

Rubin Observatory

Vera C. Rubin Observatory
Data Management

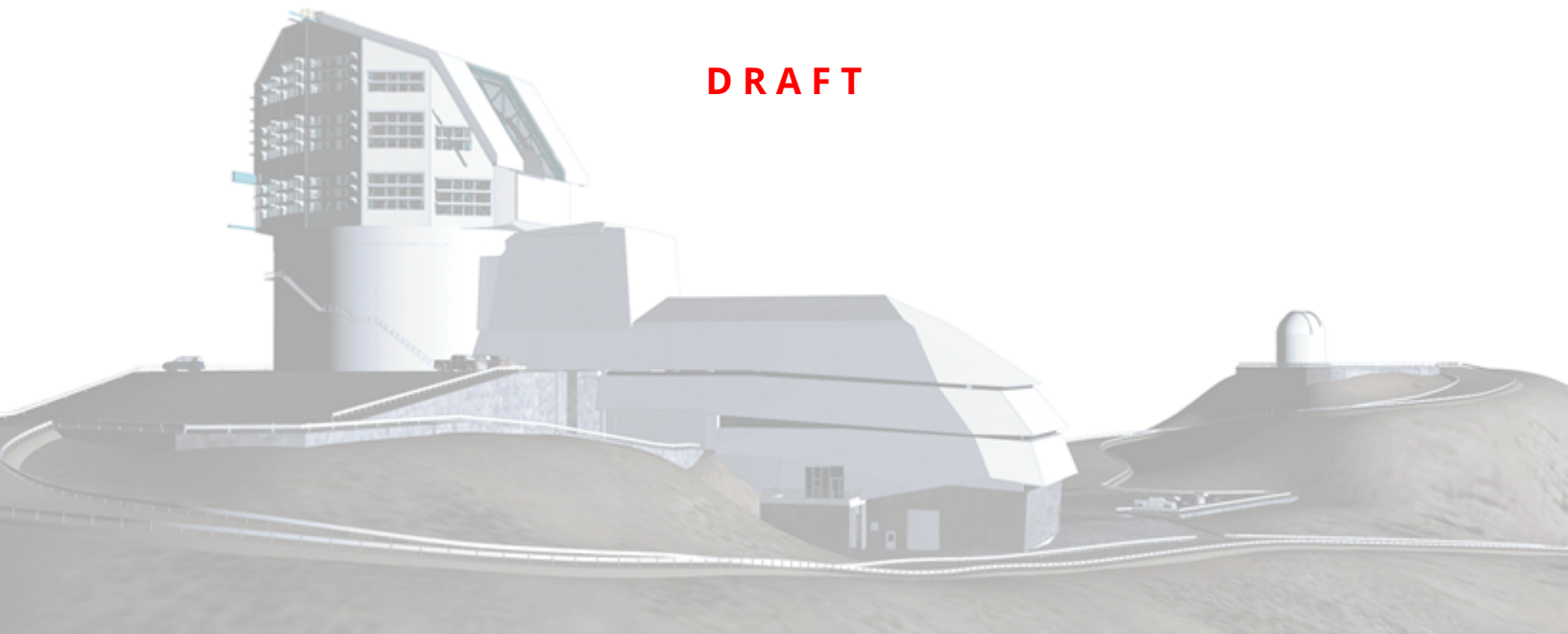
LVV-P71: Science Pipelines Release 20.0.0 Acceptance Test Campaign Test Plan and Report

Jeffrey Carlin

DMTR-261

Latest Revision: 2020-11-10

DRAFT



Abstract

This is the test plan and report for **Science Pipelines Release 20.0.0 Acceptance Test Campaign**, an LSST milestone pertaining to the Data Management Subsystem.

Draft

Change Record

Version	Date	Description	Owner name
	2020-06-24	First draft	Jeff Carlin
1.0	2020-08-05	Test plan LVM-P71 approved, test activity ready to start. DM-25646	Jeff Carlin

Document curator: Jeff Carlin

Document source location: <https://github.com/lstt-dm/DMTR-261>

Version from source repository: 3f7fb95

Draft

Contents

1 Introduction	1
1.1 Objectives	1
1.2 System Overview	1
1.3 Document Overview	2
1.4 References	2
2 Test Plan Details	3
2.1 Data Collection	3
2.2 Verification Environment	3
2.3 Entry Criteria	3
2.4 Related Documentation	3
2.5 PMCS Activity	3
3 Personnel	4
4 Test Campaign Overview	5
4.1 Summary	5
4.2 Overall Assessment	8
4.3 Recommended Improvements	9
5 Detailed Test Results	11
5.1 Test Cycle LWV-C153	11
5.1.1 Software Version/Baseline	11
5.1.2 Configuration	11
5.1.3 Test Cases in LWV-C153 Test Cycle	11
5.1.3.1 LWV-T28 - Verify implementation of measurements in catalogs from PVIs	11
5.1.3.2 LWV-T1946 - Verify implementation of measurements in catalogs from coadds	13

5.1.3.3	LVV-T1947 - Verify implementation of measurements in catalogs from difference images	15
5.1.3.4	LVV-T40 - Verify implementation of Generate WCS for Visit Images	18
5.1.3.5	LVV-T1232 - Verify Implementation of Catalog Export Formats From the Portal Aspect	23
5.1.3.6	LVV-T151 - Verify Implementation of Catalog Export Formats From the Notebook Aspect	25
5.1.3.7	LVV-T133 - Verify implementation of Provide Beam Projector Coordinate Calculation Software	30
5.1.3.8	LVV-T1759 - Verify calculation of photometric outliers in gri bands	33
5.1.3.9	LVV-T1758 - Verify calculation of photometric outliers in uzy bands	36
5.1.3.10	LVV-T1756 - Verify calculation of photometric repeatability in uzy filters	39
5.1.3.11	LVV-T1757 - Verify calculation of photometric repeatability in gri filters	42
5.1.3.12	LVV-T125 - Verify implementation of Simulated Data	45
5.1.3.13	LVV-T36 - Verify implementation of Difference Exposures	47
5.1.3.14	LVV-T39 - Verify implementation of Generate Photometric Zero-point for Visit Image	49
5.1.3.15	LVV-T38 - Verify implementation of Processed Visit Images	51
5.1.3.16	LVV-T42 - Verify implementation of Processed Visit Image Content	53
5.1.3.17	LVV-T1264 - Verify implementation of archiving camera test data	56
5.1.3.18	LVV-T146 - Verify implementation of DMS Initialization Component	58
5.1.3.19	LVV-T144 - Verify implementation of Task Specification	59
5.1.3.20	LVV-T145 - Verify implementation of Task Configuration	60

5.1.3.21	LW-T45 - Verify implementation of Prompt Processing Data Quality Report Definition	61
5.1.3.22	LW-T46 - Verify implementation of Prompt Processing Performance Report Definition	63
A Acronyms used in this document		65
B Traceability		67

Draft

LVV-P71: Science Pipelines Release 20.0.0 Acceptance Test Campaign Test Plan and Report

1 Introduction

1.1 Objectives

This Acceptance Test campaign aims to verify a subset of DMSR (LSE-61) requirements related to the LSST Science Pipelines. It will be executed in conjunction with the release of Science Pipelines Version 20.0.0, but the pipeline release is not contingent upon this test campaign. This Test Plan aims to demonstrate that the included requirements have been met by Version 20.0.0 of the Pipelines, and to thus fully verify their completion and readiness for LSST Operations.

1.2 System Overview

The tests to be executed are intended to verify that the DM system satisfies a subset of the requirements outlined in the Data Management System Requirements (DMSR; LSE-61). This subset of requirements is related to pipeline algorithms, and was selected for this campaign to coincide with the release of a new version of the LSST Science Pipelines. Additional DMSR requirements will be verified in later Acceptance Test Campaigns.

Applicable Documents:

LSE-61 Data Management System Requirements

LDM-503 Data Management Test Plan

LDM-639 LSST Data Management Acceptance Test Specification (issue 2.1)

The tests will be performed using the HSC-RC2 dataset (as defined in DM-11345). When possible, we will start our tests with the data products resulting from processing HSC-RC2 with the v20_0_0_rc1 pipelines release candidate (DM-24478) that was used to create v20 of

the Science Pipelines.

1.3 Document Overview

This document was generated from Jira, obtaining the relevant information from the LVV-P71 Jira Test Plan and related Test Cycles (LVV-C153).

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 5 provides detailed results for each step in each test case.

The current status of test plan LVV-P71 in Jira is **Approved** .

1.4 References

- [1] **[LSE-61]**, Dubois-Felsmann, G., Jenness, T., 2018, *LSST Data Management Subsystem Requirements*, LSE-61, URL <https://1s.st/LSE-61>
- [2] **[LDM-639]**, Guy, L., 2018, *DM Acceptance Test Specification*, LDM-639, URL <https://1s.st/LDM-639>
- [3] **[LDM-503]**, O'Mullane, W., Swinbank, J., Jurić, M., Economou, F., 2018, *Data Management Test Plan*, LDM-503, URL <https://1s.st/LDM-503>

2 Test Plan Details

2.1 Data Collection

Observing is not required for this test campaign.

2.2 Verification Environment

The “lsst-lsp-stable” instance of the LSST Science Platform (LSP), hosted at the LDF, and the “lsst-dev” development cluster at NCSA. In particular, we will use Release 20.0.0 of the Pipelines.

2.3 Entry Criteria

Release and availability of Science Pipelines version 20.

2.4 Related Documentation

Docushare collection where additional relevant documentation can be found:

- Jupyter notebooks in which many of these tests were executed are archived at this github repository. The ‘executions/’ directory contains the rendered, executed notebooks, while the ‘notebooks/’ directory houses the un-executed notebooks.

