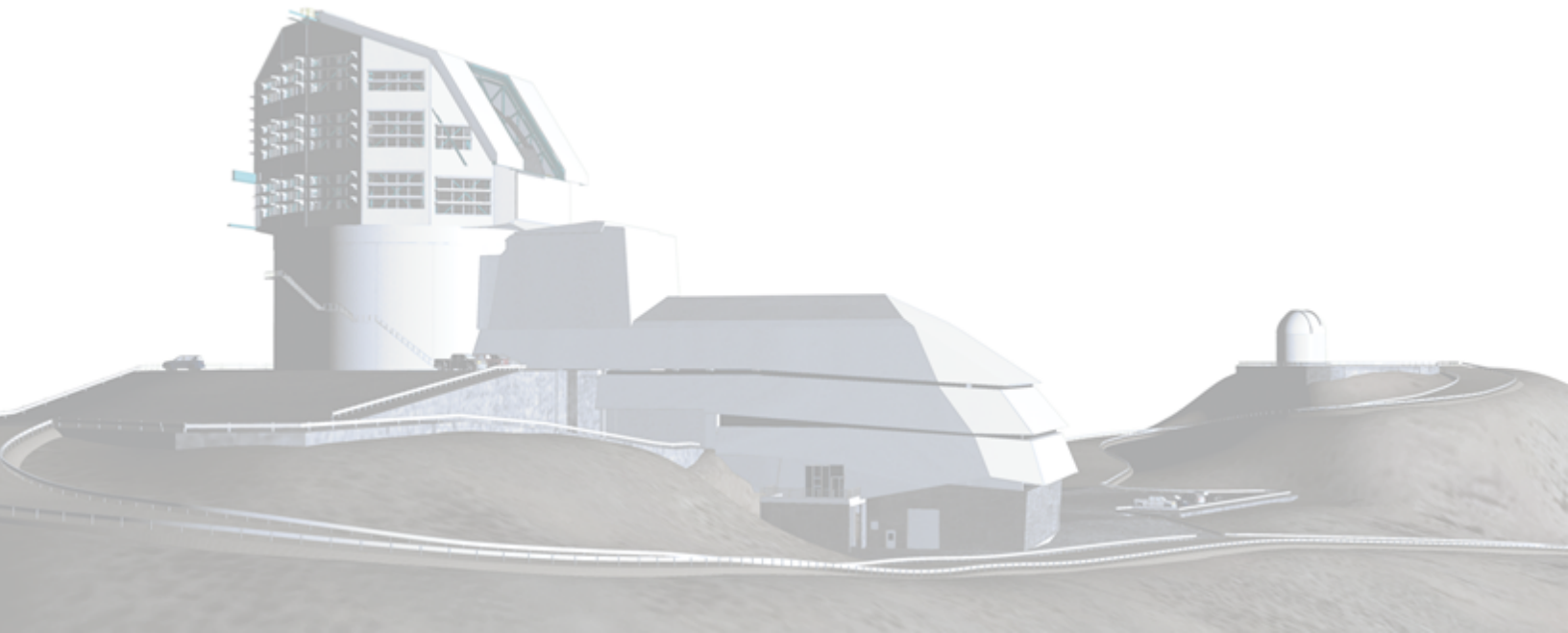
Vera C. Rubin Observatory
Rubin Observatory Operations

Data Preview 0.2 and Operations rehearsal for DRP.

William O'Mullane, Yusra Alsayyad, Hsin-Fang Chiang, Frossie Economou, Fritz Mueller, Tim Jenness, Colin Slater

RTN-041

Latest Revision: 2022-09-13



Abstract

DM delivered software to operations to perform processing of the DESC DC2 data as well as enhancements to the portal and Qserv for interaction with the results. The release of this was called Data Preview 0.2 and the production of the data products and publication of them were carried out in an operational manner. This provides valuable insights for operational data releases.

Change Record

Version	Date	Description	Owner name
1	2022-08-02	Unreleased.	William O'Mullane

Document source location: `https://github.com/lsst/rtn-041`

Contents

Data Preview 0.2 and Operations rehearsal for DRP.

