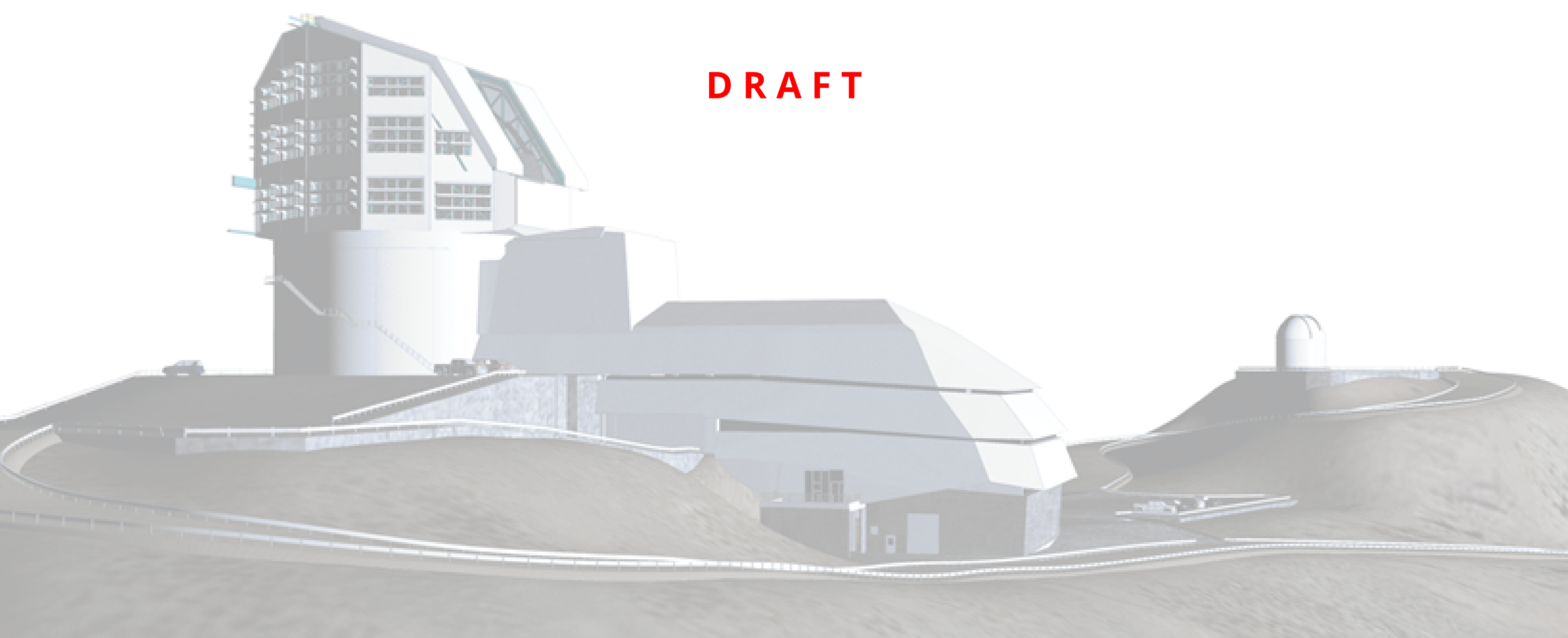
Vera C. Rubin Observatory
Data Management

LVV-P106: Data Management Acceptance Test Campaign, Fall 2023 Test Plan

Jeffrey Carlin

DMTR-401

Latest Revision: 2023-08-04



DRAFT

Abstract

This is the test plan for **Data Management Acceptance Test Campaign, Fall 2023**, an LSST milestone pertaining to the Data Management Subsystem.
This document is based on content automatically extracted from the Jira test database on 2023-08-04 . The most recent change to the document repository was on 2023-08-04.

Change Record

Version	Date	Description	Owner name
	2023-07-01	First draft	Jeffrey Carlin

Document curator: Jeffrey Carlin
Document source location: `https://github.com/lsst-dm/DMTR-401`
Version from source repository: 0640251

Contents

LVV-P106: Data Management Acceptance Test Campaign, Fall 2023 Test Plan

1 Introduction

1.1 Objectives

The primary goal of this DM acceptance test campaign will be to verify priority 1a DMSR (LSE-61) requirements that have not been verified as part of prior testing and milestones. Any priority 1b, 2, or 3 requirements that have been completed will also be verified.

1.2 System Overview

This test campaign is intended to verify that the DM system satisfies at least half of the priority 1a requirements outlined in the Data Management System Requirements (DMSR; LSE-61), ensuring that we are progressing toward readiness for the installation and operation of LSSTCam. Additional DMSR requirements will be verified in later Acceptance Test Campaigns.

Applicable Documents:
LSE-61: Data Management System (DMS) Requirements
LDM-503 Data Management Test Plan
LDM-639: Data Management Acceptance Test Specification

Tests in this campaign will use data products and artifacts from Data Preview 0.2, which consists of DESC Data Challenge 2 (DC2) simulated data reprocessed using the LSST Science Pipelines. Additional on-sky data from auxTel imaging campaigns, and camera test-stand data, will be used when appropriate.

1.3 Document Overview

This document was generated from Jira, obtaining the relevant information from the LVV-P106 Jira Test Plan and related Test Cycles (LVV-C260).

Section 1 provides an overview of the test campaign, the system under test (Acceptance), the applicable documentation, and explains how this document is organized. Section 2 provides additional information about the test plan, like for example the configuration used for this test or related documentation. Section 3 describes the necessary roles and lists the individuals assigned to them.

Section 4 provides a summary of the test results, including an overview in Table 2, an overall assessment statement and suggestions for possible improvements. Section **??** provides detailed results for each step in each test case.

The current status of test plan LVV-P106 in Jira is **Draft** .

