LARGE SYNOPTIC SURVEY TELESCOPE

Large Synoptic Survey Telescope (LSST)
Data Management

LVV-P65 Fall 2019 Pipelines Release Acceptance Test Campaign Test Plan and Report

Jeffrey Carlin

DMTR-201

Latest Revision: 2020-03-01

Abstract

This is the test plan and report for LVV-P65 (Fall 2019 Pipelines Release Acceptance Test Campaign), an LSST level 2 milestone pertaining to the Data Management Subsystem.

Change Record

Version	Date	Description	Owner name
	2019-12-03	First Draft	Jeff Carlin
1.0	2020-01-17	Document approved after review in DM-22457	Jeff Carlin
2.0	2020-03-02	Test Campaign finalized. Approved in DM-22456	Jeff Carlin

Document curator: Jeff Carlin
Document source location: `https://github.com/lsst-dm/DMTR-201`
Version from source repository: 2f5f77b

Contents

LVV-P65 Fall 2019 Pipelines Release Acceptance Test Campaign Test Plan and Report

1 Introduction

1.1 Objectives

This Acceptance Test campaign aims to verify a small number of DMSR (LSE-61) requirements related to the LSST Science Pipelines. It will be executed in conjunction with the release of Science Pipelines Version 19.0.0, 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 19.0.0 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 w_2019_46 weekly pipelines release (DM-22223) that was used to create v19 of the

Science Pipelines.

1.3 Document Overview

This document was generated from Jira, obtaining the relevant information from the LVV-P65 Jira Test Plan and related Test Cycles (LVV-C115).

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 describes the configuration used for this test. Section 3 describes the necessary roles and lists the individuals assigned to them. including all relevant information that fully describes the test campaign.

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-P65 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://ls.st/LSE-61

[2] **[LDM-639]**, Guy, L., 2018, *DM Acceptance Test Specification*, LDM-639, URL https://ls.st/LDM-639

[3] **[LDM-503]**, O'Mullane, W., Swinbank, J., Jurić, M., Economou, F., 2018, *Data Management Test Plan*, LDM-503, URL https://ls.st/LDM-503

2 Test Configuration

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 19.0.0 of the Pipelines, whose release is DM Milestone LDM-503-11b (Test Plan located here).

2.3 Entry Criteria

Release and availability of Science Pipelines version 19.

3 Personnel

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

Test Plan (LVV-P65) owner:	**Jeffrey Carlin**		
LVV-C115 owner:	**Jeffrey Carlin**		
Test Case	**Assigned to**	**Executed by**	**Additional Test Personnel**
LVV-T41	Jim Bosch	Jeffrey Carlin	
LVV-T62	Jim Bosch	Jeffrey Carlin	
LVV-T40	Jim Bosch	Jeffrey Carlin	
LVV-T1240	Jim Bosch	Jeffrey Carlin	
LVV-T28	Colin Slater	Jeffrey Carlin	
LVV-T43	Jim Bosch	Jeffrey Carlin	
LVV-T132	Robert Gruendl	Jeffrey Carlin	
LVV-T1745	Jeffrey Carlin	Jeffrey Carlin	
LVV-T1746	Jeffrey Carlin	Jeffrey Carlin	
LVV-T1747	Jeffrey Carlin	Jeffrey Carlin	
LVV-T1748	Jeffrey Carlin	Jeffrey Carlin	
LVV-T1749	Jeffrey Carlin	Jeffrey Carlin	
LVV-T1750	Jeffrey Carlin	Jeffrey Carlin	
LVV-T1751	Jeffrey Carlin	Jeffrey Carlin	
LVV-T1752	Jeffrey Carlin	Jeffrey Carlin	
LVV-T1753	Jeffrey Carlin	Jeffrey Carlin	
LVV-T1754	Jeffrey Carlin	Jeffrey Carlin	
LVV-T1755	Jeffrey Carlin	Jeffrey Carlin	
LVV-T1758	Jeffrey Carlin	Jeffrey Carlin	
LVV-T1759	Jeffrey Carlin	Jeffrey Carlin	

4 Test Campaign Overview

4.1 Summary

Test Plan **LVV-P65: Fall 2019 Pipelines Release Acceptance Test Campaign**			Approved
Test Cycle **LVV-C115: Fall 2019 Pipelines Release Acceptance Test Campaign**			Done
test case	**status**	**comment**	**issues**
LVV-T41	Pass	Test executed on lsst-lsp-stable, using Release v19.0.0, with a Large container.	
LVV-T62	Pass	Test executed on lsst-lsp-stable, using Release v19.0.0, with a Large container.	
LVV-T40	Pass w/ Deviation	Test executed on lsst-lsp-stable, using Release v19.0.0, with a Large container.	DM-23412
LVV-T1240	Pass	Test executed on lsst-lsp-stable, using Release v19.0.0, with a Large container.	
LVV-T28	Pass w/ Deviation	Test executed on lsst-lsp-stable, using Release v19.0.0, with a Large container.	DM-23413 DM-23415
LVV-T43	Pass	Test executed on lsst-lsp-stable, using Release v19.0.0, with a Large container.	
LVV-T132	Pass		
LVV-T1745	Pass	Tests executed on lsst-dev, using Release v19.0.0.	DM-23416
LVV-T1746	Pass	Tests executed on lsst-dev, using Release v19.0.0.	
LVV-T1747	Pass	Tests executed on lsst-dev, using Release v19.0.0.	
LVV-T1748	Pass	Tests executed on lsst-dev, using Release v19.0.0.	
LVV-T1749	Pass	Tests executed on lsst-dev, using Release v19.0.0.	
LVV-T1750	Pass	Tests executed on lsst-dev, using Release v19.0.0.	
LVV-T1751	Pass	Tests executed on lsst-dev, using Release v19.0.0.	
LVV-T1752	Pass	Tests executed on lsst-dev, using Release v19.0.0.	
LVV-T1753	Pass	Tests executed on lsst-dev, using Release v19.0.0.	
LVV-T1754	Pass	Tests executed on lsst-dev, using Release v19.0.0.	
LVV-T1755	Pass	Tests executed on lsst-dev, using Release v19.0.0.	

LVV-T1758	Fail	Tests executed on lsst-dev, using Release v19.0.0.	DM-23417
LVV-T1759	Pass	Tests executed on lsst-dev, using Release v19.0.0.	

Table 2: Test Campaign Summary

4.2 Overall Assessment

This campaign successfully demonstrated that the DM system is meeting many of its requirements, primarily using the HSC-RC2 dataset. The requirements that were verified include some related to basic image processing and data products (e.g., PSF and background modeling, astrometric solutions and WCS), 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:

- 'validate_drp' – while powerful and essential to this test campaign – can be challenging to work with, and its architecture may make it difficult to implement some of the more complicated metrics in the future.
- It may be difficult to scale up 'validate_drp' to run on larger datasets. It took at least two days of watchful processing to ensure that all 15 tract/filter combinations from RC2 completed their validate_drp runs. Part of this (and the first comment above) may be related to the tester's inexperience with validate_drp, but at least in part it is due to the memory-intensive long runtimes of validate_drp on not-so-small datasets.
- Many requirements specify that they must be met for PVIs, coadds, and/or difference images (i.e., Data Release (DRP) *and* Alert Production (AP) data products), but currently difference imaging is not a robustly functioning capability. When these tests are executed in the future, the AP components should be included in addition to DRP.
- Test Cases for requirements that involve comparison to external datasets (e.g., Gaia DR2 for astrometry) should more clearly specify quality cuts to extract a robust and relevant

comparison dataset. The presence of outliers and/or other “bad data” in cross-matched catalogs sometimes compromises the results for these external comparisons.

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 PSF 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).