1 Introduction

Between December 2021 and May 2022 the DESC DC2 (LSST Dark Energy Science Collaboration (LSST DESC) et al., 2021) was reprocessed with Rubin Science pipelines V23 [DMTR-351].[1] Between May and June the catalogs were ingested to Qserv, tutorials and documentation were updated and the Data Preview 0.2 data release was made on time at the end of June. A number of procedures were developed and practiced to achieve this. Planning for DP0.1 and DP0.2 are in RTN-001. We shall discuss the process in the following sections:

- Management and communication is discussed in Section 2
- An overview of the processing is given in Section 3
- Quality assurance and feedback to processing is discussed in Section 4
- Community engagement, tutorials and documentation are discussed in Section 6

2 Management and communication

Here we cover the management structures in place for DP0.2 this includes the groups and meetings like the change control for the pipeline version.

2.1 Oversight

The Data Production Leadership Team (DPLT) consisted of representatives from all the teams involved in the data preview process as well as the data facilities. The DPLT met fortnightly to discuss any issues and minutes were recorded on Confluence.

The membership of the DPLT was:

- William O'Mullane

[1] https://pipelines.lsst.io/v/v23_0_2/index.html

- Bob Blum
- Leanne Guy
- Frossie Economou
- Tim Jenness
- Yusra AlSayyad
- Hsin-Fang Chiang
- Michelle Butler (NCSA)
- Richard Dubois (USDF)
- George Beckett (UKDF)
- Fabio Hernandez (FrDF)

This encompasses practically the entire operations DPLT as depicted in Figure 2

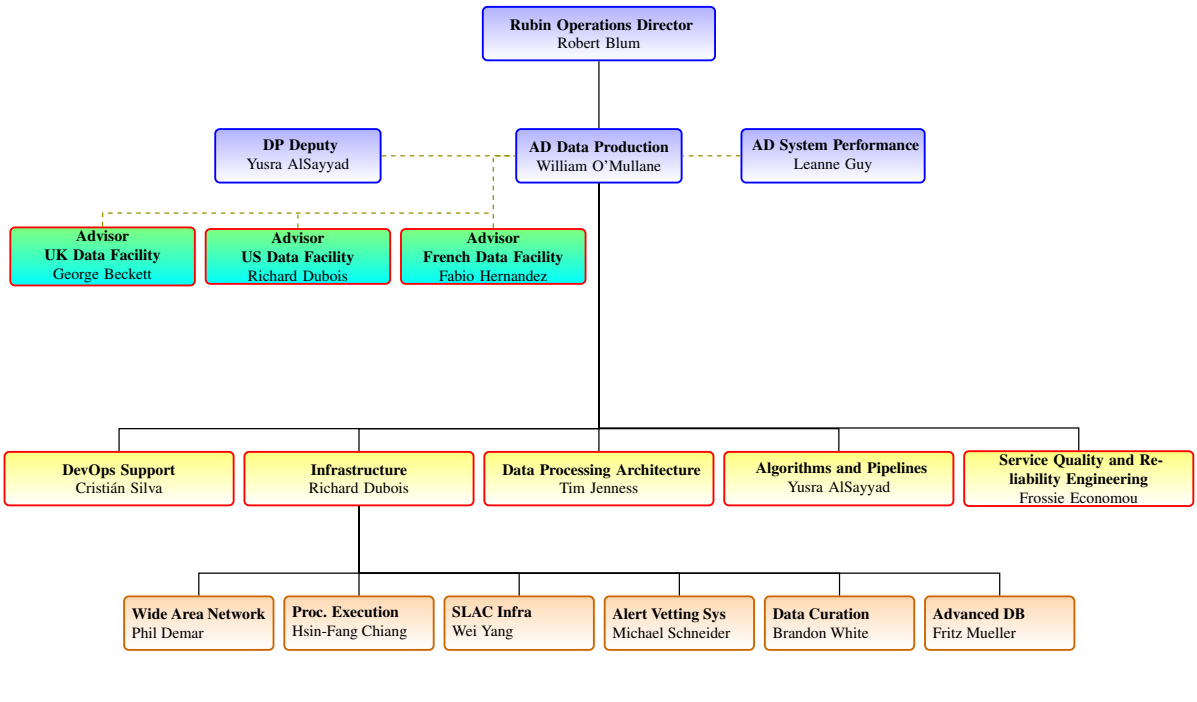


FIGURE 1: Data Production Leadership in operations - most of which was involved in DP0.2.

