Vera C. Rubin Observatory
Data Management

LSST Data Management Acceptance Test Specification

L.P. Guy, W.M. Wood-Vasey, E. Bellm, J.F. Bosch, M. Butler, J.L. Carlin, H.-F. Chiang, G. Comoretto, G.P. Dubois-Felsmann, M. Gower, M.L. Graham, R. Gruendl, K.S. Krughoff, K.-T. Lim, R.H. Lupton, F. Mueller, C. Slater, J.D. Swinbank

LDM-639

Latest Revision: 2022-09-19

Abstract

This document describes the detailed acceptance test specification for the LSST Data Management System.

Change Record

Version	Date	Description	Owner name
1.0	2019-02-07	Document first approved release. Approved on RFC-495.	L. Guy
2.0	2019-07-29	Add test cases for all Priority 1a requirements. Approved in RFC-622.	Document: J. Carlin; Approver: W. O'Mullane
2.1	2020-02-17	Consolidated test cases for DMTR-201 test activity. Approved in RFC-669.	J. Carlin
2.2	2020-08-21	Merge test specifications LDM-534, LDM-534, LDM-538, LDM-538. Update author list. RFC-721	L. Guy
3.0	2022-09-19	Add test cases for DMTR-301, DMTR-271, DMTR-371, DMTR-381 test campaigns. Tag for JSR-2022	L. Guy

Document curator: Leanne Guy
Document source location: `https://github.com/lsst/ldm-639` from Jira
Version from source repository: 602f7f4

Contents

LSST Data Management Acceptance Test Specification

1 Introduction

This document specifies the acceptance test procedures for the LSST Data Management System. It is a living document that is updated as new functionality is delivered and acceptance testing proceeds. A full description of the LSST Data Management System is provided in the Data Management System Design document, LDM-148 with the science requirements detailed in the LSST Science Requirements Document LPM-17.

1.1 Objectives

This document builds on the description of LSST Data Management's approach to testing as described in LDM-503 to describe the detailed test cases that will be performed to verify the Data Management System.

It provides test designs, test cases and procedures for the tests, and the corresponding pass/-fail criteria for each test.

1.2 Scope

This document provides the acceptance test plan for the Data Management System (DMS), as described by the Data Management System Requirements in LSE-61.

1.3 Applicable Documents

LPM-17	LSST Science Requirements Document
LDM-148	LSST Data Management System Design
LDM-294	LSST DM Organization & Management
LDM-503	LSST DM Test Plan
LSE-61	LSST DM Subsystem Requirements
LSE-163	LSST Data Products Definition Document
LDM-151	LSST DM Science Pipelines Design
LSE-180	Level 2 Photometric Calibration for the LSST Survey
LSE-30	LSST Observatory System Specifications

1.4 References

[1] **[LSE-30]**, Claver, C.F., The LSST Systems Engineering Integrated Project Team, 2018, *Observatory System Specifications (OSS)*, LSE-30, URL `https://ls.st/LSE-30`

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

[3] **[LDM-554]**, Dubois-Felsmann, G., Ciardi, D., Mueller, F., Economou, F., 2019, *Data Management LSST Science Platform Requirements*, LDM-554, URL `https://ldm-554.lsst.io/`,
Vera C. Rubin Observatory Data Management Controlled Document

[4] **[LPM-17]**, Ivezić, Ž., The LSST Science Collaboration, 2018, *LSST Science Requirements Document*, LPM-17, URL `https://ls.st/LPM-17`

[5] **[LSE-131]**, Jacoby, S., Emmons, B., Selvy, B., 2017, *Interface between Data Management and Education and Public Outreach*, LSE-131, URL `https://ls.st/LSE-131`

[6] **[LSE-180]**, Jones, L., 2013, *Level 2 Photometric Calibration for the LSST Survey*, LSE-180, URL `https://ls.st/LSE-180`

[7] **[LSE-163]**, Jurić, M., Axelrod, T., Becker, A., et al., 2021, *Data Products Definition Document*, LSE-163, URL `https://lse-163.lsst.io/`,
Vera C. Rubin Observatory

[8] **[LDM-142]**, Kantor, J., 2017, *Network Sizing Model*, LDM-142, URL `https://ls.st/LDM-142`

[9] **[LDM-148]**, Lim, K.T., Bosch, J., Dubois-Felsmann, G., et al., 2018, *Data Management System Design*, LDM-148, URL `https://ls.st/LDM-148`

[10] **[LDM-294]**, O'Mullane, W., Swinbank, J., Juric, M., Guy, L., DMLT, 2022, *Data Management Organization and Management*, LDM-294, URL `https://ldm-294.lsst.io/`, Vera C. Rubin Observatory Data Management Controlled Document

