&A 559, A96 (2013)

To run this
- Turn on
from top
- Press th

To run on a
- Click on
All subdi
- If desire
- Press th

To monitor
- Open "W
starting

A&A 559, A96 (2013)
DOI: [10.1051/0004-6361/201322494](https://doi.org/10.1051/0004-6361/201322494)
© ESO 2013

**Astronomy
&
Astrophysics**

Automated data reduction workflows for astronomy

The ESO Reflex environment

W. Freudling, M. Romaniello, D. M. Bramich, P. Ballester, V. Forchi, C. E. García-Dabló, S. Moehler, and M. J. Neeser

European Southern Observatory, Karl-Schwarzschild-Str. 2, 85748 Garching, Germany
e-mail: wfreudli@eso.org

Received 16 August 2013 / Accepted 14 October 2013

ABSTRACT

Context. Data from complex modern astronomical instruments often consist of a large number of different science and calibration files, and their reduction requires a variety of software tools. The execution chain of the tools represents a complex workflow that needs to be tuned and supervised, often by individual researchers that are not necessarily experts for any specific instrument.

Aims. The efficiency of data reduction can be improved by using automatic workflows to organise data and execute a sequence of data reduction steps. To realize such efficiency gains, we designed a system that allows intuitive representation, execution and modification of the data reduction workflow, and has facilities for inspection and interaction with the data.

Methods. The European Southern Observatory (ESO) has developed Reflex, an environment to automate data reduction workflows. Reflex is implemented as a package of customized components for the Kepler workflow engine. Kepler provides the graphical user interface to create an executable flowchart-like representation of the data reduction process. Key features of Reflex are a rule-based data organiser, infrastructure to re-use results, thorough book-keeping, data progeny tracking, interactive user interfaces, and a novel concept to exploit information created during data organisation for the workflow execution.

Results. Automated workflows can greatly increase the efficiency of astronomical data reduction. In Reflex, workflows can be run non-interactively as a first step. Subsequent optimization can then be carried out while transparently re-using all unchanged intermediate products. We found that such workflows enable the reduction of complex data by non-expert users and minimizes mistakes due to book-keeping errors.

Conclusions. Reflex includes novel concepts to increase the efficiency of astronomical data processing. While Reflex is a specific

= actor with
interactive option

DB and LOGS, DB
Mode won't work anymore;

set_3 This is set automatically
This is set automatically

Step 5:
Output
Organisation





Command line interface examples

Workflow Instructions

To run the workflow, do the following:

- Then click right mouse button to open context menu.
- Press the "Run" button OR ctrl+R to start the workflow.

To run on a different file set:

- esoreflex -l appropriate.
- All sub-directories of RAW_DATA_DIR will be searched for data.
- If desired, change END_PRODUCTS_DIR.
- Press the "Run" button OR ctrl+R to start the workflow.

To monitor progress of a workflow:

- Open RunTimeView window by clicking "RunTimeView" icon in the status bar starting the workflow.

Setup Directories

RAWDATA_DIR \$ROOT_DATA_DIR/reflex_keepingives_Blu

Working Directories

BOOKKEEPING_DIR \$ROOT_DATA_DIR/reflex_bookkeepingives_Blu

LOGS_DIR \$ROOT_DATA_DIR/reflex_logsives_Blu

TMP_PRODUCTS_DIR \$ROOT_DATA_DIR/reflex_tmp_productsives_Blu

Output:

\$ROOT_DATA_DIR/reflex_end_products

Global Parameters

emms_viewer: tv
It's viewer to use for the inspection of input/output products

esoreflex --suppress-prefix=TRUE esoreflex arguments

eraseDir: false
Change "EraseDir" to true to erase BOOKKEEPING_DIR, TMP_PRODUCTS_DIR and LOGS_DIR each time the workflow is run (Lazy Mode won't work anymore)

END_PRODUCT_SUFFIX: 2010-07-07T18:10:28[Science_DataSet_3]

GLOBAL_TIMESTAMP: 2010-07-07T18:10:29

= actor with interactive option

List all available workflows:

Load FORS workflow:

```
esoreflex fors_imaging
```

Process all new data in my directory without interaction:

```
esoreflex -n -RAW_DATA_DIR /data/fors_data fors_imaging
```

Rerun all previously failed datasets with nonstandard workflow parameter:

```
esoreflex -n -RAW_DATA_DIR /data/muse_data \
          -SelectDatasetMode failed -recomputeWCS true muse
```



file:/Users/wfreudli/KeplerData/workflows/MyWorkflows/fors_imaging_multitab.kar

Workflow

Components Data Outline

Search Components

Advanced ... Sources Cancel

All Ontologies and Folders

- Components
- Projects
- Statistics
- Demos
- Actors
- Dataturbine
- Directors
- Esoreflex
- Job
- MyWorkflows
- Opendap

0 results found.

FORS Workflow For Imaging Data (v. 5.1.4)

Workflow Overview:

- Step 1: Data Organisation
- Step 2: Creation of Virtual Calibration Tree
- Step 3: Science Reduction
- Step 4: Output Organisation

Global Parameters

= actor with interactive option

_data/
e/esopipes/datademo/fors/
_DIR if you do NOT want to use the
with the pipeline:
re/esopipes/datastatic/fors-5.1.4/

• 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 will be presented.'Continue' means that the workflow will ignore errors and continue. 'Stop' means the workflow will stop.
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)

• EraseDirs: false

• FITS_VIEWER: fv
Program to use for the inspection of input/output products.
Use full path name if it is not in the standard path.

• 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

Step 3: Science Reduction

Step 4: Output Organisation

execution finished: 9836 ms. Memory: 487424K Free: 121548K (25%)



Lazy mode: Don't redo unnecessary steps

Workflow Instructions

Setup Directories

Global Parameters

= actor with interactive option

- Lazy mode for recipes.

- It works by comparing the input of the current execution with *all* the previous recipe executions:

- All files must be the same
- All files must have the same checksum
- All files must have the same date
- All recipe parameters must be the same

- If a recipe at the beginning of the workflow is set to *Not-Lazy mode*, the input of the next recipes will be new and lazy mode will not be triggered.

- Lazy mode for DataOrganizer.

- It avoids the organization of all the data in subsequent workflow runs.
- It works similar to lazy mode for recipes

- Lazy mode for PythonActor:

- Checks input/outputs, without parsing them, e.g. checking files dates



Configuring ESOReflex

Workflow Instructions

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 cntri-R to start the workflow.

To run on a different data set:

- Change ROOT_DATA_DIR and set appropriate Starts At value in the Data Organisation tab.
- If needed, change MODIFICATION_DATE and
- Press the "Run" button OR cntri-R to start the workflow.

To monitor the progress of the workflow in more detail:

- esoreflex --create-config
- starting the workflow.

Setup Directories

Input:

- ROOT_DATA_DIR: /home/ffred/keflexdata
- KARDATA_DIR: \$ROOT_DATA_DIR/eflex_inputs_Ves_Blu

Working Directories:

- BOOKKEEPING_DIR: \$ROOT_DATA_DIR/reflex_junk/keepingVes_Blu

Output:

- END_PRODUCTS_DIR: \$ROOT_DATA_DIR/reflex_end_products

Global Parameters

- ERTS_VIEWER: tv

ERTS viewer to use for the inspection of input/output products

- ESOReflex --suppress-reflex=TRUE esoreflex arguments

- ESOReflex --quiet

RCOOLSPNG_DB, TMP_PRODUCTS_DB and LOGS_DB
each time the workflow is run (Lazy Mode won't work anymore)

- END_PRODUCT_SUBDIR: 2010-07-07T18:10:28/Science_dataset_3

- GLOBAL_TIMESTAMP: 2010-07-07T18:10:29

= actor with interactive option

Create configuration files `~/.esoreflex/esoreflexrc` :

Step 1:
Data Organisation
and Selection

Step 2:
Creation of Master
Calibration Files

Step 3:
Wavelength and Response
Calibration

Step 4:
Spectrum
Extraction

Step 5:
Output
Organisation

Some interesting parameters:

`esoreflex.python-command`

`esoreflex.inherit-environment`

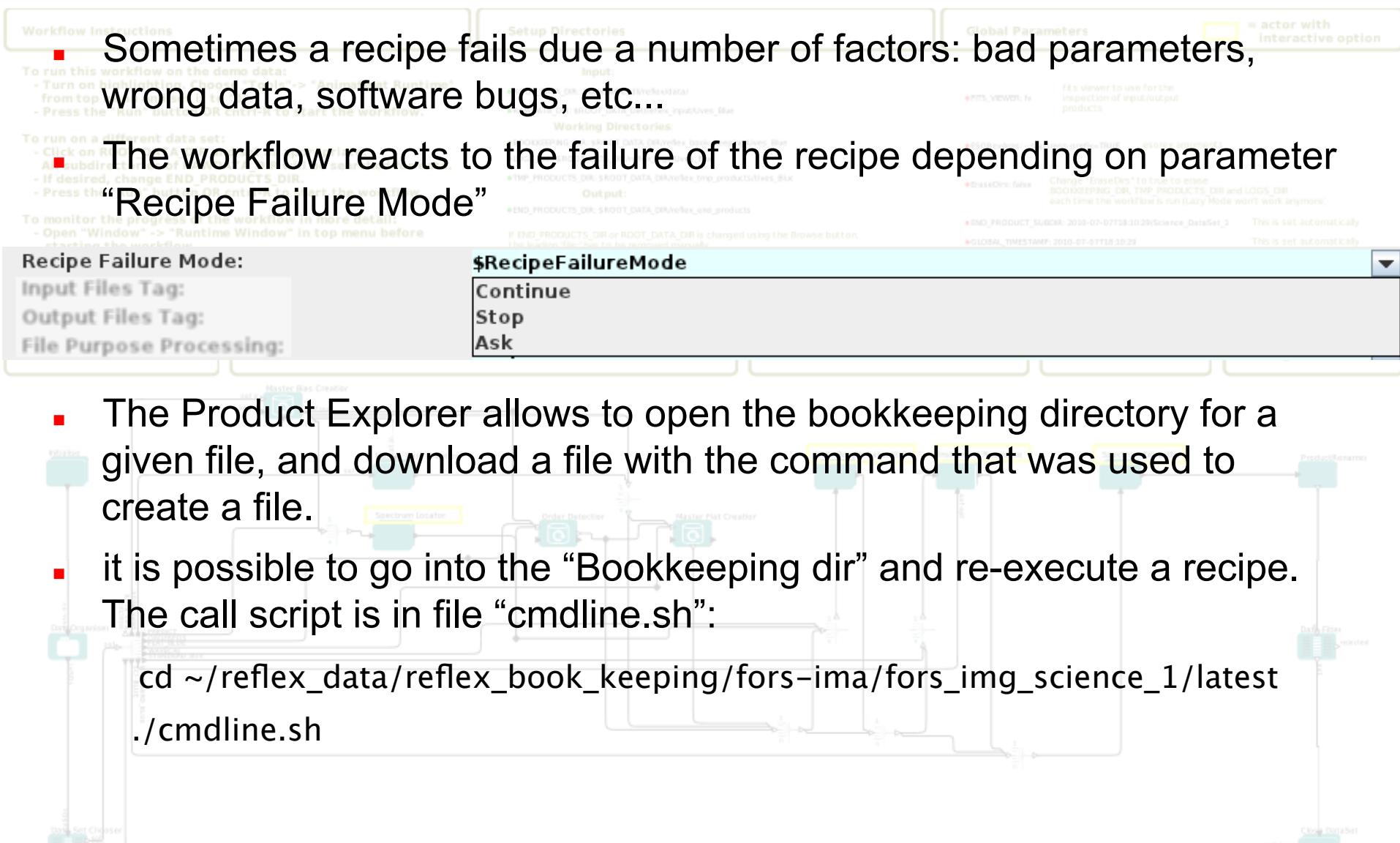
`esoreflex.python-esoreflex.python-path`



Re-executing a recipe

- Sometimes a recipe fails due to a number of factors: bad parameters, wrong data, software bugs, etc...