2.5 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-P71 owner:		Jeffrey Carlin	
T. Cycle LVV-C153 owner:		Jeffrey Carlin	
Test Cases	Assigned to	Executed by	Additional Test Personnel
LVV-T28	Colin Slater	Jeffrey Carlin	
LVV-T1946	Jeffrey Carlin	Jeffrey Carlin	
LVV-T1947	Jeffrey Carlin	Jeffrey Carlin	
LVV-T40	Jim Bosch	Jeffrey Carlin	
LVV-T1232	Colin Slater	Jeffrey Carlin	
LVV-T151	Colin Slater	Jeffrey Carlin	
LVV-T133	Robert Lupton	Jeffrey Carlin	
LVV-T1759	Jeffrey Carlin	Jeffrey Carlin	
LVV-T1758	Jeffrey Carlin	Jeffrey Carlin	
LVV-T1756	Jeffrey Carlin	Jeffrey Carlin	
LVV-T1757	Jeffrey Carlin	Jeffrey Carlin	
LVV-T125	Robert Lupton	Jeffrey Carlin	
LVV-T36	Eric Bellm	Jeffrey Carlin	
LVV-T39	Jim Bosch	Jeffrey Carlin	
LVV-T38	Eric Bellm	Jeffrey Carlin	
LVV-T42	Jim Bosch	Jeffrey Carlin	
LVV-T1264	Robert Gruendl	Jeffrey Carlin	
LVV-T146	Robert Gruendl		
LVV-T144	Kian-Tat Lim		
LVV-T145	Robert Lupton		
LVV-T45	Eric Bellm		
LVV-T46	Eric Bellm		

4 Test Campaign Overview

4.1 Summary

T. Plan LVV-P71:		Science Pipelines Release 20.0.0 Acceptance Test Campaign			Approved
T. Cycle LVV-C153:		Pipelines v20 Release DM Acceptance Test Campaign			In Progress
Test Cases	Ver.	Status	Comment	Issues	
LVV-T28	1	Pass	Test executed on lsst-lsp-stable, using Release v20.0.0, with a Large container.		
LVV-T1946	1	Pass	Test executed on lsst-lsp-stable, using Release v20.0.0, with a Large container.		
LVV-T1947	1	Fail	Test executed on lsst-lsp-stable, using Release v20.0.0, with a Large container.	DM-26372 DM-26372	
LVV-T40	1	Pass	Test executed on lsst-lsp-stable, using Release v20.0.0, with a Large container. The results of this test are contained in a Jupyter notebook named 'test_LVV-T40_T1240.ipynb'.		
LVV-T1232	1	Pass	Executed using the Portal aspect at: https://lsst-lsp-stable.ncsa.illinois.edu/portal/app/		
LVV-T151	1	Pass	Executed using the Portal aspect at: https://lsst-lsp-stable.ncsa.illinois.edu/nb . Results appear in Jupyter notebook test_LVV-T151.ipynb.		
LVV-T133	1	Pass	Test successfully executed on 'lsst-devl'.		
LVV-T1759	1	Pass	Note that this test is being executed using 'validate_drp', which will be deprecated and replaced with a new metrics framework once the Gen3 butler becomes available. Nonetheless, the test is valid, because the algorithm(s) to calculate the metrics are the same, and will be moved into the new framework.		

LWV-T1758	1	Initial Pass	<p>Note that this test is being executed using 'validate_drp', which will be deprecated and replaced with a new metrics framework once the Gen3 butler becomes available. Nonetheless, the test is valid, because the algorithm(s) to calculate the metrics are the same, and will be moved into the new framework.</p> <p>Also note that because there are no sufficient u-band data for testing, we currently are only able to test this requirement on y- and z-band data. This is the reason the status is "Initial Pass" rather than "Pass."</p>
LWV-T1756	1	Initial Pass	<p>Note that this test is being executed using 'validate_drp', which will be deprecated and replaced with a new metrics framework once the Gen3 butler becomes available. Nonetheless, the test is valid, because the algorithm(s) to calculate the metrics are the same, and will be moved into the new framework.</p> <p>Also note that because there are no sufficient u-band data for testing, we currently are only able to test this requirement on y- and z-band data. This is the reason the status is "Initial Pass" rather than "Pass."</p>
LWV-T1757	1	Pass	<p>Note that this test is being executed using 'validate_drp', which will be deprecated and replaced with a new metrics framework once the Gen3 butler becomes available. Nonetheless, the test is valid, because the algorithm(s) to calculate the metrics are the same, and will be moved into the new framework.</p>
LWV-T125	1	Pass	<p>This test demonstrates that the injected fake sources are visible, and that their measured magnitudes match the input values. However, future exploration should probably test the same for a variety of artificial sources (i.e., including galaxies).</p>
LWV-T36	1	Pass	<p>See associated Jupyter notebooks 'test_LWV-T36.ipynb' and 'test_LWV-T36-HITS.ipynb'.</p>
LWV-T39	1	Pass	<p>See associated Jupyter notebook 'test_LWV-T39.ipynb'.</p>

LWV-T38	1	Pass	In this test, we have used precursor data from DECam for verification, but any suitable precursor data would be sufficient. When ComCam data become available, this test should be executed on them.
LWV-T42	1	Pass	In this test, we have used precursor data from DECam for verification, but any suitable precursor data would be sufficient. When ComCam data become available, this test should be executed on them.
LWV-T1264	1	Pass	See associated Jupyter notebook in 'test_LWV-T1264.ipynb'. This test used data products created and ingested as part of the Operations Rehearsal detailed in ? : "LDM-503-11a: ComCam OPS Readiness Test Plan and Report".
LWV-T146	1	Not Executed	After some discussion, it was decided to defer this test to a later campaign. This is due in part to the lack of on-site personnel at some sites to execute the test. More importantly, however, is the need to understand whether this should be divided into multiple tests that address hardware at each of the various sites that are part of the DMS, and whether those systems should all be taken down simultaneously, or tested separately. We hope to execute this test as part of the next Operations Rehearsal.
LWV-T144	1	Not Executed	The Gen3 Butler rollout was delayed, so this test was not executed. Because many aspects of task specification and execution are in active development, we will defer this test to a later campaign that takes place after Gen3 has stabilized.
LWV-T145	1	Not Executed	The Gen3 Butler rollout was delayed, so this test was not executed. Because many aspects of task configuration, specification, and execution are in active development, we will defer this test to a later campaign that takes place after Gen3 has stabilized.

LWV-T45	1	Not Executed	This test was included because it was anticipated that the infrastructure to produce the Prompt Processing Data Quality Report would be available by the time of this Test Campaign. Because the report has not yet been defined, nor the automated pipeline to ingest and process data, then generate the report, we defer this test to a later campaign.
LWV-T46	1	Not Executed	This test was included because it was anticipated that the infrastructure to produce the Prompt Processing Performance Report would be available by the time of this Test Campaign. Because the report has not yet been defined, nor the automated pipeline to ingest and process data, then generate the report, we defer this test to a later campaign.

Table 2: Test Campaign Summary

4.2 Overall Assessment

This test campaign successfully demonstrated that the DM system is satisfying an additional 14 newly-verified Test Cases. These correspond to 9 unique DMSR requirements that are now completely verified, and 2 that are partially verified. The requirements that were verified include some related to basic image processing and data products (e.g., difference imaging, well-formed processed visit images, WCS solution, source catalog quantities and properties), and a number of tests related to DM algorithms provided to calculate performance metrics. (Note that the DM system is not required to verify that the performance metrics are meeting their required thresholds, but simply that DM must provide the tools to calculate/evaluate the metrics.) The tests demonstrated that a subset of these metrics can be measured at present, and that the framework (namely, 'validate_drp') exists to enable the addition of other metrics.

Some minor issues that arose include:

- An issue was uncovered in LVV-T1937 where the catalog quantities are not presented with the correct units. A ticket was filed to fix this, and this test will be revisited in a future campaign.
- Two tests (LVV-T1756 and LVV-T1758) were given a status of “Initial Pass” because the required software tools to execute the tests exist and work fine, but there is no u-band data to test the algorithms on. Thus even though the software meets the requirement, we will wait to fully sign off on this requirement until we have u-band data for testing.
- A few of the tests that were originally planned were not executed because of delays in the required tooling to become available (or initial misunderstandings about the timelines for them to be ready). These will be deferred to later test campaigns.

4.3 Recommended Improvements

The most important improvement that could be made to this process would be to automate many of the tests for data processing products rather than executing them in Jupyter notebooks. While the notebooks are good for visualization and documentation, the requirements (e.g., “generate WCS for Visit Images”) should be verified via code that checks every image rather than just a subset, and that reports a single pass/fail result in the end. The current notebook based approach should be adapted for future test campaigns (especially as tests are performed on larger datasets).

Testing of requirements related to performance metrics currently relies on ‘validate_drp’. While it is satisfactory for confirming that many of the performance metrics *can* be calculated, ‘validate_drp’ is (a) not capable of calculating some types of metrics, and (b) will not scale up to larger datasets. A replacement based on the Gen3 Butler is currently being implemented, and will form the basis of performance metric testing and monitoring henceforth. This new tooling, which incorporates and/or updates many of the ‘validate_drp’ algorithms, will be used for future acceptance test campaigns.

LVV-T125 as written only requires testing that artificial sources can be inserted into the data and recovered after the fact. In a future test campaign, we recommend updating this Test Case to require insertion of artificial stars and galaxies, and perhaps to place some requirements on the level of accuracy with which their properties can be recovered.

Finally, this testing highlighted the need to clarify some requirements, or at least to be explicit and clear about how they are being interpreted when executing tests.

Draft

5 Detailed Test Results

5.1 Test Cycle LVV-C153

Open test cycle *Pipelines v20 Release DM Acceptance Test Campaign* in Jira.

Test Cycle name: Pipelines v20 Release DM Acceptance Test Campaign

Status: In Progress

This test cycle verifies a subset of DMSR (LSE-61) requirements related to the LSST Science Pipelines, 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).

5.1.1 Software Version/Baseline

All tests will be performed with LSST Science Pipelines release version 20.0.0, including its algorithms and resulting science data products.

5.1.2 Configuration

Not provided.

5.1.3 Test Cases in LVV-C153 Test Cycle

5.1.3.1 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:

Execution status: **Pass**

Final comment:

Test executed on lsst-lsp-stable, using Release v20.0.0, with a Large container.