As noted above, it is possible that ‘validate_drp‘ will not scale well to larger datasets, and that its architecture will make implementation of some metrics challenging. To address this, we are developing an improved framework (currently dubbed the ‘SV-distiller‘) that is more flexible and modular.

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.

5 Detailed Test Results

5.1 Test Cycle LVV-C115

Open test cycle *Fall 2019 Pipelines Release Acceptance Test Campaign* in Jira.

Fall 2019 Pipelines Release Acceptance Test Campaign
Status: Done

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 19.0.0, including its algorithms and resulting science data products.

5.1.2 Configuration

Not provided.

5.1.3 Test Cases in LVV-C115 Test Cycle

5.1.3.1 Test Case LVV-T41 - Verify implementation of Generate PSF for Visit Images

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:

Execution status: **Pass**

Final comment:
Test executed on lsst-lsp-stable, using Release v19.0.0, with a Large container.

Detailed steps results:

Step	Step Details
1	Description Identify a dataset with processed visit images in multiple filters. Expected Result Actual Result We used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223. The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'. Status: **Pass**

2	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 in a notebook named 'test_LVV-T41.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/w_2019_46/DM-22223'
butler = dafPersist.Butler(rc2_repo)
```

Further steps were needed to parse the filenames in the repository to extract the available tract/-patch/visit combinations. (These steps may not be necessary with future versions of the data butler.)

Status: **Pass**

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.

Actual Result

CCD/tract/patch/visit combinations were selected at random and the corresponding dataIds (datarefs) created. To extract the background, the following line was executed for each dataId:

calexp = butler.get('calexp', dataId = dataref)
psf= calexp.getPsf()

Random XY points were generated for each image, and the image of the PSF was extracted at those positions:

```
xpt = random.random()*xsize
ypt = random.random()*ysize
psfimage = psf.computeImage(geom.PointD(xpt, ypt))
```

These were displayed in the notebook.

In addition, images of stars that were used for creating the PSF and for photometric calibration were extracted. Their images were displayed alongside the scaled PSF model evaluated at the star's position, and the residuals remaining after subtracting the scaled PSF.

Additionally, a larger subset of dataIds was selected, from which we test whether *all* calexps have an associated PSF model. The result of this test, seen in the test notebook, is as follows:

```
All CCDs have an associated PSF model:  True
```

If any of the 100 randomly selected CCDs did not have an associated PSF model, this statement would not return True.

The executed notebook was saved in the lsst-dm/DMTR-201 repository as "executions/LVV-T41/test_LVV-T41.ipynb".

Status: **Pass**

5.1.3.2 Test Case LVV-T62 - Verify implementation of Provide PSF for Coadded Images

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.

Execution status: **Pass**

Final comment:
Test executed on lsst-lsp-stable, using Release v19.0.0, with a Large container.

Detailed steps results:

Step	Step Details
1	Description Identify a dataset with coadded images in multiple filters. Expected Result Multi-band data that has been processed through the coaddition stage. Actual Result We used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223. The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'. Status: **Pass**

2	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 in a notebook named 'test_LVV-T62.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/w_2019_46/DM-22223'
butler = dafPersist.Butler(rc2_repo)
```

Further steps were needed to parse the filenames in the repository to extract the available tract/patch combinations. (These steps may not be necessary with future versions of the data butler.)

Status: **Pass**

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.

Actual Result

Tract/patch combinations were selected at random and the corresponding dataIds (datarefs) created. To extract the background, the following line was executed for each dataId:

calexp = butler.get('deepCoadd_calexp', dataId = dataref)
psf= calexp.getPsf()

Random XY points were generated for each image, and the image of the PSF was extracted at those positions:

```
xpt = random.random()*xsize
ypt = random.random()*ysize
psfimage = psf.computeImage(geom.PointD(xpt, ypt))
```

These were displayed in the notebook.

In addition, images of random bright stars (between the 85th-90th percentile in flux) were extracted. Their images were displayed alongside the scaled PSF model evaluated at the star's position, and the residuals remaining after subtracting the scaled PSF.

Additionally, a larger subset of dataIds was selected, from which we test whether *all* deepCoadd_calexps have an associated PSF model. The result of this test, seen in the test notebook, is as follows:

```
All patches have an associated PSF model:  True
```

If any of the 100 randomly selected patches did not have an associated PSF model, this statement would not return True.

The executed notebook was saved in the lsst-dm/DMTR-201 repository as "executions/LVV-T62/test_LVV-T62.ipynb".

Status: **Pass**

5.1.3.3 Test Case LVV-T40 - Verify implementation of Generate WCS for Visit Images

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:

Execution status: **Pass w/ Deviation**

Final comment:
Test executed on lsst-lsp-stable, using Release v19.0.0, with a Large container.

Detailed steps results:

Step	Step Details
1	Description Identify an appropriate processed dataset for this test. Expected Result A dataset with Processed Visit Images available. Actual Result We used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223. The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'. Status: **Pass**

2	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 in a notebook named 'test_LVV-T40_T1240.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/w_2019_46/DM-22223'
butler = dafPersist.Butler(rc2_repo)
```

Further steps were needed to parse the filenames in the repository to extract the available tract/-patch/visit combinations. (These steps may not be necessary with future versions of the data butler.)

Status: **Pass**

---

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.

Actual Result

This was done 500 times, extracting the source list and WCS for 500 randomly-selected CCD/visit combinations from the repository.

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

It was confirmed that all CCDs selected had more than astrometricMinStandards=5 standards used in their WCS solutions. This was done using the following code to extract the number of astrometric standards for each image:

```
astrom_selection = np.where(src['calib_astrometry_used'] == True)
num_calib_astrom.append(np.size(astrom_selection))
```

In the end, we calculate the fraction of fields that met this requirement, using:

```
wcsFlagsPercent = (np.size(np.where(haswcs_flags))/np.size(haswcs_flags))*100.0*u.percent
```

The result (from the notebook) is:

```
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

Executed the following (for each CCD/visit) to create a list of RA, Dec coords from XY:

```
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])
```

This yields an array with RA, Dec coordinates.

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 <2" 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 < 2.0)
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

Figures shown in the notebook. Rather than histograms, we used comparisons of the various extracted parameters.

In addition to figures, we calculated the percentage of images that satisfied the requirement on **astrometricAccuracy**. This was 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:  81.89134808853119 %  --  False
```

Some brief exploration into reasons for a few images having large astrometric residuals is included in the notebook ('test_LVV-T40_T1240.ipynb').

Pending exploration of the small fraction of "failing" images, we grant this test a "Pass With Deviation."

Issues found executing this step:

DM-23412 Determine reason for large HSC-RC2 vs. Gaia astrometry residuals

Status: **Pass w/ Deviation**

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

Upon examination, we find that many images have only ~50 Gaia matches over the entire frame. This is too few stars to get statistically meaningful results from subregions, so we did not perform this portion of the test.

We denote this as "Pass With Deviation," as it is likely the requirement text will need to be changed to account for the paucity of Gaia sources in many fields.

Status: **Pass w/ Deviation**

5.1.3.4 Test Case LVV-T1240 - Verify implementation of minimum astrometric standards per CCD

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:

Execution status: **Pass**

Final comment:
Test executed on lsst-lsp-stable, using Release v19.0.0, with a Large container.

Detailed steps results:

Step	Step Details
1	Description Identify an appropriate processed dataset for this test. Expected Result A dataset with Processed Visit Images. Actual Result We used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223. The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'. Status: **Pass**

2	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 in a notebook named 'test_LVV-T40_T1240.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/w_2019_46/DM-22223'
butler = dafPersist.Butler(rc2_repo)
```

Further steps were needed to parse the filenames in the repository to extract the available tract/-patch/visit combinations. (These steps may not be necessary with future versions of the data butler.)

Status: **Pass**

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.

