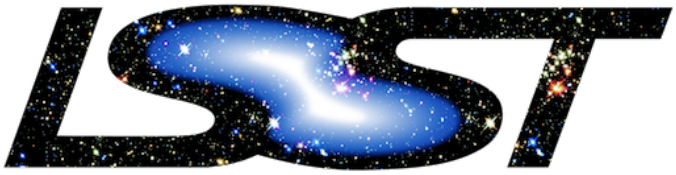

LARGE SYNOPTIC SURVEY TELESCOPE

Large Synoptic Survey Telescope (LSST)
Data Management

LDM-503-07 (Camera Data Processing) Test Plan and Report

John Swinbank

DMTR-112

Latest Revision: 2019-01-17

Abstract

This is the test plan and report for LDM-503-07 (Camera Data Processing), an LSST level 2 milestone pertaining to the Data Management Subsystem.

Change Record

Version	Date	Description	Owner name
	2018-11-26	First Draft	Swinbank, Comoretto
1.0	2019-01-17	Approved on DM-16734	Swinbank, Comoretto

Document curator: John Swinbank
Document source location: `https://github.com/lsst-dm/DMTR-112`
Version from source repository: 5bce6d7

Contents

LDM-503-07 (Camera Data Processing) Test Plan and Report

1 Introduction

1.1 Objectives

This test plan demonstrates that the LSST Science Pipelines can successfully be used to load and perform basic processing on data from the LSST Camera test systems.

In particular, it will demonstrate that:

- Data from the Camera test systems has been made available at the LSST Data Facility;
- Data from the Camera test systems can be accessed using the "Data Butler" I/O abstraction, and loaded into the LSST Science Platform Notebook Aspect for processing and inspection;
- Basic LSST Science Pipelines Tasks can be used to process and manipulate Camera data;
- Camera data can be sent to the Firefly visualization tool for display.

Verification that the data processing is "correct" falls outside the scope of this test plan: both Camera data and DM code is evolving rapidly, and this exercise will not demonstrate that particular thresholds in terms of data processing fidelity have been reached. Rather, the focus here is on demonstrating successful integration and interface compatibility.

1.2 Scope

The overall strategy for testing and verification with LSST Data Management is described in LDM-503.

Success in this test plan is intended to demonstrate completion of the milestone LDM-503-07 “Camera Data Processing”.

1.3 System Overview

This test plan addresses primarily the integration between early data coming from the LSST Camera and the data access facilities provided by the LSST Data Management system.

In the process, it will exercise:

- The “Data Butler” I/O abstraction provided by Data Management;
- The Notebook Aspect of the LSST Science Platform;
- Algorithmic code provided by the LSST Science Pipelines;
- The Firefly image display tool provided by the Science User Interface and Tools group.

1.4 Applicable Documents

LDM-503 Data Management Test Plan
LDM-151 Data Management Science Pipelines Design
LDM-152 Data Management Middleware Design
LDM-542 Science Platform Design

1.5 Document Overview

This document was generated from Jira, obtaining the relevant information from the LVV-P16 Jira Test Plan and related Test Cycles (LVV-C19).

Section 1 provides an overview of the test campaign, the system under test (Data Management), the applicable documentation, and explains how this document is organized. Section 2 describes the configuration used for this test. Section 3 lists all the individuals involved and describes their roles.

Section 4 provides a summary of the test results, including an overview in Table 1, an overall assessment statement and suggestions for possible improvements. Section 5 provides detailed results for each step in each test case.

The current status of test plan LVV-P16 in Jira is Approved.

1.6 References

[1] **[LDM-542]**, Dubois-Felsmann, G., Lim, K.T., Wu, X., et al., 2017, *LSST Science Platform Design*, LDM-542, URL https://ls.st/LDM-542

[2] **[LDM-152]**, Lim, K.T., Dubois-Felsmann, G., Johnson, M., Jurić, M., Petravick, D., 2017, *Data Management Middleware Design*, LDM-152, URL https://ls.st/LDM-152

[3] **[LDM-503]**, O'Mullane, W., Swinbank, J., Jurić, M., Economou, F., 2018, *Data Management Test Plan*, LDM-503, URL https://ls.st/LDM-503

[4] **[LDM-151]**, Swinbank, J.D., et al., 2017, *Data Management Science Pipelines Design*, LDM-151, URL https://ls.st/LDM-151

2 Test Configuration

2.1 Data Collection

Observing is not required for this test campaign.