1.4 References

[1] **[DMTN-140]**, Comoretto, G., 2021, *Documentation Automation for the Verification and Validation of Rubin Observatory Software*, DMTN-140, URL `https://dmtn-140.lsst.io/`, Vera C. Rubin Observatory Data Management Technical Note

[2] **[DMTN-178]**, Comoretto, G., 2021, *Docsteady Usecases for Rubin Observatory Constructions*, DMTN-178, URL `https://dmtn-178.lsst.io/`, Vera C. Rubin Observatory Data Management Technical Note

[3] **[LSE-61]**, Dubois-Felsmann, G., Jenness, T., 2019, *Data Management System (DMS) Requirements*, LSE-61, URL `https://lse-61.lsst.io/`, Vera C. Rubin Observatory

[4] **[LDM-639]**, Guy, L., Wood-Vasey, W., Bellm, E., et al., 2022, *LSST Data Management Acceptance Test Specification*, LDM-639, URL `https://ldm-639.lsst.io/`, Vera C. Rubin Observatory Data Management Controlled Document

[5] **[LDM-503]**, O'Mullane, W., Swinbank, J., Juric, M., et al., 2022, *Data Management Test Plan*, LDM-503, URL `https://ldm-503.lsst.io/`, Vera C. Rubin Observatory Data Management Controlled Document

[6] **[LSE-160]**, Selvy, B., 2013, *Verification and Validation Process*, LSE-160, URL `https://ls.st/LSE-160`

2 Test Plan Details

2.1 Data Collection

Observing is not required for this test campaign.

2.2 Verification Environment

Most testing will be performed using the Rubin Science Platform (RSP) and the development cluster at the USDF. In particular, we will use version 26 of the Pipelines for most tests; some tests will use more recent weekly builds of the Pipelines.

2.3 Related Documentation

2.4 PMCS Activity

Primavera milestones related to the test campaign:

- None

3 Personnel

The personnel involved in the test campaign is shown in the following table.

	T. Plan LVV-P106 owner:	**Jeffrey Carlin**	
	T. Cycle LVV-C260 owner:	**Jeffrey Carlin**	
Test Cases	**Assigned to**	**Executed by**	**Additional Test Personnel**
LVV-T191	Robert Gruendl [X]		
LVV-T1986	Leanne Guy		
LVV-T159	Simon Krughoff		
LVV-T132	Robert Gruendl [X]		
LVV-T62	Jim Bosch		
LVV-T168	Robert Gruendl [X]		
LVV-T41	Jim Bosch		
LVV-T97	Kian-Tat Lim		
LVV-T183	Gregory Dubois-Felsmann		
LVV-T2177	Leanne Guy		
LVV-T1755	Jeffrey Carlin		
LVV-T2176	Leanne Guy		
LVV-T1754	Jeffrey Carlin		
LVV-T376	Leanne Guy		
LVV-T1946	Jeffrey Carlin		
LVV-T1947	Jeffrey Carlin		
LVV-T28	Colin Slater		
LVV-T124	Jeffrey Carlin		
LVV-T142	Leanne Guy		
LVV-T1748	Jeffrey Carlin		
LVV-T1759	Jeffrey Carlin		
LVV-T1758	Jeffrey Carlin		
LVV-T149	Leanne Guy		
LVV-T40	Jim Bosch		
LVV-T129	Jeffrey Carlin		
LVV-T115	Kian-Tat Lim		
LVV-T1862	Jeffrey Carlin		
LVV-T89	Robert Lupton		
LVV-T88	Robert Lupton		
LVV-T85	Robert Lupton		
LVV-T83	Robert Lupton		

4 Test Campaign Overview

4.1 Summary

T. Plan LVV-P106:	**Data Management Acceptance Test Campaign, Fall 2023**	Draft
T. Cycle LVV-C260:	**Data Management Acceptance Test Campaign, Fall 2023**	Not Executed

Test Cases	Ver.
LVV-T191	1
LVV-T1986	1
LVV-T159	1
LVV-T132	1
LVV-T62	2
LVV-T168	1
LVV-T41	1
LVV-T97	1
LVV-T183	1
LVV-T2177	1
LVV-T1755	1
LVV-T2176	1
LVV-T1754	1
LVV-T376	1
LVV-T1946	1
LVV-T1947	1
LVV-T28	1
LVV-T124	1
LVV-T142	1
LVV-T1748	1
LVV-T1759	1
LVV-T1758	1
LVV-T149	1
LVV-T40	1
LVV-T129	1
LVV-T115	1
LVV-T1862	1
LVV-T89	1
LVV-T88	1
LVV-T85	1
LVV-T83	1

Table 2: Test Campaign Summary

4.2 Overall Assessment

Not yet available.

4.3 Recommended Improvements

5 Detailed Tests

5.1 Test Cycle LVV-C260

Open test cycle *Data Management Acceptance Test Campaign, Fall 2023* in Jira.

Test Cycle name: Data Management Acceptance Test Campaign, Fall 2023
Status: Not Executed

This test cycle verifies a subset of DMSR (LSE-61) requirements in order to verify their completion and readiness for LSST Operations (i.e., that the requirements laid out in LSE-61 have been met by the DM Systems). Testing will use data products and artifacts from Data Preview 0.2 reprocessing of DESC DC2 data, Auxtel data, and other data products housed at the U.S. Data Facility (USDF).

5.1.1 Software Version/Baseline

Primarily using Science Pipelines version 26 at the USDF.

5.1.2 Configuration

Not provided.

5.1.3 Test Cases in LVV-C260 Test Cycle

5.1.3.1 LVV-T191 - Verify implementation of Commissioning Cluster

Version **1**. Open *LVV-T191* test case in Jira.

Verify that the Commissioning Cluster has sufficient Compute/Storage/LAN at the Base Facility to support Commissioning.

Preconditions:

Final comment:

Detailed steps :

Step 1	Description

Analyze design and budget

Expected Result

5.1.3.2 LVV-T1986 - Mini DC2 processing capability

Version **1**. Open *LVV-T1986* test case in Jira.

Demonstrate that a typical 3-tract DC2 data processing is possible using the Gen3 system and the nascent Batch Production Service (BPS). This test is meant to extend LVV-T1983 (Mini RC2 processing capability) by demonstrating Gen3 + BPS systems are capable of supporting future Data Previews (which have been specified to use the DC2 image sim data rather than HSC data).

Preconditions:

Final comment:

Detailed steps :

Step 1	Description

Expected Result

5.1.3.3 LVV-T159 - Verify implementation of Regenerating Data Products from Previous Data Releases

Version **1**. Open *LVV-T159* test case in Jira.