Detailed steps results:

Step	Step Details
1	<p>Description</p> <p>Identify the path to the data repository, which we will refer to as 'DATA/path', then execute the following:</p> <p>Example Code</p> <pre>import lsst.daf.persistence as dafPersist butler = dafPersist.Butler(inputs='DATA/path')</pre> <p>Expected Result</p> <p>Butler repo available for reading.</p> <p>Actual Result</p> <p>We used the output repo from HSC-RC2 data processing, as executed using v20_0_0 of the science pipelines. It is tagged with ticket number DM-25349, and is located at this path on lsst-dev: '/datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349'</p> <p>Status: Pass</p>
2	<p>Description</p> <p>Identify and read an appropriate processed precursor dataset containing coadds with the Butler.</p> <p>Expected Result</p> <p>Actual Result</p>

The test was executed in a notebook named 'test_LVV-T28.ipynb'. Within the notebook, initialization of the Butler repo was done as follows:

```
import lsst.daf.persistence as dafPersist
rc2_repo = '/datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349'
butler = dafPersist.Butler(rc2_repo)
```

An arbitrary dataId for a processed visit image within this repo was chosen.

The specific code used:

```
filter = 'HSC-I'
# PVI:
ccd = 14
visit = 35890
dataIdPVI = {'filter':filter, 'visit':visit, 'ccd':ccd}
src = butler.get('src', dataId = dataIdPVI)
```

Status: **Pass**

3 Description

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

Expected Result

Confirmation of measurements in catalogs encoded in flux units.

Actual Result

In the notebook, we extracted the schema for the source catalog. Source flux measurements all contain the string "instFlux" (i.e., "instrumental flux") in their names, so we subselect on this string. We then confirm that all measurements with "instFlux" in their names have units of "count" using a simple assert statement. The results (seen in notebook 'test_LVV-T28.ipynb') are as follows:

```
All src instFlux entries have units of counts: True
```

Status: **Pass**

5.1.3.2 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:

Execution status: **Pass**

Final comment:

Test executed on lsst-lsp-stable, using Release v20.0.0, with a Large container.

Detailed steps results:

Step	Step Details
1	<p>Description</p> <p>Identify the path to the data repository, which we will refer to as 'DATA/path', then execute the following:</p> <hr/> <p>Example Code</p> <pre>import lsst.daf.persistence as dafPersist butler = dafPersist.Butler(inputs='DATA/path')</pre> <hr/> <p>Expected Result</p> <p>Butler repo available for reading.</p> <hr/> <p>Actual Result</p> <p>We used the output repo from HSC-RC2 data processing, as executed using v20_0_0 of the science pipelines. It is tagged with ticket number DM-25349, and is located at this path on lsst-dev: '/datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349'</p> <hr/> <p>Status: Pass</p>
2	<p>Description</p> <p>Identify and read an appropriate processed precursor dataset containing coadds with the Butler.</p> <hr/> <p>Expected Result</p> <hr/> <p>Actual Result</p>

The test was executed in a notebook named 'test_LVV-T1946.ipynb'. Within the notebook, initialization of the Butler repo was done as follows:

```
import lsst.daf.persistence as dafPersist
rc2_repo = '/datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349'
butler = dafPersist.Butler(rc2_repo)
```

An arbitrary dataId for a coadded image within this repo was chosen.

The specific code used:

```
filter = 'HSC-I'
# Coadd
tract = 9615
patch = '5,3'
dataIdCoadd = {'tract':tract, 'filter':filter, 'patch':patch}
forced_src = butler.get('deepCoadd_forced_src', dataId = dataIdCoadd)
```

Status: **Pass**

3 Description

Verify that the coadd catalog provides measurements in flux units.

Expected Result

Confirmation of measurements in catalogs encoded in flux units.

Actual Result

In the notebook, we extracted the schema for the source catalog. Source flux measurements all contain the string "instFlux" (i.e., "instrumental flux") in their names, so we subselect on this string. We then confirm that all measurements with "instFlux" in their names have units of "count" using a simple assert statement. The results (seen in notebook 'test_LVV-T1946.ipynb') are as follows:

```
All forced_src instFlux entries have units of counts: True
```

Status: **Pass**

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

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

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

Preconditions:

Execution status: **Fail**

Final comment:

Test executed on *lsst-lsp-stable*, using Release v20.0.0, with a Large container.

Issues found during the execution of LVV-T1947 test case:

- DM-26372 Units wrong in *deepDiff_diaSrc* catalog

Detailed steps results:

Step	Step Details
1	<p>Description</p> <p>Identify the path to the data repository, which we will refer to as 'DATA/path', then execute the following:</p> <p>Example Code</p> <pre>import lsst.daf.persistence as dafPersist butler = dafPersist.Butler(inputs='DATA/path')</pre> <p>Expected Result</p> <p>Butler repo available for reading.</p> <p>Actual Result</p> <p>We used the output repo from DECam HiTS data processing regularly performed by Meredith Rawls, as recently (Aug. 2020) executed using v20_0_0 of the science pipelines. It is located at this path on <i>lsst-dev</i>: <code>diffim_repo = 'project/mrawls/hits2015/rerun/cw_2020_08'</code></p> <p>Status: Pass</p>

2 Description

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

Expected Result

Actual Result

The test was executed in a notebook named 'test_LVV-T1947.ipynb'. Within the notebook, initialization of the Butler repo was done as follows:

```
import lsst.daf.persistence as dafPersist
```

An arbitrary dataId for a difference image catalog within this repo was chosen.

```
diffim_repo = '/project/mrawls/hits2015/rerun/cw_2020_08'  
butler_dia = dafPersist.Butler(diffim_repo)  
dataId_dia = {'visit': 411406, 'ccd': 15}  
deepDiff_dia_src = butler_dia.get('deepDiff_diaSrc', dataId = dataId_dia)
```

Status: **Pass**

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.

Actual Result

In the notebook, we extracted the schema for the source catalog. Source flux measurements all contain the string “in

AssertionError: FALSE: not all instFlux entries have units of counts.

Specifically, the fields that did not meet the requirement are these. A Jira ticket has been filed to fix these.

Column	Units
	pixel
ip_diffim_DipoleFit_pos_instFluxErr	
	pixel
ip_diffim_DipoleFit_neg_instFluxErr	
	none
ip_diffim_forced_PsfFlux_instFlux	
	none
ip_diffim_forced_PsfFlux_instFluxErr	

Issues found executing this step:

DM-26372 Units wrong in deepDiff_diaSrc catalog

Status: **Fail**

5.1.3.4 LUV-T40 - Verify implementation of Generate WCS for Visit Images

Version 1. Open *LUV-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:

Execution status: **Pass**

Final comment:

Test executed on lsst-lsp-stable, using Release v20.0.0, with a Large container. The results of this test are contained in a Jupyter notebook named 'test_LVV-T40_T1240.ipynb'.

Detailed steps results:

Step	Step Details
1	<p>Description</p> <p>Identify an appropriate processed dataset for this test.</p> <hr/> <p>Expected Result</p> <p>A dataset with Processed Visit Images available.</p> <hr/> <p>Actual Result</p> <p>We used the output repo from HSC-RC2 data processing, as executed using v20_0_0 of the science pipelines.</p> <hr/> <p>Status: Pass</p>
2	<p>Description</p> <p>Identify the path to the data repository, which we will refer to as 'DATA/path', then execute the following:</p> <hr/> <p>Example Code</p> <pre>import lsst.daf.persistence as dafPersist butler = dafPersist.Butler(inputs='DATA/path')</pre> <hr/> <p>Expected Result</p> <p>Butler repo available for reading.</p> <hr/> <p>Actual Result</p> <p>The repository is tagged with ticket number DM-25349, and is located at this path on lsst-dev: '/datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349'</p> <hr/> <p>Status: Pass</p>
3	<p>Description</p> <p>Select a single visit from the dataset, and extract its WCS object and the source list.</p> <hr/> <p>Expected Result</p> <p>A table containing detected sources, and a WCS object associated with that catalog.</p>

Actual Result

In practice, we iterated this process, randomly selecting 1000 visits and extracting the 'src' catalog and the WCS associated with the 'calexp' for each one.

Status: **Pass**

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.

Actual Result

This is captured by the "hasWcs" method on the 'calexp'. We extract this for all of the 1000 random visits.

```
if butler.datasetExists('calexp', dataId = dataref):
    calexp = butler.get('calexp', dataId = dataref)
    haswcs_flags.append(butler.get('calexp', dataId = dataref).hasWcs())
```

Result of the test:

```
Percentage of fields with > astrometricMinStandards=5: 100.0 % -- True
```

Status: **Pass**

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.

Actual Result

This was done using the following code:

```
# Start from XY, then use the WCS to convert to RA, Dec:
xxx = src.getX()
yyy = src.getY()
radec = [wcs.pixelToSky(xxx[i], yyy[i]) for i in range(len(xxx))]
radec_arr = np.array([(coo.getRa().asDegrees(), coo.getDec().asDegrees()) for coo in radec])
```

Status: **Pass**

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.

Actual Result

Used astroquery to extract Gaia sources, then Astropy utilities to match the catalogs:

```
gaia_mch = Gaia.query_object_async(coordinate=cen, width=width, height=height)
sc_src = SkyCoord(radec_arr[:,0]*u.deg, radec_arr[:,1]*u.deg)
sc_gaia = SkyCoord(gaia_mch['ra'], gaia_mch['dec'])
src_match = sc_src.match_to_catalog_sky(sc_gaia)
sep_match = src_match[1]
```

Filtered the matched catalog to keep only matches with <0.5” separation and with magnitude difference < 1.0 (relative to the median magnitude difference of all sources, to account for different filters):

```
okmch = (sep_match.arcsec < 0.5)
matchsep = sep_match[okmch]
```

Require the matches to have similar magnitudes:

```
gaia_gmag = gaia_mch['phot_g_mean_mag']
magdiff = src_mag[okmch][:,0]-gaia_gmag[src_match[0][okmch]]
okmagdiff = (np.abs(magdiff - np.median(magdiff)) < 1.0)
okmatchsep = matchsep[okmagdiff]
```

This yields the final matched list.

Status: **Pass**

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.

Actual Result

We retained only CCDs with >10 matches to Gaia sources. Rather than histograms, we used comparisons of the various extracted parameters. Figures are shown in the notebook ('test_LVV-T40_T1240.ipynb').

In addition to figures, we calculated the percentage of images that satisfied the requirement on **astrometricAccuracy**. This was slightly less than 100% in all trials, likely due to some deep (or problematic) images having few Gaia matches. The results printed in the notebook are as follows:

```
Percentage of fields meeting the threshold: 98.7551867219917 % -- False
```

Because a small fraction of fields do not meet the criterion, the test for LVV-T40 technically does not pass. Given that ~98% of the selected CCDs met the astrometric accuracy criterion, and that the small number not passing may be due to a lack of Gaia reference catalog objects, or data quality issues with the input images (i.e., *not* due to a deficiency in DM algorithms), we consider the result of this test **"Pass."** However, we recommend that the required fraction of images meeting the astrometric accuracy criterion should be clarified, and that the data quality selection needed to test this requirement should be better defined.