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

[12] **[LDM-151]**, Swinbank, J., Axelrod, T., Becker, A., et al., 2020, *Data Management Science Pipelines Design*, LDM-151, URL `https://ldm-151.lsst.io/`, Vera C. Rubin Observatory Data Management Controlled Document

1.5 Acronyms

Acronym	Description
AP	Alerts Production
C	Specific programming language (also called ANSI-C)
CPP	C++ Programming language
DAC	Data Access Center
DB	DataBase
DBB	Data BackBone
DM	Data Management
DMCCB	DM Change Control Board
DMS	Data Management Sub-system
DR	Data Release
DRP	Data Release Production
EFD	Engineering Facilities Database
IT	Integration Test
IVOA	International Virtual-Observatory Alliance
K	Kelvin; SI unit of temperature
LAN	Local Area Network
LDM	LSST Data Management (handle for controlled documents)

LPM	LSST Project Management (Document Handle)
LSE	LSST Systems Engineering (Document Handle)
LSP	LSST Science Platform
LSST	Large Synoptic Survey Telescope
M	Mega; SI units prefix for 1E6
MOPS	Moving Object Pipeline System
OCS	Observatory Control System
PDAC	Prototype Data Access Center
S	Strip (CCD chip along-scan coordinate identifier in focal plane)
SODA	SCOS ORATOS Distributed Access
SQL	Structured Query Language
STS	System Test Specification
W	Watt; SI unit of power
p	pico; SI units prefix for 1E-12

2 Approach

This document describes the acceptance tests for the Data Management System, with a focus on whether the data products, functionality and services satisfy the requirements described in LSE-61.

The requirements from LSE-61 are extracted into the Jira "LSST Verification and Validation" Project, managed through the Jira Test Management Plugin system. Each LSE-61 requirement leads to a "LSST Verification and Validation" (LVV) Element. Each LVV Element comprises one or more more Test Cases. Each Test Case describes a Test Script to be executed, the coverage, pre-conditions, configuration, test results, and other details as specified by LDM-503. Test Scripts may have common set up and analysis steps. The Jira system allows for these steps to be shared by other Test Scripts. This improves clarity and consistency across all Test Cases.

In this document, each Test Case is listed here with the LVV Element it tests, a summary of the Test Items exercised by the Test Case, and the detailed steps to be executed by the Test Case. Shared steps between Test Scripts have been explicitly written out to appear fully in each Test Case.

2.1 Features to be tested

All top-level requirements for the LSST Data Management System described in LSE-61 are to be tested, including

- Rubin Data Products, including their production, scientific fidelity and persistence,
- Alert, Calibration and Data Release Production pipelines and the execution of payloads,
- Middleware,
- Qserv, the LSST parallel distributed database,
- Services provided by the Rubin Data Facility,
- Rubin facilities including the data archive, base, summit, and the communications between them to accept science and engineering data.

2.2 Features not to be tested

This document does not describe facilities for periodically generating or collecting key performance metrics (KPMs), except insofar as those KPMs are incidentally measured as part of executing the documented test cases.

2.3 Pass/fail criteria

The results of all tests will be assessed using the criteria described in LDM-503 §4.

Note that when executing pipelines, tasks, or individual algorithms, any unexplained or unexpected errors or warnings appearing in the associated log or on screen output must be described in the documentation for the system under test. Any warning or error for which this is not the case must be filed as a software problem report and filed with the DMCCB.

2.4 Suspension criteria and resumption requirements

Refer to individual test cases where applicable.

2.5 Naming convention

LVV : Is the label for the "LSST Verification and Validation" project in Jira.

LVV-XXX : Are Verification Elements, where XXX is the Verification Element identifier. Each Verification Element has at least one Test Case.

LVV-TYYY : Are Test Cases. Each Test Case is associated with a Verification Element, where YYY is the Test Case identifier.

The Verification Elements are drawn from LSE-61 requirements which have names of the form DMS-REQ-ZZZZ.

