

# The ODI-PPA Calibration Library System

F. Valdes<sup>1</sup>, R. Swaters<sup>1</sup>, Y. Sun<sup>2</sup>

**National Optical Astronomy Observatory  
Science Data Management**

DRAFT: May 11, 2012

<sup>1</sup>NOAO Science Data Management, P.O. Box 26732, Tucson, AZ 85732

<sup>2</sup>Data-to-Insight Center, Pervasive Technology Institute, Indiana University, Bloomington, IN 47408

## Table of Contents

<b>Purpose of this Document</b>	<b>2</b>
<b>1 Overview</b>	<b>2</b>
<b>2 The Calibration Database</b>	<b>3</b>
<b>3 The Data Engine</b>	<b>7</b>
<b>4 The Calibration Planner</b>	<b>8</b>
4.1 The Calibration Database Query . . . . .	9
4.2 Calibration Selection . . . . .	10
<b>5 The VOTables</b>	<b>12</b>

## List of Figures

1 Calibration System Overview . . . . .	4
2 Schematic query to select a calibration . . . . .	11
3 Initial Raw Exposure VOTable . . . . .	14
4 Bias Output VOTable . . . . .	15
5 Flat Output VOTable . . . . .	16
6 Static Exposure Output VOTable . . . . .	17
7 Guided Exposure Output VOTable . . . . .	18

## List of Tables

1 Catalog Fields for Table <i>CalCat</i> . . . . .	5
2 Target Exposure Metadata . . . . .	9
3 Calibration Query Parameters . . . . .	9

## Abstract

The ODI-PPA Calibration Library manages calibration file metadata. It has two primary functions, ingesting metadata about calibration files and responding to requests for the best available calibration for individual ODI exposures. The system also provides interfaces for pipeline scientists and operators to manage the metadata such as inserting, deleting, and updating metadata. The primary components of the system within the PPA are the data engine, calibration planner, and calibration database. The interfaces described here are the database schema for the calibration database and VOTables used between the data engine, calibration planner, and workflow system.

**Keywords:** ODI, AuCaP, pipeline calibration, data engine, calibration planner, calibration database

# Purposes of this Document

This document serves as a design specification for the ODI-PPA calibration library system. It includes an overview and specifications for the calibration library database and the VOTable interface files.

## 1 Overview

Almost by definition one of the most fundamental aspects of an instrument calibration pipeline is the application of the most appropriate calibration algorithms and data to the raw instrument data. The algorithms are the provenance of the pipeline system and the selection of calibration data is the provenance of the calibration library system. The purpose of this document is to describe the latter for the ODI-PPA (One Degree Imager Pipeline, Portal, and Archive). As the library metaphor implies, the calibration library system provides a way to check-in and check-out calibration information. It includes a "catalog" for the metadata, "shelves" for storing the resources, and a "reference librarian" to identify the best resource.

In the ODI-PPA the calibration library system involves the data engine, DataCapacitor/HPSS, XMCCat, calibration planner, and CalDB components. There are also user interfaces to allow operators and pipeline scientists to manage the library such as for inserting, updating, reviewing and deleting calibration metadata and files.

Figure 1 shows a sequence overview of the two main function of the calibration library system: adding calibration files (aka PutCal) and selecting them for processing (aka GetCal). Calibration files, for example master biases and flat fields, are generally created by the Automatic Calibration Pipeline (AuCaP) but other sources, typically involving the instrument and pipeline scientists, are also possible. In the PutCal function, the data engine is responsible for receiving the files and saving them in the PPA archive system. This is the same process and system used for all raw and processed data. Saving files, the "shelving" metaphor, consists of ingesting metadata into the archive metadata catalog (XMCCat) and putting the files into the PPA file storage system (data capacitor and high performance storage system). For the calibration library system, an additional step is the ingestion of a small subset of metadata specific to the calibration files into a calibration library database (CalDB). In particular into a table, CalCat, which corresponds to the calibration library "catalog" alluded to earlier.

The association of calibration files with raw data, for processing by AuCaP, is done by the calibration planner. Since the data engine is also the high level orchestrator for pipeline processing, it is charged with extracting a table of all the raw exposures to be processed for a particular observing block. This table contains file reference URIs as well as all the metadata needed to select a calibration for each exposure. The metadata includes things such as when the exposure was taken and what filter was used. This table is supplied to the calibration planner to make calibration assignments. The calibration planner accesses the calibration catalog to get metadata for all calibrations which satisfy constraints imposed by the exposures. It then applies rules to select the "best" calibration for each exposure. The result of the calibration planner is then a table of expo-

tures to process along with the calibration files associated with the exposures. This information is then passed on by the data engine to AuCaP, through the OGCE workflow engine, for processing. The "planning" part of the calibration planner refers to organizing the exposures into a sequence of workflows to be "orchestrated" by the data engine, which includes the step of retrieving and staging both the raw exposures and the calibration files identified by the calibration planner.

