LARGE SYNOPTIC SURVEY TELESCOPE

Large Synoptic Survey Telescope (LSST)

LDM-503-2 (HSC Reprocessing) Test Report

Jim Bosch, Hsin-Fang Chiang, Michelle Gower, Mikolaj Kowalik, Timothy Morton, John D. Swinbank

DMTR-51

Latest Revision: 2018-02-12

DRAFT

Abstract

This is the test report for LDM-503-2 (HSC Reprocessing), an LSST DM level 2 milestone pertaining to the LSST Level 2 System.

Change Record

Version	Date	Description	Owner name
1	2018-01-11	Initial version.	Bosch, Chiang, Gower, Kowalik, Morton, Swinbank.
1.1	2018-01-11	Implementation of RFC-425	T. Jenness

Document curator: John D. Swinbank
Document source location: `https://github.com/lsst-dm/DMTR-51`
Version from source repository: 8b12d4d

Contents

LDM-503-2 (HSC Reprocessing) Test Report

1 Introduction

1.1 Objectives

This document describes the results of tests carried out in late (calendar) 2017 on the LSST Level 2 System in order to assess progress against the LSST DM level 2 milestone LDM-503-2. We report on the success or failure of applicable test cases and assess the state of the software and services tested.

1.2 Scope

The overall test plan for the LSST Data Management system is described in LDM-503. This document specifically refers to the late (calendar) 2017 milestone LDM-503-2, which tests the LSST Level 2 System. The overall LSST Level 2 System test specification is defined in LDM-534. The test plan for LDM-503-2 involves the execution of the entire DRP-00 (Small Scale Data Release Processing) specification, including the following test cases:

DRP-00-00 Installation of the Data Release Production science payload

DRP-00-05 Execution of the DRP Science Payload by the Batch Production Service

DRP-00-10 Data Release Includes Required Data Products

DRP-00-15 Scientific Verification of Source Catalog

DRP-00-25 Scientific Verification of Object Catalog

DRP-00-30 Scientific Verification of Processed Visit Images

DRP-00-35 Scientific Verification of Coadd Images

1.3 System Overview

The LSST Level 2 System is that part of the LSST Data Management system which will be responsible for scheduled, periodic data processing during LSST operations. The most prominent example of such processing is the generation of LSST's annual releases of both catalog and image data. However, the LSST Level 2 System is also responsible for the generation of template images (used in the nightly processing system) and calibration products (used in both nightly and annual processing), and for the Level 2 Quality Control Service (LDM-148). The LDM-503-2 milestone focuses only on the data release production part of the system.

Note that we may broadly think of the LSST Level 2 System as consisting of two independent parts: the Batch Production Service, which provides scheduling and workflow services, and the Science Payloads, which contain the algorithmic content. The LDM-503-2 milestone exercises both parts of the system.

1.4 Applicable Documents

LDM-294	LSST DM Project Management Plan
LDM-503	DM Test Plan
LDM-534	LSST Level 2 System Test Specification

1.5 References

[1] **[DMTR-31]**, Chiang, H.F., Daues, G., Thrush, S., The NCSA Team, 2017, *S17B HSC PDR1 Reprocessing Report*, DMTR-31, URL `https://ls.st/DMTR-31`

[2] **[LDM-556]**, Dubois-Felsmann, G., 2017, *Data Management Middleware Requirements*, LDM-556, URL `https://ls.st/LDM-556`

[3] **[DMTN-059]**, Gower, M., 2017, *Batch Processing Facade Prototype 0.1*, DMTN-059, URL `https://dmtn-059.lsst.io`,
LSST Data Management Technical Note

[4] **[LDM-148]**, Lim, K.T., Bosch, J., Dubois-Felsmann, G., et al., 2017, *Data Management System Design*, LDM-148, URL `https://ls.st/LDM-148`

[5] Moreau, L., Clifford, B., Freire, J., et al., 2011, Future Generation Computer Systems, 27, 743, URL `https://eprints.soton.ac.uk/271449/`

[6] **[LDM-503]**, O'Mullane, W., Jurić, M., Economou, F., 2017, *Data Management Test Plan*, LDM-503, URL `https://ls.st/LDM-503`

