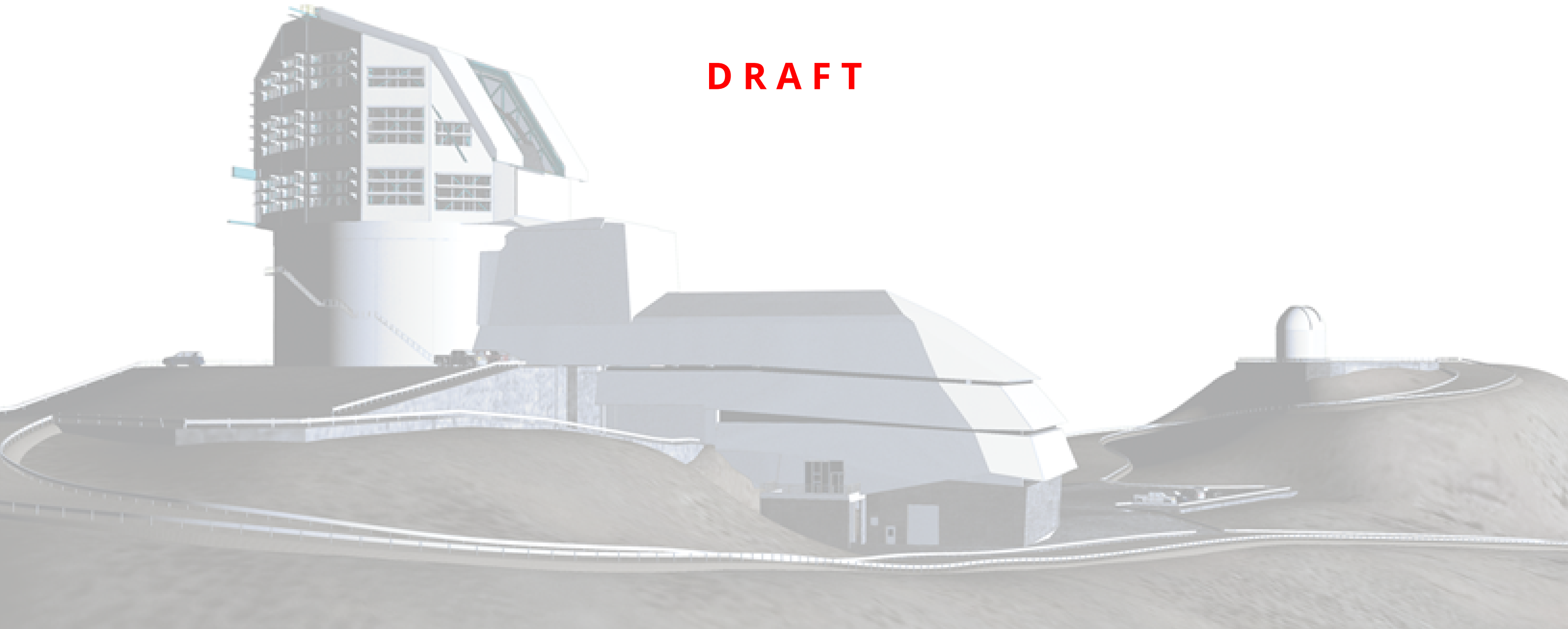
Vera C. Rubin Observatory
Data Management

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

Jeffrey Carlin

DMTR-401

Latest Revision: 2024-05-09



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 2024-05-09 . The most recent change to the document repository was on 2024-05-09.

Change Record

Version	Date	Description	Owner name
	2023-07-01	First draft	Jeffrey Carlin
1.0	2024-05-08	Test campaign LVV-P106 completed and results approved. DM-40311	Jeffrey Carlin

Document curator: Jeffrey Carlin
Document source location: `https://github.com/lsst-dm/DMTR-401`
Version from source repository: 74fef86

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 LSST-Cam. 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 **Completed** .