There is one aspect of the calibration workflow that merits some discussion. For some of the workflow steps, such as creating master bias and dome/twilight flat fields, new calibrations are generated which may then be applicable to the subsequent steps. The data engine orchestration takes this into account by having it archive calibrations from one step after which the calibration planner is called again to select calibrations for the next step. At this point the new calibration will already be in the calibration system. This bootstrapping of calibrations is the reason the calibration planner is not simply called once at the very beginning with just the list of raw exposures (which include raw calibration exposures).

There is one subtlety concerning dark sky calibrations. One of the main workflows orchestrated by the data engine, the one that calibrates static on-sky exposures, produces dark sky calibrations from the ensemble of exposures if the data allows. These calibrations may then be applied to the individual exposures. Rather than break the workflow to return data products at the point the calibrations are created and before they would be applied so that the data engine can ingest them and the calibration planner can select them, the calibration will suggest dark sky calibrations previously ingested. The pipeline will then use the same selection logic discussed in this document to select between the suggested calibrations and the one generated in the middle of the workflow and apply the result. Only at the end of the complete workflow are the new dark sky calibration added to the calibration library. This has the same effect as breaking up the workflow but it is much more efficient in not having to stage the raw data, retrieve calibrations and the intermediate exposure data, and then restage these data files for the application of the dark sky calibration in a new workflow.

One final point to keep in mind in this document is that when we refer to a file, whether raw or calibration, this means for a single OTA. So while conceptually thinks of calibration processing and calibration data in terms of exposures, the number of files is actually 64 per exposure for the full focal plane or some lesser number for the partial focal plane. This means there will be many records in the calibration database or in the VOTables for an exposure or calibration. However, the consistent application of calibrations requires that when OTA calibrations are selected for an exposure, they all be from the same parent data. For example, the selection of a master dome flat from a number of candidate master dome flats means that all the OTAs from the same master dome flat (that is derived from the same dome flat sequence) be applied to the matching OTAs of the exposure being calibrated.

## 2 The Calibration Database

The library catalog is the heart of the calibration system. It is a database table containing metadata about each calibration. The metadata fall effectively into three classes; constraints, quality metrics,