[7] **[LDM-294]**, O'Mullane, W., Swinbank, J., Jurić, M., DMLT, 2017, *Data Management Organization and Management*, LDM-294, URL https://ls.st/LDM-294

[8] **[LDM-534]**, Swinbank, J.D., 2017, *Level 2 System Software Test Specification*, LDM-534, URL https://ls.st/LDM-534

1.6 Document Overview

Section 2 of this document provides details of the LSST Level 2 System baseline used for this test, including relevant hardware and software configurations. Section 3 lists the individuals involved in performing the tests. Section 4 provides an overview of the test results, while Section 5 provides more detailed results from each individual test case.

2 Test Configuration

2.1 Documents

This test report refers to the execution of tests DRP-00-00 through DRP-00-35 in LDM-534 version 3.1.

2.2 Hardware

All tests were executed on systems in the LSST Data Facility.

Software installation (DRP-00-00) and scientific analysis work (DRP-00-10 through DRP-00-35) were carried out on `lsst-dev01.ncsa.illinois.edu`. At time of text execution, this was a Dell PowerEdge R730 with 24 physical Intel Xeon E5-2690v3 CPU cores at 2.60 GHz and 256 GB of RAM.

Bulk data processing (DRP-00-05) was carried out on the LSST Verification Cluster (VC). At the time of test execution, the VC provided 48 Dell C6320 nodes, each with 24 physical Intel Xeon E5-2680v3 CPU cores at 2.50 GHz and 128 GB of RAM.

2.3 Software

All systems used for testing — including both `lsst-dev01` and the VC nodes — were running CentOS Linux release 7.4.1708. The `devtoolset-6`[1] toolchain, including GCC[2] version 6.3.1, was enabled for all tests, with the exception DRP-00-00 for reasons described in §6.1.

2.4 Input Data

Input data for all tests was based on the Hyper Suprime-Cam "RC1" dataset, as described in the appendix to LDM-534. Calibration dataset 20170105 was used, defined as per DMTR-31.

3 Personnel

Test case DRP-00-00 was executed by John Swinbank (University of Washington).

Test case DRP-00-05 was executed by Hsin-Fang Chiang (NCSA), Michelle Gower (NCSA), Mikolaj Kowalik (NCSA), Greg Daues (NCSA) and Rob Kooper (NCSA).

Test cases DRP-00-10, DRP-00-15, DRP-00-25, DRP-00-30 and DRP-00-35 were executed by Jim Bosch (Princeton), Lauren MacArthur (Princeton) and Tim Morton (Princeton).

[1] https://access.redhat.com/documentation/en-us/red_hat_developer_toolset/6/html/6.0_release_notes/
[2] https://gcc.gnu.org/

4 Overview of the Test Results

4.1 Summary Table

TEST CASE ID	PASS/FAIL	COMMENTS
DRP-00-00	Pass	Refer to §5.1.
DRP-00-05	Pass	Refer to §5.2.
DRP-00-10	Pass	Refer to §5.3.
DRP-00-15	Pass	Refer to §5.4.
DRP-00-25	Partial Pass	An issue was identified when comparing the aperture corrections generated for different photometry algorithms. Refer to §5.5.
DRP-00-30	Pass	Refer to §5.6.
DRP-00-35	Pass	Refer to §5.7.

4.2 Overall Assessment

All but one test cases were completely passed. This successfully demonstrated the ability to install the LSST Science Pipelines on a production system, to perform at-scale data processing using existing execution middleware, to generate scientifically meaningful results, and to check those results for validity.

A single test case was recorded as a partial pass. This was due to a discrepancy in the aperture corrections being calculated for different photometric algorithms. This is not regarded as fundamental to the goals of the LDM-503-2 milestone. However, investigation of this issue has been ticketed as DM-13058, and will be scheduled as part of upcoming development work.

4.3 Impact of Test Environment

The DRP-00-05 test was performed — as per its specification — using the DESDM framework, as described in DMTN-059. The DESDM system is still under evaluation for use within LSST data processing, and may not be deployed during operations. The results of this test therefore establish a baseline system, rather than testing a true operational deployment.