3 Test Cases Summary

Test Id	Test Name	
LVV-T29	Verify implementation of Raw Science Image Data Acquisition	Defined
LVV-T30	Verify implementation of Wavefront Sensor Data Acquisition	Defined
LVV-T32	Verify implementation of Raw Image Assembly	Defined
LVV-T33	Verify implementation of Raw Science Image Metadata	Defined
LVV-T34	Verify implementation of Guider Calibration Data Acquisition	Defined
LVV-T45	Verify implementation of Prompt Processing Data Quality Report Definition	Defined
LVV-T47	Verify implementation of Prompt Processing Calibration Report Definition	Defined
LVV-T48	Verify implementation of Exposure Catalog	Defined
LVV-T61	Verify implementation of Associate Sources to Objects	Defined
LVV-T65	Verify implementation of Source Catalog	Defined
LVV-T66	Verify implementation of Forced-Source Catalog	Defined
LVV-T82	Verify implementation of Tracking Characterization Changes Between Data Releases	Defined
LVV-T83	Verify implementation of Bad Pixel Map	Defined
LVV-T85	Verify implementation of Crosstalk Correction Matrix	Defined
LVV-T88	Verify implementation of Calibration Data Products	Defined
LVV-T89	Verify implementation of Calibration Image Provenance	Defined
LVV-T97	Verify implementation of Uniqueness of IDs Across Data Releases	Defined
LVV-T98	Verify implementation of Selection of Datasets	Defined
LVV-T103	Verify implementation of Generate Data Quality Report Within Specified Time	Defined
LVV-T112	Verify implementation of Alert Filtering Service	Defined
LVV-T113	Verify implementation of Performance Requirements for LSST Alert Filtering Service	Defined
LVV-T114	Verify implementation of Pre-defined alert filters	Defined
LVV-T124	Verify implementation of Software Architecture to Enable Community Re-Use	Defined
LVV-T131	Verify implementation of Provide User Interface Services	Defined
LVV-T136	Verify implementation of Image Data Product Access	Defined
LVV-T140	Verify implementation of Production Orchestration	Defined
LVV-T141	Verify implementation of Production Monitoring	Defined

Test Id	Test Name	
LVV-T150	Verify implementation of Maintain Archive Publicly Accessible	Defined
LVV-T153	Verify implementation of Provide Engineering and Facility Database Archive	Defined
LVV-T183	Verify implementation of DMS Communication with OCS	Defined
LVV-T385	Verify implementation of minimum number of simultaneous retrievals of CCD-sized coadd cutouts	Defined
LVV-T1252	Verify number of simultaneous alert filter users	Defined
LVV-T1332	Verify implementation of maximum time for retrieval of CCD-sized coadd cutouts	Defined
LVV-T28	Verify implementation of measurements in catalogs from PVIs	Approved
LVV-T38	Verify implementation of Processed Visit Images	Approved
LVV-T39	Verify implementation of Generate Photometric Zeropoint for Visit Image	Approved
LVV-T40	Verify implementation of Generate WCS for Visit Images	Approved
LVV-T41	Verify implementation of Generate PSF for Visit Images	Approved
LVV-T42	Verify implementation of Processed Visit Image Content	Approved
LVV-T43	Verify implementation of Background Model Calculation	Approved
LVV-T62	Verify implementation of Provide PSF for Coadded Images	Approved
LVV-T74	Verify implementation of Template Coadds	Approved
LVV-T77	Verify implementation of Best Seeing Coadds	Approved
LVV-T78	Verify implementation of Persisting Data Products	Approved
LVV-T84	Verify implementation of Bias Residual Image	Approved
LVV-T90	Verify implementation of Dark Current Correction Frame	Approved
LVV-T91	Verify implementation of Fringe Correction Frame	Approved
LVV-T115	Verify implementation of Calibration Production Processing	Approved
LVV-T125	Verify implementation of Simulated Data	Approved
LVV-T126	Verify implementation of Image Differencing	Approved
LVV-T127	Verify implementation of Provide Source Detection Software	Approved
LVV-T129	Verify implementation of Provide Calibrated Photometry	Approved
LVV-T132	Verify implementation of Pre-cursor and Real Data	Approved
LVV-T133	Verify implementation of Provide Beam Projector Coordinate Calculation Software	Approved
LVV-T137	Verify implementation of Data Product Ingest	Approved
LVV-T144	Verify implementation of Task Specification	Approved