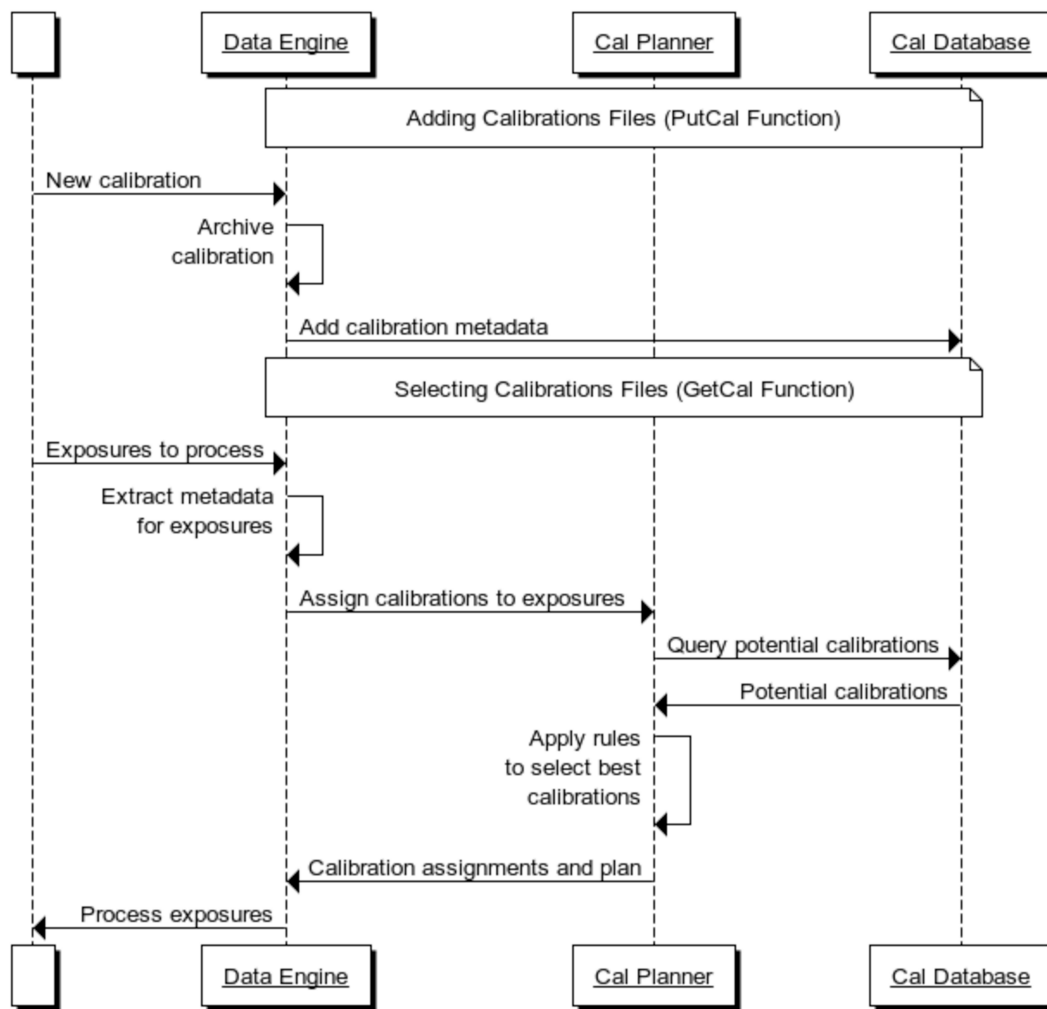


Figure 1: Calibration System Overview

and documentation. The calibration information is inserted into the catalog by the data engine whenever it receives calibration files. It harvests the data from the image headers or (equivalently) from XMCCat, after archive ingestion, and turns it into insertion records. Because the number of fields and number of calibrations is relatively small, only a single, simple table schema is required. The database engine used can also be simple, e.g. SQLite, though any engine providing APIs may be used.

The proposed schema for the *CalCat* table is shown in table 1. It is possible that additional fields may be identified during development but the goal is to not refactor the catalog once it is in production use.

**fileref**

The calibration datum the catalog is designed to provide is the file reference field. This should be a URI, which can be resolved by a storage system such as irods, rather than a physical path to allow the file to migrate.

Table 1: Catalog Fields for Table *CalCat*

<b>fileref</b>	char	File reference for retrieval
<b>createtime</b>	timestamp	Calibration creation time
<b>caltpe</b>	char	Calibration type (bias, dflat, etc)
<b>mjdobs</b>	float	MJD of calibration observation
<b>dmjdlow</b>	float	Delta MJD for validity window
<b>dmjdhigh</b>	float	Delta MJD for validity window
<b>otaid</b>	char	OTA identifier
<b>ccdbin1</b>	int	Binning factor
<b>ccdbin2</b>	int	Binning factor
<b>filter</b>	char	Filter identifier
<b>exptime</b>	float	Exposure time
<b>snr</b>	float	Avg signal-to-noise ratio or sigma
<b>ncombine</b>	int	Number of exposures used
<b>refcomp</b>	float	Reference comparison metric
<b>skycover</b>	float	Sky coverage metric in night stacks
<b>plflag</b>	float	Pipeline quality flag
<b>opflag</b>	float	Operator/scientist quality flag

**createtime**

A timestamp for when the calibration was created. This is not used for the automatic selection of a calibration but it is useful documentation and potentially of use for operator/scientist purposes.

**caltpe**

The calibration type is one of:

- bias* - Bias
- dark* - Dark of a particular exposure time
- dflat* - Dome flat field
- tflat* - Twilight flat field
- fringe* - Fringe template
- iflat* - Illumination flat field
- refbias* - Reference bias
- refdflat* - Reference dome flat field
- reffringe* - Reference fringe template
- refiflat* - Reference illumination flat field
- refskymap* - Reference sky map

The reference calibrations are used during creation of new calibrations to identify unexpected changes. These files may in a format used for the comparison; for instance, binned and smoothed to a smaller size.

**mjdobs**

The effective modified Julian Date of the data producing the calibration. This is useful documentary information but it is important in the calibration evaluation process as the fiducial point for the window of validity for the calibration.

**mjdlow**

This is a value to be *subtracted* from the mjdobs value to define a limit for the the window of validity. It is in decimal days. It is legal, though very unusual, for the value to be negative. It is often the same as the high value to define a centered window around the date of observation. It may also be zero to define a validity window strictly after the calibration data was taken.

**mjdhigh**

This is a value to be *added* to the mjdobs value to define a limit for the window of validity. It is in decimal days. It is legal, though very unusual, for the value to be negative. It is often the same as the low value to define a centered window around the date of observation.

**otaid**

The OTA identifier used to match calibration files to the appropriate data OTA.

**ccdbin1, ccdbin2**

The binning factors along the detector axes (1=serial direction, 2=parallel direction). Generally calibrations are matched to the same binning factors as the data being calibrated.

**filter**

The filter identifier which must be one of the standardized strings. This is used for those calibrations taken with the shutter open.

**exptime**

The exposure time associated with the calibration. Generally this only applies to dark calibrations and the time might be the effective dark time though the difference is typically insignificant.

**snr**

The signal-to-noise or statistical weight (inverse variance) of pixels in the calibration is a fairly direct indicator of the quality of the calibration. This is a single metric of the typical value estimated as an average over all the pixels in the calibration file. This is the most important metric since the goal of applying a calibration is to introduce as little noise as possible. A signal-to-noise value is used for flat fields and a statistical weight is used for bias and dark exposures.

**ncombine**

Most calibrations are derived by combining individual exposures. The field records the number of exposures used in creating the calibration. A larger number is often an indicator of a better calibration.



**refcomp**

Calibrations are compared to other, "reference", calibrations and this quantity is a measure of the similarity. A high degree of similarity is an indicator of a good calibration.