4.4 Recommended Improvements

4.4.1 Operational

- For LDM-503-2, some tests were carried out manually through spot checks. Further development of verification tools to perform these and additional checks is needed.
- The handoff process from Pipelines to the Data Facility needs to be improved. Improvements include providing details about changes in the codebase. Currently the operator is digging through `pipe_drivers` code to figure out some of these details in addition to repeatedly consulting pipeline developers. Providing change details during handoff will not only speed up configuring and executing the pipeline, but will also provide the information needed to configure the verification tests themselves. The Data Facility should provide a first proposal on what information they would like provided.
- Making what can seem like minor changes to the pipeline definition can require much integration time. Examples of pipeline changes that affect the pipeline execution configuration and verification tests:
  - Pipeline code changes or task configuration changes can alter the required input files or what output files are created.
  - Changing what pipeline steps should be executed may require a pipeline task configuration change in addition to modifying the overall pipeline definition.

  One of the goals of the new Butler and SuperTask effort is to minimize the amount of integration effort. When running the exact same configuration as the developer, the new Butler and SuperTask effort could definitely help. However, it is not yet clear how much integration/configuration effort will still be required with the new Butler+SuperTask if the operator wishes to run something different than the default pipeline configuration.
- At the time of this test, it is not yet understood exactly how the Data Facility would configure the execution of the pipeline for best scalability and management. This execution may not match small-scale developer tests where getting individual results faster is more important. The desire is to have the test match the expected release campaign execution configuration as much as possible. As work is done on both the pipeline science and the Batch Processing Service, effort should be undertaken to determine the release campaign execution configuration.

4.4.2 Infrastructure

Problems with the Verification Cluster infrastructure were encountered during the execution of DRP-00-05 as described in §6.2. These problems may be mitigated by increasing the scratch storage space available on the cluster and adding dedicated production job queues to the Slurm system (IHS-612) for future tests.

4.4.3 DESDM Execution Framework

A number of improvements to the DESDM framework are suggested by this work. These include:

- The step in DESDM framework which configures each step for each compute job does not scale well at the size of the larger tract. In DESDM Operations the unit of work in a single submission is smaller, so this has not been an issue there. For short term use, this code could be profiled to see if there are straight forward ways to speed it up.
- In DESDM Operations, the single submission's unit of work is chosen to be the amount of work the operator is willing to start over from the beginning. In this test, this unit of work is tract using the current operational configuration of the pipeline. Restarting a tract from the very beginning adds a lot of extra compute time and resources to complete the test (especially when including the scaling issue mentioned in the previous item). Short term solutions include modifying operational configuration of the pipeline to make smaller units of work (which currently means human overhead in managing more submissions). Also, DESDM allows for manually starting a pipeline execution using outputs in the Data Backbone from a previous submission. But this requires an operational pipeline configuration change which takes human knowledge, effort and time.
- The workflow definition (wcl) that describes a pipeline to the DESDM framework includes how to name output files. These naming patterns need to match the manually-made Butler policies. For this test, this was done manually and was very error-prone. Until this issue is resolved by the Batch Processing Service integrating updated DM Middleware (following LDM-556), modifying the DESDM framework to automatically create the Butler policy file from the pipeline definition would speed up the integration process.

At the time of writing, the long term execution framework for LSST processing has not yet been selected; we note that it may make sense only to implement these improvements to

DESDM if it is ultimately chosen for this role.

4.4.4 Scientific Analysis

Most of the analysis performed in support of the DRP-00-10 through DRP-00-35 milestones was based on manual inspection of pregenerated plots. In future, much of this could be automated by means of specific numerical thresholds which must be crossed to trigger manual intervention. This is especially significant for those tests which involve inspecting visit images and source catalogues; there are many more of these than there are objects and coadd images. Work to devise the appropriate metrics is scheduled as DM-11312.

5 Detailed Test Results

5.1 DRP-00-00

The string `Ok` was returned when executing

```
$ python bin/compare expected/Linux64/detected-sources.txt
```

after running the `demo.sh` on both the cluster head node (`lsst-dev01`) and on an example compute node.