Actual Result

It was confirmed that all CCDs selected had more than astrometricMinStandards=5 standards used in their WCS solutions. This was done using the following code to extract the number of astrometric standards for each image:

```
astrom_selection = np.where(src['calib_astrometry_used'] == True)
num_calib_astrom.append(np.size(astrom_selection))
```

In the end, we calculate the fraction of fields that met this requirement, using:

```
wcsFlagsPercent = (np.size(np.where(haswcs_flags))/np.size(haswcs_flags))*100.0*u.percent
```

The result (from the notebook 'test_LVV-T40_T1240.ipynb') is:

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

Status: **Pass**

5.1.3.5 Test Case LVV-T28 - Verify implementation of Measurements in catalogs

Open *LVV-T28* test case in Jira.

Verify that source measurements in catalogs are in flux units.

Preconditions:

Execution status: **Pass w/ Deviation**

Final comment:
Test executed on lsst-lsp-stable, using Release v19.0.0, with a Large container.

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
	We used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223.
	
	The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'.
	Status: **Pass**

2	Description Identify and read appropriate processed precursor datasets with the Butler, including one containing single-visit images, one with coadds, and one with difference imaging.

Expected Result

Actual Result

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/w_2019_46/DM-22223'
butler = dafPersist.Butler(rc2_repo)
```

Arbitrary dataIds within this repo were chosen – one for a processed visit image, and one for a deep coadd image (note that no difference images were available, so we deem this step a "Pass With Deviation"). The specific code used:

```
filter = 'HSC-I'
# PVI:
ccd = 14
visit = 35890
# Coadd
tract = 9615
patch = '5,3'
```

```
dataIdPVI = {'filter':filter, 'visit':visit, 'ccd':ccd}
dataIdCoadd = {'tract':tract, 'filter':filter, 'patch':patch}
src = butler.get('src', dataId = dataIdPVI)
forced_src = butler.get('deepCoadd_forced_src', dataId = dataIdCoadd)
```

Issues found executing this step:

DM-23413 Update LVV-178 (Measurements in Catalogs) description
DM-23415 Create separate Test Cases for different catalog types from LVV-178

Status: **Pass w/ Deviation**

3 Description

Verify that each of the single-visit, coadd, and difference image catalogs provide 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 each of the source catalogs. 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 forced_src instFlux entries have units of counts:  True

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

Status: **Pass**

5.1.3.6 Test Case LVV-T43 - Verify implementation of Background Model Calculation

Open *LVV-T43* test case in Jira.

Verify that Processed Visit Images produced by the DRP and AP pipelines have had a model of the background subtracted, and that this model is persisted in a way that permits the background subtracted from any CCD to be retrieved along with the image for that CCD.

Preconditions:

Execution status: **Pass**

Final comment:
Test executed on lsst-lsp-stable, using Release v19.0.0, with a Large container.

Detailed steps results:

Step	Step Details
1	Description Identify a dataset with processed visit images in multiple filters. Expected Result

Actual Result

We used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223.

The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'.

Status: **Pass**

2 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 in a notebook named 'test_LVV-T43.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/w_2019_46/DM-22223'
butler = dafPersist.Butler(rc2_repo)

Further steps were needed to parse the filenames in the repository to extract the available tract/-patch/visit combinations. (These steps may not be necessary with future versions of the data butler.)

Status: **Pass**

3 Description

Display an image of the background model for a full CCD. Repeat this for all available filters, and confirm that the background is smoothly varying and defined over the full CCD.

Expected Result

Well-formed background covering the entire CCD for all CCDs in all filters.

Actual Result

CCD/tract/patch/visit combinations were selected at random and the corresponding dataIds (datarefs) created. To extract the background, the following line was executed for each dataId:

bb = butler.get('calexpBackground', dataId = datarefs[ii])

These were displayed alongside the final calexp image.

Additionally, a larger subset of dataIds was selected, from which we test whether (a) *all* calexps have an associated background model, and (b) the background model is well-formed and populated with finite values for all pixels. The result of this test, seen in the test notebook, is as follows:

```
All CCDs have an associated background.
```

If any of the 100 randomly selected CCDs had a malformed (or non-existent) background model, this statement would not return True.

The executed notebook was saved in the lsst-dm/DMTR-201 repository as "executions/LVV-T43/test_LVV-T43.ipynb".

Status: **Pass**

5.1.3.7 Test Case LVV-T132 - Verify implementation of Pre-cursor and Real Data

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:

Execution status: **Pass**

Final comment:

Detailed steps results:

Step	Step Details
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 Actual Result Because the outputs from CI jobs are not persisted, we instead use the same HSC RC2 data that we have used for many of the tests in this campaign. Specifically, we used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223. The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'. Status: **Pass**
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. Actual Result

The Test Cases that were executed on this dataset for this test campaign demonstrate that this requirement is satisfied. Specifically, the following data products from HSC RC2 were used (among others) in the Test Cases from this campaign:

LVV-T28: 'src' and 'deepCoadd_forced_src'
LVV-T62: 'deepCoadd_calexp' and associated PSF
LVV-T41: 'calexp' (calibrated PVI) and associated PSF
LVV-T40, LVV-T1240: 'calexp,' 'calexp_photoCalib,' and associated WCS
LVV-T43: 'calexpBackground'

Status: **Pass**

5.1.3.8 Test Case LVV-T1745 - Verify calculation of median relative astrometric measurement error on 20 arcminute scales

Open *LVV-T1745* test case in Jira.

Verify that the DM system has provided the code to calculate the median relative astrometric measurement error on 20 arcminute scales and assess whether it meets the requirement that it shall be no more than AM2 = 10 milliarcseconds.

Preconditions:

Execution status: **Pass**

Final comment:
Tests executed on lsst-dev, using Release v19.0.0.

Detailed steps results:

Step	Step Details
1	Description

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

Expected Result

A dataset that has been ingested into a Butler repository.

Actual Result

We used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223.

The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'.

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

On lsst-dev, the setup was done as follows:

```
source /software/lsstsw/stack/loadLSST.bash
source scl_source enable devtoolset-8
setup -t w_2019_46 lsst_distrib
setup obs_subaru
setup validate_drp
```

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

Batch jobs were sent to via slurm jobs to the batch processing cluster using a script that looked like (submitted via ‘sbatch SCRIPTNAME‘):

#!/bin/bash -l

#SBATCH -p normal
#SBATCH -N 1
#SBATCH –ntasks-per-node=1
#SBATCH -t 18:00:00
#SBATCH -J v9697
#SBATCH –output=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log
#SBATCH –error=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log

srun validateDrp.py /datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223 –configFile ‘cfg9697_all.yaml’ –outputPrefix=’tract9697’

...where the configuration file (cfg9697_all.yaml) contained something like this:

Configuration information for validate_drp to
build the list of data IDs to analyze
tracts: [9697]
visits: [6320,34338,34342,34362,34366,34382,34384,34400,34402,34412,34414,34422,34424,34448,34450,34464,34468
filter: [’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-
G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-I’,’HSC-I’,’HSC-
I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-
I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-
I’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-
R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-
Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-
Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-
Y’,’HSC-Y’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-
Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-
Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’]
ccd: [0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103]
instrument: ‘HSC’

Status: **Pass**

4 Description

Confirm that the metric AM2 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 AM2 has been calculated.

Actual Result

This was confirmed by

a. loading the JSON and printing a report from within a Jupyterlab notebook on the LSP (see attached rendering of notebook; the notebook is saved in as 'test_KPMs_validate_drp.ipynb' in the DMTR-201 github repository), and
b. dispatching the metric measurements to the SQuaSH chronograf dashboard (see attached screen shot).

NOTE: these attached illustrations are relevant to all Test Cases numbered LVV-T1745-1759 in this Test Cycle, but have only been attached to this execution of LVV-T1745.

The calculated value of AM2 for the example case (tract 9615, HSC-I filter) illustrated in the attached document is AM2 = 10.9 mas, which is near the design threshold of 10.0 mas.

A summary table (extracted from the Jupyter notebook) illustrating all evaluated metrics is seen below.

Status	Specification	Measurement	Test	Metric Tags	Spec. Tags
✅	validate_drp.AA1.design	11.2 marcsec	x <= 50.0 marcsec	astrometry	AA1, achromatic, design
❌	validate_drp.AB1.design	14.4 marcsec	x <= 10.0 marcsec	astrometry	AB1, achromatic, design
❌	validate_drp.AB2.design	20.9 marcsec	x <= 20.0 marcsec	astrometry	AB2, achromatic, design
❌	validate_drp.ABF1.design	41.1 %	x <= 10.0 %	astrometry	ABF1, achromatic, design
✅	validate_drp.AD1_design.srd	15.6 marcsec	x <= 20.0 marcsec	astrometry	AD1, achromatic, design
✅	validate_drp.AD2_design.srd	14.5 marcsec	x <= 20.0 marcsec	astrometry	AD2, achromatic, design
✅	validate_drp.AD3_design.srd	14.9 marcsec	x <= 30.0 marcsec	astrometry	AD3, achromatic, design
✅	validate_drp.AF1_design.srd	6.0 %	x <= 10.0 %	astrometry	AF1, achromatic, design
✅	validate_drp.AF2_design.srd	5.1 %	x <= 10.0 %	astrometry	AF2, achromatic, design
✅	validate_drp.AF3_design.srd	1.9 %	x <= 10.0 %	astrometry	AF3, achromatic, design
❌	validate_drp.AM1.design	11.1 marcsec	x <= 10.0 marcsec	astrometry	AM1, achromatic, design
❌	validate_drp.AM2.design	10.9 marcsec	x <= 10.0 marcsec	astrometry	AM2, achromatic, design
✅	validate_drp.AM3.design	10.3 marcsec	x <= 15.0 marcsec	astrometry	AM3, achromatic, design
❌	validate_drp.PA1.design_gri	21.1 mmag	x <= 5.0 mmag	photometry	PA1, chromatic, design
❌	validate_drp.PA1.design_uzy	21.1 mmag	x <= 7.5 mmag	photometry	PA1, chromatic, design
❌	validate_drp.PA2_design_gri.srd	39.2 mmag	x <= 15.0 mmag	photometry	PA2, chromatic, design
❌	validate_drp.PF1_design_gri.srd	47.8 %	x <= 10.0 %	photometry	PF1, chromatic, design
✅	validate_drp.TE1.design	0.0	x <= 2e-05	image_quality	TE1, achromatic, design
✅	validate_drp.TE2.design	0.0	x <= 5e-05	image_quality	TE2, achromatic, design

Issues found executing this step:

DM-23416 Fix units formatting error in verify.job report

Status: **Pass**

5.1.3.9 Test Case LVV-T1746 - Verify calculation of fraction of relative astrometric measurement error on 5 arcminute scales exceeding outlier limit

Open *LVV-T1746* test case in Jira.

Verify that the DM system has provided the code to calculate the maximum fraction of relative astrometric measurements on 5 arcminute scales that exceed the 5 arcminute outlier limit **AD1 = 20 milliarcseconds**, and assess whether it meets the requirement that it shall be less than **AF1 = 10 percent.**

Preconditions:

Execution status: **Pass**

Final comment:
Tests executed on lsst-dev, using Release v19.0.0.

Detailed steps results:

Step	Step Details
1	Description Identify a dataset containing at least one field with multiple overlapping visits. Expected Result A dataset that has been ingested into a Butler repository. Actual Result We used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223. The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'.

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

On lsst-dev, the setup was done as follows:

source /software/lsstsw/stack/loadLSST.bash
source scl_source enable devtoolset-8
setup -t w_2019_46 lsst_distrib
setup obs_subaru
setup validate_drp

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

Batch jobs were sent to via slurm jobs to the batch processing cluster using a script that looked like (submitted via 'sbatch SCRIPTNAME'):

```
#!/bin/bash -l

#SBATCH -p normal
#SBATCH -N 1
#SBATCH –ntasks-per-node=1
#SBATCH -t 18:00:00
#SBATCH -J v9697
#SBATCH –output=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log
#SBATCH –error=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log

srun validateDrp.py /datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223 –configFile 'cfg9697_all.yaml'
–outputPrefix='tract9697'
```

...where the configuration file (cfg9697_all.yaml) contained something like this:

Configuration information for validate_drp to
build the list of data IDs to analyze
tracts: [9697]
visits: [6320,34338,34342,34362,34366,34382,34384,34400,34402,34412,34414,34422,34424,34448,34450,34464,34468
filter: ['HSC-G','HSC-I','HSC-R','HSC-Y','HSC-Z','HSC-Z']
ccd: [0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103]
instrument: 'HSC'

	Status: **Pass**
4	Description Confirm that the metric AF1 has been calculated using the outlier limit AD1, and that its values are reasonable. Expected Result A JSON file (and/or a report generated from that JSON file) demonstrating that AF1 has been calculated (and used the limit AD1). Actual Result This was confirmed by a. loading the JSON and printing a report from within a Jupyterlab notebook on the LSP (see attached rendering of notebook; the notebook is saved in as 'test_KPMs_validate_drp.ipynb' in the DMTR-201 github repository), and b. dispatching the metric measurements to the SQuaSH chronograf dashboard (see attached screen shot). See the documents attached to LVV-T1745 for illustration of the results. The adopted value for AD1 was verified by inspecting the validate_drp source code. Note that AD1 is also calculated in validate_drp as the value of the residual at the 10th percentile (i.e., the design value of AF1). The calculated value of AF1 for the example case (tract 9615, HSC-I filter) illustrated in the attached document is AF1 = 6.0%, which is near the design threshold of 10.0%. Status: **Pass**

5.1.3.10 Test Case LVV-T1747 - Verify calculation of relative astrometric measurement error on 5 arcminute scales

Open *LVV-T1747* test case in Jira.

Verify that the DM system has provided the code to calculate the relative astrometric measurement error on 5 arcminute scales, and assess whether it meets the requirement that it shall be less than **AM1 = 10 milliarcseconds.**

Preconditions:

Execution status: **Pass**

Final comment:
Tests executed on lsst-dev, using Release v19.0.0.

Detailed steps results:

Step	Step Details
1	Description Identify a dataset containing at least one field with multiple overlapping visits. Expected Result A dataset that has been ingested into a Butler repository. Actual Result We used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223. The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'. 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

Status: **Not Executed**

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

Batch jobs were sent to via slurm jobs to the batch processing cluster using a script that looked like (submitted via ‘sbatch SCRIPTNAME‘):

#!/bin/bash -l

#SBATCH -p normal
#SBATCH -N 1
#SBATCH –ntasks-per-node=1
#SBATCH -t 18:00:00
#SBATCH -J v9697
#SBATCH –output=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log
#SBATCH –error=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log

srun validateDrp.py /datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223 –configFile ‘cfg9697_all.yaml’ –outputPrefix=’tract9697’

...where the configuration file (cfg9697_all.yaml) contained something like this:

Configuration information for validate_drp to
build the list of data IDs to analyze
tracts: [9697]
visits: [6320,34338,34342,34362,34366,34382,34384,34400,34402,34412,34414,34422,34424,34448,34450,34464,34468
filter: [’HSC-G’,’HSC-I’,’HSC-R’,’HSC-Y’,’HSC-Z’,’HSC-Z’]
ccd: [0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103]
instrument: ‘HSC’

	Status: **Pass**
4	Description Confirm that the metric AM1 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 AM1 has been calculated. Actual Result This was confirmed by a. loading the JSON and printing a report from within a Jupyterlab notebook on the LSP (see attached rendering of notebook; the notebook is saved in as 'test_KPMs_validate_drp.ipynb' in the DMTR-201 github repository), and b. dispatching the metric measurements to the SQuaSH chronograf dashboard (see attached screen shot). See the documents attached to LVV-T1745 for illustration of the results. The calculated value of AM1 for the example case (tract 9615, HSC-I filter) illustrated in the attached document is AM1 = 11.1 mas, which is near the design threshold of 10.0 mas. Status: **Pass**

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

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:

Execution status: **Pass**

Final comment:
Tests executed on lsst-dev, using Release v19.0.0.

Detailed steps results:

Step	Step Details
1	Description Identify a dataset containing at least one field with multiple overlapping visits. Expected Result A dataset that has been ingested into a Butler repository. Actual Result We used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223. The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'. 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

On lsst-dev, the setup was done as follows:

source /software/lsstsw/stack/loadLSST.bash
source scl_source enable devtoolset-8
setup -t w_2019_46 lsst_distrib
setup obs_subaru
setup validate_drp

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

Batch jobs were sent to via slurm jobs to the batch processing cluster using a script that looked like (submitted via ‘sbatch SCRIPTNAME‘):