2.2 Coordination

During the data preview 0.2 process there was a regular coordination meeting every two weeks (out of phase with the DPLT meeting) with minutes recorded on Confluence. This meeting was attended by all the people directly involved in the data preview process: management, the processing infrastructure team, the science platform team, the execution team, the pipelines team, the verification and validation team, and the community engagement team.

This allowed the different teams to report on status and bring up any issues that needed to be addressed and made everyone aware of progress. Data Preview 0.1 had been released during this period and this allowed us to also include feedback from the Community Engagement Team as they interacted with the existing delegates and prepared updated tutorials for DP0.2.

Data Facility representatives were present at this meeting even though all the processing was being done on the Interim Data Facility at Google (O'Mullane et al., 2021).

2.3 Work Management

We used Jira to track work related to the Data Preview. Epics and milestones were created in the PREOPS Jira Project. Story tickets were then attached to each epic but in order to properly integrate the work into existing Data Management processes, any tickets that would result in code changes in pipelines software or middleware packages were created in the Data Management Jira project.

The status of the epics and how they related to the relevant milestones was monitored as part of the weekly coordination or DPLT meetings.

2.4 Change Control

Data Preview 0.2 used v23.0.x of the LSST Science Pipelines Software and that was derived from a weekly release from September 2021 (`w.2022.40`). We decided to group processing into distinct "steps" that allowed updates to the software used in later stages of processing to be worked on whilst earlier steps were executing.

We continued to want to use the v23 release for all data processing and that required that

we had a process to determine which patches would need to be back-ported to the release branch as needed before each step could begin.

The Data Management Change Control Board (DMCCB) and DPLT delegated authority to a new Data Release Steering Committee that had the following membership:

- Yusra AlSayyad, representing the pipelines team.
- Leanne Guy and Colin Slater, representing the verification and validation team.
- Hsin-Fang Chiang, representing the execution team.
- Tim Jenness, representing the data processing architecture team.

The Board met weekly on Tuesday at 8:30am Pacific Time and also had a Slack channel to discuss any issues that would come up between meetings. Minutes for the meetings were recorded on Confluence.

The process for deciding on a back-port is as follows:

1. A request is made that a ticket should be applied to the release branch by applying a `backport-v23` tag to the Jira ticket.
2. The board would then discuss the relative merits of the back-port and if approved a `backport-approved` label would be added.
3. The work on the back-port would then be scheduled by the relevant T/CAM following instructions in the developer guide.[2]
4. Once the code is on the `v23.0.x` branch a `backport-done` label would be applied.

A Jira query was constructed to find all the tickets and track their porting status. There were 61 tickets approved for back-porting as part of the Data Preview 0.2 and version 23 release process. If a ticket was rejected its label was removed, making it hard to determine counts for the number of tickets in that category. Three tickets were left in the requesting state in case

[2] `https://developer.lsst.io/work/backports.html`

they were needed, one is for a clean-up to the database schema that was discovered after we had finalized the processing; another was for an improvement to the graph-building efficiency but would have involved a very difficult back-port because there had been a package reorganization since the release branch had been created; and the final ticket was an improvement to the matched catalog filtering.

Once all the necessary back-porting has been completed for a specific step, the release manager would be instructed to start the process of creating a new patch release of the Science Pipelines. During DP0.2 we made two additional formal releases of the version 23 software: v23.0.1 and v23.0.2. This allowed us to state which release was used for each step, although we ensured that changes in later patch releases would not affect the processing from steps that were already completed using older patch releases.

3 DP0.2 processing on Google

Description of the workflow and the Jira system - refer to RTN-039

Discuss some of the issues which came up and how they were resolved.

4 Data Product Quality Assurance

During DP0.2, the Verification and Validation Team was responsible for identifying problems and bugs in the pipelines and data products. There were there main phases of V&V work:

- A period of analysis using a “pilot run” before the start of production, which ran a single tract through all steps of the pipeline, using the codebase planned for the release.
- Two “gates”, one at the end of single frame processing and another after coadd construction, where production was halted for V&V to confirm that all the data products were ready before moving on to the next step of processing.
- Spot checks during processing, and follow-up of unexpected errors or failed tasks.