**skycoverage**

Calibrations created from on-sky exposures depend on sources not covering a particular pixel in every exposure. In other words, a pixel needs to sample "blank" sky in some exposures. This metric provides an indicator of how well sky is sampled over the pixels in the calibration. Better sky coverage is an indicator of better on-sky derived calibrations.

**plflag**

A numeric data quality flag provided by the pipeline. The flag is defined such that a higher value is a better choice. A value of zero indicates the pipeline vetoes this calibration.

**opflag**

A numeric data quality flag provided by the pipeline operator or scientist. The flag is defined such that a higher value is better choice. Calibrations will normally not be graded by a person and will have a default value of 1. A value of zero indicates the operator vetoes this calibration. A value greater than one is a way to promote use of particular calibrations. There may be other operator mechanisms which override the calibration library to allow explicit use of a particular calibration.

As noted in the overview, a calibration "exposure" is actually composed of some number of OTA files. In order for the selection process to be able to choose all OTAs from a calibration exposure as a unit the metadata for all OTAs must be the same. AuCaP, when it creates calibrations is required to do this. So a quantity like the SNR is not only some composite over all pixels in an OTA it must also be a composite over all OTAs.

### 3 The Data Engine

The data engine is a central component of the PPA. The full functionality of this component is outside the scope of this document. Here we describe the functions of the data engine which are part of the calibration library system.

Adding calibrations to the system consists of archiving them, just as with all data products, and inserting metadata extracted from the files into the calibration database. This was described in the overview. This step occurs whenever new calibrations are available. This is normally as the result of a workflow completing. However, it can also be through a operator initiated submission where the calibrations are produced, or improved, by someone such as an instrument or pipeline scientist. Another operator function which is related is to cause the calibration database to be modified (updates or deletions). Typically, this consists of changing the operator flag to upgrade or downgrade the use of a particular calibration.

The process of assigning calibrations to raw exposures was also outlined earlier. The starting point is a list of exposures. In the larger context of the PPA workflow the list is first defined as

exposures from a set of nights. This is then grouped into sublists associated with a sequence of workflows by the calibration planner.

When the initial request to process a set of nights is received the data engine resolves this into a table of exposures. The table consists of the file references and calibration database query metadata. The information is extracted from XMCCat. The table is constructed as a VOTable as described in more detail in section 5.

The data engine executes the calibration planner component with the VOTable as input. The output is another VOTable file containing two tables, again as described in §5. The first table, called the workflow table, includes a parameter defining the workflow to be executed and an association table with the file references for raw exposure to be processed by the workflow and the file references for the calibrations assigned to that exposure. The second table, called the remainder table, is simply the subset of the input table not contained in the workflow table. The remainder table is used as input to the calibration planner after the specified workflow is completed and any calibrations produced are archived, thus starting the next cycle in the orchestration of a sequence of workflows. The remainder table will be absent when the workflow table for the last workflow is produced.

The data engine and/or the workflow engine need to read the workflow table and find all the unique files referenced (since one calibration file is typically associated with many raw exposures) and handle the staging of the data for the workflow. This involves retrieving the actual files from the PPA archive and placing them where the workflow code will access them. The workflow table is ultimately passed on as input to the requested workflow. If the file references need to be expanded for the workflow, i.e. dereferenced into actual pathnames, the table would need to be transformed.

## 4 The Calibration Planner

The calibration planner has two roles, organizing exposures into groups and directing the data engine as to which workflow to run for each group and assigning the best calibration files for each exposure from those available in the calibration database. It does this based on metadata for the raw exposures to be processed. It is called by the data engine after each workflow is completed so as to have access to any calibrations produced and stored by the data engine.

The input is a VOTable containing a table listing the file references of files to be processed and metadata for the exposures needed to assign calibrations. It also may contain a parameter identifying the last workflow completed.

The planner groups the exposures by observation type; such as bias, dark, dome flat, static sky, and guided sky exposures. From this it determines the next workflow needed. For that group it uses the metadata to locate potential calibrations and then assign the best one to each exposure.

As just implied, there are two steps to assigning calibrations to an exposure. The first step is a query to the calibration database with constraints based on the metadata for the exposure and some system parameters. The query returns zero or more calibrations satisfying the constraints along with data quality metrics. If there are no calibration then a potential error action may be taken or there will simply be no calibration assigned and the response deferred to workflow to decide what

to do.

The second step when there are multiple candidates is to rank them based on the data quality metrics. The way the metrics are combined is an algorithmic detail. The highest ranked calibration is then chosen as the "best" calibration and assigned to the exposure.

The raw exposure file references and the assigned calibrations for the exposure form a table row. The table is the group of exposures that is to be processed by the next workflow. The table is formatted as a resource in a VOTable. All the exposures not in the current group to be processed are added to the VOTable as a second resource. The data engine will separate the tables into one for the workflow and one for input to the next execution of the calibration planner.