5.2 DRP-00-05

1. Check the existence of the expected files: PASSED
   (a) Table 1 list all release data products per tract. For each product, we provide the expected number of files to be generated (where available) and the number of files generated in practice. Each of these files was checked to ensure that it contained some data (i.e., the size of the file was non-zero).
   (b) To verify the physical location of files on the filesystem match the location information tracked in the Data Backbone database tables, we used the tool `compare_db.py` from the DESDM `FileMgmt` package. Paths, file sizes, and checksums (MD5) were compared. The test results were that both the database and filesystem matched with 50656 files in tract 8766, 52041 files in tract 8767, and 273375 files in tract 9813.
2. Check existence of the expected metadata: PASSED
   The following metadata is expected to have been saved
   - calexp: tract, visit, filter, ccd
   - deepCoadd_calexp: tract, patch, filter

   It was verified that the above mentioned metadata had non-NULL values stored for the data products in the Data Backbone database tables.
3. Check existence of the expected provenance: PASSED
   (a) For each file, the provenance system was checked to ensure that there were no:

i. Missing direct association of output files with the processing attempt;
ii. Missing "was-generated-by" association (per the Open Provenance Model, [5]);
iii. Instances in which the "was-generated-by" association did not belong to the specificed processing attempt.

(b) Via manual spot checks, it was verified that information linking input files to each step was saved to the Data Backbone database tables.

4. Check (existence) runtime metrics: PASSED

(a) Table 2 shows wall-clock time for running the entire pipeline for each tract and total cpu time per execution.
(b) Table 3 provides details of execution time and memory usage at a per-process level.

5.3 DRP-00-10

The week 44 reprocessing of the Hyper Suprime-Cam RC1 dataset was used to execute this test case. Details of this reprocessing—including failures, which are acceptable per the test specification—are available at DM-12388. All expected products were found to exist.

5.4 DRP-00-15

The week 44 reprocessing of the Hyper Suprime-Cam RC1 dataset was used to execute this test case. Details of this reprocessing—including failures, which are acceptable per the test specification—are available at DM-12388.

Scientific assessment was carried out using the `qa-image-browser.ipynb` Jupyter notebook, made available from `https://github.com/lsst-dm/pipe_analysis/`. The version of the notebook from commit `8705ef7` was used.

All plots produced by that notebook were scrutinized by the test team. It was noted that:

- When comparing aperture corrections across photometry algorithms, some scatter was observed at the bright end and that the narrow-band (NB9021) observations had more outliers than others.
  - This was assessed as falling within normal tolerances, and no further action is required.

- When comparing photometric measurements with the reference catalog, a significant (20 mmag) offset was observed in tracts 8766 and 8767 in the R band.
  - This offset is regarded as falling within normal tolerances, but worthy of further investigation.
  - Ticket DM-13056 has been filed.
- When comparing astrometric measurements with the reference catalog, a small but systematic offset was observed in tracts 8766 and 8767 in the I band, and in tract 9813 in the R band.
  - This offset is regarded as falling within normal tolerances, but worthy of further investigation.
  - Ticket DM-13057 has been filed.

The test team agreed that all measurements fall within acceptable tolerances, and therefore regard the test case as having been passed. DM-13056 and DM-13057 will be scheduled for further investigation as part of the regular development cycle.

5.5 DRP-00-25

The input dataset and analysis tools used for this test case were the same as for DRP-00-15 (see §5.4).

The test team agreed that:

- PSF models correctly predict the ellipticities of stars over the field of view.
- Photometric and astrometric measurements are consistent with reference catalogs.
- Forced and unforced measurements are consistent.
- The slope of the stellar locus appears appropriate.

However, aperture corrections were not found to be consistent between photometry algorithms when performing forced measurement. This inconsistency was outside acceptable tolerances, and has been ticketed for further investigation as DM-13058.

This test case is regarded as "partially passed": its overall goal has been accomplished, but understanding the origin of the inconsistent aperture corrections will require further work which will be undertaken during the upcoming developing cycle.

