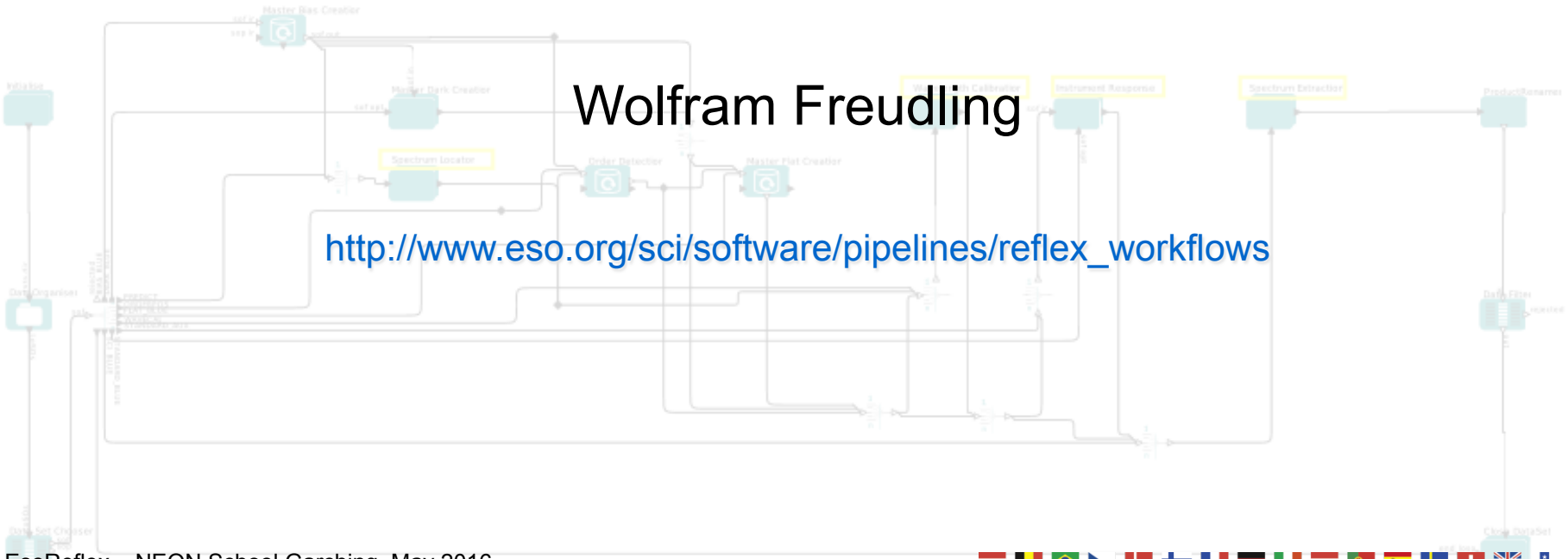
The ESO Recipe Flexible Workbench
EsoReflex
II. Advanced Features

Wolfram Freudling

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



Workflow driven Data Reduction

A&A 559, A96 (2013)

A&A 559, A96 (2013)
DOI: 10.1051/0004-6361/201322494

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

Command line interface examples

List all available workflows:

```
esoreflex -l
```

Load FORS workflow:

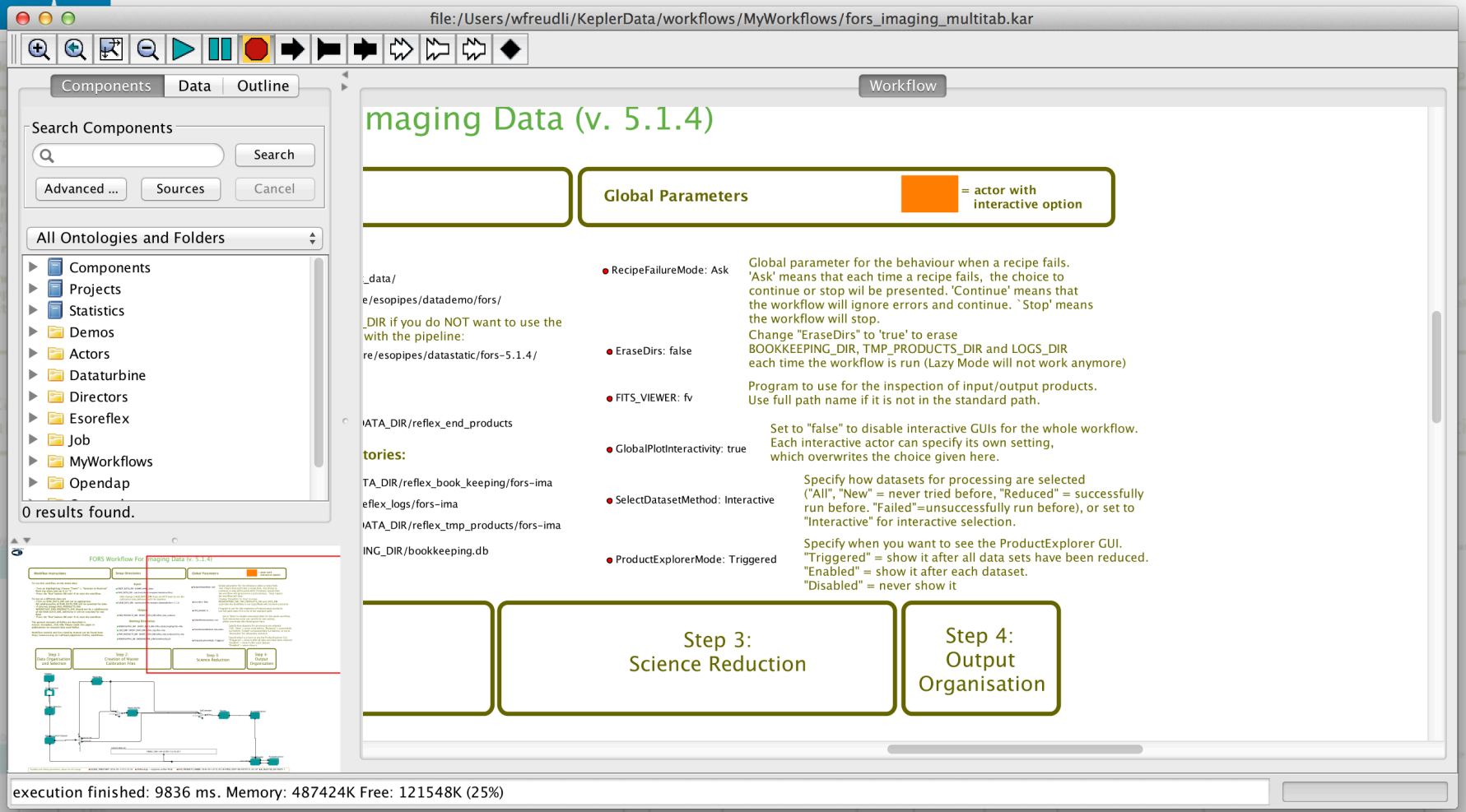
```
esoreflex fors_imaging
```

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
Components
Data
Outline
Search Components
Search
Advanced ...
Sources
Cancel
All Ontologies and Folders
Components
Projects
Statistics
Demos
Actors
Dataturbine
Directors
Esoreflex
Job
MyWorkflows
Opendap
0 results found.
Workflow
maging Data (v. 5.1.4)
Global Parameters
= actor with interactive option
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
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

- 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

Create configuration files ~/.esoreflex/esoreflexrc :

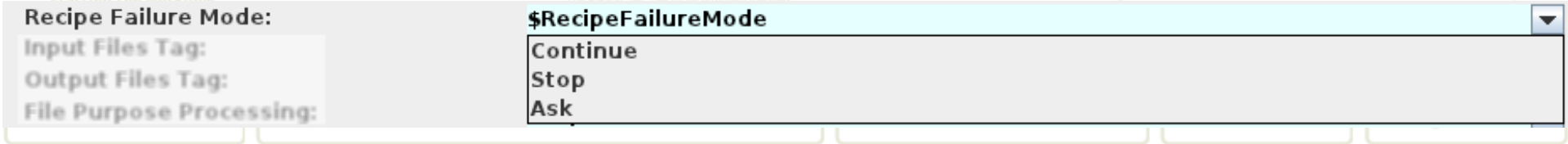
```
esoreflex -create-config
```

Some interesting parameters:

- esoreflex.python-command
- esoreflex.inherit-environment
- esoreflex.python-esoreflex.python-path

Re-executing a recipe

- Sometimes a recipe fails due 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":

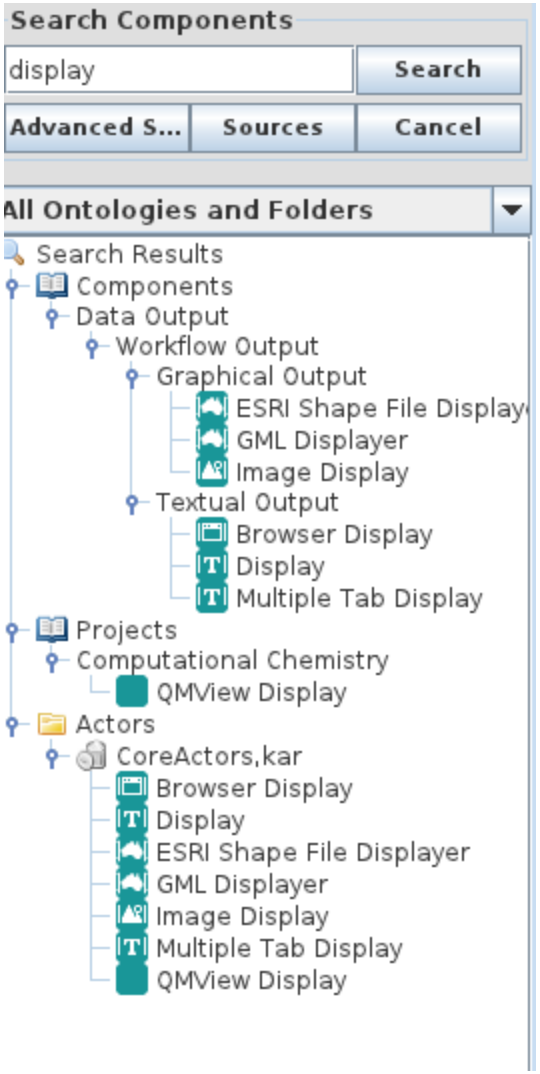
```
cd ~/reflex_data/reflex_book_keeping/fors-ima/fors_img_science_1/latest
./cmdline.sh
```

Customizing workflows

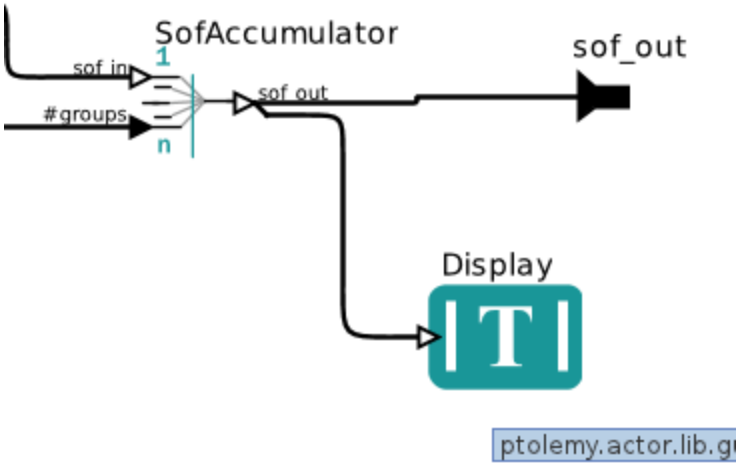
Get Script:

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

Using the components library



Debuging:
Using the Text Display



The Python actor

- 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)
```