## 4.1 The Calibration Database Query

The assignment of calibration files to raw exposures (by OTA) is done by the calibration planner component. As part of this the planner queries the CalCat catalog for calibrations which meet constraints imposed by both the target raw exposures and system parameters. Table 2 shows the target metadata used in queries and supplied to the calibration planner in a VOTable input by the data engine. Table 3 presents calibration system parameters provided as part of the PPA parameter mechanism.

Table 2: Target Exposure Metadata

<b>obstype</b>	-	observation type
<b>mjdobs</b>	-	observation MJD
<b>otaid</b>	-	OTA identifier
<b>ccdbin1</b>	-	serial binning
<b>ccdbin2</b>	-	parallel binning
<b>filter</b>	-	filter identifier
<b>exptime</b>	-	exposure time

Table 3: Calibration Query Parameters

<b>opflag</b>	-	minimum value for operator flag
<b>dmjd</b>	-	MDJ half-window
<b>dexptime</b>	-	exposure time percentage window

### **mjdobs**

The observation MJD of the target exposure.

### **otaid**

The OTA identifier of the piece of the target exposure.

**ccdbin1, ccdbin2**

The binning used for reading out the target exposure.

**filter**

The filter identifier of the target exposure.

**exptime**

The exposure time of the target exposure.

**opflag**

The minimum acceptable operator flag.

**caltype**

The calibration types required. This is determine based on the observation type of the target exposure. For example, a dome flat field calibration is required for a science exposure but not a dome flat field sequence exposure.

**dmjd**

The acceptable window about the target exposure's observation MJD. Note that this is different than the window of validity of a calibration. Calibrations selected must satisfy both of these windows.

**dexptime**

The acceptable window of calibration exposure times relative to the target's exposure time. This is expressed as a fraction of the target's exposure time; e.g. 10%. This comes into play for finding a dark calibration exposure for a target exposure which was not taken with exactly the same exposure time.

The calibration planner is free to define the query to the calibration database as is required. For example whether a constraint clause is needed for a particular type of metadata or whether the constrain expression is an equality, list selection, or pattern. Figure 2 shows a schematic query using all the constraints. In the figure those quantities in brackets are the exposure and parameter values defined earlier. In addition [caltypes] is a list of required calibrations types defined in the context of the exposure. Note that the details of the syntax may vary depending on the database engine adopted.

## 4.2 Calibration Selection

The selection of the "best" calibration for a particular exposure is based on data quality flags and metrics retrieved from the calibration database. This selection can be made as elaborate as desired using A.I. or other decision methodologies. The complication is that when there are multiple data quality parameters and metrics it is challenging to incorporate them into a selection criterion. For example, how important is proximity in time relative to the signal-to-noise.

One of the design strategies for the PPA calibration system is to isolate the calibration selection decision to a single component, namely the calibration planner. This isolation and modularity

Figure 2: Schematic query to select a calibration

```
SELECT fileref, snr, ncombine, refcomp, skycover,
       abs([mjdobs]-mjdobs) as dmjd, abs([exptime]-exptime) as dt
FROM CalCat
WHERE caltype in ([caltypes])
AND (mjdobs-[mjdobs]) <= min(dmjdlow,[dmjd])
AND ([mjdobs]-mjdobs) <= min(dmjdhigh,[dmjd])
AND otaid = [otaid]
AND filter = [filter]
AND dt <= ([dexptime]*[exptime]/100)
AND plflag > 0.
AND opflag >= [opflag]
```

allows evolution of the decision logic. We start with a simple method based on the operator flag, the signal-to-noise / statistical weight metric, and proximity in time. The calibrations which satisfied the constraint selection are ordered by OPFLAG, then by SNR, and finally by the difference in MJD between the calibration and target exposure. From this ranking the top choice is selected. In words this means that the highest operator flag value is chosen but if these are equal, the usual default case, the best signal-to-noise value is chosen and, in the even of nearly equal SNR then the closest in time is selected.

In order for the time proximity to apply the SNR metric has to be transformed into discrete values. This is done by rounding the logarithm of the SNR, digitizing into discrete values of some accuracy, say 10%, and applying a ceiling value. That is, values above some threshold are treated equally and below that threshold values that are within 10% are considered comparable. This logic is similar to one used with the NOAO pipelines except that instead of the discretized SNR the number of combined exposures is used, again with a ceiling.

So why are there more metrics included in the calibration library and why are they not used? Firstly, by recording these parameters the possibility of making use of them in the future is preserved. Secondly, we are actually making use of some of them through the constraint selection. The key parameter here is the pipeline flag. This parameter is defined to allow the pipeline to veto a calibration, using a value of zero, which is enforced through the constraint query. The pipeline vetoes a calibration itself based on the metrics of similarity to a reference and on the evaluation sky coverage for dark sky generated calibrations.

Returning again to the subject of exposures vs. OTAs, it was noted earlier that the metadata used in the selection must be the same for all OTAs from a particular calibration exposure. This is expected to be enforced by the pipeline that generates the calibration. However, if this is not the case the calibration planner can be responsible for making a composite value (e.g. an average over all the OTAs). But in addition, the calibration planner has to ensure that when a OTA calibration is assigned to one OTA of an exposure, all OTAs of the exposure are assigned corresponding OTAs from the same calibration.