2.2 Verification Environment

Tests of the Data Butler, the Science Pipelines and the Firefly image display tool will take place within the Notebook Aspect of the LSST Science Platform, as deployed at https://lsst-lspdev.ncsa.illinois.edu/nb and documented at https://nb.lsst.io. This provides a flexible and configurable environment with access to large-capacity filesystems at the LSST Data Facility.

Individual tests will be based on specific machine images provided within the Notebook Aspect, as documented in the relevant test cases.

3 Personnel

The following personnel are involved in this test activity:

- Test Plan (LVV-P16) owner: John Swinbank
- Test Cycles:
  - LVV-C19 owner: Undefined
    * Test case LVV-T374 tester: John Swinbank
    * Test case LVV-T368 tester: John Swinbank
- Additional Test Personnel involved: None

4 Overview of the Test Results

4.1 Summary

Test Cycle **LVV-C19: LDM-503-07: Camera Data Processing**		
test case	**status**	**comment**
LVV-T368	Pass	
LVV-T374	Pass	

Table 1: Test Results Summary

4.2 Overall Assessment

The test campaign has been successfully concluded.

All steps in the accompanying test scripts have been completed successfully. One minor typo was discovered in LVV-T368, but since it was a flaw in the test script rather than the system being tested, it is not regarded as germane to the success of this test campaign.

4.3 Recommended Improvements

The typo in LVV-T368 step 2 (a missing “~/”) should be resolved.

5 Detailed Test Results

5.1 Test Cycle LVV-C19

Open test cycle *LDM-503-07: Camera Data Processing* in Jira.

LDM-503-07: Camera Data Processing
Status: Done

This test cycle defines tests to be performed in late 2018 to demonstrate the current state of integration between the Data Management System and current Camera test datasets.

5.1.1 Software Version/Baseline

This test will be carried out on whatever versions of the LSST Science Platform Notebook Aspect and Portal Aspect are installed on lsst-lspdev.ncsa.illinois.edu at the time of testing. No particular version requirements are specified on those tools.

Pipelines tests will be based on the weekly w_2018_45 version of the codebase.

5.1.2 Configuration

No specific configuration is required beyond the availability of the GPFS filesystem at the LSST Data Facility and access to the Notebook and Portal Aspects of the Science Platform as described above.

5.1.3 Test Cases in LVV-C19 Test Cycle

5.1.3.1 Test Case LVV-T368

Open *LVV-T368* test case in Jira.

This test will check:

- That Camera test data is available for processing in the LSST Data Facility, and accessible through the LSST Science Platform;

- That the Data Management I/O abstraction (the "Data Butler") can load that data into the Science Platform environment;
- That Data Management algorithmic "tasks" can be executed to process that data;
- That results can be displayed in the Firefly display tool.

Preconditions:
Appropriate data — to include a "raw" and a "bias" exposure — from the Camera test systems must be available in a Butler data repository on a filesystem accessible to the Notebook Aspect of the Science Platform.

For the purposes of the following discussion, we assume that:

- Visit 258334666 from RTM (Raft Tower Module) 007 will be used;
- The data is available in a repository at /project/bootcamp/repo_RTM-007/ on the Data Facility GPFS filesystem

In the test script, we refer to "258334666" as "$VISIT_ID" and "/project/bootcamp/repo_RTM-007/" as "$REPOSITORY_PATH"; other data may be substituted as appropriate.

Execution status: **Pass**

Final comment:

Detailed step results:

Step		Description, Results and Status
1	Description	Connect to the Notebook Aspect of the Science Platform following the instructions at https://nb.lsst.io/. Log in, and "spawn" a new machine with image "Weekly 2018_45" and size "small".
	Expected Result	The JupyterLab environment appears.

	Actual Result	• The test was carried out at the University of Washington, which has a Data Facility firewall exemption, so instructions at https://nb.lsst.io/getting-started/logging-in.html describing installing, configuring and connecting with a VPN client were skipped. • The "Weekly 2018_45" image was not listed — only the two most recent weeklies are called out explicitly. However, "w201845" was chosen as a close analog. • The JupyterLab environment appeared as expected.
	Status	Pass

2	Description	Create a terminal session. Use it to set up the LSST tools, then download and build version 5c12b06e6 of obs_lsst:
		
		$ source /opt/lsst/software/stack/loadLSST.bash
		$ setup lsst_distrib
		$ git clone https://github.com/lsst/obs_lsst.git
		$ cd obs_lsst
		$ git checkout 5c12b06e6
		$ setup -k -r .
		$ scons
		
		Arrange for obs_lsst to automatically be added to the environment when starting a new notebook:
		
		$ echo "setup -j -r ~/obs_lsst" >> notebooks/.user_setups
		
		Exit the terminal.