Show that un-archived data products from previous data releases can be generated using through the LSST Science Platform.

Preconditions:

Final comment:

Detailed steps :

Step 1	Description
Delegate to LSP	

Expected Result

5.1.3.4 LVV-T132 - Verify implementation of Pre-cursor and Real Data

Version **1**. Open *LVV-T132* test case in Jira.

Demonstrate that pixel-oriented data from astronomical imaging cameras (precursor or otherwise) can be processed using LSST Science Algorithms and organized for access through the Data Butler Access Client.

Preconditions:

Final comment:

Detailed steps :

Step 1	Description

Confirm that the CI jobs used to test DRP processing successfully run. These jobs use precursor datasets from cameras other than LSST.

Expected Result

Step 2	Description

For the precursor dataset, instantiate the Butler, load the data products, and confirm that they exist as expected.

Expected Result
Processed images, catalogs, calibration information, and other related data products are present and accessible via the Butler.

5.1.3.5 LVV-T62 - Verify implementation of Provide PSF for Coadded Images

Version **2**. Open *LVV-T62* test case in Jira.

Verify that all coadd images produced by the DRP pipelines include a model from which an image of the PSF at any point on the coadd can be obtained.

Preconditions:
Fully covered by preconditions for LVV-T16.

Final comment:

Detailed steps :

Step 1 Description

Identify a dataset with coadded images in multiple filters.

Expected Result

Multi-band data that has been processed through the coaddition stage.

Step 2 Description

Identify the path to the data repository, which we will refer to as 'DATA/path', then execute the following:

Example Code

```
from lsst.daf.butler import Butler
repo = 'Data/path'
collection = 'collection'
butler = Butler(repo, collections=collection)
```

Expected Result

Butler repo available for reading.

Step 3 Description

Load the exposures, then select Objects classified as point sources on at least 10 different coadd images (including all bands). Evaluate the PSF model at the positions of these Objects, and verify that subtracting a scaled version of the PSF model from the processed visit image yields residuals consistent with pure noise.

Expected Result

Images with the PSF model subtracted, leaving only residuals that are consistent with being noise.

5.1.3.6 LVV-T168 - Verify design of Data Access Services allows Evolution of the LSST Data Model

Version **1**. Open *LVV-T168* test case in Jira.

Verify that the design of the Data Access Services are able to accommodate changes/evolution of the LSST data model from one release to another.

Preconditions:

Final comment:

Detailed steps :

Step 1	Description
Delegate to LSP	

Expected Result

5.1.3.7 LVV-T41 - Verify implementation of Generate PSF for Visit Images

Version **1**. Open *LVV-T41* test case in Jira.

Verify that Processed Visit Images produced by the DRP and AP pipelines are associated with a model from which one can obtain an image of the PSF given a point on the image.

Preconditions:

Final comment:

Detailed steps :

Step 1 Description

Identify a dataset with processed visit images in multiple filters.

Expected Result

Step 2 Description

Identify the path to the data repository, which we will refer to as 'DATA/path', then execute the following:

Example Code

```
from lsst.daf.butler import Butler
repo = 'Data/path'
collection = 'collection'
butler = Butler(repo, collections=collection)
```

Expected Result

Butler repo available for reading.

Step 3 Description

Select Objects classified as point sources on at least 10 different processed visit images (including all bands). Evaluate the PSF model at the positions of these Objects, and verify that subtracting a scaled version of the PSF model from the processed visit image yields residuals consistent with pure noise.

Expected Result

Images with the PSF model subtracted, leaving only residuals that are consistent with being noise.

5.1.3.8 LVV-T97 - Verify implementation of Uniqueness of IDs Across Data Releases

Version **1**. Open *LVV-T97* test case in Jira.

Verify that the IDs of Objects, Sources, DIAObjects, and DIASources from different Data Releases are unique.

Preconditions:

Final comment:

Detailed steps :

Step 1	Description

Identify an appropriate precursor dataset to be processed through Data Release Production.

Expected Result

Step 2	Description

Process data with the Data Release Production payload, starting from raw science images and generating science data products, placing them in the Data Backbone.

Expected Result

Step 3	Description

Identify the path to the data repository, which we will refer to as 'DATA/path', then execute the following:

Example Code

```
from lsst.daf.butler import Butler
repo = 'Data/path'
collection = 'collection'
butler = Butler(repo, collections=collection)
```

Expected Result

Butler repo available for reading.

Step 4 Description

After running the DRP payload multiple times, load the resulting data products (both data release and prompt products) using the Butler.

Expected Result

Multiple datasets resulting from processing of the same input data.

Step 5 Description

Inspect the IDs in the multiple data products and confirm that all IDs are unique.

Expected Result

No IDs are repeated between multiple processings of the identical input dataset.

5.1.3.9 LVV-T183 - Verify implementation of DMS Communication with OCS

Version **1**. Open *LVV-T183* test case in Jira.

Verify that the DMS at the Base Facility can receive commands from the OCS and send command responses, events, and telemetry back. Verified by Early Integration activities and during AuxTel commissioning.

Preconditions:

Final comment:

Detailed steps :

Step 1 Description

From the Base Site, connect to the (simulated) OCS telemetry stream.

Expected Result

Step 2	Description

Send a command to the OCS, and observe that the command has been executed.

Expected Result
Confirmation that the OCS command successfully executed.

Step 3	Description

Extract information from the telemetry being broadcast by the OCS, and ensure that these data are readable.

Expected Result
A readable extract from the OCS telemetry stream.

5.1.3.10 LVV-T2177 - Per-image limit on the median residual ellipticity correlations at scales less than to 5 arcmin.

Version **1**. Open *LVV-T2177* test case in Jira.

Verify that the per-image limit on the median residual ellipticity correlations at scales less than 5 arcmin (TE3) can be configured in the DMS and applied to the appropriate metrics.

Preconditions:

Final comment:

Detailed steps :

Step 1	Description

Check that the correct value for the TE3 threshold has been encoded in the faro package.

Expected Result

5.1.3.11 LVV-T1755 - Verify calculation of residual PSF ellipticity correlations for separations less than 1 arcmin

Version **1**. Open *LVV-T1755* test case in Jira.