## 5 The VOTables

The interface between the data engine and the calibration planner, as well as the subsequent input to the workflow from the data engine, are VOTables ([1]). These VOTables all have the same basic structure consisting of one or more resources each containing one or two tables. The tables in a resource are of two types; a table used as input to a workflow and a table used as input to the calibration planner.

Figures 3 to 7 illustrate the form of the files for the initial call to the calibration planner (fig. 3) and the tables returned by the calibration planner at each stage of the workflow sequence. The example content follows the expected conventions for dataset names, workflow names, raw exposure filenames, and calibration filenames. But the important specification are in the tags and attributes. Below are details on some of the key tags and attributes.

### utype

The "utype" (usage-specific type) is provided to identify the type of resource or table without requiring the consumer to parse the content. The utypes are defined in the following descriptions.

### name (RESOURCE and TABLE)

The "name" attribute in the RESOURCE and TABLE tags provide a unique name for each of these elements. The names are the same as "dataset" names used in the workflow processing. These names are built from the following pieces:

```
<semester>_<start date>_<process id>[-<workflow>[<filter>]]
```

```
semester - semester as 2 digit year and A or B
start date - calendar starting data of block
process id - process id formed as a hex time stamp
workflow - one of ftrB, ftrF, ftrS, ftrG
filter - filter abbreviation for filter dependent workflows
```

### RESOURCE

A resource has a "name" and a "utype". The name is a unique dataset identifier. The utype is one of:

```
A - initial input exposure data for a block of nights
B - output for bias workflow
F[filter] - output for flat field workflow for a filter
S[filter] - output for static science exposure workflow for a filter
G[filter] - output for guided science exposure workflow for a filter
```

### TABLE

A table has a "name" and a "utype". The name is a unique dataset identifier matching. It is the same as the resource name. The utype is one of:

```
WorkflowTable - table defining data for a workflow
```

RemainderTable table of exposure data for the next call to the calibration planner

These are useful for the data engine to separating these tables when handling them. Note there is no workflow table in the initial file for the calibration planner and there is no remainder table when the workflow requested is the last one for some subset of data.

### **RemainderTable**

A remainder table has a standard set of parameters and columns. The filename is the URI to be used by the data engine to setup data for the workflow engine. The remaining columns are metadata used for calibration selection by the calibration planner.

### **WorkflowTable**

The workflow tables vary in the number of columns. There is a parameter specifying the workflow that it applies to. The columns are the filename of the raw exposure to be processed and then file references for calibrations associated with the exposure. Since different workflows require different sets of calibration files the number of columns vary by the target workflow. Note that some workflows in the standard sequence of workflows may be skipped because of lack of data. So the data engine should not depend on a hardwired sequence of workflows.

As noted in the overview, the number of table rows for the raw exposures and for the calibrations is based on the number of OTAs. The examples show a single OTA for a single exposure or calibration.

## **References**

- [1] F. Ochsenbein and R. Williams. VOTable Format Definition: Version 1.2. IVOA Proposed Recommendation V1.2, IVOA, Nov 2009.  
<http://www.ivoa.net/internal/IVOA/IvoaVOTable/PR-VOTable-1.2-20091102.pdf>.

Figure 3: Initial Raw Exposure VOTable

```

<?xml version="1.0"?>
<VOTABLE version="1.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.ivoa.net/xml/VOTable/v1.2"
  xmlns:stc="http://www.ivoa.net/xml/STC/v1.30" >
<RESOURCE name="11B_20111031_7decd3a" utype="A">
  <TABLE name="11B_20111031_7decd3a" utype="RemainderTable">
    <PARAM name="dataset" datatype="char" value="11B_20111031_7decd3a"/>
    <PARAM name="nights" datatype="char" value="2011-10-31,2011-11-01,2011-11-02"/>
    <FIELD name="filename" datatype="char"/>
    <FIELD name="fileLocation" datatype="char"/>
    <FIELD name="fileID" datatype="char"/>
    <FIELD name="obstype" datatype="char"/>
    <FIELD name="mjdobs" datatype="double"/>
    <FIELD name="otaid" datatype="char"/>
    <FIELD name="ccdbin1" datatype="int"/>
    <FIELD name="ccdbin2" datatype="int"/>
    <FIELD name="filter" datatype="char"/>
    <FIELD name="exptime" datatype="float"/>
    <FIELD name="otmode" datatype="char"/>
    <FIELD name="ra" datatype="char" />
    <FIELD name="dec" datatype="char" />
    <DATA><TABLEDATA>
    <TR><TD>b20111031T230758.00.07.fits</TD>
      <TD>TBD</TD>
      <TD>TBD</TD>
      <TD>zero</TD>
      <TD>55865.96386574</TD>
      <TD>1107</TD>
      <TD>1</TD>
      <TD>1</TD>
      <TD>g SDSS k1017</TD>
      <TD>0.</TD>
      <TD>STATIC</TD></TR>
    <!-- ... -->
    </TABLEDATA></DATA>
  </TABLE>
</RESOURCE>
</VOTABLE>