```
#!/bin/bash -l

#SBATCH -p normal
#SBATCH -N 1
#SBATCH –ntasks-per-node=1
#SBATCH -t 18:00:00
#SBATCH -J v9697
#SBATCH –output=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log
#SBATCH –error=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log

srun validateDrp.py /datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223 –configFile ‘cfg9697_all.yaml’
–outputPrefix=’tract9697’
```

...where the configuration file (cfg9697_all.yaml) contained something like this:

```
# Configuration information for validate_drp to
# build the list of data IDs to analyze
tracts: [9697]
visits: [6320,34338,34342,34362,34366,34382,34384,34400,34402,34412,34414,34422,34424,34448,34450,34464,34468
filter:     [’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-
G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-I’,’HSC-I’,’HSC-
I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-
I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-
I’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-
R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-
Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-
Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-
Y’,’HSC-Y’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-
Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-
Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’]
ccd: [0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59,
60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88,
89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103]
instrument: ‘HSC’
```

	Status: **Pass**
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.
	Actual Result This was confirmed by a. loading the JSON and printing a report from within a Jupyterlab notebook on the LSP (see attached rendering of notebook; the notebook is saved in as 'test_KPMs_validate_drp.ipynb' in the DMTR-201 github repository), and b. dispatching the metric measurements to the SQuaSH chronograf dashboard (see attached screen shot). See the documents attached to LVV-T1745 for illustration of the results. The calculated value of AA1 for the example case (tract 9615, HSC-I filter) illustrated in the attached document is AA1 = 11.2 mas, which is well below the design threshold of 50.0 mas.
	Status: **Pass**

5.1.3.12 Test Case LVV-T1749 - Verify calculation of fraction of relative astrometric measurement error on 20 arcminute scales exceeding outlier limit

Open *LVV-T1749* test case in Jira.

Verify that the DM system has provided the code to calculate the maximum fraction of relative astrometric measurements on 20 arcminute scales that exceed the 20 arcminute outlier limit **AD2 = 20 milliarcseconds**, and assess whether it meets the requirement that it shall be less than **AF2 = 10 percent.**

Preconditions:

Execution status: **Pass**

Final comment:
Tests executed on lsst-dev, using Release v19.0.0.

Detailed steps results:

Step	Step Details
1	Description Identify a dataset containing at least one field with multiple overlapping visits. Expected Result A dataset that has been ingested into a Butler repository. Actual Result We used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223. The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'. 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

On lsst-dev, the setup was done as follows:

source /software/lsstsw/stack/loadLSST.bash
source scl_source enable devtoolset-8
setup -t w_2019_46 lsst_distrib
setup obs_subaru
setup validate_drp

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

Batch jobs were sent to via slurm jobs to the batch processing cluster using a script that looked like (submitted via 'sbatch SCRIPTNAME'):

#!/bin/bash -l

#SBATCH -p normal
#SBATCH -N 1
#SBATCH –ntasks-per-node=1
#SBATCH -t 18:00:00
#SBATCH -J v9697
#SBATCH –output=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log
#SBATCH –error=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log

srun validateDrp.py /datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223 –configFile 'cfg9697_all.yaml' –outputPrefix='tract9697'

...where the configuration file (cfg9697_all.yaml) contained something like this:

Configuration information for validate_drp to
build the list of data IDs to analyze
tracts: [9697]
visits: [6320,34338,34342,34362,34366,34382,34384,34400,34402,34412,34414,34422,34424,34448,34450,34464,34468
filter: ['HSC-G','HSC-I','HSC-R','HSC-Y','HSC-Z','HSC-Z']
ccd: [0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103]
instrument: 'HSC'

Status: **Pass**

4 Description

Confirm that the metric AF2 has been calculated using the outlier limit AD2, and that its values are reasonable.

Expected Result

A JSON file (and/or a report generated from that JSON file) demonstrating that AF2 has been calculated (and used the limit AD2).

Actual Result

This was confirmed by

a. loading the JSON and printing a report from within a Jupyterlab notebook on the LSP (see attached rendering of notebook; the notebook is saved in as 'test_KPMs_validate_drp.ipynb' in the DMTR-201 github repository), and
b. dispatching the metric measurements to the SQuaSH chronograf dashboard (see attached screen shot).

See the documents attached to LVV-T1745 for illustration of the results.

The adopted value for AD2 was verified by inspecting the validate_drp source code. Note that AD2 is also calculated in validate_drp as the value of the residual at the 10th percentile (i.e., the design value of AF2).

The calculated value of AF2 for the example case (tract 9615, HSC-I filter) illustrated in the attached document is AF2 = 5.1%, which is near the design threshold of 10.0%.

Status: **Pass**

5.1.3.13 Test Case LVV-T1750 - Verify calculation of separations relative to r-band exceeding color difference outlier limit

Open *LVV-T1750* test case in Jira.

Verify that the DM system has provided the code to calculate the separations measured relative to the r-band that exceed the color difference outlier limit **AB2 = 20 milliarcseconds**, and assess whether it meets the requirement that it shall be less than **ABF1 = 10 percent.**

Preconditions:

Execution status: **Pass**

Final comment:
Tests executed on lsst-dev, using Release v19.0.0.

Detailed steps results:

Step	Step Details
1	Description Identify a dataset containing at least one field with multiple overlapping visits, and including at least one visit in r-band. Expected Result A dataset that has been ingested into a Butler repository. Actual Result We used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223. The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'. 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

On lsst-dev, the setup was done as follows:

source /software/lsstsw/stack/loadLSST.bash
source scl_source enable devtoolset-8
setup -t w_2019_46 lsst_distrib
setup obs_subaru
setup validate_drp

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

Batch jobs were sent to via slurm jobs to the batch processing cluster using a script that looked like (submitted via ‘sbatch SCRIPTNAME‘):

```
#!/bin/bash -l

#SBATCH -p normal
#SBATCH -N 1
#SBATCH –ntasks-per-node=1
#SBATCH -t 18:00:00
#SBATCH -J v9697
#SBATCH –output=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log
#SBATCH –error=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log

srun validateDrp.py /datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223 –configFile ‘cfg9697_all.yaml’
–outputPrefix=’tract9697’
```

...where the configuration file (cfg9697_all.yaml) contained something like this:

```
# Configuration information for validate_drp to
# build the list of data IDs to analyze
tracts: [9697]
visits: [6320,34338,34342,34362,34366,34382,34384,34400,34402,34412,34414,34422,34424,34448,34450,34464,34468
filter: ['HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z']
ccd: [0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103]
instrument: 'HSC'
```

Status: **Pass**

4 Description

Confirm that the metric ABF1 has been calculated using the outlier limit AB2, and that its values are reasonable.

Expected Result

A JSON file (and/or a report generated from that JSON file) demonstrating that ABF1 has been calculated (and used the limit AB2).

Actual Result

This was confirmed by

a. loading the JSON and printing a report from within a Jupyterlab notebook on the LSP (see attached rendering of notebook; the notebook is saved in as 'test_KPMs_validate_drp.ipynb' in the DMTR-201 github repository), and
b. dispatching the metric measurements to the SQuaSH chronograf dashboard (see attached screen shot).

See the documents attached to LVV-T1745 for illustration of the results.

The adopted value for AB1 was verified by inspecting the validate_drp source code. Note that AB1 is also calculated in validate_drp as the value of the residual at the 10th percentile (i.e., the design value of ABF1).

The calculated value of ABF1 for the example case (tract 9615, HSC-I filter) illustrated in the attached document is ABF1 = 41.1%, which is well above the design threshold of 10.0%. However, the large value of this metric is likely due in part due to the small input dataset used in its calculation.

Status: **Pass**

5.1.3.14 Test Case LVV-T1751 - Verify calculation of median relative astrometric measurement error on 200 arcminute scales

Open *LVV-T1751* test case in Jira.

Verify that the DM system has provided the code to calculate the median relative astrometric measurement error on 200 arcminute scales and assess whether it meets the requirement that it shall be no more than AM3 = 15 milliarcseconds.

Preconditions:

Execution status: **Pass**

Final comment:
Tests executed on lsst-dev, using Release v19.0.0.

Detailed steps results:

Step	Step Details
1	Description Identify a dataset containing at least one field with multiple overlapping visits, and that covers an area larger than 200 arcminutes. Expected Result A dataset that has been ingested into a Butler repository. Actual Result We used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223. The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'. 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

On lsst-dev, the setup was done as follows:

source /software/lsstsw/stack/loadLSST.bash
source scl_source enable devtoolset-8
setup -t w_2019_46 lsst_distrib
setup obs_subaru
setup validate_drp

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

Batch jobs were sent to via slurm jobs to the batch processing cluster using a script that looked like (submitted via ‘sbatch SCRIPTNAME‘):

```
#!/bin/bash -l

#SBATCH -p normal
#SBATCH -N 1
#SBATCH –ntasks-per-node=1
#SBATCH -t 18:00:00
#SBATCH -J v9697
#SBATCH –output=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log
#SBATCH –error=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log

srun validateDrp.py /datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223 –configFile ‘cfg9697_all.yaml’
–outputPrefix=’tract9697’
```

…where the configuration file (cfg9697_all.yaml) contained something like this:

```
# Configuration information for validate_drp to
# build the list of data IDs to analyze
tracts: [9697]
visits: [6320,34338,34342,34362,34366,34382,34384,34400,34402,34412,34414,34422,34424,34448,34450,34464,34468
filter:     [’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-
G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-G’,’HSC-I’,’HSC-I’,’HSC-
I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-
I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-I’,’HSC-
I’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-
R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-R’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-
Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-
Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-Y’,’HSC-
Y’,’HSC-Y’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-
Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-
Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’,’HSC-Z’]
ccd: [0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59,
60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88,
89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103]
instrument: ‘HSC’
```

	Status: **Pass**
4	Description Confirm that the metric AM3 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 AM3 has been calculated. Actual Result This was confirmed by a. loading the JSON and printing a report from within a Jupyterlab notebook on the LSP (see attached rendering of notebook; the notebook is saved in as 'test_KPMs_validate_drp.ipynb' in the DMTR-201 github repository), and b. dispatching the metric measurements to the SQuaSH chronograf dashboard (see attached screen shot). See the documents attached to LVV-T1745 for illustration of the results. The calculated value of AM3 for the example case (tract 9615, HSC-I filter) illustrated in the attached document is AM3 = 10.3 mas, which is near the design threshold of 15.0 mas. Status: **Pass**

5.1.3.15 Test Case LVV-T1752 - Verify calculation of fraction of relative astrometric measurement error on 200 arcminute scales exceeding outlier limit

Open *LVV-T1752* test case in Jira.

Verify that the DM system has provided the code to calculate the maximum fraction of relative astrometric measurements on 200 arcminute scales that exceed the 200 arcminute outlier limit **AD3 = 30 milliarcseconds**, and assess whether it meets the requirement that it shall be less than **AF3 = 10 percent.**

Preconditions:

Execution status: **Pass**

Final comment:
Tests executed on lsst-dev, using Release v19.0.0.

Detailed steps results:

Step	Step Details
1	Description Identify a dataset containing at least one field with multiple overlapping visits, and that covers an area larger than 200 arcminutes. Expected Result A dataset that has been ingested into a Butler repository. Actual Result We used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223. The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'. 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

On lsst-dev, the setup was done as follows:

source /software/lsstsw/stack/loadLSST.bash
source scl_source enable devtoolset-8
setup -t w_2019_46 lsst_distrib
setup obs_subaru
setup validate_drp

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

Batch jobs were sent to via slurm jobs to the batch processing cluster using a script that looked like (submitted via ‘sbatch SCRIPTNAME‘):

#!/bin/bash -l

#SBATCH -p normal
#SBATCH -N 1
#SBATCH –ntasks-per-node=1
#SBATCH -t 18:00:00
#SBATCH -J v9697
#SBATCH –output=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log
#SBATCH –error=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log

srun validateDrp.py /datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223 –configFile ‘cfg9697_all.yaml’ –outputPrefix=’tract9697’

...where the configuration file (cfg9697_all.yaml) contained something like this:

Configuration information for validate_drp to
build the list of data IDs to analyze
tracts: [9697]
visits: [6320,34338,34342,34362,34366,34382,34384,34400,34402,34412,34414,34422,34424,34448,34450,34464,34468
filter: [’HSC-G’,’HSC-I’,’HSC-R’,’HSC-Y’,’HSC-Z’,’HSC-Z’]
ccd: [0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103]
instrument: ‘HSC’

Status: **Pass**

4 Description

Confirm that the metric AF3 has been calculated using the outlier limit AD3, and that its values are reasonable.

Expected Result

A JSON file (and/or a report generated from that JSON file) demonstrating that AF3 has been calculated (and used the limit AD3).

Actual Result

This was confirmed by

a. loading the JSON and printing a report from within a Jupyterlab notebook on the LSP (see attached rendering of notebook; the notebook is saved in as 'test_KPMs_validate_drp.ipynb' in the DMTR-201 github repository), and
b. dispatching the metric measurements to the SQuaSH chronograf dashboard (see attached screen shot).

See the documents attached to LVV-T1745 for illustration of the results.

The adopted value for AD3 was verified by inspecting the validate_drp source code. Note that AD3 is also calculated in validate_drp as the value of the residual at the 10th percentile (i.e., the design value of AF3).

The calculated value of AF3 for the example case (tract 9615, HSC-I filter) illustrated in the attached document is AF3 = 1.9%, which is near the design threshold of 10.0%.

Status: **Pass**

5.1.3.16 Test Case LVV-T1753 - Verify calculation of RMS difference of separations relative to r-band

Open *LVV-T1753* test case in Jira.

Verify that the DM system has provided the code to calculate the separations measured relative to the r-band, and assess whether it meets the requirement that it shall be less than **AB1 = 10 milliarcseconds.**

Preconditions:

Execution status: **Pass**

Final comment:
Tests executed on lsst-dev, using Release v19.0.0.

Detailed steps results:

Step	Step Details
1	Description Identify a dataset containing at least one field with multiple overlapping visits, and including at least one visit in r-band.
	Expected Result A dataset that has been ingested into a Butler repository.
	Actual Result We used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223. The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'.
	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

On lsst-dev, the setup was done as follows:

source /software/lsstsw/stack/loadLSST.bash
source scl_source enable devtoolset-8
setup -t w_2019_46 lsst_distrib
setup obs_subaru
setup validate_drp

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

Batch jobs were sent to via slurm jobs to the batch processing cluster using a script that looked like (submitted via ‘sbatch SCRIPTNAME‘):

#!/bin/bash -l

#SBATCH -p normal
#SBATCH -N 1
#SBATCH –ntasks-per-node=1
#SBATCH -t 18:00:00
#SBATCH -J v9697
#SBATCH –output=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log
#SBATCH –error=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log

srun validateDrp.py /datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223 –configFile ‘cfg9697_all.yaml’ –outputPrefix=’tract9697’

...where the configuration file (cfg9697_all.yaml) contained something like this:

Configuration information for validate_drp to
build the list of data IDs to analyze
tracts: [9697]
visits: [6320,34338,34342,34362,34366,34382,34384,34400,34402,34412,34414,34422,34424,34448,34450,34464,34468
filter: [’HSC-G’,’HSC-I’,’HSC-R’,’HSC-Y’,’HSC-Z’,’HSC-Z’]
ccd: [0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103]
instrument: ‘HSC’

	Status: **Pass**
4	Description Confirm that the metric AB1 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 AB1 has been calculated. Actual Result This was confirmed by a. loading the JSON and printing a report from within a Jupyterlab notebook on the LSP (see attached rendering of notebook; the notebook is saved in as 'test_KPMs_validate_drp.ipynb' in the DMTR-201 github repository), and b. dispatching the metric measurements to the SQuaSH chronograf dashboard (see attached screen shot). See the documents attached to LVV-T1745 for illustration of the results. The calculated value of AB1 for the example case (tract 9615, HSC-I filter) illustrated in the attached document is AB1 = 14.4 mas, which is near the design threshold of 10.0 mas. Status: **Pass**

5.1.3.17 Test Case LVV-T1754 - Verify calculation of residual PSF ellipticity correlations for separations less than 5 arcmin

Open *LVV-T1754* test case in Jira.

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

Preconditions:

Execution status: **Pass**

Final comment:
Tests executed on lsst-dev, using Release v19.0.0.

Detailed steps results:

Step	Step Details
1	Description Identify a dataset containing at least one field with multiple overlapping visits. Expected Result A dataset that has been ingested into a Butler repository. Actual Result We used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223. The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'. 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

On lsst-dev, the setup was done as follows:

source /software/lsstsw/stack/loadLSST.bash
source scl_source enable devtoolset-8
setup -t w_2019_46 lsst_distrib
setup obs_subaru
setup validate_drp

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

Batch jobs were sent to via slurm jobs to the batch processing cluster using a script that looked like (submitted via ‘sbatch SCRIPTNAME‘):

