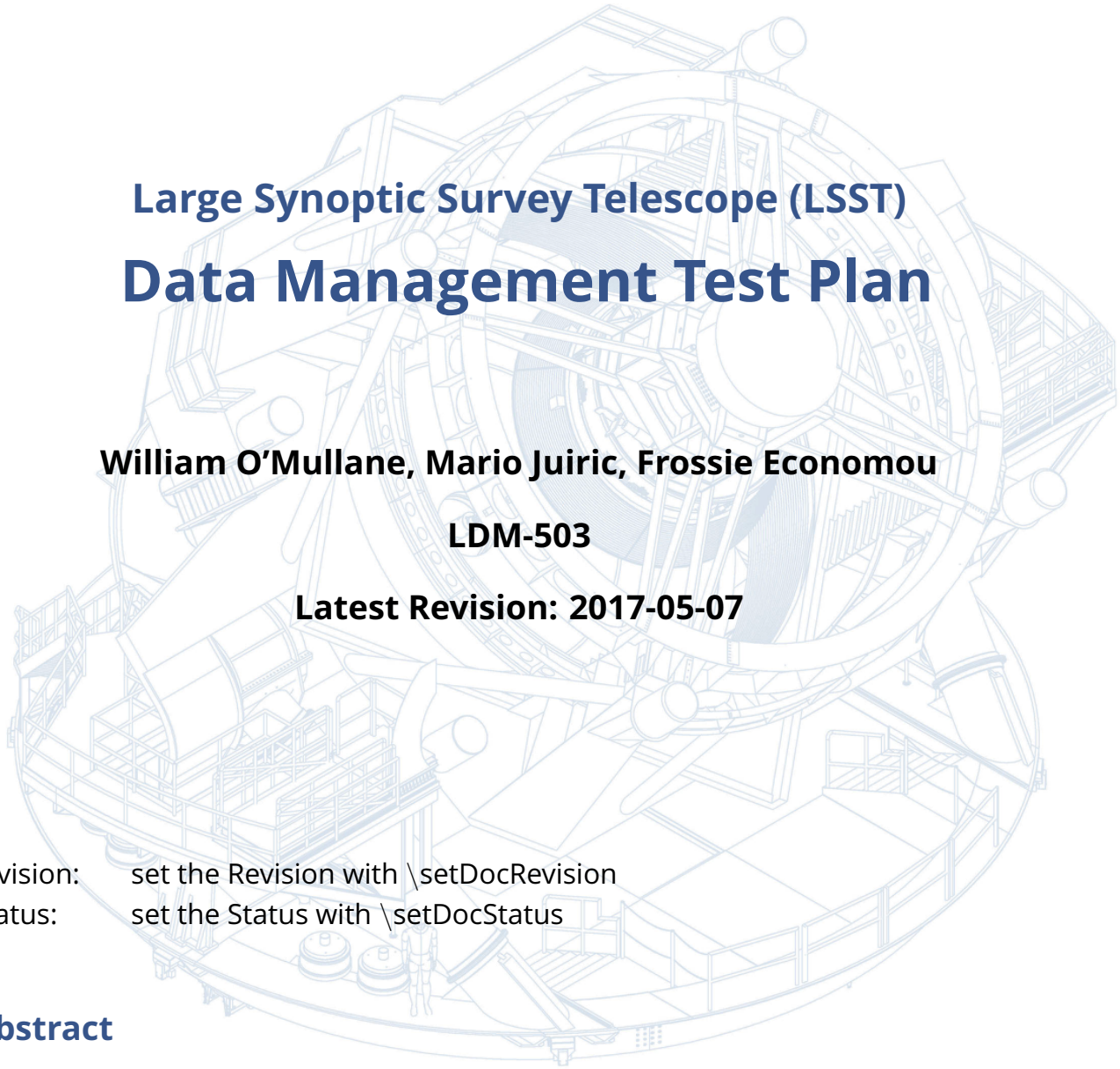
LARGE SYNOPTIC SURVEY TELESCOPE



Large Synoptic Survey Telescope (LSST)

Data Management Test Plan

William O'Mullane, Mario Juiric, Frossie Economou

LDM-503

Latest Revision: 2017-05-07

revision: set the Revision with \setDocRevision
status: set the Status with \setDocStatus

Abstract

This is the Test Plan for Data Management. In it we define terms associated with testing and further test specifications for specific items.

Change Record

Version	Date	Description	Owner name
D	2017-01-13	First draft	William O'Mullane

Contents

1 Introduction

In this document we outline the verification and validation approach for LSST Data Management. In addition we outline some of the high level test milestones.