Verify that the DM system has provided the code to calculate the median residual PSF ellipticity correlations averaged over an arbitrary field of view for separations less than 1 arcmin, and assess whether it meets the requirement that it shall be no greater than **TE1 = 2.0e-5[arcminuteSeparationCorrelation].**

Preconditions:

Final comment:

Detailed steps :

Step 1	Description

Identify a dataset containing at least one field with multiple overlapping visits.

Expected Result
A dataset that has been ingested into a Butler repository.

Step 2	Description

The 'path' that you will use depends on where you are running the science pipelines. Options:

- local (newinstall.sh - based install):[path_to_installation]/loadLSST.bash
- development cluster ("lsst-dev"): /software/lsstsw/stack/loadLSST.bash
- LSP Notebook aspect (from a terminal): /opt/lsst/software/stack/loadLSST.bash

From the command line, execute the commands below in the example code:

Example Code

```
source 'path'
setup lsst_distrib
```

Expected Result

Science pipeline software is available for use. If additional packages are needed (for example, 'obs' packages such as 'obs_subaru'), then additional 'setup' commands will be necessary.

To check versions in use, type:
eups list -s

Step 3 Description

Execute 'faro' on a repository containing processed data. Identify the path to the data, which we will call 'DATA/-path', then execute something similar to the following (with paths, datasets, and flags replaced or additionally specified as needed):

Example Code

```
pipetask –long-log run -j 2 -b DATA/path/butler.yaml –register-dataset-types -p $FARO_DIR/pipelines/metrics_pipeline.yaml -d "band in ('g', 'r', 'i') AND tract=9813 AND skymap='hsc_rings_v1' AND instrument='HSC'" –output u/username/-faro_metrics -i HSC/runs/RC2/w_2021_06 2>&1 | tee w06_2021_tract9813_faro.txt
```

Expected Result

The output collection (in this case, "u/username/faro_metrics") containing metric measurements and any associated extras and metadata is available via the butler.

Step 4 Description

Confirm that the metric TE1 has been calculated, and that its values are reasonable.

Expected Result
A JSON file (and/or a report generated from that JSON file) demonstrating that TE1 has been calculated.

5.1.3.12 LVV-T2176 - Per-image limit on the median residual ellipticity correlations at scales greater than or equal to 5 arcmin.

Version **1**. Open *LVV-T2176* test case in Jira.

Verify that the per-image limit on the median residual ellipticity correlations at scales greater than or equal to 5 arcmin (TE4) can be configured in the DMS and applied to the appropriate metrics

Preconditions:

Final comment:

Detailed steps :

Step 1	Description

Check that the correct value for the TE4 threshold has been encoded in the faro package.

Expected Result

5.1.3.13 LVV-T1754 - Verify calculation of residual PSF ellipticity correlations for separations greater than or equal to 5 arcmin

Version **1**. Open *LVV-T1754* test case in Jira.

Verify that the DM system has provided the code to calculate the median residual PSF ellipticity correlations averaged over an arbitrary field of view for separations greater than or equal to 5 arcmin, and assess whether it meets the requirement that it shall be no greater than **TE2 = 1.0e-7[arcminuteSeparationCorrelation].**

Preconditions:

Final comment:

Detailed steps :

Step 1 Description

Identify a dataset containing at least one field with multiple overlapping visits.

Expected Result

A dataset that has been ingested into a Butler repository.

Step 2 Description

The 'path' that you will use depends on where you are running the science pipelines. Options:

- local (newinstall.sh - based install):[path_to_installation]/loadLSST.bash
- development cluster ("lsst-dev"): /software/lsstsw/stack/loadLSST.bash
- LSP Notebook aspect (from a terminal): /opt/lsst/software/stack/loadLSST.bash

From the command line, execute the commands below in the example code:

Example Code
source 'path'
setup lsst_distrib

Expected Result
Science pipeline software is available for use. If additional packages are needed (for example, 'obs' packages such as 'obs_subaru'), then additional 'setup' commands will be necessary.

To check versions in use, type:
eups list -s

Step 3 Description
Execute 'faro' on a repository containing processed data. Identify the path to the data, which we will call 'DATA/-path', then execute something similar to the following (with paths, datasets, and flags replaced or additionally specified as needed):

Example Code
pipetask –long-log run -j 2 -b DATA/path/butler.yaml –register-dataset-types -p $FARO_DIR/pipelines/metrics_pipeline.yaml -d "band in ('g', 'r', 'i') AND tract=9813 AND skymap='hsc_rings_v1' AND instrument='HSC"' –output u/username/-faro_metrics -i HSC/runs/RC2/w_2021_06 2>&1 | tee w06_2021_tract9813_faro.txt

Expected Result
The output collection (in this case, "u/username/faro_metrics") containing metric measurements and any associated extras and metadata is available via the butler.

Step 4 Description
Confirm that the metric TE2 has been calculated, and that its values are reasonable.

Expected Result
A JSON file (and/or a report generated from that JSON file) demonstrating that TE2 has been calculated.

5.1.3.14 LVV-T376 - Verify the Calculation of Ellipticity Residuals and Correlations

Version **1**. Open *LVV-T376* test case in Jira.

Verify that the DMS includes software to enable the calculation of the ellipticity residuals and correlation metrics defined in the OSS.

Preconditions:

Final comment:

Detailed steps :

Step 1 Description

Identify the path to the data repository, which we will refer to as 'DATA/path', then execute the following:

Example Code

```
from lsst.daf.butler import Butler
repo = 'Data/path'
collection = 'collection'
butler = Butler(repo, collections=collection)
```

Expected Result

Butler repo available for reading.

Step 2 Description

Point the butler to an appropriate (precursor or simulated) dataset containing data in all filters, that is sufficient for the purposes of measuring astrometric performance metrics.

Expected Result

Step 3 Description

Execute the LSST Stack package 'validate_drp' (or an alternate package that is relevant) on this dataset to perform the measurements of the metrics.

Expected Result

Measurements of validation metrics and the presence of QA plots resulting from the validation pipeline.