5.6 DRP-00-30

The input dataset used for this test case was the same as for DRP-00-15 (see §5.4).

The test was carried out as described. All required data products were accessible. Data was inspected by the Data Release Production Science Lead, who certified that artifacts were correctly matched and that backgrounds were not over-subtracted.

This test case is therefore successfully passed.

5.7 DRP-00-35

The input dataset used for this test case was the same as for DRP-00-15 (see §5.4).

The test was carried out as described. All required data products were accessible. Data was inspected by the Data Release Production Science Lead, who certified that artifacts were correctly matched, satellite trails and ghosts were properly rejected, and that backgrounds were not over-subtracted.

It was noted that:

- We note that the background model is stored separately from the image, but is readily accessible;
- Pixels marked as contaminated by cosmic rays that had been successfully interpolated were not being included in the coadd, unnecessarily making much of the coadd unusable for precision science. This issue had already been identified and ticketed as DM-9953; it has subsequently been resolved.

This test case is therefore successfully passed.

6 Problems encountered

6.1 DRP-00-00

Per §2.3, the initial attempt to execute this test case was based on the `devtoolset-6` toolchain, as is current LSST best practice. However, although the code executed without error, the numerical values recorded did not fall within the expected tolerances. This is due to a known compatibility issue between the version 14.0 release of the Pipelines and `devtoolset-6` which was fixed on DM-10902. The test was repeated without `devtoolset-6` enabled and was then completed successfully.

6.2 DRP-00-05

- During execution, several processing attempts failed due to unstable GPFS availability resulted from hardware problems and network issues. We retried the attempts after the transient problems were resolved, following the Test Specifications. The Data Facility's Network Group replaced the broken routers and purchased a hot spare.
- The scratch filesystem, where the prototype DBB stores the files, was close to full several times during test execution. We had to manually purge files to ensure sufficient space for the execution.
- The Verification Cluster can be very busy making it difficult to acquire nodes to do the test work (both initial configuration test and actual full test). A production reservation was added to Slurm to support the test execution.
- We needed to apply a patch on top of the Stack release to workaround the issue that commas are part of the data IDs; this has been a known problem as discussed in RFC-361. The fix has been agreed in RFC-365 for future implementation in DM-11874, DM-11875, and DM-11876.
- The current implementation of the "Data Butler" I/O abstraction in the DM Stack is hard to adapt to file-based production data management systems that do not utilize a single global filesystem, for example the production DES Data Management System used for the Data Backbone and Batch Processing Service in this test. This caused lengthy integration issues and fragility in the system design. It is expected that future work on the DM middleware will resolve these concerns (LDM-556).

6.3 DRP-00-10

An initial attempt was made to execute this test case based on the week 46 reprocessing of the Hyper Suprime-Cam "RC" dataset[3]. However an (independently discovered) bug in the version of the code used to process this data rendered it unusable for these tests. The bug was captured as DM-12882 and has subsequently been resolved. For the purpose of executing this test case (and others which depend upon in), the week 44 reprocessing[4] was used instead.

6.4 DRP-00-15

None.

6.5 DRP-00-25

None.

6.6 DRP-00-30

None.

6.7 DRP-00-35

None.

7 Deviations from test cases/procedures

7.1 DRP-00-00

None.

7.2 DRP-00-05

- For the first processing attempts of tracts 8766 and 8767, there was a misconfiguration which executed both attempts on the same set of worker nodes simultaneously. This

[3] Refer to DM-12811 and `https://confluence.lsstcorp.org/display/DM/Reprocessing+of+the+HSC+RC+dataset`.
[4] See DM-12388.

does not match the procedure in the Test Specification. While tract 8766 finished successfully, it took longer wall-clock time than expected. Because it produced expected outputs, tract 8766 was not rerun. The first attempt of tract 8767 failed due to a network problem, and the time reported in Table 2 is the result of a new attempt (based on the Test Specification regarding resubmission of intermittent infrastructure problems) configured correctly using non-shared nodes. Thus, the wall-clock time for tract 8767 in Table 2 is closer to the expected time.

7.3 DRP-00-10

None.