Test Id	Test Name	
LVV-T145	Verify implementation of Task Configuration	Approved
LVV-T146	Verify implementation of DMS Initialization Component	Approved
LVV-T149	Verify implementation of Catalog Queries	Approved
LVV-T151	Verify Implementation of Catalog Export Formats From the Notebook Aspect	Approved
LVV-T190	Verify implementation of Base Facility Co-Location with Existing Facility	Approved
LVV-T199	Verify implementation of Archive Center Co-Location with Existing Facility	Approved
LVV-T216	Installation of the Alert Distribution payloads.	Approved
LVV-T217	Full Stream Alert Distribution	Approved
LVV-T218	Simple Filtering of the LSST Alert Stream	Approved
LVV-T283	RAS-00-00: Writing well-formed raw image	Approved
LVV-T285	RAS-00-10: Raw images in Observatory Operations Data Service	Approved
LVV-T286	RAS-00-20: Raw image are part of the permanent record of survey via DBB	Approved
LVV-T287	RAS-00-30: Raw Image Archiving Availability, Throughput, Reliability, and Heterogeneity	Approved
LVV-T362	Installation of the LSST Science Pipelines Payloads	Approved
LVV-T363	Science Pipelines Release Documentation	Approved
LVV-T368	Loading and processing Camera test data	Approved
LVV-T374	Ingesting Camera test data	Approved
LVV-T376	Verify the Calculation of Ellipticity Residuals and Correlations	Approved
LVV-T377	Verify Calculation of Photometric Performance Metrics	Approved
LVV-T454	LDM-503-8 Enable LSP viewing of spectrograph data.	Approved
LVV-T1085	Short Queries Functional Test	Approved
LVV-T1086	Full Table Scans Functional Test	Approved
LVV-T1087	Full Table Joins Functional Test	Approved
LVV-T1088	Concurrent Scans Scaling Test	Approved
LVV-T1089	Load Test	Approved
LVV-T1090	Heavy Load Test	Approved
LVV-T1168	Verify Summit - Base Network Integration	Approved
LVV-T1232	Verify Implementation of Catalog Export Formats From the Portal Aspect	Approved

Test Id	Test Name	
LVV-T1240	Verify implementation of minimum astrometric standards per CCD	Approved
LVV-T1264	Verify implementation of archiving camera test data	Approved
LVV-T1549	LDM-503-6 Comcam verification readiness	Approved
LVV-T1550	LDM-503-10 DAQ Validation	Approved
LVV-T1556	LDM-503-10B Large Scale CCOB Data Access	Approved
LVV-T1745	Verify calculation of median relative astrometric measurement error on 20 arcminute scales	Approved
LVV-T1746	Verify calculation of fraction of relative astrometric measurement error on 5 arcminute scales exceeding outlier limit	Approved
LVV-T1747	Verify calculation of relative astrometric measurement error on 5 arcminute scales	Approved
LVV-T1748	Verify calculation of median error in absolute position for RA, Dec axes	Approved
LVV-T1749	Verify calculation of fraction of relative astrometric measurement error on 20 arcminute scales exceeding outlier limit	Approved
LVV-T1750	Verify calculation of separations relative to r-band exceeding color difference outlier limit	Approved
LVV-T1751	Verify calculation of median relative astrometric measurement error on 200 arcminute scales	Approved
LVV-T1752	Verify calculation of fraction of relative astrometric measurement error on 200 arcminute scales exceeding outlier limit	Approved
LVV-T1753	Verify calculation of RMS difference of separations relative to r-band	Approved
LVV-T1754	Verify calculation of residual PSF ellipticity correlations for separations greater than or equal to 5 arcmin	Approved
LVV-T1755	Verify calculation of residual PSF ellipticity correlations for separations less than 1 arcmin	Approved
LVV-T1756	Verify calculation of photometric repeatability in uzy filters	Approved
LVV-T1757	Verify calculation of photometric repeatability in gri filters	Approved
LVV-T1758	Verify that the repeatability outlier limit for isolated bright non-saturated point sources in the u, z, and y filters (PA2uzy) can be applied.	Approved
LVV-T1759	Verify that the repeatability outlier limit for isolated bright non-saturated point sources in the g, r, and i filters (PA2gri) can be applied.	Approved