1.1 Objectives

We describes the test and verification approach for DM and describe constraints and limitations in the testing to be performed. We also describe the validation tests to be performed on the partially and fully integrated system. We do not describe all tests in details but leave that to dedicated test plans.

1.2 Scope

This provides the approach and plan for all od DM. It covers interfaces to DM but nothing outside of DM. This document will be updated in response to any requirements updates.

1.3 Assumptions

We will run large scale Science Validations. A large amount of informal science validation will be done in the the teams and documented in technical notes, in this test plan we are looking for broad validation and specifically *operaability* i.e. can we run this system everyday for a long period of time (years).

1.4 Applicable Documents

When applicable documents change a change may be required in this document.

LPM-55	LSST Quality Assurance Plan
LDM-294	DM Project Management Plan
LDM-148	DM Architecture

1.5 References

[1] **[LDM-148]**, Kantor, J., Axelrod, T., 2013, *Data Management System Design*, LDM-148, URL https://ls.st/LDM-148

[2] **[LDM-294]**, O'Mullane, W., DMLT, 2017, *Data Management Project Management Plan*, LDM-294, URL https://ls.st/LDM-294

[3] **[LPM-55]**, Sweeney, D., McKercher, R., 2013, *Project Quality Assurance Plan*, LPM-55, URL https://ls.st/LPM-55

1.6 Definitions, acronyms, and abbreviations

The following table has been generated from the on-line Gaia acronym list:

Acronym	Description
API	Application Programming Interface
CAM	CAMera
CU	Coordination Unit (in DPAC)
DM	Data Management
DMLT	DM Leadership Team
DPAC	Data Processing and Analysis Consortium
DPC	Data Processing Centre
ESA	European Space Agency
HSC	Hyper Supreme-Cam
LSST	Large-aperture Synoptic Survey Telescope
NCSA	National Center for Supercomputing Applications
OPS	OPerationS
SP	Software Product
SPR	Software Problem Report
SUIT	Science User Interface and Tools
SVN	SubVersioN
TBD	To Be Defined (Determined)
WISE	Wide-field Survey Explorer

2 Test Items

The test items covered in this test plan are Data Management and its constituent components:

- All the product - from KT diagrams
- Interfaces
- Procedures like Data release

3 Roles and Reporting

Tester report issues through Jira, but what other mechanisms will be used?

WHo directs OPS rehersals .. ?

Reports on rehersals .. issues and

Handeling failures - time ines for fix.

Not my section but a thought: System Engineering intends to capture commissioning tests through JIRA testing addons/plugins such as Kanoah. We could use some of these to capture functional tests for repeatability between operation rehearsals (they might be overkill for most tests though). –FE

Note that downstream text refers to "Software Review Board". We don't have such an entity so we need to either identify an existing entity or define the constitution of that Board

The Software Review Board will meet once a full run of all Test Cases has been performed, and subsequently after a complete run of all outstanding Test Cases. *We don't actually have a software review board, not sure what the equivalent would be? is it an in-system, in-project or independent body?*

3.1 Pass/Fail Criteria

Pass/Fail criteria vary slighty depending on the type of test being performed.

A Test Case will be considered "Passed" when:

- All of the test steps of the Test Case are completed and
- All open SPRs from this Test Case agreed in Software Review Board are considered non-critical.

A Test Case will be considered "Partially Passed" when:

- Only a subset of all of the test steps in the Test Case are completed but the overall purpose of the test has been met and
- Any critical SPRs from this Test Case agreed in Software Review Board are still not closed.

A Test Case will be considered "Failed" when:

- Only a subset of all of the test steps in the Test Case are completed and the overall purpose of the test has not been met and
- Any critical SPRs from this Test Case agreed in Software Review Board are still not closed.

3.1.1 Key Performance Metrics

Note: Given the incomplete 1:1 match between Key Performance Metrics listed in LDM-240 (which was a spreadsheet) and LSE-30 (aka OSS), we could theoretically have a situation where we pass our KPMs but fail an OSS metric. I think this is unlikely, but we would need to complete the OSS flowdown to be able to demonstate that to a skeptic. If we re-flowdown and come up with KPM 2.0s though, this section would stand as written. We just have to surrender the pedantic point that they wouldn't be "Key" at that point, they would be ***the*** *performance metrics.*

Key Performance Metrics

4 Constraints and Limitations

Wil: Describes the limitations and the constraints which apply to CU level tests of the system. lack of computing resources may mean that datasets are smaller or that full accuracy cannot be achieved.

