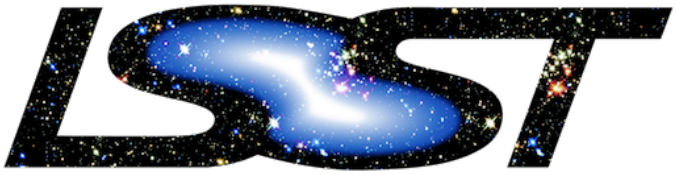

LARGE SYNOPTIC SURVEY TELESCOPE

Large Synoptic Survey Telescope (LSST)
Data Management

LDM-503-07 (Camera data processing) Test Plan and Report

John Swinbank

DMTR-112

Latest Revision: 2018-11-26

DRAFT

Abstract

This is the test plan and report for LDM-503-07 (Camera data processing), an LSST DM level 1 milestone pertaining to the Data Management.

Change Record

Version	Date	Description	Owner name
	2018-11-26	First Draft	Swinbank, Comoretto

Document curator: John Swinbank
Document source location: `https://github.com/lsst-dm/DMTR-112`
Version from source repository: 7984567

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.

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

1.4 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

1.5 Document Overview and Procedure

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

The following general sections are completed before the start of the test activity.

Section 1 provides an overview of the test campaing, the system under test (Data Management), the documentation, and explains how this document is organized. Section 2 describes the configuration used for this test. Section 3 lists all people and roles involved. Section 4 provides the list of planned test cycles and test cases, including all relevant informatino that fully describe the the test campaign. The content provided by the above sections shall be sufficient to prove that that the test campaign is ready to start.

Once the above sections are completed, this document can be reviewed by the product leader, the involved opersonel (section 3 and by who requested the test campaign. If everybody agree that the test is ready to start, the Jira Test Plan shall be set to **Approved** by the Data Management product leader. A first issue of this document can be uploaded in docushare for record.

Section 5 is filled after the test activity is completed and the Jira Test Cycles involved have been set to **Done**. The first subsection 5.1 provides a summary view of the results, in table 3, an overall assessment statement and suggestions on possible improvements. The subsection 5.2 provides detailed results for each step in each test case.

When completed, this document has to be approved by who requested the test activity and the final issued uploaded in docushate. The status of the Jira Test Plan shall then be set to **Completed**.

The actual status of the Jira test plan LVV-P16 is Draft.

2 Test Configuration

Observing is not required for this test campaign.

2.1 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

Following personnel is involved in the test activity:

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

4 Planned Test Activities

4.1 Test Cycle LVV-C19

LDM-503-07: Camera Data Processing
Status: Not Executed

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.

4.1.1 LVV-T374

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").

Step	Description
1	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".

2	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`

Step	Description
3	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.
4	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)

4.1.2 LVV-T368

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.

Step	Description
1	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".

2	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.

Step	Description

3	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.

4	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)`

5	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.

6	Display the raw image data in the Firefly window: `afw_display.mtv(raw)`

Step	Description

7	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)`

8	Display the corrected image data in the Firefly window: afw_display.mtv(result.exposure)

5 Test Results

5.1 Overview of the Test Results

5.1.1 Summary Table

Test Cycle **LVV-C19: LDM-503-07: Camera Data Processing**		
test case id	**status**	**comment**
LVV-T374	Not Executed	
LVV-T368	Not Executed	

Table 3: Test Results Summary Table

5.1.2 Overall Assessment

5.1.3 Recommended Improvements

5.2 Detailed Test Results