Step 4 Description

Compare measured ellipticity correlations to known (for simulated data) or measured (if using precursor data) values from input (precursor or simulated) data, and confirm that the output values for all of the ellipticity performance metrics are as expected.

Expected Result

Measured ellipticity metrics that are within reasonable values given the (known) input dataset.

5.1.3.15 LVV-T1946 - Verify implementation of measurements in catalogs from coadds

Version **1**. Open *LVV-T1946* test case in Jira.

Verify that source measurements in catalogs containing measurements from coadd images are in flux units.

Preconditions:

Final comment:

Detailed steps :

Step 1 Description

Identify the path to the data repository, which we will refer to as 'DATA/path', then execute the following:

Example Code

```
from lsst.daf.butler import Butler
repo = 'Data/path'
collection = 'collection'
butler = Butler(repo, collections=collection)
```

Expected Result
Butler repo available for reading.

Step 2 Description
Identify and read an appropriate processed precursor dataset containing coadds with the Butler.

Expected Result

Step 3 Description
Verify that the coadd catalog provides measurements in flux units.

Expected Result
Confirmation of measurements in catalogs encoded in flux units.

5.1.3.16 LVV-T1947 - Verify implementation of measurements in catalogs from difference images

Version **1**. Open *LVV-T1947* test case in Jira.

Verify that source measurements in catalogs containing measurements from difference images are in flux units.

Preconditions:

Final comment:

Detailed steps :

Step 1 Description
Identify the path to the data repository, which we will refer to as 'DATA/path', then execute the following:

Example Code

```
from lsst.daf.butler import Butler
repo = 'Data/path'
collection = 'collection'
butler = Butler(repo, collections=collection)
```

Expected Result
Butler repo available for reading.

Step 2 Description
Identify and read an appropriate processed precursor dataset containing difference images with the Butler.

Expected Result

Step 3 Description
Verify that the difference image source catalog provides measurements in flux units.

Expected Result
Confirmation of measurements in catalogs encoded in flux units.

5.1.3.17 LVV-T28 - Verify implementation of measurements in catalogs from PVIs

Version **1**. Open *LVV-T28* test case in Jira.

Verify that source measurements in catalogs containing measurements from processed visit images are in flux units.

Preconditions:

Final comment:

Detailed steps :

Step 1 Description

Identify the path to the data repository, which we will refer to as 'DATA/path', then execute the following:

Example Code

```
from lsst.daf.butler import Butler
repo = 'Data/path'
collection = 'collection'
butler = Butler(repo, collections=collection)
```

Expected Result

Butler repo available for reading.

Step 2 Description

Identify and read an appropriate processed precursor dataset containing coadds with the Butler.

Expected Result

Step 3 Description

Verify that the single-visit catalog provides measurements in flux units.

Expected Result

Confirmation of measurements in catalogs encoded in flux units.

5.1.3.18 LVV-T124 - Verify implementation of Software Architecture to Enable Community Re-Use

Version **1**. Open *LVV-T124* test case in Jira.

Show that the LSST software is capable of being executed in multiple contexts: single user instance, batch processing, continuous integration.
Also show that the algorithms can be reconfigured and, if desired, completely replaced at run time.

Preconditions:

Final comment:

Detailed steps :

Step 1	Description

The 'path' that you will use depends on where you are running the science pipelines. Options:

- local (newinstall.sh - based install):[path_to_installation]/loadLSST.bash
- development cluster ("lsst-dev"): /software/lsstsw/stack/loadLSST.bash
- LSP Notebook aspect (from a terminal): /opt/lsst/software/stack/loadLSST.bash

From the command line, execute the commands below in the example code:

Example Code

```
source 'path'
setup lsst_distrib
```

Expected Result

Science pipeline software is available for use. If additional packages are needed (for example, 'obs' packages such as 'obs_subaru'), then additional 'setup' commands will be necessary.

To check versions in use, type:
eups list -s

Step 2 Description

Using curated test datasets for multiple precursor instruments, verify and log that the prototype DRP pipelines execute successfully in three contexts:
1. The CI system
2. On a single user system: laptop, desktop, or notebook running in the Notebook aspect of the LSP.
3. Project workflow system.

Expected Result

Step 3 Description

Using a template testing notebook in the Notebook aspect of the LSP, verify and log the following:
1. Individual pipeline steps (tasks) are importable and executable on their own. this is not comprehensive, but demonstrative.
2. Individual pipeline steps may be overridden by configuration.
3. Users can implement a custom pipeline step and insert i into the processing flow via configuration.

Expected Result

Step 4 Description

Identify the path to the data repository, which we will refer to as 'DATA/path', then execute the following:

Example Code

```
from lsst.daf.butler import Butler
repo = 'Data/path'
collection = 'collection'
butler = Butler(repo, collections=collection)
```

Expected Result

Butler repo available for reading.

Step 5 Description

Read the resulting dataset using the Bulter, and confirm that it produced the desired data products.

Expected Result

Step 6 Description

Run subset of full DRP from previous step on an individual node. Was this organizationally easy? Did the performance scale appropriately?

Expected Result

Step 7 Description

Re-run aperture correction on subset. Verify that same results as DRP run are achieved.

Expected Result

Step 8 Description

Re-run photometric redshift estimation algorithm on subset coadd catalogs. Verify that same results are achieved as from full DRP.

Expected Result

5.1.3.19 LVV-T142 - Verify implementation of Production Fault Tolerance

Version **1**. Open *LVV-T142* test case in Jira.

Demonstrate production systems report faults in pipeline executions and that system is able to recover. Where recovery can mean the ability to provide production artifacts for examination, return production elements ready for subsequent use, and/or reset and repeat production attempts.

Preconditions:

Final comment:

Detailed steps :

Step 1	Description

Execute AP and DRP, simulate failures, observe correct processing

Expected Result

5.1.3.20 LVV-T1748 - Verify calculation of median error in absolute position for RA, Dec axes

Version **1**. Open *LVV-T1748* test case in Jira.

Verify that the DM system has provided the code to calculate the median error in absolute position for each axis, RA and DEC, and assess whether it meets the requirement that it shall be less than **AA1 = 50 milliarcseconds**.

Preconditions:

Final comment:

Detailed steps :

Step 1 Description

Identify a dataset containing at least one field with multiple overlapping visits.