During these main phases, the V&V team made extensive use of the plotting capabilities in

analysis_drp along with adding new diagnostic plots. Much of the analysis was performed by writing notebooks to test out new diagnostics for data products that were recently added to the pipelines. The team also drew on experience from many prior processings (particularly of Hypersuprime-cam) to quickly distinguish "known" problems from new problems.

A notable success occurred during coadd construction, when as part of the spot checks during processing the team noticed some regions inside successfully-processed patches had no coadd sources detected. One of the plots that lead to this discovery is shown below. This was particularly unexpected because entire patches are expected to succeed or fail entirely, it was highly unusual for portions to fail silently.

The eventual explanation was that the coaddition code operated on sub-patch-sized regions sequentially, in order to limit peak memory usage, and so it would read from disk different portions of the input warp images as it progressed. On a typical POSIX filesystem these reads typically either all succeed or all fail, but in the cloud environment the object store would sometimes deny individual requests as a form of rate-limiting. The coaddition code could have caught this, but since that type of failure was never encountered in prior usage, it mistakenly proceeded without raising an exception. Because this issue was identified early during coaddition, only a few days worth of processing had to be redone.

The data previews are also a chance to learn from the issues that we didn't catch during V&V; most notably were two cases where invalid flux calibrations were being applied to certain measurement algorithms, resulting in NaNs in the output. This case shows the value of having a wide breadth of testing coverage. For future releases we will ensure that we have some form of testing for every column in every user-facing data product. Even if the tests are relatively simple, they may identify significant issues.

As one further example of potential issues encountered during production, a few days after the start of single frame processing the pipelines began to show hundreds of failures with the error message `Exception ValueError: No reference objects supplied`. This was not seen in the pilot run, so production was paused while we investigated. By plotting the locations of these failures on the sky, we determined that these sensors fell outside the footprint of the stellar input catalog, and hence there were no stars available for calibrating these images. These were thus "unprocessable" and no corrective action was required, but it illustrated the value of realtime error collection and monitoring during production.

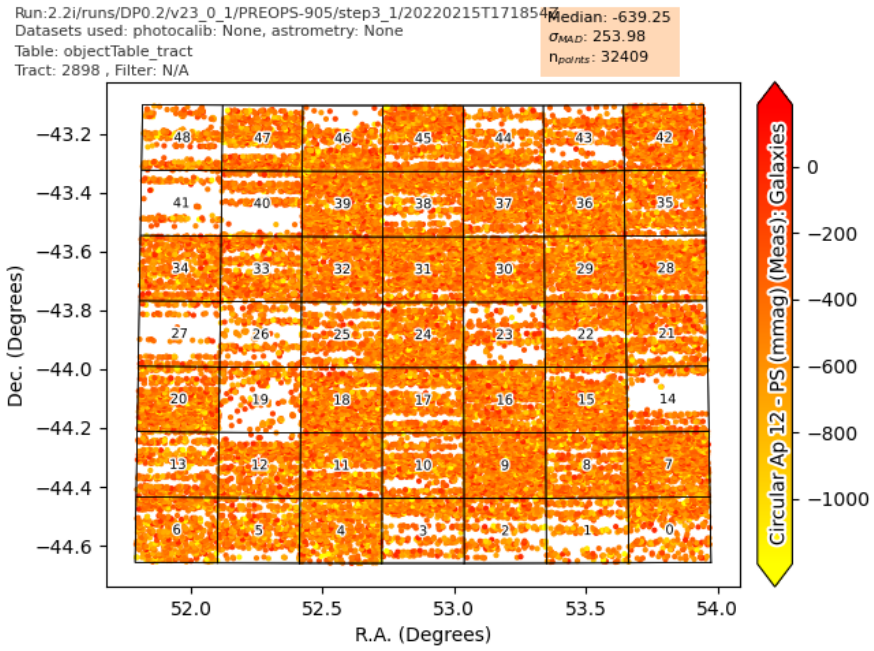


FIGURE 2: The regions of the coadd that lack any Objects were caused by an unexpected failure mode in accessing the input images, which was possible on cloud infrastructure but not seen on POSIX files.

