

# CalSelector 2.0 implementation

---

Version 2.0 of the CalSelector will have the following new features:

1. Implement the Raw2Master use case
2. Re-implement the Raw2Raw, using the same OCA rules of point 1 and making use of real and virtual products
3. Mark certified associations.

A consequence of point 1 and 2 is that the current “raw2raw” OCA rules in the DB will be replaced by standard QC rules, as close as possible to the operational ones, ideally the same.

## Virtual products

Point 2 requires to create the virtual products in the DB: a virtual product is a pipeline product that would be generated if the pipeline was run with a given set of input files, e.g. a master bias or a master flat. A different tool will take care of creating these products, but from a DB point of view they will be stored in the same table where QC products are currently stored (the products IST), because they have the same structure. This tool will also store an entry in qc\_metadata..qc\_products for bookkeeping and the raw files that generated the virtual product in the table qc\_metadata..provenance.

The products will be presented to the CalSelector in two tables, the products IST and the products provenance: the first table will contain two flag that will tell whether the product is real and/or certified, the second table will contain the provenance information for all products.

## Raw2Master mode

In this mode the CalSelector will query the table products IST for real products and stop after the first level of association, i.e. there is no recursion.

## Reimplementation of the Raw2Raw use case

When the OCA rules select a product, the CalSelector 2.0 will query the products IST, preferring real and certified files (i.e. ORDER BY VIRTUAL ASC, CERTIFIED DESC). It will then look into the provenance table for the raw files that went into that product and repeat the OCA evaluation until no raw files are found.

Certified association will be marked in the XML.

The top level association will be considered certified if all the first level associations are certified (the higher level associations are certified by construction), and this information will be passed to the NGRH at the end of the execution of the CalSelector.

## Definition of the input files

CalSelector 1.x uses some logic to expand the input file set and to include other files that might be of interest to the user: this has a number of consequences on the code and is error prone.

This behavior can be kept in CalSelector 2.0, but the ideal solution would be to define also the virtual science products in the DB, and let the CalSelector follow the associations defined there. This requires to define the science products in the OCA rules, because they are not currently there.

## Remove obsolete options

The following features will be discontinued in CalSelector 2

- Static associations
- Night logs

## Virtual Products Generator

This tool will generate the entries associated to virtual products in the products IST table. This is the workflow:

- Input: instrument and time interval
- Get the metadata mappings for the ISTs
- Get the OCA rules for all the files (there could be more than one set, depending on the time span of the input files)
- Get the metadata for the input files from the IST
- Classify the input files
- Organize the input files
- Generate the association blocks
- Store the virtual products and the provenance. The name of the virtual product will be the name of the first input raw file prepended by "V.". In case an AB generates multiple products a suffix (-1, -2...) will be appended.

### Note: definition of input files

The default way to define the input files would be "all the files arrived from last execution", so the tool has to store the timestamp of the last execution: the location will be in the new table `qc_metadata..timestamps` (columns `process_name`, `timestamp`), values ("last\_virtual\_products\_generation", `getdate()`).

Another option will be to get all the files in a time interval, useful for reprocessing.

Either way the tool should remove files belonging to the template closest in time (because it might be incomplete) and try to complete the first template (unless it was already processed).

Files will NOT be overwritten, in case of reprocessing the operator has to make sure old entries are properly cleaned up.

### **Database changes**

1. DB qc\_metadata: move here ISTs and calselector\_metadata from phase3\_metadata
2. DB obs\_metadata: add column product\_ingestion\_table to calselector\_metadata (we read from a view, we have to write to the real table)