	Expected Result	No errors are seen during execution of the provided commands.

	Actual Result	No expected result is provided. Executing the final command returned: bash: notebooks/.user_setups: No such file or directory This is an obvious typo in the test script. Correcting the command to echo “setup -j -r ~/obs_lsst” >> ~/notebooks/.user_setups caused it to execute successfully.
	Status	Pass
3	Description	Create a new “LSST” notebook. Import the standard libraries required for the rest of this test: import os import lsst.afw.display as afwDisplay from lsst.daf.persistence import Butler from lsst.ip.isr import IsrTask from firefly_client import FireflyClient from IPython.display import IFrame and execute the cell.
	Expected Result	Nothing is printed.
	Actual Result	The imports completed successfully.
	Status	Pass

4	Description	Create a Data Butler client, and use it to retrieve the data which will be used for this test. butler = Butler($REPOSITORY_PATH) raw = butler.get("raw", visit=$VISIT_ID, detector=2) bias = butler.get("bias", visit=$VISIT_ID, detector=2)
	Expected Result	Nothing is printed.
	Actual Result	The commands were executed successfully.
	Status	Pass
5	Description	Initialize the Firefly display system: my_channel = '{}_test_channel'.format(os.environ['USER']) server = 'https://lsst-lspdev.ncsa.illinois.edu' ff='{}/firefly/slate.html?__wsch={}'.format(server, my_channel) IFrame(ff,800,600) afwDisplay.setDefaultBackend('firefly') afw_display = afwDisplay.getDisplay(frame=1, name=my_channel) Click on the link provided after executing the above.
	Expected Result	A Firefly window is shown.
	Actual Result	Firefly window appears.
	Status	Pass
6	Description	Display the raw image data in the Firefly window: afw_display.mtv(raw)

	Expected Result	Raw image data is displayed.
	Actual Result	Image data was seen in the Firefly window.
	Status	Pass

7	Description	Configure and run an Instrument Signature Removal (ISR) task on the raw data. Most corrections are disabled for simplicity. but the bias frame is applied.
		`isr_config = IsrTask.ConfigClass()`
		`isr_config.doDark=False`
		`isr_config.doFlat=False`
		`isr_config.doFringe=False`
		`isr_config.doDefect=False`
		`isr_config.doAddDistortionModel=False`
		`isr_config.doLinearize=False`
		`isr = IsrTask(config=isr_config)`
		`result = isr.run(raw, bias=bias)`

	Expected Result	Nothing is printed.
	Actual Result	Commands successfully executed.
	Status	Pass
8	Description	Display the corrected image data in the Firefly window: `afw_display.mtv(result.exposure)`
	Expected Result	Processed (trimmed, bias-subtracted) image data is displayed.
	Actual Result	Processed data was displayed in the Firefly window.

Status	Pass

5.1.3.2 Test Case LVV-T374

Open *LVV-T374* test case in Jira.

This test will check:

- That raw Camera test data is available on a filesystem in the LSST Data Facility;
- That raw Camera test data can be ingested and made available through the Data Management I/O abstraction (the "Data Butler").

Preconditions:
Appropriate raw data from Camera test systems must be available on a filesystem within the LSST Data Facility. This test data is assumed to include visit 258334666 (hereafter referred to as $VISIT_ID) from RTM (Raft Tower Module) 007 for the purposes of this test, but other, equivalent, may be substituted.

At time of writing, suitable data may be found on the GPFS filesystem at /project/bootcamp/data/LCA-11021_RTM-007/7086/fe55_raft_acq/v0/44981. In future, as data transport procedures to the Data Facility become more streamlined and formalised, this data may be moved elsewhere or made available through some other system. Throughout the test script, we use the string "$INPUT_DATA_DIR" as an alias for "/project/bootcamp/data/LCA-11021_RTM-007/7086/fe55_raft_acq/v0/44981" or wherever this data has been moved to.

