



Image services for DP0.2 (and commissioning and beyond)

A significant incremental goal for DP0.2 is the initiation of the service of image data products from the production.

This means **making images available through IVOA-style services in the API Aspect**, and thereby through the Portal Aspect's existing and future abilities to work with such services. (We have already provided access to image data via the Butler in the Notebook Aspect in DP0.1.)

We would like to provide **image services that are functional in the rapid-turnaround commissioning context** as much as in the "static release" environment of the Data Previews and the future DRs. This helps us prepare for live-data service for AP outputs at the start of operations, as well as (I hope) actually being useful for commissioning.

Planned elements of image services – DP0.2

- DP0.2 requirements:
 - ObsCore data model
 - ObsTAP service (“ivoa.ObsCore” table in a TAP service) for image metadata
 - Image retrieval via https: linked to ObsTAP contents – “links services” *a la* CADC
 - IVOA SODA service for cutouts
 - DataLink annotations to the ObsTAP service for access to the SODA service, etc.
 - HiPS images generated from coadds, deployed on (trivial) HiPS service
- DP0.2 desired:
 - Links from ObsCore table to native Exposure/Visit tables
 - DataLink annotations to the catalog and ObsTAP services allowing navigation between images and associated catalog entries; microservices supporting these links
 - ObsTAP on same TAP service as catalogs (NB: not required for the previous item)
 - Well-defined bidirectional link between:
 - Opaque data product IDs used in IVOA services
 - DataIDs usable with a Butler’s Python API / in the Notebook Aspect

Planned elements of image services – beyond DP0

- Data-taking-oriented services:
 - Mechanism for making new data visible promptly in services
 - Integration of EFD data and observing log with image metadata tables
 - Periodic (at least nightly) MOC generation to document survey coverage
- Additional capabilities required in operations:
 - SIAv2 service for image metadata (overlaps ObsTAP to a large extent)
 - IVOA Registry entries for our services
 - CAOM2 data model, exposed in TAP service, in addition to ObsCore
 - SODA-like mosaic-making service
 - PVI compression and mechanism for purging original PVIs and representing them in services
 - Difference-image and PVI re-creation services
 - “virtual data product” services
 - Forced-photometry-on-demand service
 - “HiPS progenitors” links back to the input coadd tiles

New architecture for calculational image services

SODA, diffIm/PVI re-creation, etc.

- Key design principle is separation of concerns:
Keep the VO-service-implementation codebase (URL API parsing, XML generation, ...) separate from the afw-based computational code
- Allow each piece of the system to be maintained by suitable experts:
 - SQuaRE develops and maintains service framework
 - Science Pipelines develops and maintains algorithmic code
- Service framework being developed by Russ Allbery
 - Starting with a SODA implementation but intended to be more general
 - Solves all the IVOA-interface problems “once”, facilitating deploying more services
- afw-based back-end for SODA service prototyped
 - Some architectural refinement still in progress
 - Analysis showed that a pure PipelineTask/Butler-based architecture did not fit all use cases
 - Evaluating:
 - * Extend Butler to support output to opaque non-coordinate IDs
 - * Perform I/O one level below Butler: Datastore (initial solution)

Status toward minimal DP0.2 needs

- Done
 - Portal Aspect is ~ready to provide the basic service required for DP0.2
 - ObsTAP service has been prototyped, but not integrated with Rubin-produced image data
 - Basic DataLink annotation capability has been developed for the CADC TAP service codebase
 - ObsCore data generation for Rubin images spec'd in 2020 and prototype developed, not deployed
 - Demonstrated an example of a DataLink-supporting microservice in the SQuaRE framework
- In active work
 - Re-architected (see previous slide) SODA service has been implemented and demonstrated, refinements still in progress
 - HiPS tile generation in afw context developed, but not integrated into generation of a full HiPS map
- Not yet available
 - “Assembly line” for data from from pipeline outputs to ObsCore data deployed in ObsTAP service
 - “Links services” to support the CADC-style data access model
 - https: service for the image products themselves
 - Actual DataLink annotations and associated microservices

Near-term needs

- Complete architecture for calculational-service back ends & release the cutout service
 - Discussing packaging and release principles (see RFC-828)
- Complete definition of opaque-ID to Butler DataID connection and user access
- Schedule work to complete basic HiPS generation from coadds and deploy in an RSP service
- Schedule (mostly SQuaRE) work on the links services and DataLink microservices
 - Gregory and Frossie to precisely specify what's expected for DP0.2
 - Frossie believes these are small tasks given the whole *acquis* of SQuaRE tooling
- Implement the pipeline from DP0.2 pipeline outputs to image metadata deployed in ObsTAP
 - Major decision needed:
 1. Statically deploy an ObsCore table for the DP0.2 data products on a database (Postgres), **OR**
 2. Implement ObsCore as a view on the Butler registry database

ObsCore as a database view on Butler registry

- Enables trivial deployment of image services on any Butler repository
 - Therefore enables image services on the OODS
 - Therefore supports quasi-real-time use of image services in commissioning
 - Significantly simplifies the overall system: no need for regular updates to a separate ObsCore table triggered by data transfers
- Notes from DMLT discussion:

 - * Worries about maintainability/portability of UDFs across database impl's
 - * Can the s_region be created at ingest time?
 - * Can the s_region be computed by the TAP layer?
 - * Some skepticism about the usefulness of the image services in the Summit environment
 - * Attempt to define view using SQLAlchemy / Python APIs rather than direct DB interaction
- What is needed?
 - Check that all basic needed data is either available in Butler Registry or mappable from it
 - Many ObsCore attributes can be set based on DatasetType (calib_level, dataproduct_type, ...) so a view can be generated from a JOIN against a small static table holding those mappings
 - Generation of text-based s_region (STC-S-like) from Registry might require a stored procedure or a UDF – architectural work required to determine how repositories would be supplied with these
 - Gregory working on establishing how close the mapping already is to being doable

Next big steps

- Integration of “transformed EFD”-style data with the image metadata services
 - Gregory, K-T, and Frossie working on this architecture, technote coming soon
 - Will also include access to the observing log
- Extend use of DataLink to catalog-to-image links
- Implementation of a data-product-(re-)creation service as a test particle for the architecture

Portal Aspect image-access capabilities – now and future

- ObsCore-based image browsing was added to Firefly before the “freeze”
- ObsTAP-specific search capabilities were added to the generic Portal (Firefly) TAP query screens during 2021 by Brian van Klaveren, and are being regularly exercised against image services at public archives (CADC, MAST)
- DataLink-following capabilities were added to Firefly in 2020-21
- DP0.2 image services in the Portal Aspect will be based on these capabilities

Next steps

- Refinements to DataLink-based service access (better on-screen documentation, tooltips, assistance in specifying values in a variety of units)
- ObsTAP search screen enhanced with additional data-discovery features
 - E.g., pre-query service for valid values of the obs_collection attribute and provide them as a pick-list
- MOC-based display of dataset coverage integrated with image/catalog search
 - Intent is to document survey coverage on a daily basis via MOC