Status: **Pass**

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.

Actual Result

This is implicit in the random selection of 1000 CCDs in the analysis. By its nature, the process samples a variety of subregions in the focal plane.

Status: **Pass**

5.1.3.5 LVV-T1232 - Verify Implementation of Catalog Export Formats From the Portal Aspect

Version 1. Open *LW-T1232* test case in Jira.

Verify that catalog data is exportable from the portal aspect in a variety of community-standard formats.

Preconditions:

Execution status: **Pass**

Final comment:

Executed using the Portal aspect at: <https://lsst-lsp-stable.ncsa.illinois.edu/portal/app/>

Detailed steps results:

Step	Step Details
1	<p>Description</p> <p>Navigate to the Portal Aspect endpoint. The stable version should be used for this test and is currently located at: https://lsst-lsp-stable.ncsa.illinois.edu/portal/app/.</p> <hr/> <p>Expected Result</p> <p>A credential-entry screen should be displayed.</p> <hr/> <p>Actual Result</p> <p>While logged into the NCSA VPN, two-factor authentication credentials were entered at the prompts.</p> <hr/> <p>Status: Pass</p>
2	<p>Description</p> <p>Enter a valid set of credentials for an LSST user with LSP access on the instance under test.</p> <hr/> <p>Expected Result</p> <p>The Portal Aspect UI should be displayed following authentication.</p> <hr/> <p>Actual Result</p> <p>Splash page of Portal Aspect displayed.</p> <hr/> <p>Status: Pass</p>
3	<p>Description</p>

Select query type "ADQL".

 Expected Result

 Actual Result

Selected via the "ADQL" button on the Portal splash page, which brought up a box in which to type queries.

 Status: **Pass**

4 Description

Execute the example query given in the example code below by entering the text in the ADQL Query box, then clicking "Search" at the lower left corner of the page.

 Example Code

```
SELECT cntr, ra, decl, w1mpro_ep, w2mpro_ep, w3mpro_ep FROM wise_00.allwise_p3as_mep WHERE CONTAINS(POINT('ICRS', ra, decl), CIRCLE('ICRS', 192.85, 27.13, .2)) = 1
```

 Expected Result

A new page will load with the search results as a table, with some plots as well.

 Actual Result

Query executed quickly, and returned a table with 50000 results.

 Status: **Pass**

5 Description

Click the icon that looks like a floppy disk (it says "Save the content as an IPAC, CSV, or TSV table" when you mouse over it).

 Expected Result

 Actual Result

Save menu pops up with options.

 Status: **Pass**

6 Description

- Select "CSV", then specify a destination to save the file on your local computer.
- Select "VOTable", then specify a destination to save the file on your local computer.
- Select "FITS", then specify a destination to save the file on your local computer.

 Expected Result

 Actual Result

Files are present on disk:

```
$ ls -ltrh
total 20232
-rw-r-r-@ 1 jcarlin staff 3.7M Aug 19 16:31 LVV-T1232_test.csv
-rw-r-r-@ 1 jcarlin staff 3.1M Aug 19 16:33 LVV-T1232_test_vot.xml
-rw-r-r-@ 1 jcarlin staff 3.1M Aug 19 16:33 LVV-T1232_test.fits
```

Status: **Pass**

7

Description

Open each of the files (either in TOPCAT, or using Astropy io tools). Confirm that the data tables are well-formed, and that each table contains the same columns and the same number of rows.

Expected Result

Actual Result

Each file was opened in TOPCAT, and it was confirmed that each of them contains the same 6 columns and 50000 rows.

Status: **Pass**

8

Description

Currently, there is no logout mechanism on the portal.
This should be updated as the system matures.

Simply close the browser window.

Expected Result

Closed browser window. When navigating to the portal endpoint, expect to execute the steps in LVV-T849.

Actual Result

Exited the Portal Aspect and closed the browser window.

Status: **Pass**

5.1.3.6 LVV-T151 - Verify Implementation of Catalog Export Formats From the Notebook Aspect

Version 1. Open *LW-T151* test case in Jira.

Verify that catalog data is exportable from the notebook aspect in a variety of community-standard formats.

Preconditions:

Execution status: **Pass**

Final comment:

Executed using the Portal aspect at: <https://lsst-lsp-stable.ncsa.illinois.edu/nb>. Results appear in Jupyter notebook test_LVV-T151.ipynb.

Detailed steps results:

Step	Step Details
1	<p>Description</p> <p>Authenticate to the notebook aspect of the LSST Science Platform (NB-LSP). This is currently at https://lsst-lsp-stable.ncsa.illinois.edu/nb.</p> <hr/> <p>Expected Result</p> <p>Redirection to the spawner page of the NB-LSP allowing selection of the containerized stack version and machine flavor.</p> <hr/> <p>Actual Result</p> <p>While logged into the NCSA VPN, two-factor authentication credentials were entered at the prompts. Notebook Aspect authentication credentials were entered at the prompt.</p> <hr/> <p>Status: Pass</p>
2	<p>Description</p> <p>Spawn a container by:</p> <ol style="list-style-type: none"> 1) choosing an appropriate stack version: e.g. the latest weekly. 2) choosing an appropriate machine flavor: e.g. medium 3) click "Spawn" <hr/> <p>Expected Result</p> <p>Redirection to the JupyterLab environment served from the chosen container containing the correct stack version.</p> <hr/> <p>Actual Result</p> <p>For this test, we use Release v20.0.0, with a Large container. The JupyterLab environment is presented.</p> <hr/> <p>Status: Pass</p>
3	<p>Description</p>

Open a new launcher by navigating in the top menu bar "File" -> "New Launcher"

Expected Result

A launcher window with several sections, potentially with several kernel versions for each.

Actual Result

A launcher appeared with "Python 3" and "LSST" options under "Notebook" and "Console", and various options in the "Other" category.

Status: **Pass**

4 Description

Select the option under "Notebook" labeled "LSST" by clicking on the icon.

Expected Result

An empty notebook with a single empty cell. The kernel show up as "LSST" in the top right of the notebook.

Actual Result

Notebook was displayed. We renamed the notebook "test_LVW-T151.ipynb".

Status: **Pass**

5 Description

Execute a query in a notebook to select a small number of stars. In the example code below, we query the WISE catalog, then extract the results to an Astropy table.

Example Code

```
import pandas
import pyvo
service = pyvo.dal.TAPService('http://lsst-lsp-stable.nsa.illinois.edu/api/tap')

results = service.search("SELECT ra, decl, w1mpro_ep, w2mpro_ep, w3mpro_ep FROM
wise_00.allwise_p3as_mep WHERE CONTAINS(POINT('ICRS', ra, decl), CIRCLE('ICRS', 192.85, 27.13,
.2)) = 1")
tab = results.to_table()
```

Expected Result

Actual Result

The Notebook Aspect has some tools implemented that execute queries using 'pyvo', so we use those instead of directly accessing the TAPService with pyvo as follows:

```
import pandas
from jupyterlabutils.notebook import get_catalog, retrieve_query
```

```
pandas.set_option('display.max_rows', 1000)
service = get_catalog()
```

```
results = service.search("SELECT ra, decl, w1mpro_ep, w2mpro_ep, w3mpro_ep FROM
wise_00.allwise_p3as_mep WHERE CONTAINS(POINT('ICRS', ra, decl), CIRCLE('ICRS', 192.85, 27.13,
.2)) = 1")
tab = results.to_table()
```

This returns a table with 97058 entries.

Status: **Pass**

6 Description

Using the example code below, save the files to your storage space on the LSP Notebook Aspect.

Confirm that non-empty output files appear on disk.

Example Code

```
tab.write('test.csv', format='ascii.csv')
tab.write('test.vot', format='votable')
tab.write('test.fits', format='fits')
```

Expected Result

For the example given here, there should be the following files with the file size as listed:

- test.csv 5.7M
- test.vot 16M
- test.fits 4.5M

Actual Result

The files appear on disk. Note, however, that the predicted file sizes listed above are not what resulted. Here are the results from a command line:

```
$ ls -ltrh test.*
-rw-r--r- 1 jcarlin 3.8M Aug 20 00:37 test.csv
-rw-r--r- 1 jcarlin 13M Aug 20 00:37 test.vot
-rw-r--r- 1 jcarlin 3.8M Aug 20 00:37 test.fits
```

Status: **Pass**

7 Description

Check that these files contain the same number of rows:

Example Code

```
from astropy.table import Table
dat_csv = Table.read('test.csv', format='ascii.csv')
dat_vot = Table.read('test.vot', format='votable')
dat_fits = Table.read('test.fits', format='fits')
```

```
import numpy as np
print(np.size(dat_csv), np.size(dat_vot), np.size(dat_fits))
```

Expected Result

Print statement produces output "97058 97058 97058".

Actual Result

Executing the above code yields the following output:
97058 97058 97058

Status: **Pass**

8 Description

Under the 'File' menu at the top of your Jupyter notebook session, select one of the following:

- Save All, Exit, and Log Out
- Exit and Log Out Without Saving

Expected Result

You will be returned to the LSP landing page: <https://lsst-lsp-stable.ncsa.illinois.edu/> It is now safe to close the browser window.

Actual Result

Logged out, exited the Notebook Aspect, and closed the browser window.

Status: **Pass**

5.1.3.7 LVV-T133 - Verify implementation of Provide Beam Projector Coordinate Calculation Software

Version 1. Open *LW-T133* test case in Jira.

Verify that the DMS provides software to calculate coordinates relating the collimated beam projector position and telescope pupil position to the illumination position on the telescope optical elements and focal plane.

Preconditions:

Execution status: **Pass**

Final comment:

Test successfully executed on 'lsst-devl'.

Detailed steps results:

Step	Step Details
1	<p>Description</p> <p>On the LSST development cluster or notebook aspect, git clone the repo containing the CBP package: https://github.com/lsst/cbp</p> <hr/> <p>Expected Result</p> <hr/> <p>Actual Result</p> <p>Working in a terminal on the command line logged into <i>lsst-devl02</i>. With the Rubin Obs/LSST stack set up: Make the directory: <code>/project/jcarlin/cbp</code>, and change to that directory. Then: <code>git clone https://github.com/lsst/cbp.git</code></p> <hr/> <p>Status: Pass</p>

2 Description

Follow the steps in the package README to install the package.