Expected Result

A dataset that has been ingested into a Butler repository.

Step 2 Description

The 'path' that you will use depends on where you are running the science pipelines. Options:

- local (newinstall.sh - based install):[path_to_installation]/loadLSST.bash
- development cluster ("lsst-dev"): /software/lsstsw/stack/loadLSST.bash
- LSP Notebook aspect (from a terminal): /opt/lsst/software/stack/loadLSST.bash

From the command line, execute the commands below in the example code:

Example Code

```
source 'path'
setup lsst_distrib
```

Expected Result

Science pipeline software is available for use. If additional packages are needed (for example, 'obs' packages such as 'obs_subaru'), then additional 'setup' commands will be necessary.

To check versions in use, type:
eups list -s

Step 3 Description

Execute 'faro' on a repository containing processed data. Identify the path to the data, which we will call 'DATA/-path', then execute something similar to the following (with paths, datasets, and flags replaced or additionally specified as needed):

Example Code

pipetask –long-log run -j 2 -b DATA/path/butler.yaml –register-dataset-types -p $FARO_DIR/pipelines/metrics_pipeline.yaml -d "band in ('g', 'r', 'i') AND tract=9813 AND skymap='hsc_rings_v1' AND instrument='HSC'" –output u/username/faro_metrics -i HSC/runs/RC2/w_2021_06 2>&1 | tee w06_2021_tract9813_faro.txt

---

Expected Result

The output collection (in this case, "u/username/faro_metrics") containing metric measurements and any associated extras and metadata is available via the butler.

Step 4	Description

Confirm that the metric AA1 has been calculated, and that its values are reasonable.

Expected Result

A JSON file (and/or a report generated from that JSON file) demonstrating that AA1 has been calculated.

5.1.3.21 LVV-T1759 - Verify that the repeatability outlier limit for isolated bright non-saturated point sources in the g, r, and i filters (PA2gri) can be applied.

Version **1**. Open *LVV-T1759* test case in Jira.

Verify that the DM system has provided the code to apply the repeatability outlier limit for isolated bright non-saturated point sources in the g, r, and i filters(PA2gri) to to computed values of the PF1 metric.

Preconditions:

Final comment:

Detailed steps :

Step 1 Description

Identify a dataset containing at least one field in each of the g, r, and i filters with multiple overlapping visits.

Expected Result

A dataset that has been ingested into a Butler repository.

Step 2 Description

The 'path' that you will use depends on where you are running the science pipelines. Options:

- local (newinstall.sh - based install):[path_to_installation]/loadLSST.bash
- development cluster ("lsst-dev"): /software/lsstsw/stack/loadLSST.bash
- LSP Notebook aspect (from a terminal): /opt/lsst/software/stack/loadLSST.bash

From the command line, execute the commands below in the example code:

Example Code

```
source 'path'
setup lsst_distrib
```

Expected Result

Science pipeline software is available for use. If additional packages are needed (for example, 'obs' packages such as 'obs_subaru'), then additional 'setup' commands will be necessary.

To check versions in use, type:
eups list -s

Step 3 Description

Execute 'faro' on a repository containing processed data. Identify the path to the data, which we will call 'DATA/-path', then execute something similar to the following (with paths, datasets, and flags replaced or additionally specified as needed):

Example Code

```
pipetask –long-log run -j 2 -b DATA/path/butler.yaml –register-dataset-types -p $FARO_DIR/pipelines/metrics_pipeline.yaml -d "band in ('g', 'r', 'i') AND tract=9813 AND skymap='hsc_rings_v1' AND instrument='HSC'" –output u/username/faro_metrics -i HSC/runs/RC2/w_2021_06 2>&1 | tee w06_2021_tract9813_faro.txt
```

Expected Result

The output collection (in this case, "u/username/faro_metrics") containing metric measurements and any associated extras and metadata is available via the butler.

Step 4 Description

Confirm that the PA2gri threshold has been applied to the assessment of the computed values of PF1 for filters g,r,i.

Expected Result

A JSON file (and/or a report generated from that JSON file) demonstrating that PA2gri has been calculated (and that it used PF1).

5.1.3.22 LVV-T1758 - Verify that the repeatability outlier limit for isolated bright non-saturated point sources in the u, z, and y filters (PA2uzy) can be applied.

Version **1**. Open *LVV-T1758* test case in Jira.

Verify that the DM system has provided the code to apply the repeatability outlier limit for isolated bright non-saturated point sources in the u, z, and y filters(PA2uzy) to to computed values of the PF1 metric.

Preconditions:

Final comment:

Detailed steps :

Step 1 Description

Identify a dataset containing at least one field in each of the u, z, and y filters with multiple overlapping visits.

Expected Result

A dataset that has been ingested into a Butler repository.

Step 2 Description

The 'path' that you will use depends on where you are running the science pipelines. Options:

- local (newinstall.sh - based install):[path_to_installation]/loadLSST.bash
- development cluster ("lsst-dev"): /software/lsstsw/stack/loadLSST.bash
- LSP Notebook aspect (from a terminal): /opt/lsst/software/stack/loadLSST.bash

From the command line, execute the commands below in the example code:

Example Code

```
source 'path'
setup lsst_distrib
```

Expected Result

Science pipeline software is available for use. If additional packages are needed (for example, 'obs' packages such as 'obs_subaru'), then additional 'setup' commands will be necessary.

To check versions in use, type:
eups list -s

Step 3 Description

Execute 'faro' on a repository containing processed data. Identify the path to the data, which we will call 'DATA/-path', then execute something similar to the following (with paths, datasets, and flags replaced or additionally specified as needed):

Example Code

pipetask –long-log run -j 2 -b DATA/path/butler.yaml –register-dataset-types -p $FARO_DIR/pipelines/metrics_pipeline.yaml -d "band in ('g', 'r', 'i') AND tract=9813 AND skymap='hsc_rings_v1' AND instrument='HSC'" –output u/username/faro_metrics -i HSC/runs/RC2/w_2021_06 2>&1 | tee w06_2021_tract9813_faro.txt

Expected Result

The output collection (in this case, "u/username/faro_metrics") containing metric measurements and any associated extras and metadata is available via the butler.