1.4 References

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

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

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

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

[5] **[LDM-503]**, O'Mullane, W., Swinbank, J., Juric, M., et al., 2023, Data Management Test Plan, URL `https://ldm-503.lsst.io/`, Vera C. Rubin Observatory Data Management Controlled Document LDM-503

[6] **[LSE-160]**, Selvy, B., 2013, Verification and Validation Process, URL `https://ls.st/LSE-160`, Vera C. Rubin Observatory 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-T146	Leanne Guy	Leanne Guy	
LVV-T1240	Jim Bosch	Jeffrey Carlin	
LVV-T132	Leanne Guy	Jeffrey Carlin	
LVV-T62	Jim Bosch	Jeffrey Carlin	
LVV-T41	Jim Bosch	Jeffrey Carlin	
LVV-T97	Kian-Tat Lim	Jeffrey Carlin	
LVV-T1946	Jeffrey Carlin	Jeffrey Carlin	
LVV-T1947	Jeffrey Carlin	Jeffrey Carlin	
LVV-T28	Colin Slater	Jeffrey Carlin	
LVV-T142	Colin Slater	Jeffrey Carlin	
LVV-T1748	Jeffrey Carlin	Jeffrey Carlin	
LVV-T1759	Jeffrey Carlin	Jeffrey Carlin	
LVV-T1758	Jeffrey Carlin	Jeffrey Carlin	
LVV-T149	Leanne Guy	Jeffrey Carlin	
LVV-T40	Jeffrey Carlin	Jeffrey Carlin	

4 Test Campaign Overview

4.1 Summary

T. Plan LVV-P106:	**Data Management Acceptance Test Campaign, Fall 2023**	Completed
T. Cycle LVV-C260:	**Data Management Acceptance Test Campaign, Fall 2023**	Done

Test Cases	**Ver.**
LVV-T146	1
LVV-T1240	1
LVV-T132	1
LVV-T62	2
LVV-T41	1
LVV-T97	1
LVV-T1946	1
LVV-T1947	1
LVV-T28	1
LVV-T142	1
LVV-T1748	1
LVV-T1759	1
LVV-T1758	1
LVV-T149	1
LVV-T40	1

Table 2: Test Campaign Summary

4.2 Overall Assessment

In this test campaign, we have successfully verified 9 unique requirements from LSE-61 via the execution of 15 Test Cases (all of which passed). Of these requirements, 6 are of priority 1a, and 3 are priority 1b. The set of requirements tested in this campaign mostly cover aspects of the DM system related to the software pipelines and the facilities for data processing and handling. These tests were performed at the US Data Facility (USDF) using precursor HSC and Auxtel data.

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: Done

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-T146 - Verify implementation of DMS Initialization Component

Version **1**. Open *LVV-T146* test case in Jira.

Demonstrate that all components of the DM system have a defined deployment configuration within the DM deployment strategy

Preconditions:

Final comment:

Detailed steps :

Step 1 Description
Inspect each service component to check if it has a deployment configuration defined

Expected Result
All systems have a defined deployment configuration as part of the DM deployment strategy

5.1.3.2 LVV-T1240 - Verify implementation of minimum astrometric standards per CCD

Version **1**. Open *LVV-T1240* test case in Jira.

Verify that each CCD in a processed dataset had its astrometric solution determined by at least **astrometricMinStandards = 5** astrometric standards.

Preconditions:

Final comment:
Test executed with science pipelines version w_2023_37 in the RSP Notebook aspect at the USDF.

The executed notebook was saved in the repository associated with this campaign's test report as "notebooks/test_LVV-T40_T1240.ipynb".

Detailed steps :

Step 1 Description

Identify an appropriate processed dataset for this test.

Expected Result

A dataset with Processed Visit Images.

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 calibration data. For a subset of CCDs, check how many astrometric standards contributed to the solution. Confirm that this number is at least **astrometricMinStandards = 5.**

Expected Result

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

5.1.3.3 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.

5.1.3.4 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:
Test executed with science pipelines version w_2023_34 in the RSP Notebook aspect at the USDF.

The executed notebook was saved in the repository associated with this campaign's test report as "notebooks/test_LVV-T62.ipynb".

Detailed steps :

Step 1 Description
Identify a repo containing RC2 data 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.

5.1.3.5 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:
Test executed with science pipelines version w_2023_37 in the RSP Notebook aspect at the USDF.

The executed notebook was saved in the repository associated with this campaign's test report as "notebooks/test_LVV-T41.ipynb".

Detailed steps :

Step 1	Description

Identify a repo containing data 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.

5.1.3.6 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:
Executed at the USDF using the w_2023_43 version of the Science Pipelines.

In addition to demonstrating that changing the "RELEASE_ID" results in unique IDs, we further examine the code itself to demonstrate that the uniqueness of IDs is ensured by the way the code is implemented.

