**NOAO DPP Document PL012**

# NOAO Science Pipelines Operations Model

F. Valdes[1], D. Scott[1], N. Zarate[1], R. Swaters[2]

**National Optical Astronomy Observatory**
**Data Products Program**

January 18, 2008

[1]NOAO Data Products Program, P.O. Box 26732, Tucson, AZ 85732
[2]Department of Astronomy, University of Maryland, College Park, MD 20742

# Table of Contents

**1 Introduction** **2**

**2 Science Pipeline Operations Model** **2**

**3 Pipeline Datasets** **3**

**4 Pipeline Scheduling System** **3**
4.1 Pipeline Scheduling Queue Database . . . . . 4
4.2 Pipeline Scheduling Agent . . . . . 4
4.3 Populating the PSQDB . . . . . 5
4.4 Managing the PSQDB . . . . . 7

**5 Pipeline Parameters** **7**

**6 Pipeline Data Manager and Calibration Library** **7**

**7 Data Management and Science Support** **8**

**8 Pipeline FTP Distribution** **9**

# Abstract

This document describes the operational context, strategies, and components of the NOAO science pipelines applications. Currently NOAO has science pipelines for its Mosaic and NEWFIRM cameras. These NOAO instruments are used for a variety of programs requiring the pipeline applications to handle heterogeneous observing protocols, conditions, and fields. This leads to an operations model where blocks of nights, possibly spanning different programs, are processed as a single dataset to maximize the likelihood of including sufficient calibration data. Therefore, a key component of the pipelines is a pipeline scheduling database and agent to define and submit these types of datasets. Other aspects of the operational model discussed are pipeline operations parameters, the calibration library, and interactions with the NOAO Data Management Science Support system which supports the NOAO Science Archive and Portal.

**Keywords:** pipeline, operations

# 1 Introduction

This technical report presents a high-level description of the operational model for the NOAO science pipelines. These pipelines are applications of the NOAO High-Performance Pipeline System (NHPPS: Valdes et al., 2007). The intended audience includes pipeline operators, collaborators, and developers. One class of developers of particular note are those working on pipelines to be used, at least in part, at NOAO telescopes by the general NOAO community where issues of program and data heterogeneity must be addressed.

NOAO operates pipelines in two contexts: at telescopes for quick-look evaluation by observers and at data centers for full processing and final distribution to the community. Pipelines in the first context are referred to as *quick-reduce pipelines*. These aim for quick processing of the raw data into basic data products for data quality evaluation and monitoring. The calibration and data products are not complete or the best possible in this context. Pipelines in the second context are referred to as *science pipelines*. Their aim is to fully calibrate the data and produce advanced data products for the principle investigators and, after any proprietary period, the astronomical community. This report is focuses on science pipelines.

A fundamental requirement for the NOAO science pipelines is that they process observational data from various science programs (sometimes multiplexed by shared nights or queue scheduling), taken by a variety of observers using different observing protocols and strategies, and obtained under a range of observing conditions. This heterogeneity leads to pipeline and operational strategies for making the best attempt at calibrating and producing useful data products.

One method to ensure proper calibration for these heterogeneous NOAO observations is to mandate observers follow a specific observing protocol; e.g. the Mosaic Pipeline Observing Protocol (Dickinson et al., 2006). Currently, NOAO does not mandate such observing protocols.

The key operational strategy is based on the concept of a *dataset*. Section 3 discusses this concept. Subsequent sections then expand on how datasets are defined, submitted to the pipeline, and some details about how the dataset definitions allow for self-calibration.

The report presents a high level description of the operational interfaces and databases. Some implementation details are provided to clarify them.

# 2 Science Pipeline Operations Model

The science pipelines run continuously on a cluster of machines at a data center; currently a single center in Tucson. One component of the pipeline, the *pipeline scheduling agent*, has knowledge of the telescope and instrument schedules. After completion of a block of one or more nights of observations with a pipeline supported instrument, this component initiates a query to the NOAO Data Management and Science Support (DMaSS) services for the data identifiers (a kind of URI). These URIs, which define a dataset, are used to trigger the appropriate pipeline application.

