

Calibration Metadata Table

We will be handling calibration information for both EVLA and ALMA (with the potential for adding VLBA some time further in the future). Unfortunately, while EVLA calibrations can easily be linked to their execution_block of origin, that is not the case with ALMA. For ALMA, calibrations are done at the MOUS level (and thus are tied to multiple EBs).

As part of creating the ALMA OUS structure within the AAT/PPI & the generalization of archive products (constituents?), here is a draft for discussion of the calibrations (not calibration_tables) table:

Column Name	Column Type	Comments
calibration_id	integer	Automatically generated unique Id
execution_block_id	integer	(nullable) Foreign Key to the Execution Block from which this calibration was derived.
alma_ous_id	integer	(nullable) Foreign Key to the Alma OUS from which the calibration was derived.
products_id	integer	Foreign Key to the products table for the generalized system.
filegroup_id	integer	Filegroup for the file(s) of this calibration.
project_code	varchar	Project to which the calibration belongs.
casa_version	varchar	Which version of CASA was used.
qa_comments	varchar	Comments provided by the DA upon Ingestion. Optionally, this could be replaced with a workflow_id if the comments are stored there.

A calibration is the results of either: An Execution Block (EVLA) or an MOUS (Alma). So one (and only one) of those two structural columns should be null for any column (will need to be enforced via software design, I believe).

From Here To There:

Currently, the calibration_tables table consists of: calibration_table_id, file_id, and a blank 'TBD' column. We'll require a number of transformations for the 8k+ entries already existing (all EVLA):

- The table itself, the primary key column, and the generating sequence need to be renamed
 - details to be figured out
- file_id filegroup_id should be manageable since their directly tied to each other in the files table
 - SELECT filegroup FROM files WHERE file_id=?;
- alma_ous_id will be null for all existing entries.
- execution_block_id is obtainable via the existing filegroup structure, as is the project_code
 - SELECT [execution_block_id, project_code] FROM execution_blocks JOIN filegroups ON execution_blocks.filegroup_id = filegroups.parent_filegroup_id WHERE filegroups.filegroup_id=?;
- TODO: products_id
- The other two metadata fields can be blank
- Drop/reuse metadata_tbd

ALMA Calibrations:

Calibration of ALMA data happens after an MOUS has been fully observed. At that point, the pipeline is run, and the results are analyzed for QA2 purposes. An ALMA calibration is not official until there are science images archived for it, therefore the MOUSes of interest to us are those with:

```
select count(*) from ALMA.ASA_PRODUCT_FILES where FILE_CLASS='science' and ASA_OUS_ID=?; > 0
```

However, there are cases (roughly 15%) where the DAs must take an active role in performing the calibration. Those results are not necessarily able to be restored by the pipeline, and therefore we should reject calibrations where:

```
select count(*) from ASA_PRODUCT_FILES where ASA_OUS_ID='.....' and FILE_CLASS='script' and NGAS_FILE_ID LIKE '%scriptFor%Cal%'; <> 0
```

We should consider what calibration status to assign in this case, as the data seems to be inappropriate for pipeline processing.

In order to complete the calibrations table, we need to create a filegroup, and populate it with the relevant file information. The list of files we desire is generated by:

```
select FILE_ID,FILE_SIZE from ngas.ngas_files where file_id in (select ngas_file_id from alma.asa_product_files where ASA_OUS_ID=? and
FILE_CLASS in ('calibration','script'));
```

That will (with a potential translation between ALMA's standard for file sizes and ours), allow us to populate the files table. Note that the filename and ngas_id are identical, and both are needed for the table.

NOTE: With the NAASC processing moving toward a split imaging & calibration system, it might simplify matters to screen out some of the extraneous files from the imaging pipeline which are caught in the above query. In particular, there can be auxproducts and pipeline_manifests for the hifa_image pipeline, but those can also easily be ignored in the restore set-up software.