# cdbIngest replacement: implementation information

This document contains some implementation info about the replacement of cdbIngest: it is not a design document, but it aims on giving a high level description of the implementation decisions.

## Name:

The current names of the tools are misleading, because they refer to a db which will be replaced and whose name is also misleading (calib, but it contains more than just calibrations), therefore the tools shall be renamed to dp[Ingest|Query|Delete], where dp stands for data product. This will be the name of the scripts invoking the jar, not the name of the classes themselves.

## Deliverables:

The new tools will be written in Java, they will all be contained in one jar file. There will be three wrapper scripts to easily invoke them.

## Database:

The current database (calib) will be discontinued, and a new one (qc_products) will be created. Figure 1 shows the new DB schema.

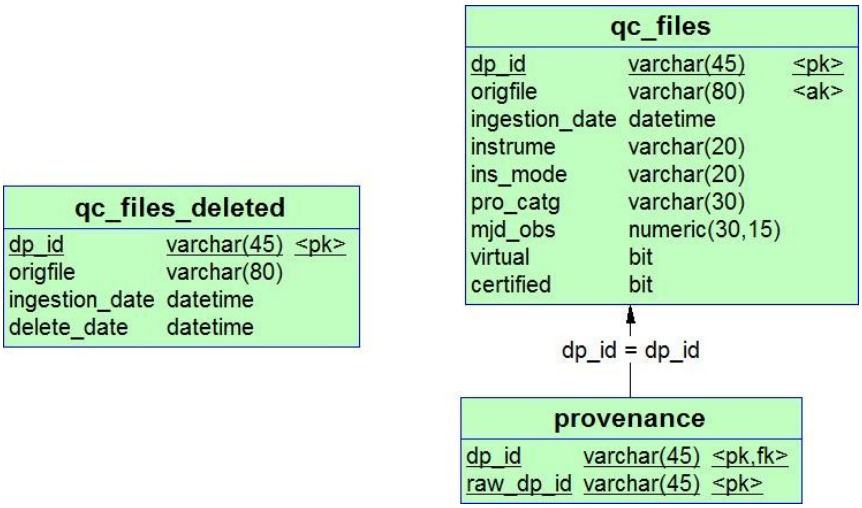


**Figure 1: DB schema**

- qc_files contains the ingested products, including:
    - the metadata required for querying: instrume, ins_mode, pro_catg, mjd_obs, origfile
    - virtual: false for real products, inserted by dpIngest, true for virtual products, inserted by the CalSelector

    - certified: true if all the raw files that generated this product are used in at least one real product, i.e. dpIngest files are certified by definition.

- provenance contains the raw files that went into that product, the contents are read from the keywords `HIERARCH ESO PRO REC1 RAWn NAME` removing the trailing `.fits`
- qc_files_deleted keeps track of the old versions of the products.

There will be a stored procedure (unique_datetime) to generate unique dp_ids.

## Certified flag:

This flag is crucial for the new CalSelector version. Every insert or delete into qc_files shall trigger an update of all the virtual products that use the raw files used by the affected product.

## Infrastructure:

The implementation will reuse as much as possible all the tools already used in the phase3 infrastructure (e.g. keywords extraction, archiving, keywords update…), with the exception of the DB layer.

## FITS verification:

FITS verification will be skipped for auxiliary (A.) files.

## Configuration:

The current tools reads the db access configuration from $HOME/.dbrc. This approach will be kept, but the tool will support only clear text passwords. There should be no need for a configuration file.

## Query:

The required metadata will be extracted by dpIngest and written into the qc_files table at ingestion time. Note that possible modifications to these metadata in the keywords repository shall NOT be propagated here, this is however acceptable because we are using fundamental keywords, that are not likely to change.

## Delete:

Every delete event will be stored in the table qc_files_deleted, and will trigger a db procedure that will hide the file from the archive.

Removing a file from qc_files shall also trigger a re-evaluation of the certified flag.

## Replace:

When a file replace is requested the current file is deleted and then a new entry is created in the files table.

## Ingestion Process:

Ideally the process should be atomic, but given the number of involved systems and the duration of the single steps (it is not feasible to lock a db table during the ingestion of a 3GB file) this is not possible. This is the proposed workflow (pseudo code):

```
try:
        insert in qc_files
catch:
        return
try:
        ingest in NGAS
catch:
        delete from qc_files
        return
try:
        extract_keywords
catch:
        try:
                set ignore flag in NGAS
        catch:
                log "cannot set ignore flag for file <dp_id>, please notify AOG"
        log "file_id <dp_id> was ingested in NGAS but the keyword extraction failed, please
        notify AOG"
        delete from qc_files
        return
try:
        update certified flag in qc_files
catch:
        log "cannot update virtual flag, please contact DBCM"
```

## Notes:

- Historical data: the current DB contents will be migrated to the new DB. As a second step the provenance table will be populated from the keywords repository.
- Certification: The requirements document states that the tool should mark raw frames as certified, in order for the CalSelector to properly create associations. This has been replaced by the certified flag in qc_files.