The pipeline application gets the raw data (through another transaction with the DMaSS), processes the data on the pipeline cluster, and finally queues the pipeline processed data products for incorporation by the DMaSS. The pipeline also supports an option to directly stage data products

as tar files in a password protected FTP portal staging area. The end user, or customer, gets the pipeline data products through the NOAO Portal, which access the DMaSS, or through the FTP portal staging area.

Note that a basic aspect of this model for science pipelines is that they do not run either as data is being acquired or after each night of observing, though the latter strategy could be provided by defining dataset blocks as single nights for specific programs or for special requirements and priorities. For the general heterogeneous observing requirement noted earlier, single night datasets do not work as well as larger blocks of nights.

# 3 Pipeline Datasets

The NOAO pipelines operate on *datasets*. These are arbitrary collections data, possibly of different types, which are processed as a group.

In the quick-reduce context datasets are either single exposures or sequences of exposures. In the science pipeline context the datasets are larger groups of exposures. In particular science datasets need to include sufficient data for complete calibration.

The pipeline applications process datasets in a kind of hierarchical decomposition. This means that a top level dataset is broken down into smaller datasets for processing by various pipelines which may, in turn, further decompose the dataset for even lower level pipelines. The applications also regroup data into different kinds of datasets such as for basic calibration by exposure and for stacking by dither sequence.

For the NOAO science pipelines the top level datasets consist of blocks of consecutive nights irrespective of the proposal. The size of the blocks is a trade-off between active disk space and the potential for cloudy nights and shared calibrations between data from different proposals.

Using more than one night in a block allows the pipeline application to decide at a lower level how to break up data depending on the number of exposures. This is most important for defining data from which dark sky self-calibrations (e.g. sky flats) are made. For example, suppose one night was cloudy or, for one filter, the observers took only a few exposures per night. The pipeline then makes datasets for a single night when there are some minimum number of exposures, say 20, in the same filter and groups several nights for filters with just one or a few exposures per night.

The strategy for making the top level datasets is described further in the section on the Pipeline Scheduling System (§4). Briefly, the strategy is to define datasets in blocks of three to four nights while attempting to keep proposal runs together. Since the NOAO telescopes are currently scheduled ”classically” with typical runs of between 3 and 8 nights this leads to the use of the 3 to 4 night dataset blocks. However, if the telescope schedule warrants, the pipeline may process single nights (such as from engineering nights) or two night runs.

# 4 Pipeline Scheduling System

There are two approaches to running a science pipeline. One is as a service that some external system or agent directs to process data. The second is for the pipeline to actively discover and

process data. In discussions within DPP it was decided that the DPP science pipelines use the second approach. Philosophically, this is equivalent to assigning responsibility to the pipeline developers for an operational system to feed the pipelines.

The DPP pipeline team developed a pipeline scheduling system built around a *Pipeline Scheduling Queue Database* (PSQDB). The components that interact with the PSQDB are a *Pipeline Scheduling Agent* (PSA), an SQL interface, a browser UI, and a pair of tools that extract information from the NOAO telescope schedule database and insert this information into the PSQDB.

The system has a great deal of flexibility and does not impose many constraints or how data is processed.

## 4.1 Pipeline Scheduling Queue Database

The Pipeline Scheduling Queue Database has three kinds of tables. There is a single, top-level table defining logical *queues* for the operator to conveniently control multiple instruments or large scale data groupings. Each of these queues has an associated dataset and data table. Figures 1, 2, and 3 show the schema for these three types of tables along with some example values. The tables are linked by the `queue`, `data`, `dataset`, and `name` fields.