The implementation of how the 64 bits of the Source/Object IDs are apportioned is in the `_IdGeneratorBits` class, and especially its `__post_init__`:

- it gets the maximum number of bits needed to pack the data ID (based on `{visit, detector}` for Source, and `{tract, patch}` for Object) and turns that into the number of distinct data IDs it could hold (`n_data_ids`);
- it multiplies that with the `n_releases` option to identify how many "upper" bits need to be reserved for `n_data_ids*n_releases` "catalog_ids", and sets `n_counters` to be the number of values remaining in the Source or Object 64-bit ID to count sources or objects within one image.

Because some interfaces historically wanted to count bits rather than just multiply integers, there is a mix of direct multiplication and log2 addition. But you can see the result most clearly in `FullIdGenerator.arange` and `FullIdGenerator.catalog_id`; a full Source or Object ID is:

```
id = counter + n_counters * catalog_id
```

or

```
id = counter + n_counters * (packed_data_id + n_data_ids * release_id)
```

and hence the releases will get distinct IDs as long as counter values are less than `n_counters` and the packed data IDs are less than `n_data_ids`. (And there are several checks throughout the file for those criteria).

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.

5.1.3.7 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:
This test case can be executed by running the script test_LVV-T1946.py, which is available in the test report github repository's "scripts/" directory.

The tests that confirm fluxes have "reasonable" values are checking that the fluxes, if converted to magnitudes, would result in a magnitude fainter than -5.

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.

5.1.3.8 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:
This test case can be executed by running the scripts test_LVV-T1947_DiaSource.py, test_LVV-T1947_forcedSourceOnDiaObject.py, and test_LVV-T1947_DiaObject.py, which are available

in the test report github repository's "scripts/" directory.

The tests that confirm fluxes have "reasonable" values are checking that the fluxes, if converted to magnitudes, would result in a magnitude fainter than -5.

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.

5.1.3.9 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:
This test case can be executed by running the scripts test_LVV-T28.py, test_LVV-T28_forcedSource.py, and test_LVV-T28_DC2.py, which are available in the test report github repository's "scripts/" directory.

The tests that confirm fluxes have "reasonable" values are checking that the fluxes, if converted to magnitudes, would result in a magnitude fainter than -5.

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 repo with processed RC2 precursor data 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.

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

Identify and read an appropriate repo with processed DC2 precursor data containing coadds with the Butler.

Expected Result

Step 6 Description

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

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

5.1.3.10 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:
Executed at the USDF using the w_2023_43 version of the Science Pipelines.

Detailed steps :

Step 1 Description
Create a HTCondor Pool at USDF

Example Code
allocateNodes.py -c 8 -m 4-00:00:00 -q roma,milano -g 900 s3df

---

Expected Result

Step 2 Description
Submit a RC2-subset run using bps

Example Code

bps submit –output u/username/collectionname ${DRP_PIPE_DIR}/drp_pipe/pipelines/HSC/DRP-RC2_subset.yaml

Expected Result

Step 3 Description

Cancel the job while it is in progress

Example Code

bps cancel [job id]

Expected Result

Step 4 Description

Check that some output products are missing

Example Code

butler query-datasets /repo/main –collections u/username/collectionname

Expected Result

Step 5 Description

Submit a new job to finish the tasks that were not completed

Example Code

bps submit –output u/username/collectionname ${DRP_PIPE_DIR} drp_pipe/pipelines/HSC/DRP-RC2_subset.yaml

Expected Result

Step 6 Description

Confirm that all expected files are now present

Example Code

butler query-datasets /repo/main –collections u/username/collectionname

Expected Result

5.1.3.11 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 processed data.

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 'analysis_tools' 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 $ANALYSIS_TOOLS_DIR/pipelines/matchedVisitQualit
-d "band in ('g', 'r', 'i') AND tract=9813 AND skymap='hsc_rings_v1' AND instrument='HSC'" –output u/username/a-
tools_metrics -i HSC/runs/RC2/w_2023_36 –instrument lsst.obs.subaru.HyperSuprimeCam 2>&1 | tee w36_2023_tract9813_atools.tx
```

Expected Result

The output collection (in this case, "u/username/atools_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.

5.1.3.12 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:
This test used a modified version of the analysis_tools pipeline "matchedVisitQualityCore.yaml."

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 'analysis_tools' 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 $ANALYSIS_TOOLS_DIR/pipelines/matchedVisitQualit
-d "band in ('g', 'r', 'i') AND tract=9813 AND skymap='hsc_rings_v1' AND instrument='HSC'" –output u/username/a-
tools_metrics -i HSC/runs/RC2/w_2023_36 –instrument lsst.obs.subaru.HyperSuprimeCam 2>&1 | tee w36_2023_tract9813_atools.tx
```

Expected Result

The output collection (in this case, "u/username/atools_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 PF1 has been calculated (and that it used the requested threshold value of PA2gri).