Step 4 Description

Confirm that the PA2uzy threshold has been applied to the assessment of the computed values of PF1 for filters u,z,y.

Expected Result

A JSON file (and/or a report generated from that JSON file) demonstrating that PA2uzy has been calculated (and that it used PF1).

5.1.3.23 LVV-T149 - Verify implementation of Catalog Queries

Version **1**. Open *LVV-T149* test case in Jira.

Verify that SQL, or a similar structured language, can be used to query catalogs.

Preconditions:

An operational QSERV database that has been verified via LVV-T1085 and LVV-T1086 and LVV-T1087.

Final comment:

Detailed steps :

Step 1 Description

Execute a simple query (for example, the one below) and confirm that it returns the expected result.

Example Code

```
SELECT * FROM Object WHERE qserv_areaspec_box(316.582327, −6.839078, 316.653938, −6.781822)
```

Expected Result

A catalog of objects satisfying the specified constraints.

Step 2 Description

Repeat the query from all available access routes (e.g., an external VO client, internal DM tools on the development cluster, the Science Platform query tool, and from within the Notebook Aspect), confirming in each case that the results are as expected.

Expected Result

5.1.3.24 LVV-T40 - Verify implementation of Generate WCS for Visit Images

Version **1**. Open *LVV-T40* test case in Jira.

Verify that Processed Visit Images produced by the AP and DRP pipelines include FITS WCS accurate to specified **astrometricAccuracy** over the bounds of the image.

Preconditions:

Final comment:

Detailed steps :

Step 1 Description

Identify an appropriate processed dataset for this test.

Expected Result

A dataset with Processed Visit Images available.

Step 2 Description

Identify the path to the data repository, which we will refer to as 'DATA/path', then execute the following:

Example Code

```
from lsst.daf.butler import Butler
repo = 'Data/path'
collection = 'collection'
butler = Butler(repo, collections=collection)
```

Expected Result

Butler repo available for reading.

Step 3 Description

Select a single visit from the dataset, and extract its WCS object and the source list.

Expected Result

A table containing detected sources, and a WCS object associated with that catalog.

Step 4 Description

Confirm that each CCD within the visit image contains at least **astrometricMinStandards** astrometric standards that were used in deriving the astrometric solution.

Expected Result

At least **astrometricMinStandards** from each CCD were used in determining the WCS solution.

Step 5 Description

Starting from the XY pixel coordinates of the sources, apply the WCS to obtain RA, Dec coordinates.

Expected Result
A list of RA, Dec coordinates for all sources in the catalog.

Step 6 Description
We will assume that Gaia provides a source of "truth." Match the source list to Gaia DR2, and calculate the positional offset between the test data and the Gaia catalog.

Expected Result
A matched catalog of sources in common between the test source list and Gaia DR2.

Step 7 Description
Apply appropriate cuts to extract the optimal dataset for comparison, then calculate statistics (median, 1-sigma range, etc.; also plot a histogram) of the offsets in milliarcseconds. Confirm that the offset is less than **astrometricAccuracy**.

Expected Result
Histogram and relevant statistics needed to confirm that the WCS transformation is accurate.

Step 8 Description
Repeat Step 5, but for subregions of the image, to confirm that the accuracy criterion is met at all positions.

Expected Result
astrometricAccuracy requirement is met over the entire image.

5.1.3.25 LVV-T129 - Verify implementation of Provide Calibrated Photometry

Version **1**. Open *LVV-T129* test case in Jira.

Verify that the DMS provides photometry calibrated in AB mags and fluxes (in nJy) for all measured objects and sources. Must be tested for both DRP and AP products.

Preconditions:

Final comment:

Detailed steps :

Step 1 Description

Identify the path to the data repository, which we will refer to as 'DATA/path', then execute the following:

Example Code

```
from lsst.daf.butler import Butler
repo = 'Data/path'
collection = 'collection'
butler = Butler(repo, collections=collection)
```

Expected Result

Butler repo available for reading.

Step 2 Description

Ingest the data products from an appropriate DRP-processed dataset.

Expected Result

Step 3 Description

Confirm that AB-calibrated magnitudes and fluxes are available for all measured Sources and Objects. [An enhanced verification could include matching the sources to an external source catalog and comparing the magnitudes to show that they are well-calibrated.]

Expected Result

Calibrated fluxes and magnitudes are available for all sources, as well as tools to convert measured fluxes to magnitudes (and vice-versa).

Step 4 Description

Ingest the data products from an appropriate AP processing dataset.

Expected Result

Step 5 Description

Confirm that AB-calibrated magnitudes and fluxes are available for all measured Sources, DIASources, and Objects. [An enhanced verification could include matching the sources to an external source catalog and comparing the magnitudes to show that they are well-calibrated.]

Expected Result

Calibrated fluxes and magnitudes are available for all Sources, DIASources, and Objects, as well as tools to convert measured fluxes to magnitudes (and vice-versa).

5.1.3.26 LVV-T115 - Verify implementation of Calibration Production Processing

Version **1**. Open *LVV-T115* test case in Jira.

Execute CPP on a variety of representative cadences, and verify that the calibration pipeline correctly produces necessary calibration products.

Preconditions:

Final comment:

Detailed steps :

Step 1 Description

Identify a suitable set of calibration frames, including biases, dark frames, and flat-field frames.

Expected Result

Step 2 Description

Execute the Calibration Products Production payload. The payload uses raw calibration images and information from the Transformed EFD to generate a subset of Master Calibration Images and Calibration Database entries in the Data Backbone.

Expected Result

Step 3 Description

Confirm that the expected Master Calibration images and Calibration Database entries are present and well-formed.

Expected Result

Step 4 Description

Confirm that the expected data products are created, and that they have the expected properties.

Expected Result

Repos containing valid calibration products that are well-formed and ready to be applied to processed datasets.

5.1.3.27 LVV-T1862 - Verify determining effectiveness of dark current frame

Version **1**. Open *LVV-T1862* test case in Jira.

Verify that the DMS can determine the effectiveness of a dark correction and determine how often it should be updated.

Preconditions:

Final comment:

Detailed steps :

Step 1 Description

Identify the path to a dataset containing dark frames (i.e., exposures taken with the shutter closed).