The main operational queues are defined as data for an instrument at a telescope for a semester. For example, the queue C4M07B is for C(TIO) 4(-meter) M(osaic camera) data for the (20)07B semester. There are queues for KPNO Mosaic and NEWFIRM data for this and other semesters. There are also queues defining test datasets or datasets for special science verification or engineering purposes.

The tables define dataset names, their observing dates, and SQL conditional constraints (the WHERE clause terms) to form an SQL query to be submitted to the appropriate data management service or archive database. The SQL conditional constraints are distributed between two tables. In the queue table the constraints select what is common for all data in a particular queue, typically for the instrument and telescope. The other fragment is what defines a specific dataset in that queue. In the example, values of the queue constraints select the NOAO Mosaic Camera at CTIO and the data constraints select a range of calendar nights. The approach is flexible and the SQL conditions sometimes are more involved than the examples shown here.

The other operational fields of the database are the state of a queue, whether enabled or disabled, and the status of a dataset. The status of a dataset may take a number of values with the principle ones being 'pending', 'submitted', and 'completed'. The dataset table also contains fields for recording the times of submission and completion. These times make it more convenient to monitor pipeline activity although much more detailed timestamps, down to the individual pipeline stages, are recorded in the pipeline processing database.

## 4.2 Pipeline Scheduling Agent

The *Pipeline Scheduling Agent* (PSA) is the key component for autonomous operation of the science pipeline applications. It implements priorities and requirements set by NOAO management

Figure 1: PSQDB queues table: PSQ.

```
psqname     char(8)         C4M07B
queue       char(8)         C4M07B
data        char(8)         C4M07BD
application char(8)         MOSAIC
pipeline    char(8)         dir
state       char(16)        enabled
query       varchar(256)    dtinstru='mosaic_2'
```

Figure 2: PSQDB dataset table: e.g. C4M07B.

```
dataset     char(32) 20071222
priority    int 1
status      char(16) completed
submitted   char(16) 2007-12-25T02:54
completed   char(16) 2007-12-25T12:21
```

and customers. The current priorities are 1) process data shortly after the proposal "run" is over and 2) process any other runs which have been queued.

The PSA checks the PSQDB for pending datasets in the active queues of the PSQDB when it is started, when it receives a dataset processing completed event, and at times when a new dataset is scheduled to be available.

The PSA has a configuration parameter that defines how many datasets may be submitted to a particular pipeline application at one time. At this time the PSA only allows one dataset to be submitted to a pipeline at one time. It does not directly check for an active dataset but only uses the status flag of the PSQDB which records submitted datasets. If the pipeline is reinitialized for some reason, such as after a system failure, the operator uses one of the PSQDB interface tools to reset the dataset status flag back to pending. The pipeline restart command includes an option to do this automatically.

Figure 4 shows output from the PSA log. In this example the the PSA has just been started and there are no back datasets pending. It simply sets a wake up time when the next dataset is expected to be available. Note that the offset between the end of the observing block and the wake up time is set to allow time for the DPP data transport system to get the data to the archive and for the DMaSS to ingest the data so that it can be queried. This delay is expected to decrease in the future.

## 4.3 Populating the PSQDB

The tools (`getruns` and `updPSQ`) in this component are what implement the operational processing model or policy. The approach is to have an operator run a tool run once a semester on

Figure 3: PSQDB data table: e.g. C4M07BD.

```
    name        char(32) 20071222
    start       int 20071222
    end         int 20071224
    subquery    varchar(256) dtcaldat between '2007-12-22 and '2007-12-24'
```