Expected Result

Actual Result

To make the package available to use, simply execute the following:

```
cd cbp/  
git checkout 20.0.0 ### This step is to ensure that we are using the correct tagged version.  
setup -r .
```

Execute the setup of 'lsst_distrib' again to ensure that the correct version (v20.0.0) of the pipelines is loaded.

```
setup -t v20_0_0 lsst_distrib
```

Status: **Pass**

3 Description

Confirm that the package can be loaded in python, and that some of the tests in the 'tests/' folder will execute.

Expected Result

Successful execution of test scripts, which demonstrate the calculation of beam projector coordinates.

Actual Result

From the command line, launch ipython and confirm that the package can be loaded:

```
ipython
In [1]: import lsst.cbp

In [2]: dir(lsst.cbp)
Out[2]:
['BeamInfo',
'CoordinateConverter',
'CoordinateConverterConfig',
'MaskInfo',
'__builtins__',
'__cached__',
'__dependency_versions__',
'__doc__',
'__file__',
'__fingerprint__',
'__loader__',
'__name__',
'__package__',
'__path__',
'__repo_version__',
'__spec__',
'__version__',
'__version_info__',
'beamInfo',
'computeHolePositions',
'coordUtils',
'coordinateConverter',
'coordinateConverterConfig',
'maskInfo',
'version']
```

This was successful. The package loads, and we can introspect its methods via the 'dir' command.

Now confirm that the tests can be executed. We will cycle through them one by one, confirming that the unit tests (a) execute, and (b) pass with status "OK."

```
python test_computeHolePositions.py
```

```
....
```

```
-----
Ran 4 tests in 0.059s
```

Status: **Pass**

5.1.3.8 LUV-T1759 - Verify calculation of photometric outliers in gri bands

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

Verify that the DM system has provided the code to calculate the photometric repeatability in the g, r, and i filters, and assess whether it meets the requirement that no more than **PF1 = 10[percent]** of the repeatability outliers exceed the outlier limit of **PA2gri = 15 millimagnitudes**.

Preconditions:

Execution status: **Pass**

Final comment:

Note that this test is being executed using 'validate_drp', which will be deprecated and replaced with a new metrics framework once the Gen3 butler becomes available. Nonetheless, the test is valid, because the algorithm(s) to calculate the metrics are the same, and will be moved into the new framework.

Detailed steps results:

Step	Step Details
1	<p>Description</p> <p>Identify a dataset containing at least one field in each of the g, r, and i filters with multiple overlapping visits.</p> <hr/> <p>Expected Result</p> <p>A dataset that has been ingested into a Butler repository.</p> <hr/> <p>Actual Result</p> <p>We will use the HSC "RC2" data that were reprocessed with the release candidate of Science Pipelines v20.0. These are located at '/datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349' on the development machines.</p>

Status: **Pass**

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
```

Actual Result

To select the v20.0 release of the Science Pipelines, we entered the following:

```
setup -t v20_0_0 lsst_distrib
setup obs_subaru
```

Status: **Pass**

3 Description

Execute 'validate_drp' on a repository containing precursor data. Identify the path to the data, which we will call 'DATA/path', then execute the following (with additional flags specified as needed):

Example Code

```
validateDrp.py 'DATA/path'
```

Expected Result

JSON files (and associated figures) containing the Measurements and any associated "extras."

Actual Result

An example of the call to execute 'matchedVisitMetrics' (which runs the metrics portion of 'validate_drp' on data that have already been processed):

```
matchedVisitMetrics.py /datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349 -output
/datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349/validateDrp/matchedVisitMetrics/9813/HSC-
G -config doApplyExternalPhotoCalib=True doApplyExternalSkyWcs=True instrumentName=HSC
datasetName=HSC-RC2 -id tract=9813 visit=11690^11692^11694^11696^11698^11700^11702^11704^11706^11708^1
ccd=0..8^10..103
```

This process is included as part of the regular RC2 reprocessing. See the log for this and other runs of 'matchedVisitMetrics' in /datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349/qaLogs/matchedVisitMetrics/

Status: Pass

4

Description

Confirm that the metric PA2gri has been calculated using the threshold PF1, and that its values are reasonable.

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).

Actual Result

By executing "reportPerformance /datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349/qaLogs/matchedVisitMetrics/matchedVisitMetrics-9813-HSC-G-251998.log", a report is generated summarizing all of the metrics that were measured by 'validate_drp'. The relevant lines showing results for PA2 and PF1 from tract 9813 are:

```
PA2 - HSC-G - 28.8 mmag
PF1 - HSC-G - 30.2 %
PA2 - HSC-I - 32.6 mmag
PF1 - HSC-I - 34.6 %
PA2 - HSC-R - 33.4 mmag
PF1 - HSC-R - 34.4 %
```

Note that because the science pipelines are still in development, the actual resulting values for these metrics are not important. The requirement to be satisfied is that we have (a) provided the software to calculate them, and (b) provided a framework for comparing them to requirements. These are both satisfied.

Status: **Pass**

5.1.3.9 LVV-T1758 - Verify calculation of photometric outliers in uzy bands

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

Verify that the DM system has provided the code to calculate the photometric repeatability in the u, z, and y filters, and assess whether it meets the requirement that no more than **PF1 = 10[percent]** of the repeatability outliers exceed the outlier limit of **PA2uzy = 22.5 millimag-nitudes**.

Preconditions:

Execution status: **Initial Pass**

Final comment:

Note that this test is being executed using 'validate_drp', which will be deprecated and replaced with a new metrics framework once the Gen3 butler becomes available. Nonetheless, the test is valid, because the algorithm(s) to calculate the metrics are the same, and will be moved into the new framework.

Also note that because there are no sufficient u-band data for testing, we currently are only able to test this requirement on y- and z-band data. This is the reason the status is "Initial Pass" rather than "Pass."

Detailed steps results:

Step	Step Details
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.

Actual Result

We will use the HSC "RC2" data that were reprocessed with the release candidate of Science Pipelines v20.0. These are located at '/datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349' on the development machines.

Status: **Pass**

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
```

Actual Result

To select the v20.0 release of the Science Pipelines, we entered the following:

```
setup -t v20_0_0 lsst_distrib
setup obs_subaru
```

Status: **Pass**

3 Description

Execute 'validate_drp' on a repository containing precursor data. Identify the path to the data, which we will call 'DATA/path', then execute the following (with additional flags specified as needed):

Example Code

```
validateDrp.py 'DATA/path'
```

Expected Result

JSON files (and associated figures) containing the Measurements and any associated "extras."

Actual Result

An example of the call to execute 'matchedVisitMetrics' (which runs the metrics portion of 'validate_drp' on data that have already been processed):

```
matchedVisitMetrics.py /datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349 -output
/datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349/validateDrp/matchedVisitMetrics/9813/HSC-
Y -config doApplyExternalPhotoCalib=True doApplyExternalSkyWcs=True instrumentName=HSC
datasetName=HSC-RC2 -id tract=9813 visit=318^322^324^326^328^330^332^344^346^348^350^352^354^356^358^36
ccd=0..8^10..103
```

This process is included as part of the regular RC2 reprocessing. See the log for this and other runs of 'matchedVisitMetrics' in /datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349/qaLogs/matchedVisitMetrics/

Status: **Pass**

4

Description

Confirm that the metric PA2uzy has been calculated using the threshold PF1, and that its values are reasonable.

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).

Actual Result

By executing `"reportPerformance /datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349/qaLogs/matchedVisitMetrics/matchedVisitMetrics-9813-HSC-Y-252002.log"`, a report is generated summarizing all of the metrics that were measured by `'validate_drp'`. The relevant lines showing results for PA2 and PF1 from tract 9813 are:

PA2 - HSC-Y - 34.2 mmag

PF1 - HSC-Y - 20.5 %

PA2 - HSC-Z - 27.4 mmag

PF1 - HSC-Z - 15.6 %

Note that because the science pipelines are still in development, the actual resulting values for these metrics are not important. The requirement to be satisfied is that we have (a) provided the software to calculate them, and (b) provided a framework for comparing them to requirements. These are both satisfied.

This step is given a status of "Initial Pass" because there are no u-band data to test on. We have demonstrated that the capability is provided, but simply do not have data to pass through the code.

Status: **Initial Pass**

5.1.3.10 LVV-T1756 - Verify calculation of photometric repeatability in uzy filters

Version 1. Open *LW-T1756* test case in Jira.

Verify that the DM system has provided the code to calculate the RMS photometric repeatability of bright non-saturated unresolved point sources in the u, z, and y filters, and assess whether it meets the requirement that it shall be less than **PA1uzy = 7.5 millimagnitudes**.

Preconditions:

Execution status: **Initial Pass**

Final comment:

Note that this test is being executed using `'validate_drp'`, which will be deprecated and re-

placed with a new metrics framework once the Gen3 butler becomes available. Nonetheless, the test is valid, because the algorithm(s) to calculate the metrics are the same, and will be moved into the new framework.

Also note that because there are no sufficient u-band data for testing, we currently are only able to test this requirement on y- and z-band data. This is the reason the status is “Initial Pass” rather than “Pass.”