Explain what must be validated in the DPC tests

- Verification is being done on the basis of precursor data sets such as HSC, and eventually with engineering data from the LSST arrays. These are just a proxy for full-focal-plane on-site LSST data.
- Metric measurements and operational rehearsals during construction may not involve critical operational systems that are still in development. For example, while computational performance is being measured, computationally dominant algorithmic steps such as deblending and multi-fit are only modeled, since they have not yet been implemented; operational rehearsals are done without the factory LSST workflow system; etc.

4.1 Requirements Traceability Constraints

I felt a summary of the current state of play being verified could be useful to Wil. We don't have to leave it in the final document –FE

4.1.1 Scientific

Some science requirements are captured in LSE-29 (aka LSR) and flow down to LSE-30 (aka OSS) ; some also exist in LSE-163 (aka DPDD) and will flow down in LSE-61 (aka DMSR) *Flow-down is not complete, TJ is working on this*

4.1.2 Computational

There are requirements in LSE-61 (aka DMSR) which captures the LSE-30 (OSS) requirements that DM is responsible for. *In practice LSE-63 (the QA document) has not been flown down to LSE-61*. These are:

- The primary computational performance flown down from LSE-29 (LSR) is OTT1 which is the requirement to issue an alert within 60 seconds of exposure end.
- Another requirement flows down from LSE-29 is calculation of orbits within 24 hours of the end of the observing night

- There is a new (not yet baselined?) requirement for the calibration pipeline to reduce calibration observations within 1200 seconds
- A nightly report on data quality, data management system performance and a calibration report have to be generated with 4 hours of the end of the night

Note that there are no computational requirements on individual technical components eg. data processing cluster availability, database data retrieval speeds, etc. There is an upper limit on acceptable data loss, and there is a network availability requirement.

4.1.3 KPMs

As a proxy for validating the DM system, LDM-240 (aka "the spreadsheet") defined a set of Key Performance Metrics that the system could be verified against. KPMs were not formally flowed down from LSE-29 (LSR) through LSE-30 (OSS) although there is some overlap with LSE-29 requirements. [TJ is working on this]. In particular, the non-science KPMs only exist in LDM-240 *(spreadsheet/old plan)*.

[While verification was part of the SQuaRE WBS we prepared a KPM verification plan at the request of System Engineering - LDM-502. This work is now being led by Wil now I guess?]

4.2 Functional Requirements

Functional requirement are not explicitly called out as such. They are captured in LSE-61 (DMSR). *[When SQuaRE prepared the verification plan for SysEng, we were directed not to include functional requirements and limit ourselves to KPM. In general functional requirements are easy to verify by simply undertaking to perform the required functions in eg. operational rehearsals so maybe we could just say that?]*

4.3 Interfaces

There is an implicit, but not explicit, need to verify interfaces to other subsystems. The ICDs describing external interfaces are curated in Docushare Collection 5201. *[Integration used to be a Tucson role; I believe this is being led by the currently vacant Integration Scientist role? or whoever conducts the operation rehearsals?]*

Internal interfaces: currently we have no definitions and hence they are not verifiable presumably. If we did, I would propose: I

5 Master Schedule

The schedule for testing the system until operations commence (currently 2022).

Date/Freq	Location	Title, Description
Nightly	Amazon	**Nightly Tests** Run all automated tests on all DM packages automatically.
Weekly	Amazon	**Integration tests** Basic Sanity check to make sure code compiles at no regressions have occurred and also pushing though a basic data set.
TBP	NCSA	Interface tests The interface tests have to be planned and documented in a separate test plan that should include tests for each two parties on an interface (2by2 tests) as well as tests for all parties. Some of these will be covered again in E2E tests but before that we should be confident they work. **This includes internal and external interfaces.**
TBP	NCSA + IN2P3	End to End Tests ?? Freeze software for Ops .. `https://confluence.lsstcorp.org/display/DM/Data+Processing+End+to+End+Testing` What is the status of these ?
F17	NCSA	**Science Platform with WISE data in PDAC** SUIT continues PDAC development, adding the WISE data, further exercising the DAX dbserv and imgserv APIs, and taking advantage of metaserv once it becomes available
F17	NCSA	**HSC reprocessing** Validate the data products withe LSST stack match or improve the HSC products - thus validating the stack. Validate the ops platform in NCSA. Validate some procedures like installing the stack, patches, starting, stopping production. Generate validation data set for weekly integration and other tests.
S18?	NCSA	**ZTF Alerts processing** Validate L1 pipe ..