Step 5 Description
Change the value of the PA2 threshold in the pipeline yaml for analysis_tools, then rerun analysis_tools

Expected Result

Step 6 Description
Confirm that the new PA2 threshold has been applied when computing PF1.

Expected Result
A JSON file (and/or a report generated from that JSON file) demonstrating that PF1 has been calculated (and that it used the requested threshold value of PA2gri).

5.1.3.13 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 computed values of the PF1 metric.

Preconditions:

Final comment:
Note that because we do not have access to u-band data, this test was performed for only y- and z-band. The steps would be unchanged for u-band data.

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 ‘analysis_tools‘ 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 $ANALYSIS_TOOLS_DIR/pipelines/matchedVisitQualit -d "band in ('g', 'r', 'i') AND tract=9813 AND skymap='hsc_rings_v1' AND instrument='HSC"' –output u/username/a-tools_metrics -i HSC/runs/RC2/w_2023_36 –instrument lsst.obs.subaru.HyperSuprimeCam 2>&1 | tee w36_2023_tract9813_atools.tx

Expected Result

The output collection (in this case, "u/username/atools_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 PF1 has been calculated (and that it used the requested PA2uzy threshold).

Step 5 Description

Change the value of the PA2 threshold in the pipeline yaml for analysis_tools, then rerun analysis_tools

Expected Result

Step 6 Description

Confirm that the new PA2 threshold has been applied when computing PF1.

Expected Result

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

5.1.3.14 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:
Executed using the IDF Notebook, Portal, and API aspects. For the notebook execution, we used science pipelines version w_2023_34.

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 dp02_dc2_catalogs.Object as obj WHERE CONTAINS(POINT('ICRS', obj.coord_ra, obj.coord_dec), CIRCLE('ICRS', 62.0, -37.0, 0.10)) = 1
```

Expected Result
A catalog of objects satisfying the specified constraints. The catalog should contain 26,115 results.

Step 2	Description

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

Expected Result

5.1.3.15 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:
Test executed with science pipelines version w_2023_37 in the RSP Notebook aspect at the USDF.

The executed notebook was saved in the repository associated with this campaign's test report as "notebooks/test_LVV-T40_T1240.ipynb".

Detailed steps :

Step 1	Description

Identify an appropriate repo containing processed HSC data 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 DR3, 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 DR3.

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.

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
ADQL	Astronomical Data Query Language (IVOA standard)
AP	Alert Production
API	Application Programming Interface
BPS	Batch Production Service
CCD	Charge-Coupled Device
CI	Continuous Integration
DC2	Data Challenge 2 (DESC)
DEC	Declination
DESC	Dark Energy Science Collaboration
DIA	Difference Image Analysis
DM	Data Management
DMS	Data Management Subsystem
DMSR	DM System Requirements; LSE-61
DMTN	DM Technical Note
DR3	Data Release 3
DRP	Data Release Production
FITS	Flexible Image Transport System
FTS3	File Transfer Service 3
HSC	Hyper Suprime-Cam
ICRS	International Celestial Reference Frame
IDF	Interim Data Facility
JSON	JavaScript Object Notation
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
NFS	Network File System
OCS	Observatory Control System
ObsLocTAP	Observation Locator Table Access Protocol (IVOA standard)
PMCS	Project Management Controls System
PSF	Point Spread Function
PVI	Processed Visit Image
PanDA	Production ANd Distributed Analysis system
QC	Quality Control
QSERV	LSST Query Services
RA	Right Ascension
RSP	Rubin Science Platform
S3	(Amazon) Simple Storage Service
SQL	Structured Query Language
TAP	Table Access Protocol (IVOA standard)
US	United States
USDF	United States Data Facility
VO	Virtual Observatory
WCS	World Coordinate System
bps	bit(s) per second