Detailed steps results:

Step	Step Details
1	<p>Description</p> <p>Identify a dataset containing at least one field in each of the u, z, and y filters with multiple overlapping visits.</p> <hr/> <p>Expected Result</p> <p>A dataset that has been ingested into a Butler repository.</p> <hr/> <p>Actual Result</p> <p>We will use the HSC “RC2” data that were reprocessed with the release candidate of Science Pipelines v20.0. These are located at ‘/datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349’ on the development machines.</p> <hr/> <p>Status: Pass</p>
2	<p>Description</p> <p>Execute ‘validate_drp’ on a repository containing precursor data. Identify the path to the data, which we will call ‘DATA/path’, then execute the following (with additional flags specified as needed):</p> <hr/> <p>Example Code</p> <pre>validateDrp.py 'DATA/path'</pre> <hr/> <p>Expected Result</p> <p>JSON files (and associated figures) containing the Measurements and any associated “extras.”</p> <hr/> <p>Actual Result</p>

To select the v20.0 release of the Science Pipelines, we entered the following:

```
setup -t v20_0_0 lsst_distrib
setup obs_subaru
```

An example of the call to execute 'matchedVisitMetrics' (which runs the metrics portion of 'validate_drp' on data that have already been processed):

```
matchedVisitMetrics.py /datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349 -output
/datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349/validateDrp/matchedVisitMetrics/9813/HSC-
Y -config doApplyExternalPhotoCalib=True doApplyExternalSkyWcs=True instrumentName=HSC
datasetName=HSC-RC2 -id tract=9813 visit=318^322^324^326^328^330^332^344^346^348^350^352^354^356^358^36
ccd=0..8^10..103
```

This process is included as part of the regular RC2 reprocessing. See the log for this and other runs of 'matchedVisitMetrics' in /datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349/qaLogs/matchedVisitMetrics/

Status: **Pass**

3

Description

Confirm that the metric PA1uzy 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 PA1uzy has been calculated.

Actual Result

By executing `"reportPerformance /datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349/qaLogs/matchedVisitMetrics/matchedVisitMetrics-9813-HSC-Y-252002.log"`, a report is generated summarizing all of the metrics that were measured by `'validate_drp'`. The relevant lines showing results for PA2 and PF1 from tract 9813 are:

PA1 – HSC-Y – 15.5 mmag
PA1 – HSC-Z – 12.9 mmag

Note that because the science pipelines are still in development, the actual resulting values for these metrics are not important. The requirement to be satisfied is that we have (a) provided the software to calculate them, and (b) provided a framework for comparing them to requirements. These are both satisfied.

This step is given a status of "Initial Pass" because there are no u-band data to test on. We have demonstrated that the capability is provided, but simply do not have data to pass through the code.

Status: **Initial Pass**

5.1.3.11 LW-T1757 - Verify calculation of photometric repeatability in gri filters

Version 1. Open *LW-T1757* test case in Jira.

Verify that the DM system has provided the code to calculate the RMS photometric repeatability of bright non-saturated unresolved point sources in the g, r, and i filters, and assess whether it meets the requirement that it shall be less than **PA1gri = 5.0 millimagnitudes**.

Preconditions:

Execution status: **Pass**

Final comment:

Note that this test is being executed using `'validate_drp'`, which will be deprecated and replaced with a new metrics framework once the Gen3 butler becomes available. Nonetheless, the test is valid, because the algorithm(s) to calculate the metrics are the same, and will be

moved into the new framework.

Detailed steps results:

Step	Step Details
1	<p>Description</p> <p>Identify a dataset containing at least one field in each of the g, r, and i filters with multiple overlapping visits.</p> <hr/> <p>Expected Result</p> <p>A dataset that has been ingested into a Butler repository.</p> <hr/> <p>Actual Result</p> <p>We will use the HSC "RC2" data that were reprocessed with the release candidate of Science Pipelines v20.0. These are located at '/datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349' on the development machines.</p> <hr/> <p>Status: Pass</p>
2	<p>Description</p> <p>Execute 'validate_drp' on a repository containing precursor data. Identify the path to the data, which we will call 'DATA/path', then execute the following (with additional flags specified as needed):</p> <hr/> <p>Example Code</p> <pre>validateDrp.py 'DATA/path'</pre> <hr/> <p>Expected Result</p> <p>JSON files (and associated figures) containing the Measurements and any associated "extras."</p> <hr/> <p>Actual Result</p>

To select the v20.0 release of the Science Pipelines, we entered the following:

```
setup -t v20_0_0 lsst_distrib
setup obs_subaru
```

An example of the call to execute 'matchedVisitMetrics' (which runs the metrics portion of 'validate_drp' on data that have already been processed):

```
matchedVisitMetrics.py /datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349 -output
/datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349/validateDrp/matchedVisitMetrics/9813/HSC-
G -config doApplyExternalPhotoCalib=True doApplyExternalSkyWcs=True instrumentName=HSC
datasetName=HSC-RC2 -id tract=9813 visit=11690^11692^11694^11696^11698^11700^11702^11704^11706^11708^11710
ccd=0..8^10..103
```

This process is included as part of the regular RC2 reprocessing. See the log for this and other runs of 'matchedVisitMetrics' in /datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349/qaLogs/matchedVisitMetrics/

Status: **Pass**

3

Description

Confirm that the metric PA1gri 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 PA1gri has been calculated.

Actual Result

By executing `"reportPerformance /datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349/qaLogs/matchedVisitMetrics/matchedVisitMetrics-9813-HSC-G-251998.log"`, a report is generated summarizing all of the metrics that were measured by `'validate_drp'`. The relevant lines showing results for PA2 and PF1 from tract 9813 are:

PA1 – HSC-G – 13.5 mmag
 PA1 – HSC-I – 14.9 mmag
 PA1 – HSC-R – 15.5 mmag

Note that because the science pipelines are still in development, the actual resulting values for these metrics are not important. The requirement to be satisfied is that we have (a) provided the software to calculate them, and (b) provided a framework for comparing them to requirements. These are both satisfied.

 Status: **Pass**

5.1.3.12 LVV-T125 - Verify implementation of Simulated Data

Version 1. Open *LW-T125* test case in Jira.

Verify that the DMS can inject simulated data into data products for testing.

Preconditions:

Execution status: **Pass**

Final comment:

This test demonstrates that the injected fake sources are visible, and that their measured magnitudes match the input values. However, future exploration should probably test the same for a variety of artificial sources (i.e., including galaxies).

Detailed steps results:

Step	Step Details
1	Description

Identify a dataset that has been (or can be readily) processed through single-frame processing and coaddition.

Expected Result

The 'calexp' and 'deepCoadd_calexp' images and their associated source catalogs are created.

Actual Result

We executed this test on the regularly reprocessed HSC-RC2 dataset. It is now part of routine processing to insert artificial sources into the data.

Status: **Pass**

2 Description

Roughly determine the coordinates of a bounding box that is contained within the images that were processed.

Expected Result

RA, Dec boundaries of a region in which to generate fake sources.

Actual Result

We extracted a single patch, and then selected only the fake sources within that RA, Dec box.

Status: **Pass**

3 Description

Generate a catalog in the correct format for 'insertFakes' to accept. The catalog should specify positions and magnitudes of stars (and optionally, parameters specifying galaxy shape, if galaxies are also being inserted).

Expected Result

An input catalog of fake source positions and magnitudes to be inserted into the images.

Actual Result

This step was performed prior to the test. We did, however, load the fakes catalog from the butler repository and examined them using:

```
fakeSourceCat = butler_mb.get('deepCoadd_fakeSourceCat', dataId = dataIdCoadd)
```

Status: **Pass**

4 Description

Execute 'insertFakes.py' on the repository, specifying the input catalog from the previous step.

Expected Result

A repository with images that have fake sources inserted.

Actual Result

We confirmed that the number of sources in the 'forced_src' photometry catalog *including fakes* is larger than the catalog without fakes.

Status: **Pass**

5 Description

Run 'multiBandDriver.py' on the repository, specifying the fake-source repository as the input.

Expected Result

'calexp' and coadd images containing the artificial sources and sources catalogs that contain their measurements along with the sources detected in the original run.

Actual Result

See the associated notebook, test_LVV-T125.ipynb, which shows comparisons of images with and without fake sources, and differences of images (to highlight the fake sources that are left after subtracting off the original image).

Status: **Pass**

6 Description

Confirm that the injected sources appear in the images and the catalogs.

Expected Result

Fake sources and their measured properties are recoverable.

Actual Result

By extracting the detected/measured sources that correspond to the input fakes, we compared the photometry and confirmed that the magnitudes of input fake sources are recovered.

Status: **Pass**

5.1.3.13 LVV-T36 - Verify implementation of Difference Exposures

Version **1**. Open *LVV-T36* test case in Jira.

Verify successful creation of a

1. PSF-matched template image for a given Processed Visit Image
2. Difference Exposure from each Processed Visit Image

Preconditions:

Execution status: **Pass**

Final comment:

See associated Jupyter notebooks 'test_LVV-T36.ipynb' and 'test_LVV-T36-HITS.ipynb'.

Detailed steps results:

Step	Step Details
1	<p>Description</p> <p>The 'path' that you will use depends on where you are running the science pipelines. Options:</p> <ul style="list-style-type: none"> • 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 <p>From the command line, execute the commands below in the example code:</p> <hr/> <p>Example Code</p> <pre>source 'path' setup lsst_distrib</pre> <hr/> <p>Expected Result</p> <p>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.</p> <p>To check versions in use, type:</p> <pre>eups list -s</pre> <hr/> <p>Actual Result</p> <p>The test was executed on the Science Platform Notebook Aspect with version 20.0.0 set up.</p> <hr/> <p>Status: Pass</p>
2	<p>Description</p> <p>Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.</p> <hr/> <p>Expected Result</p> <p>An output dataset including difference images and DIASource and DIAObject measurements.</p> <hr/>

Actual Result

These tests were executed using processed data from DECam/HiTS and HSC PDR2 (the COSMOS field - tract 9813). The processing with the Alert Production payload was executed prior to the start of this testing. Paths to the data are given in the accompanying Jupyter notebooks.

Status: **Pass**

3 Description

Verify that the expected data products have been produced, and that catalogs contain reasonable values for measured quantities of interest.

Expected Result

Actual Result

See the accompanying notebooks, titled 'test_LVV-T36.ipynb' and 'test_LVV-T36-HiTS.ipynb'. The expected images and catalogs are present, and were inspected to ensure that they are well-formed.

Status: **Pass**

4 Description

Demonstrate successful creation of a template image from HSC PDF and DECam HiTS data. Demonstrate successful creation of a Difference Exposure for at least 10 other images from survey, ideally at a range of airmass. In particular, HiTS has 2013A u-band data. While the Blanco 4-m does have an ADC, there are still some chromatic effects and we should demonstrate that we can successfully produce Difference Exposures and templates for different airmass bins.

Expected Result

Actual Result

This was done for 11 images from HSC at a range of airmasses, and images from DECam HiTS data. The difference exposures were shown to have zero mean flux, as expected, and visually inspected to confirm that they are well-formed.

Status: **Pass**

5.1.3.14 LVV-T39 - Verify implementation of Generate Photometric Zeropoint for Visit Image

Version 1. Open *LVV-T39* test case in Jira.