Execution status: **Pass**

Final comment:

Detailed step results:

Step		Description, Results and Status
1	Description	Connect to the Notebook Aspect of the Science Platform following the instructions at https://nb.lsst.io/. Log in, and "spawn" a new machine with image "Weekly 2018_45" and size "large".
	Expected Result	The JupyterLab environment appears.
	Actual Result	• The test was carried out at the University of Washington, which has a Data Facility firewall exemption, so instructions at https://nb.lsst.io/getting-started/logging-in.html describing installing, configuring and connecting with a VPN client were skipped. • The "Weekly 2018_45" image was not listed — only the two most recent weeklies are called out explicitly. However, "w201845" was chosen as a close analog. • The JupyterLab environment appeared as expected.
	Status	Pass

2	Description	Create a terminal session. Use it to set up the LSST tools, then download and build version 5c12b06e6 of obs_lsst:
		
		$ source /opt/lsst/software/stack/loadLSST.bash
		$ setup lsst_distrib
		$ git clone https://github.com/lsst/obs_lsst.git
		$ cd obs_lsst
		$ git checkout 5c12b06e6
		$ setup -k -r .
		$ scons

	Expected Result	No errors are seen during execution of the provided commands.
	Actual Result	Commands were executed with no errors. Last line printed was "scons: done building targets".
	Status	Pass

3	Description	Ingest RTM-007 test data by executing the following commands: OUTPUT_REPO_DIR=$OUTPUT_DATA_DIR INPUT_DATA_DIR=$INPUT_DATA_DIR mkdir -p $OUTPUT_REPO_DIR echo “lsst.obs.lsst.ts8.Ts8Mapper” > $OUTPUT_REPO_DIR/_mapper ingestImages.py $OUTPUT_REPO_DIR $INPUT_DATA_DIR/*/*.fits constructBias.py $OUTPUT_REPO_DIR –rerun calibs –id imageType=BIAS –batch-type smp –cores 4 ingestCalibs.py $OUTPUT_REPO_DIR –calibType bias $OUTPUT_REPO_DIR/rerun/calibs/bias/*/*.fits –validity 9999 –output $OUTPUT_REPO_DIR/CALIB –mode=link Where: $OUTPUT_DATA_DIR is some location on shared storage to which the user has write permission; $INPUT_DATA_DIR is defined in the test case description.
	Expected Result	Many status messages are logged to screen, and the command exits with status 0.
	Actual Result	• A number of “WARN” messages were logged during constructBias.py. These are not important for the correct execution of the test, but should be suppressed or included in the test description. • All commands executed successfully.
	Status	Pass
4	Description	Demonstrate that raw and bias data for visit $VISIT_ID have been made available in the repository. Load a Python interpreter (run “python”) and execute the following: from lsst.daf.persistence import Butler visit_id = $VISIT_ID b = Butler($OUTPUT_DATA_DIR) b.get(“raw”, visit=visit_id, detector=2) b.get(“bias”, visit=visit_id, detector=2)

Expected Result	Each call to b.get() returns an instance of an ExposureF object. Warnings about lack of dark-time or WCS information may be ignored.

Actual Result	$ python
	Python 3.6.6 |Anaconda, Inc.| (default, Jun 28 2018, 17:14:51)
	[GCC 7.2.0] on linux
	Type “help”, “copyright”, “credits” or “license” for more information.
	>>> from lsst.daf.persistence import Butler
	>>> visit_id = 258334666
	>>> b = Butler(’/scratch/swinbank/ldm50307’)
	CameraMapper INFO: Loading exposure registry from /scratch/swinbank/ldm50307/registry.sqlite3
	CameraMapper INFO: Loading calib registry from /scratch/swinbank/ldm50307/CALIB/calibRegistry.sqlite3
	>>> b.get(“raw”, visit=visit_id, detector=2)
	CameraMapper WARN: Key=“DARKTIME” not in metadata
	CameraMapper INFO: darkTime is NaN/Inf; using exposureTime
	LsstCamMapper WARN: Unable to set WCS for DataId(initialdata={’visit’: 258334666, ’detector’: 2}, tag=set()) from header as RATEL/DECTEL/ROTANGLE are unavailable
	<lsst.afw.image.exposure.exposure.ExposureF object at 0x7ff07235ea78>
	>>> b.get(“bias”, visit=visit_id, detector=2)
	<lsst.afw.image.exposure.exposure.ExposureF object at 0x7ff09a8b88f0>
	>>>

Status	Pass