# Largely automated science processing (working title: *phoenix*)

Reinhard Hanuschik, 2012-05-10

## *Pre-conditions*

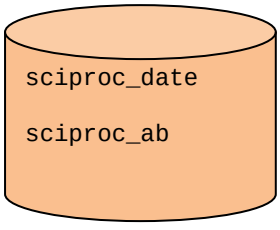
- all associations (ABs) already done, by standard QC workflow
- all associations point to certified and archived master calibrations (the ones to virtual calibrations can't be processed so they need to be filtered)
- we use only selected pipelines ("*workhorse*"/"*flagship*"/"*bread and butter*"):
    - certified (reviewed by SDP)
    - instruments with homogeneous data properties (science data types, calibration plans, headers stable over years)
    - instruments with sufficiently demanded modes (we don't aim for completeness but for optimal turnback of investment)
    - pipelines with stable and robust processing
- products get seamlessly ingested as IDPs
    - aim: no involvement by QCG needed
    - acceptable: starting set of configuration done by EDP and then reviewed/certified by QCG
    - automatic ingestion done as part of the workflow (we never had that before but it seems reasonable and doable)

## *Process*

**Model 1:**

- global solution: register date when *executable* science ABs have been created (as part of dfos workflow):
  (executable means: no virtual calibrations, and pipeline/instrument (mode) is registered as "phoenix certified")

→ database




```
sciproc_date:
instr, date, abs
```

- execute on central platform, have monitoring interface, have operator look into status

or

**Model 2:**

- local solution: fill a JOB_FILE during day as part of dfos workflow
- execute locally, over night (because then there is no load and no interference with incremental daytime QC processing)

## *Phoenix workflow*

for each scheduled date:

- download mcalibs (bulk download since many ABs might require same mcalibs) into
  $DFO_CAL_DIR/<date>
  read MCALIB list; ignore MASSOCs
  check for required gencalibs
  This is a single process, needs to be done once per date and INS
- process ABs
  download raw data within each job
  use processAB
  can be massively parallel; total execution time is T = t(single AB)*N(ABs)/N(cores)
  delete raw files immediately after processing (by post-plugin)
  have minimal score-like process to evaluate association quality, plus
  measure pipeline processing quality (could be e.g. S/N) into QC1 params
  ingest QC1 params and scores (?)
  display status, logs, scores on process monitor
- ingest products
  few (mostly primary) only, no intermediate steps (?)
  must be automatic(!), at the end of processing

after scheduled job:

- feed info on process monitor
- delete all local mcalibs (except gencalibs) and sciproducts
- store process info in database (logs, scores, ingestion logs)

All of this could be done in the background, with an operator checking the process sanity.

## *Day-to-day processing vs. back-processing*

The described scenario is applicable for day-to-day processing of new science data.

- load balancing: incremental daytime processing of calibrations is not affected; science jobs are auto-scheduled only during the night
- from experience the processing power of dfo blades was usually high enough
- but it might be wise to envisage a dedicated server for day-to-day processing (like the pre-img server dfo33)
- to monitor and schedule these tasks, the currently existing dfos tools are sufficient.

The same scenario could be applied for back-processing (to close the gap between now and 2011-10-01), or re-processing (to process backwards the entire data history of an instrument), but:

- then we need a scheduling and monitoring tool
- the process then needs to be monitored and maintained

## *IDP ingestion*

Most if not all issues with **day-to-day processing** are expected for the ingestion process.

Issues:

- which products do we select for ingestion? The trivial answer is “the final ones” but what does it mean e.g. for UVES?
    - standard setups between 2001 and 2006 had a master response curve, the non-standard ones not

      after 2006, the master response curves have not been updated;
      the 2009 detector upgrade has not seen any corresponding master response curve
      → we cannot generally provide flux calibration
      →should we give up on it (*case 1*) or support it whenever possible (*case 2*)?

      *Case 1:* final products come sometimes flux-calibrated, sometimes not
      *Case 2:* final products never come flux-calibrated, always wave-calib only
    - do we provide the error file per final product? If so, how?

- How will EDP requirements for adding keywords look like this time (for IDP/VO compliance)?
    - in the UVES reprocessing project 6 years ago, this turned out to be one of the workflow components that required most efforts, without paying back anything in the end (e.g. the lengthy discussion about the “proper” S/N value per spectrum costed a month of work, without this number being visible anywhere in the end)
    - for the GIRAFFE reprocessing project finished more than a year ago, the lack of a data model has even prevented any publication
    - this is in general true for all QC SCIENCE data products
- this **must** be kept at a minimum level
    - otherwise we are again limited by lack of standards/concepts, or
    - spend precious time with header compatibility issues rather than with science grade processing
- what does minimum mean?
    - as much metadata information as possible to be read from the header
    - additional information as far as possible from configuration files or by database processes
      for instance: if the header requires information about “pointing accuracy”, it should be possible to provide this at database level

## *What would all this mean for UVES?*

- Typical nights with 50-100 science ABs: 1-2 hours execution time,
  → performance was never an issue on dfo21
- processed modes (since pipeline certified!):
    - ECHELLE, point source (there is also EXTENDED)
    - ECHELLE, ABSORPTION-CELL
    - note: the flux calibration is not certified, strictly speaking!

    - no distinction for SM vs. VM, standard vs. non-standard setups

- no processing:
    - ECHELLE, SLICER (why not?)
    - FLAMES/UVES alias UVES/MOS: unstable/delicate pipeline, not certified

## *Impact on QC workload*

- Of course: some impact on setting this up
- **no impact** on day-to-day operations: all is done automatically, no decisions to be taken; only a process monitor required
- provided we can **automatically cdbIngest** …
- true for day-to-day processing; not for back-processing or re-processing (some more monitoring required, pus some development for monitoring and maybe scheduling tools)

Requirements:

- one attempt only: the AB either processes fine or fails
- that information should be stored in a database and be displayed on the ADP user interface
- if pipeline issues are discovered: these should be taken up by SDP
- QCG will not provide any processing comments, or feedback to users or pipeline developers or SDP
- *QCG will just provide the platform for processing (both hardware and software-wise)*
- if issues show up with ingestion or pipeline, the process could be stopped anytime and resumed after fixing the issue, without the requirement to process the backlog
- in general, there cannot be a new operational requirement on QCG related to *phoenix* (unless it would be balanced by manpower, of course)
- there can also be no commitment to the speed of the process (if monitoring reveals bottlenecks, these should be taken up efficiently by SOS but not tackled by QCG)

## *Sketch of phoenix processing workflow*

Day-to-day processing: this workflow is executed once per day and per selected INS