Expected Result

Step 2 Description

Execute the Calibration Products Production payload. The payload uses raw calibration images and information from the Transformed EFD to generate a subset of Master Calibration Images and Calibration Database entries in the Data Backbone.

Expected Result

Step 3 Description

Confirm that the expected Master Calibration images and Calibration Database entries are present and well-formed.

Expected Result

Step 4 Description

Determining whether the dark correction is being done properly will require on-sky science data. The dark correction can be applied to these frames and the results inspected to ensure that the correction was correctly measured and applied.

Expected Result

Applying the dark correction to a dataset produces noticeable differences between the original frame(s) and the corrected outputs.

5.1.3.28 LVV-T89 - Verify implementation of Calibration Image Provenance

Version **1**. Open *LVV-T89* test case in Jira.

Verify that the DMS records the required provenance information for the Calibration Data Products.

Preconditions:

Final comment:

Detailed steps :

Step 1	Description

Ingest an appropriate precursor calibration dataset into a Butler repo.

Expected Result

Step 2	Description

Execute the Calibration Products Production payload. The payload uses raw calibration images and information from the Transformed EFD to generate a subset of Master Calibration Images and Calibration Database entries in the Data Backbone.

Expected Result

Step 3	Description

Confirm that the expected Master Calibration images and Calibration Database entries are present and well-formed.

Expected Result

Step 4 Description

Load the relevant database/Butler data product, and observe that all provenance information has been retained.

Expected Result

A dataset consisting of calibration images, with provenance information recorded and properly associated with the calibration images.

5.1.3.29 LVV-T88 - Verify implementation of Calibration Data Products

Version **1**. Open *LVV-T88* test case in Jira.

Verify that the DMS can produce and archive the required Calibration Data Products: cross talk correction, bias, dark, monochromatic dome flats, broad-band flats, fringe correction, and illumination corrections.

Preconditions:

Final comment:

Detailed steps :

Step 1 Description

Identify a suitable set of calibration frames, including biases, dark frames, and flat-field frames.

Expected Result

Step 2 Description

Execute the Calibration Products Production payload. The payload uses raw calibration images and information from the Transformed EFD to generate a subset of Master Calibration Images and Calibration Database entries in the Data Backbone.

Expected Result

Step 3	Description

Confirm that the expected Master Calibration images and Calibration Database entries are present and well-formed.

Expected Result

Step 4	Description

Confirm that the expected data products are created, and that they have the expected properties.

Expected Result

A full set of calibration data products has been created, and they are well-formed.

Step 5	Description

Test that the calibration products are archived, and can readily be applied to science data to produce the desired corrections.

Expected Result

Confirmation that application of the calibration products to processed data has the desired effects.

5.1.3.30 LVV-T85 - Verify implementation of Crosstalk Correction Matrix

Version **1**. Open *LVV-T85* test case in Jira.

Verify that the DMS can generate a cross-talk correction matrix from appropriate calibration data.
Verify that the DMS can measure the effectiveness of the cross-talk correction matrix.

Preconditions:

Final comment:

Detailed steps :

Step 1 Description

Identify an appropriate calibration dataset that can be used to derive the crosstalk correction matrix.

Expected Result

Step 2 Description

Execute the Calibration Products Production payload. The payload uses raw calibration images and information from the Transformed EFD to generate a subset of Master Calibration Images and Calibration Database entries in the Data Backbone.

Expected Result

Step 3 Description

Confirm that the expected Master Calibration images and Calibration Database entries are present and well-formed.

Expected Result

Step 4 Description

Confirm that the crosstalk correction matrix is produced and persisted.

Expected Result

A correction matrix quantifying what fraction of the signal detected in any given amplifier on each sensor in the focal plane appears in any other amplifier.

Step 5 Description

Apply the crosstalk correction to simulated images, and confirm that the correction is performing as expected.

Expected Result
A noticeable difference between images before and after applying the correction.

5.1.3.31 LVV-T83 - Verify implementation of Bad Pixel Map

Version **1**. Open *LVV-T83* test case in Jira.

Verify that the DMS can produce a map of detector pixels that suffer from pathologies, and that these pathologies are encoded in at least 32-bit values.

Preconditions:

Final comment:

Detailed steps :

Step 1	Description

Interrogate the calibRegistry for the metadata associated with a bad pixel map, where the validity range contains the date of interest.

Expected Result
A bad pixel map for the requested date has been returned.

Step 2	Description

Check that the bad pixel pathologies are encoded as at least 32-bit values, and that the various pathologies are represented by different encoding.

Expected Result
Bad pixel values can be decoded to determine their pathologies using their 32-bit values.

A Documentation

The verification process is defined in LSE-160. The use of Docsteady to format Jira information in various test and planing documents is described in DMTN-140 and practical commands are given in DMTN-178.

B Acronyms used in this document

Acronym	Description
AP	Alert Production
BPS	Batch Production Service
CCD	Charge-Coupled Device
CI	Continuous Integration
CPP	Calibration Production Processing
DC2	Data Challenge 2 (DESC)
DEC	Declination
DESC	Dark Energy Science Collaboration
DM	Data Management
DMS	Data Management Subsystem
DMSR	DM System Requirements; LSE-61
DMTN	DM Technical Note
DR2	Data Release 2
DRP	Data Release Production
EFD	Engineering and Facility Database
FITS	Flexible Image Transport System
HSC	Hyper Suprime-Cam
JSON	JavaScript Object Notation
LAN	Local Area Network
LDM	LSST Data Management (Document Handle)
LSE	LSST Systems Engineering (Document Handle)
LSP	LSST Science Platform (now Rubin Science Platform)
LSST	Legacy Survey of Space and Time (formerly Large Synoptic Survey Telescope)
LVV	LSST Verification and Validation

OCS	Observatory Control System
OSS	Observatory System Specifications; LSE-30
PMCS	Project Management Controls System
PSF	Point Spread Function
QA	Quality Assurance
QSERV	LSST Query Services
RA	Right Ascension
RSP	Rubin Science Platform
SQL	Structured Query Language
USDF	United States Data Facility
VO	Virtual Observatory
WCS	World Coordinate System