Verify that Processed Visit Image data products produced by the DRP and AP pipelines include the parameters of a model that relates the observed flux on the image to physical flux units.

Preconditions:

Execution status: **Pass**

Final comment:

See associated Jupyter notebook 'test_LVV-T39.ipynb'.

Detailed steps results:

Step	Step Details
1	<p>Description</p> <p>Identify a dataset with processed visit images in multiple filters.</p> <hr/> <p>Expected Result</p> <hr/> <p>Actual Result</p> <p>We will use HSC-RC2 data that was processed with the v20.0 science pipelines. The output repo is tagged with the Jira ticket number "DM-25349":</p> <pre>rc2_repo = 'datasets/hsc/repo/rerun/RC/v20_0_0_rc1/DM-25349-sfm'</pre> <p>We also repeat each test on data from the difference image repo at the following path on the 'lsst-devl' servers:</p> <pre>diffim_repo = '/project/mrawls/cosmos/rerun/ap_pipe-g-202007'</pre> <hr/> <p>Status: Pass</p>
2	<p>Description</p> <p>Identify the path to the data repository, which we will refer to as 'DATA/path', then execute the following:</p> <hr/> <p>Example Code</p> <pre>import lsst.daf.persistence as dafPersist butler = dafPersist.Butler(inputs='DATA/path')</pre> <hr/> <p>Expected Result</p> <p>Butler repo available for reading.</p> <hr/> <p>Actual Result</p> <pre>butler = dafPersist.Butler(rc2_repo)</pre> <p>The data is available for reading.</p> <hr/>

Status: **Pass**

3 Description

Extract the photometric zeropoint from the source catalog associated with a visit image. Repeat this for all available filters, and confirm that the zeropoint has been set, and has a reasonable value.

 Expected Result

A zeropoint that enables one to convert the measured fluxes to magnitudes.

 Actual Result

In the notebook associated with this test - 'test_LWV-T39.ipynb' - we extract the zeropoints, fluxes, and magnitudes, and confirm that the conversion between fluxes and magnitudes is enabled, and that it round-trips successfully.

 Status: **Pass**

4 Description

Extract fluxes for some sources, and convert them to magnitudes. Confirm that the distribution spans a reasonable range.

 Expected Result

In most cases, well-measured magnitudes (i.e., for high S/N measurements) should be between 12 to 28 for all bands.

 Actual Result

We extracted fluxes for 500 sources at random, and confirmed that converting their fluxes to magnitudes yields reasonable values for all stars.

 Status: **Pass**

5.1.3.15 LWV-T38 - Verify implementation of Processed Visit Images

Version 1. Open *LWV-T38* test case in Jira.

Verify that the DMS

1. Successfully produces Processed Visit Images, where the instrument signature has been removed.
2. Successfully combines images obtained during a standard visit.

Preconditions:

Execution status: **Pass**

Final comment:

In this test, we have used precursor data from DECam for verification, but any suitable precursor data would be sufficient. When ComCam data become available, this test should be executed on them.

Detailed steps results:

Step	Step Details
1	<p>Description</p> <p>Identify suitable precursor datasets containing unprocessed raw images.</p> <hr/> <p>Expected Result</p> <hr/> <p>Actual Result</p> <p>The test was executed on the Science Platform Notebook Aspect with version 20.0.0 set up.</p> <p>These tests were executed using processed data from DECam/HiTS. The processing with the Alert Production payload was executed prior to the start of this testing. The path to the data are given in the accompanying Jupyter notebook entitled 'test-T38_42.ipynb'.</p> <hr/> <p>Status: Pass</p>
2	<p>Description</p> <p>Identify the path to the data repository, which we will refer to as 'DATA/path', then execute the following:</p> <hr/> <p>Example Code</p> <pre>import lsst.daf.persistence as dafPersist butler = dafPersist.Butler(inputs='DATA/path')</pre> <hr/> <p>Expected Result</p> <p>Butler repo available for reading.</p> <hr/> <p>Actual Result</p> <p>The path where Meredith regularly re-runs HiTS data through diff imaging: diffim_repo = '/project/mrawls/hits2015/rerun/cw_2020_06'</p> <pre>butler = dafPersist.Butler(diffim_repo)</pre> <hr/> <p>Status: Pass</p>
3	<p>Description</p>

Run the Prompt Processing payload on these data. Verify that Processed Visit Images are generated at correct size and with significant instrumental artifacts removed.

Expected Result

Raw precursor dataset images have been processed into Processed Visit Images, with instrumental artifacts corrected.

Actual Result

In the associated notebook, we have examined a raw image and a processed visit image (also known as a "calexp") side-by-side, and confirmed that (a) the calexp is well-formed, and that (b) the instrumental artifacts have been corrected.

Status: **Pass**

5.1.3.16 LVV-T42 - Verify implementation of Processed Visit Image Content

Version 1. Open *LVV-T42* test case in Jira.

Verify that Processed Visit Images produced by the DRP and AP pipelines include the observed data, a mask array, a variance array, a PSF model, and a WCS model.

Preconditions:

Execution status: **Pass**

Final comment:

In this test, we have used precursor data from DECam for verification, but any suitable precursor data would be sufficient. When ComCam data become available, this test should be executed on them.

Detailed steps results:

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

```
import lsst.daf.persistence as dafPersist
butler = dafPersist.Butler(inputs='DATA/path')
```

Expected Result

Butler repo available for reading.

Actual Result

The test was executed on the Science Platform Notebook Aspect with version 20.0.0 set up.

These tests were executed using processed data from DECam/HiTS. The processing with the Alert Production payload was executed prior to the start of this testing. The path to the data are given in the accompanying Jupyter notebook entitled 'test-T38_42.ipynb'.

Status: **Pass**

2 Description

Ingest the data from an appropriate processed dataset.

Expected Result

Actual Result

The path where Meredith regularly re-runs HiTS data through diff imaging:
diffim_repo = '/project/mrawls/hits2015/rerun/cw_2020_06'

```
butler = dafPersist.Butler(diffim_repo)
```

Status: **Pass**

3 Description

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

Expected Result

Actual Result

See the relevant code in the associated notebook. All of these objects were selected as "wcs", "calexp.image", "psf", and "src."

Status: **Pass**

4 Description

Inspect the calexp image to ensure that

1. A well-formed image is present,
2. The variance plane is present and well-behaved,
3. Mask planes are present and contain information about defects.

Expected Result

An astronomical image with mask and variance planes. This can be readily visualized using Firefly, which displays mask planes by default.

Actual Result

The images of both the calexp and its variance plane are well-formed. We confirmed the presence of a mask plane and the meanings of its bit values using afwDisplay within the Jupyter notebook.

Status: **Pass**

5 Description

Plot images of the PSF model at various points, and verify that the PSF differs with position.

Expected Result

A "star-like" image of the PSF evaluated at various positions. The PSF should vary slightly with position (this could be readily visualized by taking a difference of PSFs at two positions).

Actual Result

We extracted an image of the PSF at two random positions and noted that they were well-formed. We then took the difference of the two PSFs, which showed clear differences between them. We thus confirmed that the PSF differs as a function of position.

Status: **Pass**

6 Description

Starting from the XY pixel coordinates of the sources, apply the WCS to obtain RA, Dec coordinates. Plot these positions and confirm that they match the expected values from the WCS object.

Expected Result

RA, Dec coordinates that are returned should be near the central position of the visit coordinate as given in either the calexp metadata or the WCS.

Actual Result

This was executed for all sources on a CCD in a processed visit image. The XY positions converted to (RA, Dec) were confirmed to match the expected position as given by the WCS.

Status: **Pass**

7 Description

Repeat steps 2-6, but now with difference images created by the Alert Production pipeline (for example, in the 'ap_verify' test data processing).

Expected Result

Actual Result

These steps were repeated for a difference image. Results of this successful test can be seen in the associated notebook.

Status: **Pass****5.1.3.17 LVV-T1264 - Verify implementation of archiving camera test data**

Version **1**. Open *LVV-T1264* test case in Jira.

Verify that a subset of camera test data has been ingested into Butler repos and is available through standard data access tools.

Preconditions:

Execution status: **Pass**

Final comment:

See associated Jupyter notebook in 'test_LVV-T1264.ipynb'. This test used data products created and ingested as part of the Operations Rehearsal detailed in ? : "LDM-503-11a: ComCam OPS Readiness Test Plan and Report".

Detailed steps results:

Step	Step Details
1	Description Obtain some data on a camera test stand. ----- Expected Result ----- Actual Result ComCam data were obtained as part of the July 2020 Operations Rehearsal (see the report in DMTR-231: "LDM-503-11a: ComCam OPS Readiness Test Plan and Report"). ----- Status: Pass
2	Description

Wait a sufficient amount of time, then confirm that automatic transfer/ingest of the data has occurred, and a repo is available at NCSA.

Expected Result

The data is present at NCSA in non-empty repos.

Actual Result

This was verified as part of LVV-T1934, which was a test to confirm that the data were transferred and ingested successfully from the Base Facility to NCSA.

Status: **Pass**

3 Description

Identify the relevant Butler repo of ingested camera test stand data.

Expected Result

Actual Result

We used the data from July 29, 2020, at the following path:

```
repo_path = '/project/shared/comCam/rerun/CALSET_20200729_1847_v2/'
```

Status: **Pass**

4 Description

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

Example Code

```
import lsst.daf.persistence as dafPersist
butler = dafPersist.Butler(inputs='DATA/path')
```

Expected Result

Butler repo available for reading.

Actual Result

```
repo_path = '/project/shared/comCam/rerun/CALSET_20200729_1847_v2/'
butler = dafPersist.Butler(repo_path)
```

Status: **Pass**

5 Description

Read various repo data products with the Butler, and confirm that they contain the expected data.

Expected Result

Camera test stand data that is well-formed.

Actual Result

In the associated Jupyter notebook entitled 'test_LVV-T1264.ipynb', we loaded a bias and a flat image from the above repo. Each of these was displayed to confirm that it was a well-formed image, and histograms of the count values were plotted to show that the counts were as expected (i.e., that the bias values are centered on roughly 0.0, and the flat values on roughly 1.0).

Status: **Pass**

5.1.3.18 LVV-T146 - Verify implementation of DMS Initialization Component

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

Demonstrate that the DMS can be initialized in a safe state that will not allow data corruption/loss.