5 Science Platform, user front end for DP0.2

DP0.1 upgrade image services any new issues but mainly deployment and control, issue handling

5.1 Qserv

Qserv is a horizontally scalable parallel SQL database system, developed by the Rubin Observatory specifically to support the LSST production catalog use case. Qserv is further described in LDM-135.

The DP0.2 simulated survey covers about 1/60th of the anticipated on-sky area of the full survey (roughly 300/18,000 sq. degrees), at about half the anticipated observational depth of the full survey. The generated catalogs contain ~136 billion rows in total. On disk, including necessary indexes, this amounts to ~30 TiB of database storage. A Qserv cluster with one head node and five worker nodes was comissioned in the IDF cloud computing environment to host and serve this data.

5.1.1 Enhancements delivered during DP0.2

- **Ingest systems:** At these scales, a trouble-free database ingest from DRP pipeline outputs to a complete online database takes several days. All catalog data must be format converted from the parquet outputs as delivered by DRP to CSV files needed for efficient load at the Qserv worker nodes. Data after conversion must be sharded, distributed, loaded at worker nodes, and appropriate indices compiled. A significant fraction of the Qserv codebase is concerned only with the management, tracking, and automation of these ingest activities.

  Enhancement of these Qserv subsystems continued during the course of the DP0.2 exercise, and improvements were deployed continuously into production. These improvments were aimed at streamlining the ongoing ingest campaigns, and included such features as provision of fully asynchronous task APIs to upstream ingest tooling, refinements to the ingest state machine for improved observability of ongoing ingest activities, and finer-grained control over publication/retraction/replacement life-cycle of individual tables.

- **Diagnostic monitoring:** Qserv provides its own in-built administration dashboard, which has proved to be an indispensible tool for monitoring and troublshooting operating instances. This subsystem also saw lots of improvement during the course of DP0.2, including instrumentation of many additional aspects of Qserv related to ingest and query processing activities, an enhanced query monitor / query history, and overall performance improvments to the administration dashboard itself.

- **Optimized point-in-polygon spherical geometric queries:** Rather late in timeline for DP0.2 a decision was taken to host IVOA ObsCore observational metadata within Qserv, since doing so would enable a much richer interaction model with image data and image services being offered for the first time in DP0.2. This interaction model required support for point-in-polygon spherical geometry queries, and needed spatial indexing features beyond the previously established scope of Qserv for DP0.2 in order to perform acceptably. A workable strategy for efficiently processing these sorts of queries based on integration of several bits of existing functionality was developed and successfully delivered into production.

- **Photometry UDFs:** Some photometry-related UDFs were developed and contributed upstream to the SciSQL project (https://github.com/smonkewitz/scisql), along with general modernization and infrastructural improvements of SciSQL. These UDFs can simplify construction of queries against the LSST data model, which to date is committed to

provision of linear flux measurements rather than logarithmic magnitudes (DOCUMENT-27758).

5.1.2 Some lessons learned

DP0.2 proved to be a very valuable exercise in shaking down Qserv and surrounding tooling and operating processes. Some take-aways related providing catalog database service for DP0.2:

- **Timely schema metadata is hard to get.** Rubin's middleware has an extremely flexible data model. While this provides a lot of agility in pipeline development, when it comes to provision of catalog services at some point before database ingestion a concrete schema commitment must be made.

  While Rubin does have an accepted format for such schema descriptions and a sufficient source control and deployment practice around this, the schema descriptions themselves run to the many thousands of lines and nobody chomps at the bit to make sure these are maintained, current, complete, and error free. Otherwise ready-to-go catalogs have on more than one occasion been held up in release by the need for last-minute revisions to the schema descriptions.

  Remediation: much has been done already to drive schema consumers toward a single source of truth and to streamline the deployment and update of schemas, once updates are committed to source control. Since schema description deliverables have consistently been seen to lag, it remains for management to appropriately pre-load the demand for these.