Test Id	Test Name	
LVV-T1830	Verify Implementation of Scientific Visualization of Camera Image Data	Approved
LVV-T1946	Verify implementation of measurements in catalogs from coadds	Approved
LVV-T1947	Verify implementation of measurements in catalogs from difference images	Approved
LVV-T2202	Verify that the of zero-point error outlier limit threshold (PA4) can be applied.	Approved
LVV-T23	Verify implementation of Storing Approximations of Per-pixel Metadata	Draft
LVV-T24	Verify implementation of Computing Derived Quantities	Draft
LVV-T25	Verify implementation of Denormalizing Database Tables	Draft
LVV-T26	Verify implementation of Maximum Likelihood Values and Covariances	Draft
LVV-T27	Verify implementation of Data Availability	Draft
LVV-T35	Verify implementation of Nightly Data Accessible Within 24 hrs	Draft
LVV-T36	Verify implementation of Difference Exposures	Draft
LVV-T37	Verify implementation of Difference Exposure Attributes	Draft
LVV-T44	Verify implementation of Documenting Image Characterization	Draft
LVV-T46	Verify implementation of Prompt Processing Performance Report Definition	Draft
LVV-T49	Verify implementation of DIASource Catalog	Draft
LVV-T50	Verify implementation of Faint DIASource Measurements	Draft
LVV-T51	Verify implementation of DIAObject Catalog	Draft
LVV-T52	Verify implementation of DIAObject Attributes	Draft
LVV-T53	Verify implementation of SSObject Catalog	Draft
LVV-T54	Verify implementation of Alert Content	Draft
LVV-T55	Verify implementation of DIAForcedSource Catalog	Draft
LVV-T56	Verify implementation of Characterizing Variability	Draft
LVV-T57	Verify implementation of Calculating SSObject Parameters	Draft
LVV-T58	Verify implementation of Matching DIASources to Objects	Draft
LVV-T59	Verify implementation of Regenerating L1 Data Products During Data Release Processing	Draft
LVV-T60	Verify implementation of Publishing predicted visit schedule	Draft
LVV-T63	Verify implementation of Produce Images for EPO	Draft

Test Id	Test Name	
LVV-T64	Verify implementation of Coadded Image Provenance	Draft
LVV-T67	Verify implementation of Object Catalog	Draft
LVV-T68	Verify implementation of Provide Photometric Redshifts of Galaxies	Draft
LVV-T69	Verify implementation of Object Characterization	Draft
LVV-T71	Verify implementation of Detecting extended low surface brightness objects	Draft
LVV-T72	Verify implementation of Coadd Image Method Constraints	Draft
LVV-T73	Verify implementation of Deep Detection Coadds	Draft
LVV-T75	Verify implementation of Multi-band Coadds	Draft
LVV-T76	Verify implementation of All-Sky Visualization of Data Releases	Draft
LVV-T79	Verify implementation of PSF-Matched Coadds	Draft
LVV-T80	Verify implementation of Detecting faint variable objects	Draft
LVV-T81	Verify implementation of Targeted Coadds	Draft
LVV-T86	Verify implementation of Illumination Correction Frame	Draft
LVV-T87	Verify implementation of Monochromatic Flatfield Data Cube	Draft
LVV-T92	Verify implementation of Processing of Data From Special Programs	Draft
LVV-T93	Verify implementation of Level 1 Processing of Special Programs Data	Draft
LVV-T94	Verify implementation of Special Programs Database	Draft
LVV-T95	Verify implementation of Constraints on Level 1 Special Program Products Generation	Draft
LVV-T96	Verify implementation of Query Repeatability	Draft
LVV-T99	Verify implementation of Processing of Datasets	Draft
LVV-T100	Verify implementation of Transparent Data Access	Draft
LVV-T101	Verify implementation of Transient Alert Distribution	Draft
LVV-T102	Verify implementation of Solar System Objects Available Within Specified Time	Draft
LVV-T104	Verify implementation of Generate DMS Performance Report Within Specified Time	Draft
LVV-T105	Verify implementation of Generate Calibration Report Within Specified Time	Draft
LVV-T106	Verify implementation of Calibration Images Available Within Specified Time	Draft
LVV-T107	Verify implementation of Level-1 Production Completeness	Draft

Test Id	Test Name	
LVV-T108	Verify implementation of Level 1 Source Association	Draft
LVV-T109	Verify implementation of SSObject Precovery	Draft
LVV-T110	Verify implementation of DIASource Precovery	Draft
LVV-T111	Verify implementation of Use of External Orbit Catalogs	Draft
LVV-T116	Verify implementation of Associating Objects across data releases	Draft
LVV-T117	Verify implementation of DAC resource allocation for Level 3 processing	Draft
LVV-T118	Verify implementation of Level 3 Data Product Self Consistency	Draft
LVV-T119	Verify implementation of Provenance for Level 3 processing at DACs	Draft
LVV-T120	Verify implementation of Software framework for Level 3 catalog processing	Draft
LVV-T121	Verify implementation of Software framework for Level 3 image processing	Draft
LVV-T122	Verify implementation of Level 3 Data Import	Draft
LVV-T123	Verify implementation of Access Controls of Level 3 Data Products	Draft
LVV-T128	Verify implementation Provide Astrometric Model	Draft
LVV-T130	Verify implementation of Enable a Range of Shape Measurement Approaches	Draft
LVV-T134	Verify implementation of Provide Image Access Services	Draft
LVV-T138	Verify implementation of Bulk Download Service	Draft
LVV-T142	Verify implementation of Production Fault Tolerance	Draft
LVV-T147	Verify implementation of Control of Level-1 Production	Draft
LVV-T148	Verify implementation of Unique Processing Coverage	Draft
LVV-T152	Verify implementation of Keep Historical Alert Archive	Draft
LVV-T154	Verify implementation of Raw Data Archiving Reliability	Draft
LVV-T155	Verify implementation of Un-Archived Data Product Cache	Draft
LVV-T156	Verify implementation of Regenerate Un-archived Data Products	Draft
LVV-T157	Verify implementation Level 1 Data Product Access	Draft
LVV-T158	Verify implementation Level 1 and 2 Catalog Access	Draft
LVV-T159	Verify implementation of Regenerating Data Products from Previous Data Releases	Draft
LVV-T160	Verify implementation of Providing a Precovery Service	Draft
LVV-T161	Verify implementation of Logging of catalog queries	Draft
LVV-T162	Verify implementation of Access to Previous Data Releases	Draft