```



Figure 4: Bias Output VOTable

```

<?xml version="1.0"?>
<VOTABLE version="1.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.ivoa.net/xml/VOTable/v1.2"
  xmlns:stc="http://www.ivoa.net/xml/STC/v1.30" >

<RESOURCE name="11B_20111031_7decd3a" utype="B">
  <TABLE name="11B_20111031_7decd3a" utype="WorkflowTable">
    <PARAM name="workflow" datatype="char" value="caB"/>
    <FIELD name="filename" datatype="char"/>
    <FIELD name="fileLocation" datatype="char"/>
    <FIELD name="fileID" datatype="char"/>
    <FIELD name="groupname" datatype="char"/>
    <FIELD name="refbias" datatype="char"/>
    <FIELD name="refbiasLocation" datatype="char"/>
    <FIELD name="refbiasID" datatype="char"/>
    <DATA><TABLEDATA>
      <TR><TD>b20110915T152915.01.01.fits</TD>
        <TD>TBD</TD>
        <TD>TBD</TD>
        <TD>12A_20120623_1123456-20110912T152915.01_refbias.fits</TD></TR>
        <TD>TBD</TD>
        <TD>TBD</TD>
      <!-- ... -->
    </TABLEDATA></DATA>
  </TABLE>

  <TABLE name="11B_20111031_7decd3a-caB" utype="RemainderTable">
    <!-- Table 11B_20111031_7decd3a with subset of rows. -->
  </TABLE>
</RESOURCE>
</VOTABLE>

```

Figure 5: Flat Output VOTable

```

<?xml version="1.0"?>
<VOTABLE version="1.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.ivoa.net/xml/VOTable/v1.2"
  xmlns:stc="http://www.ivoa.net/xml/STC/v1.30" >

<RESOURCE name="11B_20111031_7decd3a-caFV" utype="FV">
  <TABLE name="11B_20111031_7decd3a-caFV" utype="WorkflowTable">
    <PARAM name="workflow" datatype="char" value="caF"/>
    <FIELD name="filename" datatype="char"/>
    <FIELD name="fileLocation" datatype="char"/>
    <FIELD name="fileID" datatype="char"/>
    <FIELD name="groupname" datatype="char"/>
    <FIELD name="bias" datatype="char"/>
    <FIELD name="biasLocation" datatype="char"/>
    <FIELD name="biasID" datatype="char"/>
    <FIELD name="refflat" datatype="char"/>
    <FIELD name="refflatLocation" datatype="char"/>
    <FIELD name="refflatID" datatype="char"/>
    <DATA><TABLEDATA>
    <TR><TD>f20110915T152915.01.01.fits</TD>
      <TD>TBD</TD>
      <TD>TBD</TD>
      <TD>12A_20120701_1234567-20110915T145917.01.fits</TD>
      <TD>TBD</TD>
      <TD>TBD</TD>
      <TD>12A_20120623_1123456-FV-20110912T145917.01_refflat.fits</TD></TR>
      <TD>TBD</TD>
      <TD>TBD</TD>
    <!-- ... -->
    </TABLEDATA></DATA>
  </TABLE>

  <TABLE name="11B_20111031_7decd3a-caFV" utype="RemainderTable">
    <!-- Table 11B_20111031_7decd3a with subset of rows. -->
  </TABLE>
</RESOURCE>

<RESOURCE name="11B_20111031_7decd3a-caFR" utype="FR">
  <!-- Same as previous resource but for R filter -->
</RESOURCE>

<!-- ... -->

</VOTABLE>