Figure 4: Example PSA log output.

```
PSA SUBMIT: Thu 18:08:01 27-Dec-2007
Selected Queue(s), Application(s), Pipeline(s):
        K4M07B, MOSAIC, dir
        C4M07B, MOSAIC, dir
Processing pipeline: MOSAIC:dir
        Currently submitted 0/1
        No real-time datasets queued at the moment
        Submitting 0 entries:
Looking for datasets with end dates greater than 20071225
Checking queue K4M07B K4M07BD
        No future end dates found
Checking queue C4M07B C4M07BD
        Next end date: 20080101
                selected as new next end date...
Scheduled to wake up and submit real-time datasets with end date of
20080101 on 20080102 22:00 UTC
Current time: 20071227 18:08 UTC
```

the final telescope schedule database to automatically define initial datasets. The output is text which an operator can then manually adjust as needed. Since this is only done once a semester the interactive stage is justified. The final adjusted output is then fed into another tool that creates and inserts the data and queues into the PSQDB.

The telescope extraction tool is designed around the model described in the Pipeline Dataset section. It identifies the lengths of runs for each proposal and divides them into groups of three to four nights if the runs are longer than four nights. Shorter runs are their own datasets which the operator may merge if appropriate. Split night proposals are treated as single proposals for this operations.

The use of generic SQL allows other possibilities.

### 4.4 Managing the PSQDB

The most general way to manage the PSQ database is through the native SQL interface of the DBMS. The current DBMS is `postgres` though clearly any DBMS may be used and, in an earlier version of the DMaSS it was `mysql`. Use of this interface requires modest proficiency with SQL and understanding of the PSQDB tables. While an understanding of the available DMaSS queries is useful for creating new entries or making specialized entries, an operator generally does not need this.

There are many possibilities for creating specialized GUIs using systems which provide a programmatic interface to the DBMS. DPP currently provides a browser interface through ZOPE/SOAP which allows examining the tables, navigating to linked tables through HTML links, and modifying some fields through menus. In particular this interfaces allows changing the `state` of a queue and the `status` of a dataset.

Backup and restoring the database falls under the tools provided by the DBMS.

## 5 Pipeline Parameters

It is inevitable that a pipeline will have parameters to alter its behavior. The are two operational methods provided by the NOAO Science Pipelines. The principle, and preferred, one is through a parameter file. Each pipeline application has a parameter file which defines all the available parameters and default values. There is then an operations configuration directory where an operator may place parameter files, that override the default values. Only those parameters to be changed need be in the file.

The file names are used to provide overrides for the pipeline application as a whole, for specific queues (see §4.1) and for specific datasets. Normally, pipeline parameter adjustments are only used for specific datasets when the operator is aware of special aspects of the observing since the NOAO science pipelines are already designed to provide the best reductions for all programs.

The second, less favored, method is through changes in the calibration database (see §6). This database is indexed by filter and time and only in as much as the time observation can be matched to a particular set of observations is this suitable for customizing processing for specific data. There is only entry in the database which can be thought of as an pipeline operations parameter. This parameter defines the calibration operations to be applied to data from a particular filter and detector over some period of time. Other calibration data, such as rules defining when certain processing steps are reasonable, are intended for instrument scientists to define and not as part of an operational context.

## 6 Pipeline Data Manager and Calibration Library

The *Pipeline Data Manager* (DM) is a server that interfaces distributed pipeline applications to pipeline specific data management services. This component is described in some detail in Valdes et al. (2007). The particular service of interest here is the Calibration Library. Its relevance is that

it allows use of calibrations from other datasets in the event that a dataset fails to include important calibrations such as biases and flat fields.

The calibration library maintains a database of calibration information indexed by attributes appropriate for selecting a calibration for a particular observation. The indexing attributes include the detector, image identifier, filter, exposure time, a quality rank, and starting and ending valid dates. The detector and image identifier attributes are needed to support multiple instruments and mosaics and the quality rank is used to give greater weight to calibrations which have been deemed of higher quality.

The calibration library contains a variety of static and dynamic calibration information, meaning whether or not the pipeline adds calibrations to the library. It is the dynamic calibration data that is of particular interest in our operational model. This type of data consists of biases, darks, dome flats, fringe templates, dark sky delta flats, and other self-calibration data produced by the pipeline. These calibrations are entered into the library with a period of validity sufficient to allow use with datasets from nearby times.

