

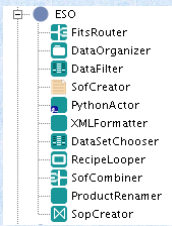


Reflex: Scientific Workflows for the ESO Pipelines

P.Ballester¹, D.Bramich¹, V.Forchi¹, W.Freudling¹, C.E.García-Dabó², M.Klein-Gebbinck¹, A.Modigliani¹, M.Romaniello¹

1: European Southern Observatory – Karl-Schwarzschild-Str. 2 D-85748 Garching bei München, Germany
2: FRACTAL SLNE, Tulipán 2, portal 13, 1. A E-28231 Las Rozas de Madrid, Spain

Reflex, a data processing workflow environment...

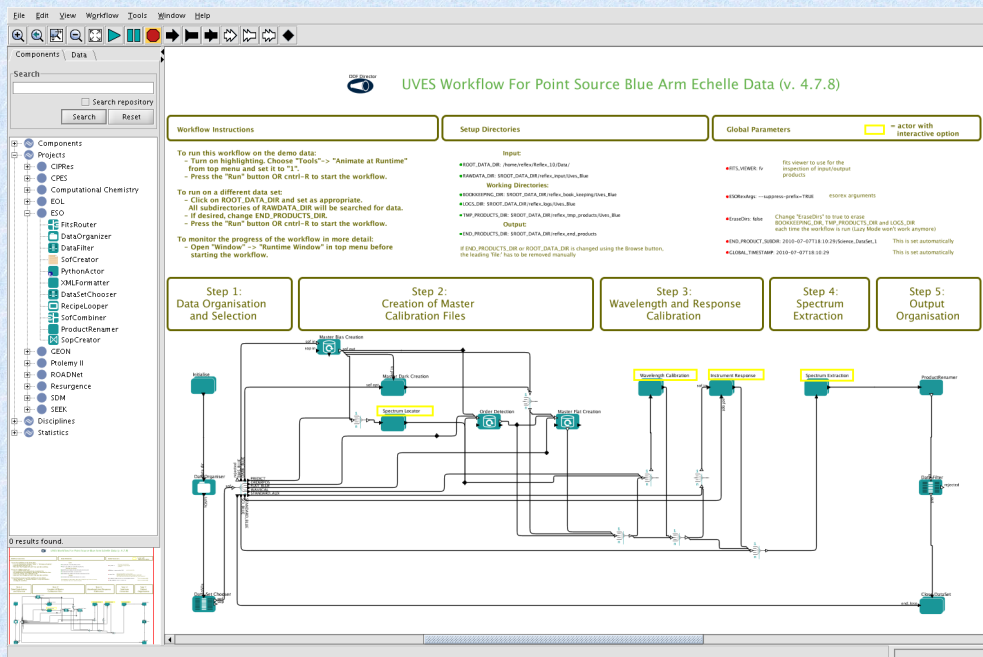


The ESO Recipe Flexible Execution Workbench (Reflex) is an environment which allows an easy and flexible way to execute VLT pipelines. It is built using the Kepler workflow engine (<https://kepler-project.org>). Reflex allows the users to process their scientific data in the following steps:

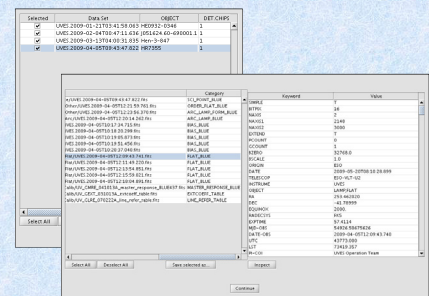
- Associate scientific files with required calibrations ,
- Choose datasets to be processed,
- Execute several pipeline recipes .

A Reflex workflow is visually represented as a sequence of interconnected boxes (actors) that process data: the workflow allows the user to follow the data reduction process, possibly interacting with it. The user can visualize the data association and the input files and decide what scientific data he wants to process. It is also possible to visualize intermediate products, using components provided by Reflex, or modify the data flow with custom components.

...based on Kepler



Data Organisation



The DataOrganizer organizes and classifies raw and reduced data: it takes as inputs all the files contained in a given directory, classifies them and organizes them into Lists of Science Observations. Each list contains science frames, all the associated calibrations needed to reduce them, and custom parameters for the reduction recipes. The user can check and modify the selection.

CPL Recipe Executer

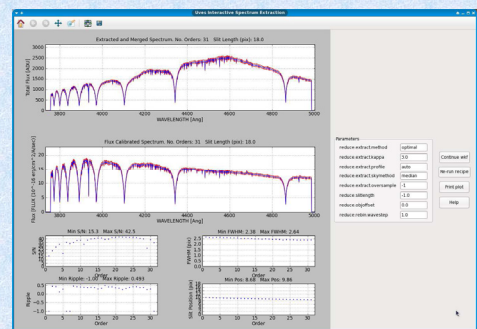
The RecipeExecuter executes a CPL recipe from the VLT pipelines (<http://www.eso.org/pipelines>). In the workflow, each parameter of the recipe is initially set to the default value. The user can change any of the parameters, or provide calculated parameter values to a special input port. The RecipeExecuter supports a variety of execution modes, facilitating the processing of large data sets.

The **Lazy mode** is particularly useful: if true the RecipeExecuter will check whether the pipeline recipe has already been executed with the same input files and with the same recipe parameters. If this is the case then the recipe will not be executed, and instead the previously generated products (which are the same as those that would have been generated if the recipe were executed, except for a timestamp) will be broadcasted to the output port.

Kepler supports **subworkflows**, which make it possible to display the data processing tasks at different levels of detail. An example of subworkflows are the **iterative subworkflows**: It is sometimes useful to be able to visualize the products of a recipe execution, tweak some recipe parameter and execute the recipe again, until the products are as expected. This can be easily achieved in Reflex by means of the RecipeLooper and the PythonActor, embedded in a Kepler subworkflow. Python actors are used to create plotting windows, graphical user interface, or to invoke tasks from external data reduction systems.

Ask for a live demonstration in the hall !

Python Actor



The PythonActor executes custom Python scripts. A module included in the Reflex release provides the functions necessary to insert scripts into the workflows. Upon selecting a script, input and output ports are automatically created, together with a parameter for each output port. Scripts can be developed for data visualisation, selection, or processing tasks, or to invoke external data reduction systems.