```

Figure 6: Static Exposure Output VOTable

```

<?xml version="1.0"?>
<VOTABLE version="1.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.ivoa.net/xml/VOTable/v1.2"
  xmlns:stc="http://www.ivoa.net/xml/STC/v1.30" >

<RESOURCE name="11B_20111031_7dec3a-ftrV" utype="SV">
  <TABLE name="11B_20111031_7dec3a-ftrV" utype="WorkflowTable">
    <PARAM name="workflow" datatype="char" value="ftr"/>
    <FIELD name="filename" datatype="char"/>
    <FIELD name="fileLocation" datatype="char"/>
    <FIELD name="fileID" datatype="char"/>
    <FIELD name="groupname" datatype="char"/>
    <FIELD name="bias" datatype="char"/>
    <FIELD name="biasLocation" datatype="char"/>
    <FIELD name="biasID" datatype="char"/>
    <FIELD name="dflat" datatype="char"/>
    <FIELD name="dflatLocation" datatype="char"/>
    <FIELD name="dflatID" datatype="char"/>
    <FIELD name="fringe" datatype="char"/>
    <FIELD name="fringeLocation" datatype="char"/>
    <FIELD name="fringeID" datatype="char"/>
    <FIELD name="iflat" datatype="char"/>
    <FIELD name="iflatLocation" datatype="char"/>
    <FIELD name="iflatID" datatype="char"/>
    <FIELD name="reffringe" datatype="char"/>
    <FIELD name="reffringeLocation" datatype="char"/>
    <FIELD name="reffringeID" datatype="char"/>
    <FIELD name="refiflat" datatype="char"/>
    <FIELD name="refiflatLocation" datatype="char"/>
    <FIELD name="refiflatID" datatype="char"/>
    <FIELD name="refskymap" datatype="char"/>
    <FIELD name="refskymapLocation" datatype="char"/>
    <FIELD name="refskymapID" datatype="char"/>
    <DATA><TABLEDATA>
    <TR><TD>f20110915T152915.01.01.fits</TD>
      <TD>TBD</TD>
      <TD>TBD</TD>
      <TD>12A_20120701_1234567-20110915T145917.01.fits</TD>
      <TD>TBD</TD>
      <TD>TBD</TD>
      <TD>12A_20120701_1234567-V-20110915T145917.01.fits</TD>
      <TD>TBD</TD>
      <TD>TBD</TD>
      <TD>12A_20120701_1234567-V-20110915T145917.01_frg.fits</TD>
      <TD>TBD</TD>
      <TD>TBD</TD>
      <TD>12A_20120701_1234567-V-20110915T145917.01_ifr.fits</TD>
      <TD>TBD</TD>
      <TD>TBD</TD>
      <TD>12A_20120623_1123456-V-20110912T145917.01_reffrg.fits</TD>
      <TD>TBD</TD>
      <TD>TBD</TD>
      <TD>12A_20120623_1123456-V-20110912T145917.01_refift.fits</TD>
      <TD>TBD</TD>
      <TD>TBD</TD>
      <TD>12A_20120623_1123456-V-20110912T145917_refskymap.fits</TD></TR>
    <TD>TBD</TD>
    <TD>TBD</TD>
    <!-- ... -->
    </TABLEDATA></DATA>
  </TABLE>

  <TABLE name="11B_20111031_7dec3a-ftrV" utype="RemainderTable">
    <!-- Table 12A_20120701_1234567 with subset of rows. -->
  </TABLE>
</RESOURCE>

<RESOURCE name="11B_20111031_7dec3a-ftrR" utype="SR">
  <!-- Same as previous resource but for R filter -->
</RESOURCE>

<!-- ... -->

</VOTABLE>

```

Figure 7: Guided Exposure Output VOTable

```

<?xml version="1.0"?>
<VOTABLE version="1.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.ivoa.net/xml/VOTable/v1.2"
  xmlns:stc="http://www.ivoa.net/xml/STC/v1.30" >

<RESOURCE name="11B_20111031_7decd3a-ftGV" utype="GV">
  <TABLE name="11B_20111031_7decd3a-ftGV" utype="WorkflowTable">
    <PARAM name="workflow" datatype="char" value="ftG"/>
    <FIELD name="filename" datatype="char"/>
    <FIELD name="fileLocation" datatype="char"/>
    <FIELD name="fileID" datatype="char"/>
    <FIELD name="groupname" datatype="char"/>
    <FIELD name="bias" datatype="char"/>
    <FIELD name="biasLocation" datatype="char"/>
    <FIELD name="biasID" datatype="char"/>
    <FIELD name="dflat" datatype="char"/>
    <FIELD name="dflatLocation" datatype="char"/>
    <FIELD name="dflatID" datatype="char"/>
    <FIELD name="fringe" datatype="char"/>
    <FIELD name="fringeLocation" datatype="char"/>
    <FIELD name="fringeID" datatype="char"/>
    <FIELD name="iflat" datatype="char"/>
    <FIELD name="iflatLocation" datatype="char"/>
    <FIELD name="iflatID" datatype="char"/>
    <DATA><TABLEDATA>
    <TR><TD>f20110915T152915.01.01.fits</TD>
      <TD>TBD</TD>
      <TD>TBD</TD>
      <TD>12A_20120701_1234567-20110915T145917.01.fits</TD>
      <TD>TBD</TD>
      <TD>TBD</TD>
      <TD>12A_20120701_1234567-V-20110915T145917.01.fits</TD>
      <TD>TBD</TD>
      <TD>TBD</TD>
      <TD>12A_20120701_1234567-V-20110915T145917.01_frg.fits</TD>
      <TD>TBD</TD>
      <TD>TBD</TD>
      <TD>12A_20120701_1234567-V-20110915T145917.01_ift.fits</TD></TR>
      <TD>TBD</TD>
      <TD>TBD</TD>
    <!-- ... -->
    </TABLEDATA></DATA>
  </TABLE>

  <!-- Note absence of RemainderTable indicates no more workflows -->

</RESOURCE>

<RESOURCE name="11B_20111031_7decd3a-ftGR" utype="GR">
  <!-- Same as previous resource but for R filter -->
</RESOURCE>

<!-- ... -->

</VOTABLE>

```