```
#!/bin/bash -l

#SBATCH -p normal
#SBATCH -N 1
#SBATCH –ntasks-per-node=1
#SBATCH -t 18:00:00
#SBATCH -J v9697
#SBATCH –output=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log
#SBATCH –error=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log

srun validateDrp.py /datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223 –configFile ‘cfg9697_all.yaml’ –outputPrefix=’tract9697’
```

...where the configuration file (cfg9697_all.yaml) contained something like this:

```
# Configuration information for validate_drp to
# build the list of data IDs to analyze
tracts: [9697]
visits: [6320,34338,34342,34362,34366,34382,34384,34400,34402,34412,34414,34422,34424,34448,34450,34464,34468
filter: ['HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z']
ccd: [0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103]
instrument: 'HSC'
```

Status: **Pass**

4 Description

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

Expected Result

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

Actual Result

This was confirmed by

a. loading the JSON and printing a report from within a Jupyterlab notebook on the LSP (see attached rendering of notebook; the notebook is saved in as 'test_KPMs_validate_drp.ipynb' in the DMTR-201 github repository), and
b. dispatching the metric measurements to the SQuaSH chronograf dashboard (see attached screen shot).

See the documents attached to LVV-T1745 for illustration of the results.

The calculated value of TE2 for the example case (tract 9615, HSC-I filter) illustrated in the attached document is TE2 = 1.26e-6, which is near the design threshold of 5e-5. (Note that this value does not appear in the attached table due to an error in formatting.

Status: **Pass**

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

Open *LVV-T1755* test case in Jira.

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

Preconditions:

Execution status: **Pass**

Final comment:
Tests executed on lsst-dev, using Release v19.0.0.

Detailed steps results:

Step	Step Details
1	**Description** Identify a dataset containing at least one field with multiple overlapping visits. **Expected Result** A dataset that has been ingested into a Butler repository. **Actual Result** We used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223. The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'. 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

On lsst-dev, the setup was done as follows:

source /software/lsstsw/stack/loadLSST.bash
source scl_source enable devtoolset-8
setup -t w_2019_46 lsst_distrib
setup obs_subaru
setup validate_drp

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

Batch jobs were sent to via slurm jobs to the batch processing cluster using a script that looked like (submitted via ‘sbatch SCRIPTNAME‘):

#!/bin/bash -l

#SBATCH -p normal
#SBATCH -N 1
#SBATCH –ntasks-per-node=1
#SBATCH -t 18:00:00
#SBATCH -J v9697
#SBATCH –output=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log
#SBATCH –error=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log

srun validateDrp.py /datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223 –configFile ‘cfg9697_all.yaml’ –outputPrefix=’tract9697’

...where the configuration file (cfg9697_all.yaml) contained something like this:

Configuration information for validate_drp to
build the list of data IDs to analyze
tracts: [9697]
visits: [6320,34338,34342,34362,34366,34382,34384,34400,34402,34412,34414,34422,34424,34448,34450,34464,34468
filter: [’HSC-G’,’HSC-I’,’HSC-R’,’HSC-Y’,’HSC-Z’,’HSC-Z’]
ccd: [0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103]
instrument: ‘HSC’

	Status: **Pass**
4	Description Confirm that the metric TE1 has been calculated, and that its values are reasonable.
	Expected Result A JSON file (and/or a report generated from that JSON file) demonstrating that TE1 has been calculated.
	Actual Result This was confirmed by a. loading the JSON and printing a report from within a Jupyterlab notebook on the LSP (see attached rendering of notebook; the notebook is saved in as 'test_KPMs_validate_drp.ipynb' in the DMTR-201 github repository), and b. dispatching the metric measurements to the SQuaSH chronograf dashboard (see attached screen shot). See the documents attached to LVV-T1745 for illustration of the results. The calculated value of TE1 for the example case (tract 9615, HSC-I filter) illustrated in the attached document is TE1 = 1.3e-5, which is near the design threshold of 2e-5. (Note that this value does not appear in the attached table due to an error in formatting.
	Status: **Pass**

5.1.3.19 Test Case LVV-T1758 - Verify calculation of photometric outliers in uzy bands

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 millimagnitudes**.

Preconditions:

Execution status: **Fail**

Final comment:
Tests executed on lsst-dev, using Release v19.0.0.

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 used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223. The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'. 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

On lsst-dev, the setup was done as follows:

source /software/lsstsw/stack/loadLSST.bash
source scl_source enable devtoolset-8
setup -t w_2019_46 lsst_distrib
setup obs_subaru
setup validate_drp

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

Batch jobs were sent to via slurm jobs to the batch processing cluster using a script that looked like (submitted via 'sbatch SCRIPTNAME'):

```
#!/bin/bash -l

#SBATCH -p normal
#SBATCH -N 1
#SBATCH –ntasks-per-node=1
#SBATCH -t 18:00:00
#SBATCH -J v9697
#SBATCH –output=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log
#SBATCH –error=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log

srun validateDrp.py /datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223 –configFile 'cfg9697_all.yaml'
–outputPrefix='tract9697'
```

...where the configuration file (cfg9697_all.yaml) contained something like this:

```
# Configuration information for validate_drp to
# build the list of data IDs to analyze
tracts: [9697]
visits: [6320,34338,34342,34362,34366,34382,34384,34400,34402,34412,34414,34422,34424,34448,34450,34464,34468
filter: ['HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-
G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-I','HSC-I','HSC-
I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-
I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-
I','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-
R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-Y','HSC-Y','HSC-Y','HSC-
Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-
Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-
Y','HSC-Y','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-
Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-
Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z']
ccd: [0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59,
60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88,
89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103]
instrument: 'HSC'
```

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

This was confirmed by

a. loading the JSON and printing a report from within a Jupyterlab notebook on the LSP (see attached rendering of notebook; the notebook is saved in as 'test_KPMs_validate_drp.ipynb' in the DMTR-201 github repository), and
b. dispatching the metric measurements to the SQuaSH chronograf dashboard (see attached screen shot).

See the documents attached to LVV-T1745 for illustration of the results.

We discovered after executing validate_drp that PA2 has only been implemented for the gri bands, and not for uzy. Thus it is not reported in the outputs, and this test has failed.

Issues found executing this step:

DM-23417 Implement PA2uzy in validate_drp

Status: **Fail**

5.1.3.20 Test Case LVV-T1759 - Verify calculation of photometric outliers in gri bands

Open *LVV-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:
Tests executed on lsst-dev, using Release v19.0.0.

Detailed steps results:

Step	Step Details
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. Actual Result We used the output repo from HSC-RC2 data processing, as executed using the weekly pipelines release (w_2019_46) that became v19.0.0. The output repo is tagged with the Jira ticket number DM-22223. The path to the dataset on the development cluster (lsst-dev) is '/datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223'. 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

On lsst-dev, the setup was done as follows:

source /software/lsstsw/stack/loadLSST.bash
source scl_source enable devtoolset-8
setup -t w_2019_46 lsst_distrib
setup obs_subaru
setup validate_drp

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

Batch jobs were sent to via slurm jobs to the batch processing cluster using a script that looked like (submitted via 'sbatch SCRIPTNAME'):

```
#!/bin/bash -l

#SBATCH -p normal
#SBATCH -N 1
#SBATCH –ntasks-per-node=1
#SBATCH -t 18:00:00
#SBATCH -J v9697
#SBATCH –output=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log
#SBATCH –error=/project/jcarlin/verify/RC2_v3/validateDrp/logs/rc2-9697-%j.log

srun validateDrp.py /datasets/hsc/repo/rerun/RC/w_2019_46/DM-22223 –configFile 'cfg9697_all.yaml' –outputPrefix='tract9697'
```

...where the configuration file (cfg9697_all.yaml) contained something like this:

```
# Configuration information for validate_drp to
# build the list of data IDs to analyze
tracts: [9697]
visits: [6320,34338,34342,34362,34366,34382,34384,34400,34402,34412,34414,34422,34424,34448,34450,34464,34468
filter: ['HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-G','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-I','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-R','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Y','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z','HSC-Z']
ccd: [0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103]
instrument: 'HSC'
```

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

This was confirmed by

a. loading the JSON and printing a report from within a Jupyterlab notebook on the LSP (see attached rendering of notebook; the notebook is saved in as 'test_KPMs_validate_drp.ipynb' in the DMTR-201 github repository), and
b. dispatching the metric measurements to the SQuaSH chronograf dashboard (see attached screen shot).

See the documents attached to LVV-T1745 for illustration of the results.

The adopted value for PA2gri was verified by inspecting the validate_drp source code. Note that PA2gri is also calculated in validate_drp as the value of the residual at the 10th percentile (i.e., the design value of PF1).

The calculated value of PF1 for the example case (tract 9615, HSC-I filter) illustrated in the attached document is PF1 = 47.8%, which is much higher than the design threshold of 10.0%. However, it is not clear whether this is due to the input dataset being used.

Status: **Pass**

A Traceability

Test Case	VE Key	VE Summary
LVV-T28	LVV-178	DMS-REQ-0347-V-01: Measurements in catalogs
LVV-T40	LVV-13	DMS-REQ-0030-V-01: Absolute accuracy of WCS
LVV-T41	LVV-30	DMS-REQ-0070-V-01: Generate PSF for Visit Images
LVV-T43	LVV-158	DMS-REQ-0327-V-01: Background Model Calculation
LVV-T62	LVV-20	DMS-REQ-0047-V-01: Provide PSF for Coadded Images
LVV-T132	LVV-127	DMS-REQ-0296-V-01: Pre-cursor, and Real Data
LVV-T1240	LVV-9741	DMS-REQ-0030-V-02: Minimum astrometric standards per CCD
LVV-T1745	LVV-3402	DMS-REQ-0360-V-01: Median astrometric error on 20 arcmin scales
LVV-T1746	LVV-9767	DMS-REQ-0360-V-02: Max fraction exceeding limit on 5 arcmin scales
	LVV-9773	DMS-REQ-0360-V-07: Outlier limit on 5 arcmin scales
LVV-T1747	LVV-9768	DMS-REQ-0360-V-03: Median astrometric error on 5 arcmin scales
LVV-T1748	LVV-9769	DMS-REQ-0360-V-04: Median absolute error in RA, Dec
LVV-T1749	LVV-9776	DMS-REQ-0360-V-10: Max fraction exceeding limit on 20 arcmin scales
	LVV-9770	DMS-REQ-0360-V-05: Outlier limit on 20 arcmin scales
LVV-T1750	LVV-9771	DMS-REQ-0360-V-06: Color difference outlier limit relative to r-band
	LVV-9777	DMS-REQ-0360-V-11: Max fraction of r-band color difference outliers
LVV-T1751	LVV-9774	DMS-REQ-0360-V-08: Median astrometric error on 200 arcmin scales
LVV-T1752	LVV-9779	DMS-REQ-0360-V-13: Max fraction exceeding limit on 200 arcmin scales

LVV-T1753	LVV-9778	DMS-REQ-0360-V-12: RMS difference between r-band and other filter separation
LVV-T1754	LVV-3404	DMS-REQ-0362-V-01: Median residual PSF ellipticity correlations on 5 arcmin scales
LVV-T1755	LVV-9782	DMS-REQ-0362-V-04: Median residual PSF ellipticity correlations on 1 arcmin scales
LVV-T1758	LVV-9758	DMS-REQ-0359-V-09: Repeatability outlier limit in uzy
	LVV-9752	DMS-REQ-0359-V-03: Max fraction of outliers among non-saturated sources
LVV-T1759	LVV-9752	DMS-REQ-0359-V-03: Max fraction of outliers among non-saturated sources
	LVV-9754	DMS-REQ-0359-V-05: Repeatability outlier limit in gri

B Acronyms used in this document

Acronym	Description
AP	Alert Production
CCD	Charge-Coupled Device
CI	Cyber Infrastructure
DEC	Declination
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
JSON	JavaScript Object Notation
LDF	LSST Data Facility
LDM	LSST Data Management (Document Handle)
LSE	LSST Systems Engineering (Document Handle)
LSP	LSST Science Platform
LSST	Legacy Survey of Space and Time (formerly Large Synoptic Survey Telescope)
LVV	LSST Verification and Validation
NCSA	National Center for Supercomputing Applications
PSF	Point Spread Function
PVI	Processed Visit Image
RA	Right Ascension
RC	Release Candidate
RC2	HSC-RC2
RMS	Root-Mean-Square
SQuaSH	Science Quality Analysis Harness
SV	Science Validation
WCS	World Coordinate System
arcmin	arcminute minute of arc (unit of angle)
arcsec	arcsecond second of arc (unit of angle)

deg	degree; unit of angle