7.4 DRP-00-15

None.

7.5 DRP-00-25

None.

7.6 DRP-00-30

None.

7.7 DRP-00-35

None.

A Analysis of Bulk Data Processing

TABLE 1: File counts. Expected values came from corresponding weekly RC exections. Symbol † dentoes dataset types which differs from weekly execution and ‡ indicates DESDM specific dataset types.

	tract 8766		tract 8767		tract 9813	
Dataset type	*expected*	*actual*	*expected*	*actual*	*expected*	*actual*
cat_corr_src†	–	3850	–	3956	–	24103
deepcoadd†	–	405	–	405	–	465
deepcoadd_calexp	405	405	405	405	465	465
deepcoadd_calexp_bkg	405	405	405	405	465	465
deepcoadd_det	405	405	405	405	465	465
deepcoadd_directwarp	–	2276	–	2389	–	15873
deepcoadd_forced_src	405	405	405	405	465	465
deepcoadd_meas	405	405	405	405	465	465
deepcoadd_measmatch	405	405	405	405	465	465
deepcoadd_measmatchfull	405	405	405	405	465	465
deepcoadd_mergedet	81	81	81	81	79	79
deepcoadd_ref	81	81	81	81	79	79
img_corr†	–	3850	–	3956	–	24103
list‡	3297	3297	3403	3403	17415	17415
log‡	7957	7957	8169	8169	42448	42448
processccd_md_boost	–	3850	–	3956	–	24103
sfm_calexpbkgd	–	3850	–	3956	–	24103
sfm_flattened_thumb	–	3850	–	3956	–	24103
sfm_oss_thumb	–	3850	–	3956	–	24103
sfm_srcmatch	–	3850	–	3956	–	24103
sfm_srcmatchfull	–	3850	–	3956	–	24103
wcl‡	3	3	3	3	3	3
totals	–	50656	–	52041	–	273375

TABLE 2: The wall-clock time and total execution time in each tract. See §7.2 for an explanation of the longer wall-clock time for tract 8766.

Tract	Wall-clock Time (hr)	Total User CPU Time in Running Executables (hr)
8766	20.98	392.792236
8767	7.82	378.250683
9813	42.10	1979.60976

TABLE 3: The CPU time used by each executable in each tract. Also shown is the maximum resident set size (RSS) — corresponding to the maximum amount of main memory — used by each executable

Executable	**Execution count**	**User Time (min)**	**System Time (min)**	**Maximum RSS (kB)**
		tract 8766		
processCcd.py	3850	9390.55	332.29	1956920
makeCoaddTempExp.py	2325	1073.35	185.30	1113296
assembleCoadd.py	405	665.03	63.54	1820116
detectCoaddSources.py	405	56.12	18.79	955692
mergeCoaddDetections.py	81	27.92	3.45	462000
measureCoaddSources.py	405	9529.12	32.96	3694980
mergeCoaddMeasurements.py	81	31.46	4.70	1741084
forcedPhotCoadd.py	405	2794.00	25.90	2196376
		tract 8767		
processCcd.py	3956	9262.97	331.62	1911480
makeCoaddTempExp.py	2431	1104.69	253.26	1117804
assembleCoadd.py	405	687.55	66.34	1894980
detectCoaddSources.py	405	56.19	19.12	953240
mergeCoaddDetections.py	81	28.14	3.56	463896
measureCoaddSources.py	405	8718.07	35.31	3594972
mergeCoaddMeasurements.py	81	30.14	4.68	2188696
forcedPhotCoadd.py	405	2807.30	25.85	2166484
		tract 9813		
processCcd.py	24103	64001.65	2091.06	1942904
makeCoaddTempExp.py	16327	6329.18	1044.22	1128420
assembleCoadd.py	465	3900.33	383.54	7476556
detectCoaddSources.py	465	70.59	23.92	1035160
mergeCoaddDetections.py	79	57.24	3.80	512504
measureCoaddSources.py	465	31375.48	83.69	12747664
mergeCoaddMeasurements.py	79	47.45	6.53	4746248
forcedPhotCoadd.py	465	12994.67	49.43	4489536