The workflow reacts to the failure of the recipe depending on parameter "Recipe Failure Mode"



- The Product Explorer allows to open the bookkeeping directory for a given file, and download a file with the command that was used to create a file.
- it is possible to go into the "Bookkeeping dir" and re-execute a recipe. The call script is in file "cmdline.sh":

```
cd ~/reflex_data/reflex_book_keeping/fors-ima/fors_img_science_1/latest  
./cmdline.sh
```

Customizing workflows





Get Script:

```
wget http://www.eso.org/~wfreudli/tmp/miniscript.py
```



Using the components library

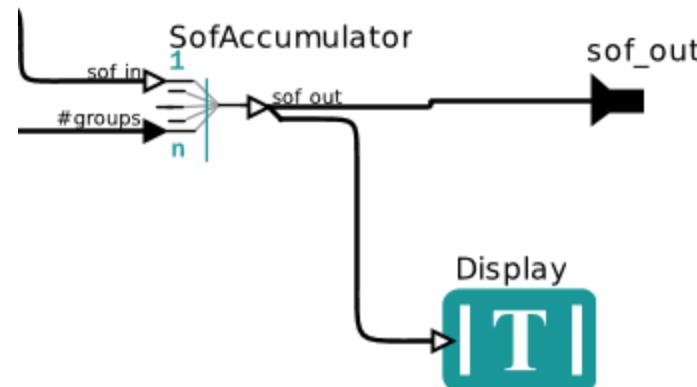
Search Components

All Ontologies and Folders

- Search Results
- Components
 - Data Output
 - Workflow Output
 - Graphical Output
 - ESRI Shape File Display
 - GML Displayer
 - Image Display
 - Textual Output
 - Browser Display
 - Display
 - Multiple Tab Display
 - Projects
 - Computational Chemistry
 - QMView Display
 - Actors
 - CoreActors,kar
 - Browser Display
 - Display
 - ESRI Shape File Displayer
 - GML Displayer
 - Image Display
 - Multiple Tab Display
 - QMView Display



Debuging: Using the Text Display



[ptolemy.actor.lib.gi](#)



The Python actor

Workflow Instructions

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 cntri-R to start the workflow.

To run on a different data set:

- Click on "File" -> "Data Set" and choose "Copy into Another workspace" or "Open in another workspace".
- If required, clean up the workspace.
- Press the "Run" button OR cntri-R to start the workflow.

To monitor the progress of the workflow in more detail:

- Open "Windows" -> "EsoReflex" and select "Monitoring" and "Monitoring the current workspace".

Setup Directories

Input:

• #ROOT_DATA_DIR: /home/ewaldk/reflexdata/
• #KANALDATA_DIR: #ROOT_DATA_DIR/reflex_inputs/ves_Blu

Working Directories:

• #BOOKKEEPING_DIR: #ROOT_DATA_DIR/reflex_wks/keeping/ves_Blu
• #LOGS_DIR: #ROOT_DATA_DIR/reflex_wks/logs/ves_Blu

Output:

• #END_PRODUCTS_DIR: #ROOT_DATA_DIR/reflex_end_products

Global Parameters

#ITS_VIEWER: tv

Its viewer to use for the inspection of input/output products

= actor with interactive option

It is able to execute generic python code.

To translate from/to Reflex ports to/from python script arguments a special syntax is used, with the help of a Python module

- To create a python actor, use the components menu on the left and search for *Eso-reflex ->Scripting.kar -> PythonActor*.



The python actor (II)

- The real data processing looks like this:

```
#Get the input files
files = inputs.in_sof.files

#Do the stuff
for file in files:
    input_image = pyfits.open(file.name).data
    output_image= input_image * 100
    pyfits.writeto(file.name.replace(".fits", "_new.fits"), output_image)
```