In the operational model, the dataset defined by a block of nights is supposed to contain all the raw data needed to generate the calibration noted in the previous paragraph. However, when this is not the case, and especially when the observational field is not suited to making dark sky self-calibrations, the calibration library may be used.

In order for this to be effective, the science pipeline needs to be operated in a sequential fashion so that at least the preceding or succeeding dataset have been processed. As noted in section 4.2 the pipeline scheduling agent normally processes new data shortly after each observing run is completed, i.e. forward in time, or otherwise the most recent backlog data, i.e. backwards in time.

The calibration database is normally an internal component of the NHPPS. It was noted earlier that it is possible for operators to modify the behavior the pipeline through this database. However, it is intended primarily for support scientists to adjust the calibration rules and calibration file priorities. The support scientists may also provide hand-crafted calibrations if desired.

# 7 Data Management and Science Support

The *Data Management and Science Support* (DMaSS) component of the NOAO/DPP integrated system is the source and sink for the pipeline applications. There are three services required by these pipeline applications; a query for data holdings, a request for staging the data, and queuing of pipeline data products for ingestion. The end consumer of the pipeline data products then obtains them through a portal into the DMaSS.

The first required service is a data holdings query. As described in section 4.1, the PSA initiates a request for data for a particular dataset (block of nights) based on the telescope schedule. The query encapsulated in the PSQDB is sent to the DMaSS query service. The service returns a list of identifiers for the raw data constituting the observational dataset. The possible results may be a list of identifiers, an empty list, or status messages such as data not yet available. An empty list is, unfortunately, an occasional result for NOAO instruments due to bad weather or instrument problems.

When the list of identifiers is not empty the pipeline applications submit the identifiers to the DMaSS data service to retrieve and stage the data. There are two current versions of this process, one where the pipeline applications pulls the data to its staging area and one where the data service stages the data to the pipeline staging area. In both approaches, the pipeline application is ready to process the dataset when the data has been placed in its staging area.

The third service is for queuing pipeline data products for assimilation into the DMaSS. The pipeline submits data data products to a data queue service. This service has an interface identical to a printer queue since, in fact, the implementation makes use of a Unix printer queue with its own daemon to handle interfacing with the DMaSS ingestion service. An important architectural feature of this in the integrated DPP system is that this is the same queuing interface used by the data acquisition systems to submit raw data to the DMaSS.

# 8 Pipeline FTP Distribution

There are circumstances where it is desirable for the pipeline to more directly provide data products to consumers. This is accomplished by a pipeline application stage that packages data products in a user oriented file format – FITS, HTML, PNG, or TAR files – and deposits them in an FTP or HTTP staging area.

This requires the pipeline application to be concerned with proprietary data issues normally handled by the portal interface to the DMaSS. This is accomplished by segregating data products by the proposal identifier associated with the data products. The pipeline uses the proposal identifier attached to the raw observation data.

The pipeline system has another database, the Pipeline FTP Database, which indexes FTP staging directories by the proposal identifier. The pipeline application places the formatted data products in the location specified in the database. DPP operations provides password protected access to these staging areas where the principle investigators are provided with a password for their proposal.

The Pipeline FTP Database is like the PSQDB in that it can be prepared in advance based on the accepted proposal database which defines the link between proposal identifiers used during observing and the PIs.

# References

Dickinson, M., Jannuzi, B., & Abbott, T., 2006, NOAO DPP Document PL005, http://dpopsn.tuc.noao.edu:8080/DPP/pipeline/pipeline-dpp-documents/drafts/pl005/pl005.pdf

Valdes, F., Cline, T., Pierfederici, F., Miller, M., Thomas, B., & Swaters, R., 2007, NOAO DPP Document PL001, http://chive.tuc.noao.edu/noaodpp/Pipeline/PL001.pdf