Transition to calSelector v2.0 (raw2master)

Version 1.0 2014-11-13 Reinhard Hanuschik

Step 1: Installation

Before installation:

- save calselManager to calselManager_v2.4.1
- save createCalibMap to createCalibMap_v1.4.1

Then install:

- calselManager v3.0_beta
- createCalibMap v2.0_beta
- verifyAB v1.0_beta
- calSelector_2.0.0 (comes with the verifyAB installation: check $DFO_BIN_DIR/OCA)

Check out:

- http://www.eso.org/~qc/dfos/calSelector.html
- same site: calselManager.html, createCalibMap.html, verifyAB.html, writeBreakpoint.html

Step 2: Migrate DFOS_OPS to CALSELECTOR (current, R2M)

Create the raw2master (R2M) rules for calSelector, for the CURRENT period (TASK 1 in http://www.eso.org/~qc/dfos/calselManager.html).

- This is a small migration from DFOS_OPS to CALSELECTOR.
- Use calselManager in v3_beta (the 'beta' because some components and some steps are not yet final)
- Call:
  - 'calselManager –I' for more information about this step
  - 'calselManager' (without parameter) to do the migration
  - follow the dialogue; the tool checks for time constraints (between for dynamic, PREVIOUS for static) and informs about a few more modifications
  - 'calselManager –C' to precompile
  - 'calselManager –U' to upload to $DFO_CONFIG_DIR/CALSELECTOR/<date>; as before, there is a dialogue about the date and the comment
  - finished: the CURRENT CALSELECTOR OCA file for R2M is in $DFO_CONFIG_DIR/CALSELECTOR/<date>
- at that stage, the new R2M file exists locally only, in addition to the R2R file, same directory

- it is available now for createCalibMap, verifyAB and finally for database ingestion

Step 3: Create CalibMap and verify the new rule

For visualization and consistency checks, create the CalibMap and do tests with the new rule. The createCalibMap tool has not changed in terms of functionality (except for the new row displaying the recipe, and the support for step2 cascades).

- Call:
  - 'createCalibMap –r <pathname to new R2M file>'
  - check the output on qcweb: as for the other OCA calibmaps, the CALSELECTOR calibmap now displays the validities and highlights errors. If you need a fix, go back to Step 2 (start from scratch), or go to Step 4 (modify existing rule).
  - If all is OK, call
    'createCalibMap –r <pathname to new R2M file> -A'
    for the calbmaps per mode.
  - done
- Note: although the CALSELECTOR rules are stored both for R2R and R2M, the calibmap for R2M replaces the one for R2R.

Step 4: Modify existing OCA rule

If in step3 a minor modification is needed, you may prefer to modify the already existing rule, rather than going thru step 2 again.

- Call:
  - 'calselManager –B' and then edit as specified in the dialogue.
  - 'calselManager –C' to precompile
  - 'calselManager –U' to upload to $DFO_CONFIG_DIR/CALSELECTOR/<date>; as before, there is a dialogue about the date and the comment
  - done

Step 5: Verify the OCA logically

Use the new tool verifyAB that has the previous "test" functionality of calselManager (in order to streamline the tools a bit). Please make sure at this stage to always call it with option –o <local OCA file>.

- Use $DFO_CONFIG_DIR/CALSELECTOR/config.verifyAB (copy it from the obsolete config.calselManager) for recording your test cases;
- OR: use the –f option
- call
  - verifyAB –f <dp_id> -o <pathname to OCA file>
    and check the DIFF file, linked as previously to your dfoMonitor
  - In most cases results should be green for the R2M case.

  - You can also put a copy of your OCA file to some other place and play with whatever change you want to test, by specifying '-o <pathname>'.
  - For more options check out http://www.eso.org/~qc/dfos/verifyAB.html

Step 6: Ingestion of current CALSELECTOR rule into the database

Once all is fine, you can ingest the CALSELECTOR OCA rule into the CALSELECTOR database. It will be ingested with a R2M flag, hence the old R2R file and the new R2M file can reside in parallel and do not disturb each other.

- For the database ingestion of a new rule, call:
  - 'calselManager –m INGEST –R <pathname to OCA file>'
    This starts the same dialogue as before for R2R. There are two more questions:
  - Q3: decide if rule to be used for R2M or for R2R. For the current period, it is certainly R2M, the default. The other option makes sense only for historical and old periods when no pipeline support existed, or the ingested master calibrations are considered bad. With this flag set to R2R you force calSelector to use that rule always in R2R mode.
  - Q5: decision about virtual products. These are usually created by various background processes (DBCM cronjobs for historical and new data) but they require an OCA rule. If you are about to ingest the first OCA rule for a selected period, no virtual products exist, and you better force the database to create them, based on the new OCA rule. This may take minutes or hours, depending on the number of files. If you replace an existing OCA rule, you can decide whether or not you want to recreate the virtual products (depending on the previous step, deletion, and depending on any changes in the organization part).
- For finding more documentation about database interaction, call:
  'calselManager –M'
- For replacing an existing OCA rule, call:
  - 'calselManager –m LIST' to see all existing ones
  - 'calselManager –m DELETE' start a dialogue, specify the file to delete, decide about virtual products (see above)
  - 'calselManager –m INGEST -R <pathname>' : see above

Step 7: Historical OCA rules for CALSELECTOR

- develop from CURRENT rule
- same versions needed as for R2R
- follow STEP4 to go from CURRENT to HISTORY; then STEP3 for the calibmaps, STEP5 for verification, STEP6 for ingestion.

Step 8: adapt the remaining dfos tools to calSelector R2M

- **THIS NEEDS AN ADDITIONAL TRIGGER, DON’T DO IT NOW!**
- turn off ‘harvestAB’
- install new versions of dfoMonitor and getStatusAB