- **Data curation – try (even) harder.** Often times, catalog data curation issues are not easily detected until the "end of the line", when a completed data release is presented for database ingestion. Representation issues, key constraints, and referential integrity issues may not be apparent until a full set of tables is available. Some curation issues may not even be detected until *after* database ingestion, when the catalogs can first conveniently be queried in the large.

  For DP0.2, there was a plan to mitigate turbulence at the end of the line by obtainin and ingesting a representationally complete subset of catalog products early in the release cycle. In practice, this had only limited success. Inevitable delays on all fronts compressed the time between availability of the subset products and the full release

products. Additionally, sequencing of DRP activity meant while some subset tables were available usefully early in the release cycle, others were not available until much later, closer to the full release. Predictably, curation issues *did* exist that thus were not identified until fairly late in the release cycle, and release of several tables in the data products was delayed while these tables were re-processed upstream and then re-ingested into the database.

Remediation: work with DRP to see if a more complete set of tables can feasibly be obtained earlier in the release cycle. Schedule subset production and ingest even more conservatively.

- **Watch out for automated test lacunae.** Qserv contains some distributed reference match machinery to facilitate performant joins between catalogs. It turns out that DP0.2 includes a truth table and associated match table which depend upon this functionality. Though this feature is not otherwise commonly in use, the Qserv development team maintained a high level of confidence that it was ready to go, because it had known coverage in the automated integration test suites, and "if it was broken, we'd know about it."

  As it turned out, the relevant integration tests had been commented out of the test suite sometime in the previous year as a temporary measure while working through some containerization/deployment issues, test coverage had never been subsequently restored, and the uncovered feature, when put into use, was found to have some minor issues.

  Remediation: a thorough survey of the test suites was conducted, and commented/disabled tests were updated as necessary and returned to active service.

6 Community engagement

Organisation and execution of the tutorials, docs, assemblies ..

A References

[LDM-135], Becla, J., Wang, D., Monkewitz, S., et al., 2017, *Data Management Database Design*, LDM-135, URL https://ldm-135.lsst.io/,
Vera C. Rubin Observatory Data Management Controlled Document

[DMTR-351], Carlin, J., 2022, *Characterization Metric Report: Science Pipelines Version 23.0.0*, DMTR-351, URL https://dmtr-351.lsst.io/,
Vera C. Rubin Observatory Data Management Test Report

[DOCUMENT-27758], Ivezić, Ž., the LSST Project Science Team, 2018, *On the Choice of LSST Flux Units*, DOCUMENT-27758, URL https://pstn-001.lsst.io/,
Vera C. Rubin Observatory

LSST Dark Energy Science Collaboration (LSST DESC), Abolfathi, B., Alonso, D., et al., 2021, ApJS, 253, 31 (arXiv:2010.05926), doi:10.3847/1538-4365/abd62c, ADS Link

[RTN-001], O'Mullane, W., 2021, *Data Preview 0: Definition and planning.*, RTN-001, URL https://rtn-001.lsst.io/,
Vera C. Rubin Observatory Technical Note

O'Mullane, W., Economou, F., Huang, F., et al., 2021, arXiv e-prints, arXiv:2111.15030 (arXiv:2111.15030), ADS Link

[RTN-039], Yanny, B., 2022, *Compute Resource Usage of DP0.2 production run*, RTN-039, URL https://rtn-039.lsst.io/,
Vera C. Rubin Observatory Technical Note

B Acronyms

Acronym	Description
CAM	CAMera
CSV	Comma Separated Values
DC2	Data Challenge 2 (DESC)
DESC	Dark Energy Science Collaboration
DM	Data Management
DMCCB	DM Change Control Board

DMTR	DM Test Report
DP0	Data Preview 0
DPLT	DP Leadership Team
DRP	Data Release Production
FrDF	French Data Facility
IDF	Interim Data Facility
IVOA	International Virtual-Observatory Alliance
LDM	LSST Data Management (Document Handle)
LSST	Legacy Survey of Space and Time (formerly Large Synoptic Survey Telescope)
NCSA	National Center for Supercomputing Applications
OPS	Operations
POSIX	Portable Operating System Interface
RTN	Rubin Technical Note
SQL	Structured Query Language
T/CAM	Technical/Control (or Cost) Account Manager
UKDF	United Kingdom Data Facility
USDF	United States Data Facility