Test Id	Test Name	
LVV-T163	Verify implementation of Data Access Services	Draft
LVV-T164	Verify implementation of Operations Subsets	Draft
LVV-T165	Verify implementation of Subsets Support	Draft
LVV-T166	Verify implementation of Access Services Performance	Draft
LVV-T167	Verify Capability to serve older Data Releases at Full Performance	Draft
LVV-T168	Verify design of Data Access Services allows Evolution of the LSST Data Model	Draft
LVV-T169	Verify implementation of Older Release Behavior	Draft
LVV-T170	Verify implementation of Query Availability	Draft
LVV-T171	Verify implementation of Pipeline Availability	Draft
LVV-T172	Verify implementation of Optimization of Cost, Reliability and Availability	Draft
LVV-T173	Verify implementation of Pipeline Throughput	Draft
LVV-T174	Verify implementation of Re-processing Capacity	Draft
LVV-T175	Verify implementation of Temporary Storage for Communications Links	Draft
LVV-T176	Verify implementation of Infrastructure Sizing for "catching up"	Draft
LVV-T177	Verify implementation of Incorporate Fault-Tolerance	Draft
LVV-T178	Verify implementation of Incorporate Autonomics	Draft
LVV-T179	Verify implementation of Compute Platform Heterogeneity	Draft
LVV-T180	Verify implementation of Data Management Unscheduled Downtime	Draft
LVV-T181	Verify Base Voice Over IP (VOIP)	Draft
LVV-T182	Verify implementation of Prefer Computing and Storage Down	Draft
LVV-T185	Verify implementation of Summit to Base Network Availability	Draft
LVV-T186	Verify implementation of Summit to Base Network Reliability	Draft
LVV-T187	Verify implementation of Summit to Base Network Secondary Link	Draft
LVV-T188	Verify implementation of Summit to Base Network Ownership and Operation	Draft
LVV-T189	Verify implementation of Base Facility Infrastructure	Draft
LVV-T191	Verify implementation of Commissioning Cluster	Draft
LVV-T192	Verify implementation of Base Wireless LAN (WiFi)	Draft
LVV-T193	Verify implementation of Base to Archive Network	Draft
LVV-T194	Verify implementation of Base to Archive Network Availability	Draft

Test Id	Test Name	
LVV-T195	Verify implementation of Base to Archive Network Reliability	Draft
LVV-T196	Verify implementation of Base to Archive Network Secondary Link	Draft
LVV-T197	Verify implementation of Archive Center	Draft
LVV-T198	Verify implementation of Archive Center Disaster Recovery	Draft
LVV-T200	Verify implementation of Archive to Data Access Center Network	Draft
LVV-T201	Verify implementation of Archive to Data Access Center Network Availability	Draft
LVV-T202	Verify implementation of Archive to Data Access Center Network Reliability	Draft
LVV-T203	Verify implementation of Archive to Data Access Center Network Secondary Link	Draft
LVV-T204	Verify implementation of Access to catalogs for external Level 3 processing	Draft
LVV-T205	Verify implementation of Access to input catalogs for DAC-based Level 3 processing	Draft
LVV-T206	Verify implementation of Federation with external catalogs	Draft
LVV-T207	Verify implementation of Access to images for external Level 3 processing	Draft
LVV-T208	Verify implementation of Access to input images for DAC-based Level 3 processing	Draft
LVV-T209	Verify implementation of Data Access Centers	Draft
LVV-T210	Verify implementation of Data Access Center Simultaneous Connections	Draft
LVV-T211	Verify implementation of Data Access Center Geographical Distribution	Draft
LVV-T212	Verify implementation of No Limit on Data Access Centers	Draft
LVV-T284	RAS-00-05: (LDM-503-8b) Writing data from CCOB to the DBB for further data processing	Draft
LVV-T1097	Verify Summit Facility Network Implementation	Draft
LVV-T1250	Verify implementation of minimum number of simultaneous DM EFD query users	Draft
LVV-T1251	Verify implementation of maximum time to retrieve DM EFD query results	Draft
LVV-T1276	Verify implementation of latency of reporting optical transients	Draft