2018	NCSA	**Spectrograph Data acquisition** ...Do we need a test BEFORE THIS?
2018	NCSA	**Operations rehearsal for commissioning** With TBD weeks commissioning (lets say a week) - pick which parts of plan we could rehearse.
2019	NCSA	**Operations rehearsal #2 for commissioning** More complete rehearsal - where do the scientist look at quality data? How do they feed it back to the Telescope ? How do we create/update calibrations ? Exercises some of the control loops.
2020	NCSA	**Operations Rehearsal Data Release (Commissioning Data)**
2021	NCSA	**Operations Rehearsal Data Release (Regular Data).**

6 Validation Tools

6.1 Introduction

To evaluate the correctness of the generated data and the systems performances a set of tools may be developed or used. These tools will provide the means to facilitate the validation tasks. Following subsections describe the various tools that can used in the Data Managementvalidation (e.g. data comparison tools, analysis tools, etc).

6.2 Data Comparison Tools

This type of test tools are used to manage products in terms of:

- Comparison of a product generated during a test execution w.r.t. the relevant reference product
- Non regression verification comparing output products generated by different versions of the same system
- Measurement of quality degradation due to perturbed inputs

It allows:

- Product analysis
- Decoding of generated product allowing to read the most significant data of the product itself
- Visualisation of the values of a single selected field
- Apply an accuracy to the comparison
- Comparing specific parts of the products
- Filtering using flags values

6.3 Data Transformation Tools

These tools allow the data to be transformed to other formatted data.

6.4 Analysis Tools

Descriptions of the performance monitoring tools, profilers, test coverage programs... used in the Performance evaluation tests.
...

7 Unit and Integration Tests

7.1 Approach

Unit and Integration Tests will be automatically executed through the JUnit test framework. The descriptions of the test below are extracted from the test cases code and documentation.The results of Unit and Integration Test to be included in the Sofwtare Test Report will be generated automatically from the output of the execution of the tests by JUnit. A script will be provided to perform thes processing steps.

Module identification? (module tag in class header? mapping file?)

7.2 Test Coverage

Test coverage goal for unit and integration testing. Each class and public method shall have a JUnit test harness that may be labelled according to their purpose (e.g. I/O, individual class/method tests, software integration, data model integration etc.). Nominal and contingency tests should be clearly identified.

Interface coverage...

The tool [insert name of unit test coverage tool here] will be used to provide metrics on the code coverage by Unit Tests for Data Management and this metric will be provide in the Test Report.

7.3 Unit and Integration Test Specification

This is a example test plan record; this should be generated automatically.

Class	Unit Test Name	Purpose
Unit Test Class	Unit Test Method	Purpose of Unit Test from method header

8 Validation Tests

Validation of the system though Oerations Rehersals (and or end to end tests)

8.1 General strategy

Description of the general verification and validation strategy, decomposition into verification testing categories (e.g. science tests, SP external interface tests, algorithms interrelation and sequence). Assessed validation tests results shall be available over the software development duration: they are stored into SVN repository along with related input data, property-file, etc.

A subset of tests are run at DPC during software release qualification process, the results of DPC runs are compared with corresponding test outputs. During DPC integration tests, these assessed outputs will also allow to verify software non-regression.

8.2 Test Designs

8.2.1 Test Design DM-Data Management-SYS-X

8.2.1.1 Objective Explain the objective of this test design

8.2.1.2 Features to be tested

- Component A
- Component B

8.2.1.3 Features not to be tested

- Component C
- Component D

8.2.1.4 Approach Description of the approach to writing this test design

8.2.1.5 Test Cases List of test cases to be specified

Test Case	Description
DM-Data Management-SYS-X-1	Description of Validation Test

8.3 Test Case Specification

8.3.1 Test Case DM-Data Management-SYS-X-1

8.3.1.1 Testable Items List the components to be tested in this test case

8.3.1.2 Purpose Explain the purpose of this test case

8.3.1.3 Input Specification Describe the inputs to this test (data, written procedures, etc.)

8.3.1.4 Output Specification Describe the outputs of this test

8.3.1.5 Environment Describe the environment (computing resources etc) required for this test.

8.3.1.6 Inter-case dependencies If this test in dependent on another test having been completed successfully (for input data for example), state that here.

8.3.1.7 Test Procedure Describe the procedure to be performed

8.3.1.8 Test Verification Describe how to verify if the test has been successful.

9 Science Validation