Preconditions:

Execution status: **Not Executed**

Final comment:

After some discussion, it was decided to defer this test to a later campaign. This is due in part to the lack of on-site personnel at some sites to execute the test. More importantly, however, is the need to understand whether this should be divided into multiple tests that address hardware at each of the various sites that are part of the DMS, and whether those systems should all be taken down simultaneously, or tested separately. We hope to execute this test as part of the next Operations Rehearsal.

Detailed steps results:

Step	Step Details
1	Description
	Power-cycle all of the DM systems at each Facility.
	Expected Result
	Restart of all DM systems.
	Actual Result

Status: **Not Executed**

2 Description

Observe each system and ensure that it has recovered in a properly initialized state.

Expected Result

Systems are all active and initialized for their designated purpose.

Actual Result

Status: **Not Executed**

5.1.3.19 LVV-T144 - Verify implementation of Task Specification

Version 1. Open *LVV-T144* test case in Jira.

Verify that the DMS provides the ability to define a new or modified pipeline task without recompilation.

Preconditions:

Execution status: **Not Executed**

Final comment:

The Gen3 Butler rollout was delayed, so this test was not executed. Because many aspects of task specification and execution are in active development, we will defer this test to a later campaign that takes place after Gen3 has stabilized.

Detailed steps results:

Step	Step Details
1	Description
	Inspect software architecture. Verify that there exist Tasks that can be run and configured without re-compilation.
	----- Expected Result

Confirmation that the software architecture has allowed for reconfiguring and running Tasks without recompilation.

Actual Result

Status: **Not Executed**

2 Description

Verify that an example science algorithm can be run through one of these Tasks. Three examples from different areas: source measurement, image subtraction, and photometric-redshift estimation.

Expected Result

Successful Task execution with different configurations, including confirmation that the outputs are different from tasks with altered configurations.

Actual Result

Status: **Not Executed**

5.1.3.20 LVV-T145 - Verify implementation of Task Configuration

Version 1. Open *LVV-T145* test case in Jira.

Verify that the DMS software provides configuration control to define, override, and verify the configuration for a DMS Task.

Preconditions:

Execution status: **Not Executed**

Final comment:

The Gen3 Butler rollout was delayed, so this test was not executed. Because many aspects of task configuration, specification, and execution are in active development, we will defer this test to a later campaign that takes place after Gen3 has stabilized.

Detailed steps results:

Step	Step Details
1	<p>Description</p> <p>Inspect software design to verify that one can define the configuration for a Task.</p> <p>-----</p> <p>Expected Result</p> <p>-----</p> <p>Actual Result</p> <p>-----</p> <p>Status: Not Executed</p>
2	<p>Description</p> <p>Run a Task with a known invalid configuration. Verify that the error is caught before the science algorithm executes.</p> <p>-----</p> <p>Expected Result</p> <p>-----</p> <p>Actual Result</p> <p>-----</p> <p>Status: Not Executed</p>
3	<p>Description</p> <p>Run a simple task with two different configurations that make a material difference for a Task. E.g., specify a different source detection threshold. Verify that the configuration is different between the two runs through difference in recorded provenance and in results.</p> <p>-----</p> <p>Expected Result</p> <p>-----</p> <p>Actual Result</p> <p>-----</p> <p>Status: Not Executed</p>

5.1.3.21 LVV-T45 - Verify implementation of Prompt Processing Data Quality Report Definition

Version 1. Open *LW-T45* test case in Jira.

Verify that the DMS produces a Prompt Processing Data Quality Report. Specifically check absolute value and temporal variation of

1. Photometric zeropoint
2. Sky brightness
3. Seeing
4. PSF
5. Detection efficiency

Preconditions:

Execution status: **Not Executed**

Final comment:

This test was included because it was anticipated that the infrastructure to produce the Prompt Processing Data Quality Report would be available by the time of this Test Campaign. Because the report has not yet been defined, nor the automated pipeline to ingest and process data, then generate the report, we defer this test to a later campaign.

Detailed steps results:

Step	Step Details
1	<p>Description</p> <p>Ingest raw data from L1 Test Stand DAQ.</p> <hr/> <p>Expected Result</p> <hr/> <p>Actual Result</p> <hr/> <p>Status: Not Executed</p>
2	<p>Description</p> <p>Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.</p> <hr/> <p>Expected Result</p> <p>An output dataset including difference images and DIASource and DIAObject measurements.</p> <hr/> <p>Actual Result</p> <hr/>

Status: **Not Executed**

-
- 3 Description
Verify that the expected data products have been produced, and that catalogs contain reasonable values for measured quantities of interest.

Expected Result

Actual Result

Status: **Not Executed**

-
- 4 Description
Load the Prompt Processing QC reports, and observe that a dynamically updated Data Quality Report has become available at the relevant UI.

Expected Result

A Prompt Processing QC report is available via a UI, and contains information about the photometric zeropoint, sky brightness, seeing, PSF, and detection efficiency, and possibly other relevant quantities.

Actual Result

Status: **Not Executed**

-
- 5 Description
Check that a static report is created and archived in a readily-accessible location.

Expected Result

Persistence of a static QC report in an accessible location, containing the same information as in the report from Step 3.

Actual Result

Status: **Not Executed**

5.1.3.22 LVV-T46 - Verify implementation of Prompt Processing Performance Report Definition

Version 1. Open *LVV-T46* test case in Jira.

Verify that the DMS produces a Prompt Processing Performance Report. Specifically check

that the number of observations that describe each of the following:

1. Successfully processed, recoverable failures, unrecoverable failures.
2. Archived
3. Result in science.

This is testing more the processing rather than the observatory system.

Preconditions:

Execution status: **Not Executed**

Final comment:

This test was included because it was anticipated that the infrastructure to produce the Prompt Processing Performance Report would be available by the time of this Test Campaign. Because the report has not yet been defined, nor the automated pipeline to ingest and process data, then generate the report, we defer this test to a later campaign.

Detailed steps results:

Step	Step Details
1	<p>Description</p> <p>Execute single-day operations rehearsal, observe report</p> <hr style="border-top: 1px dashed black;"/> <p>Expected Result</p> <hr style="border-top: 1px dashed black;"/> <p>Actual Result</p> <hr style="border-top: 1px dashed black;"/> <p>Status: Not Executed</p>

A Acronyms used in this document

Acronym	Description
ADC	atmospheric dispersion corrector
ADQL	Astronomical Data Query Language
AP	Alert Production
CBP	Collimated Beam Projector
CCD	Charge-Coupled Device
CSV	Comma Separated Values
ComCam	The commissioning camera is a single-raft, 9-CCD camera that will be installed in LSST during commissioning, before the final camera is ready.
DAQ	Data Acquisition System
DM	Data Management
DMS	Data Management Subsystem
DMS-REQ	Data Management System Requirements prefix
DMSR	DM System Requirements; LSE-61
DMTR	DM Test Report
DRP	Data Release Production
FITS	Flexible Image Transport System
HSC	Hyper Suprime-Cam
IPAC	No longer an acronym; science and data center at Caltech
ISR	Instrument Signal Removal
JSON	JavaScript Object Notation
L1	Lens 1
LDF	LSST Data Facility
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)
NCSA	National Center for Supercomputing Applications
OPS	Operations
PDF	Portable Document Format
PDR2	Public Data Release 2 (HSC)
PMCS	Project Management Controls System

PSF	Point Spread Function
PVI	Processed Visit Image
QC	Quality Control
RA	Right Ascension
RC	Release Candidate
RMS	Root-Mean-Square
TOPCAT	Tool for OPerations on Catalogues And Tables
UI	User Interface
VE	vendor estimate
VPN	virtual private network
WCS	World Coordinate System
WISE	Wide-field Survey Explorer
arcsec	arcsecond second of arc (unit of angle)
deg	degree; unit of angle

B Traceability

Test Case	VE Key	VE Summary
LWV-T28	LWV-178	DMS-REQ-0347-V-01: Measurements in catalogs
LWV-T36	LWV-7	DMS-REQ-0010-V-01: Difference Exposures
LWV-T38	LWV-29	DMS-REQ-0069-V-01: Processed Visit Images
LWV-T39	LWV-12	DMS-REQ-0029-V-01: Generate Photometric Zeropoint for Visit Image
LWV-T40	LWV-13	DMS-REQ-0030-V-01: Absolute accuracy of WCS
LWV-T42	LWV-31	DMS-REQ-0072-V-01: Processed Visit Image Content
LWV-T45	LWV-39	DMS-REQ-0097-V-01: Level 1 Data Quality Report Definition
LWV-T46	LWV-41	DMS-REQ-0099-V-01: Level 1 Performance Report Definition
LWV-T125	LWV-6	DMS-REQ-0009-V-01: Simulated Data
LWV-T133	LWV-182	DMS-REQ-0351-V-01: Provide Beam Projector Coordinate Calculation Software
LWV-T144	LWV-136	DMS-REQ-0305-V-01: Task Specification
LWV-T145	LWV-137	DMS-REQ-0306-V-01: Task Configuration
LWV-T146	LWV-128	DMS-REQ-0297-V-01: DMS Initialization Component
LWV-T151	LWV-35	DMS-REQ-0078-V-01: Catalog Export Formats
LWV-T1232	LWV-35	DMS-REQ-0078-V-01: Catalog Export Formats
LWV-T1264	LWV-9637	DMS-REQ-0372-V-01: Archiving Camera Test Data
LWV-T1756	LWV-3401	DMS-REQ-0359-V-01: RMS photometric repeatability in uzy
LWV-T1757	LWV-9759	DMS-REQ-0359-V-10: RMS photometric repeatability in gri
LWV-T1758	LWV-9758	DMS-REQ-0359-V-09: Repeatability outlier limit in uzy
	LWV-9752	DMS-REQ-0359-V-03: Max fraction of outliers among non-saturated sources
LWV-T1759	LWV-9752	DMS-REQ-0359-V-03: Max fraction of outliers among non-saturated sources

	LWV-9754	DMS-REQ-0359-V-05: Repeatability outlier limit in gri
LWV-T1946	LWV-178	DMS-REQ-0347-V-01: Measurements in catalogs
LWV-T1947	LWV-178	DMS-REQ-0347-V-01: Measurements in catalogs

Draft