Test Id	Test Name	
LVV-T1277	Verify processing of maximum number of calibration exposures	Draft
LVV-T1524	Verify Implementation of Exporting MOCs as FITS	Draft
LVV-T1525	Verify Implementation of Linkage Between HiPS Maps and Coadded Images	Draft
LVV-T1526	Verify Availability of Secure and Authenticated HiPS Service	Draft
LVV-T1527	Verify Support for HiPS Visualization	Draft
LVV-T1528	Verify Visualization of MOCs via Science Platform	Draft
LVV-T1529	Verify Production of All-Sky HiPS Map	Draft
LVV-T1530	Verify Production of Multi-Order Coverage Maps for Survey Data	Draft
LVV-T1560	Verify archiving of processing provenance	Draft
LVV-T1561	Verify provenance availability to science users	Draft
LVV-T1562	Verify availability of re-run tools	Draft
LVV-T1563	Verify re-run on different system produces the same results	Draft
LVV-T1564	Verify re-run on similar system produces the same results	Draft
LVV-T1612	Verify Summit - Base Network Integration (System Level)	Draft
LVV-T1831	Verify Implementation of Data Management Nightly Reporting	Draft
LVV-T1836	Verify calculation of resolved-to-unresolved flux ratio errors	Draft
LVV-T1837	Verify calculation of band-to-band color zero-point accuracy	Draft
LVV-T1838	Verify calculation of image fraction affected by ghosts	Draft
LVV-T1839	Verify calculation of RMS width of photometric zeropoint	Draft
LVV-T1840	Verify calculation of sky brightness precision	Draft
LVV-T1841	Verify calculation of scientifically unusable pixel fraction	Draft
LVV-T1842	Verify calculation of zeropoint error fraction exceeding the outlier limit	Draft
LVV-T1843	Verify calculation of significance of imperfect crosstalk corrections	Draft
LVV-T1844	Verify calculation of u-band photometric zero-point RMS	Draft
LVV-T1845	Verify accuracy of photometric transformation to physical scale	Draft
LVV-T1846	Verify calculation of band-to-band color zero-point accuracy including u-band	Draft
LVV-T1847	Verify calculation of sensor fraction with unusable pixels	Draft
LVV-T1862	Verify determining effectiveness of dark current frame	Draft
LVV-T1863	Verify ability to process Special Programs data alongside normal processing	Draft

Test Id	Test Name	
LVV-T1865	Verify implementation of time to L1 public release for Special Programs	Draft
LVV-T1866	Verify latency of reporting optical transients from Special Programs	Draft
LVV-T1867	Verify implementation of at least numStreams alert streams supported	Draft
LVV-T1868	Verify implementation of alert streams distributed within latency limit	Draft
LVV-T2091	Verify Fraction of Alerts Transmitted Within Latency Threshold	Draft
LVV-T2092	Verify Meeting Threshold for Max Fraction of Visits With Failed Alerts	Draft
LVV-T2093	Verify Latency of Reporting Transients	Draft
LVV-T2094	Verify Peak Number of Alerts Per Standard Visit	Draft
LVV-T2095	Verify Max Fraction of Visits With Alert Delays	Draft
LVV-T2096	Verify Handling of Peak Number of Alerts	Draft
LVV-T2097	Verify Handling of Average Number of Alerts	Draft
LVV-T2176	Per-image limit on the median residual ellipticity correlations at scales greater than or equal to 5 arcmin.	Draft
LVV-T2177	Per-image limit on the median residual ellipticity correlations at scales less than to 5 arcmin.	Draft
LVV-T2297	Verify implementation of Science Data Archive	Draft
LVV-T2302	Verify the minimum number of simultaneous users retrieving a set of postage stamp images	Draft
LVV-T2303	Verify Image Archive	Draft
LVV-T2304	Verify maximum number of stars associated with a DIASource.	Draft
LVV-T2305	Verify radius considered nearby	Draft
LVV-T2328	Verify regeneration of un-archived Data Products (Services)_1	Draft
LVV-T2329	Verify the archiving of ancilliary data	Draft
LVV-T2330	Verify that the data processing infrastructure for user computing exists	Draft
LVV-T2331	Verify the number of precovery serivce connections	Draft
LVV-T2332	Verify the time to retrieve results from a query of the prompt products database	Draft
LVV-T2333	Verify the minimum number of simultaneous users querying the prompt products database.	Draft
LVV-T2334	Verify implementation of processed visit images - snaps	Draft

Test Id	Test Name	
LVV-T2692	Verify implementation of Image Metadata Access	Draft
LVV-T2693	Verify implementation of Image Provenance Access	Draft
LVV-T2694	Verify implementation of File Data Product Access	Draft
LVV-T2695	Verify implementation of file metadata access	Draft
LVV-T2696	Verify implementation of file provenance access	Draft
LVV-T2697	Verify implementation of Catalog Data Product Access	Draft
LVV-T2698	Verify implementation of Catalog Metadata Access	Draft
LVV-T2699	Verify implementation of Catalog Provenance Access	Draft
LVV-T2700	Verify Result latency for high-volume complex queries	Draft
LVV-T2724	Verify Result latency for high-volume full-sky queries on the Object table	Draft

4 Active Test Cases

This section documents all active test cases that have a status in the Jira/ATM system of Draft, Defined or Approved.

4.1 Defined Test Cases

4.1.1 LVV-T29 - Verify implementation of Raw Science Image Data Acquisition

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Kian-Tat Lim

Open LVV-T29 in Jira

4.1.1.1 Verification Elements

None.

4.1.1.2 Test Items

Verify acquisition of raw data from L1 Test Stand DAQ while simulating all modes

4.1.1.3 Test Procedure

Step 1	Description

Ingest raw data from L1 Test Stand DAQ, simulating each observing mode

Expected Result

Step 2	Description

Observe image and its metadata is present and queryable in the Data Backbone.

Expected Result
Well-formed image data with appropriate associated metadata.

4.1.2 LVV-T30 - Verify implementation of Wavefront Sensor Data Acquisition

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Kian-Tat Lim

Open LVV-T30 in Jira

4.1.2.1 Verification Elements

None.

4.1.2.2 Test Items

Verify successful ingestion of wavefront sensor data from L1 Test Stand DAQ while simulating all modes.

4.1.2.3 Test Procedure

Step 1	Description
Ingest wavefront sensor data from L1 Test Stand DAQ while simulating all modes	
	Expected Result

Step 2	Description
Observe wavefront sensor data and metadata archived in the Data Backbone.	
	Expected Result
Well-formed wavefront sensor image data with appropriate associated metadata.	

4.1.3 LVV-T32 - Verify implementation of Raw Image Assembly

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Kian-Tat Lim

Open LVV-T32 in Jira

4.1.3.1 Verification Elements

None.

4.1.3.2 Test Items

Verify that the raw exposure data from all readout channels in a sensor can be assembled into a single image, and that all required/relevant metadata are associated with the image data.

4.1.3.3 Test Procedure

Step 1	Description
Ingest data from the L1 Camera Test Stand DAQ.	
	Expected Result

Step 2	Description
Simulate all different modes of data gathering.	
	Expected Result

Step 3	Description
Verify that a raw image is constructed in correct format.	
	Expected Result
A single raw image combining data from all readout channels for a given sensor.	

Step 4	Description

Verify that a raw image is constructed with correct metadata.

	Expected Result

Image header or ancillary table contains the required metadata about the observing context in which data were gathered.

4.1.4 LVV-T33 - Verify implementation of Raw Science Image Metadata

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Kian-Tat Lim

Open LVV-T33 in Jira

4.1.4.1 Verification Elements

None.

4.1.4.2 Test Items

Verify successful ingestion of raw data from L1 Test Stand DAQ and that image metadata is present and queryable.

4.1.4.3 Predecessors

LVV-T29, LVV-T32

4.1.4.4 Test Procedure

Step 1	Description

Identify (or gather) a dataset of raw science images.

	Expected Result

Step 2	Description

Verify that time of exposure start/end, site metadata, telescope metadata, and camera metadata are stored in DMS system.

Expected Result

Raw image data contain the required metadata.

4.1.5 LVV-T34 - Verify implementation of Guider Calibration Data Acquisition

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Kian-Tat Lim

Open LVV-T34 in Jira

4.1.5.1 Verification Elements

None.

4.1.5.2 Test Items

Verify successful
1. Ingestion of calibration frames from L1 Test Stand DAQ
2. Execution of CPP payloads
3. Availability of observed guider calibration products

4.1.5.3 Test Procedure

Step 1	Description

Ingest calibration frames for the guider sensors from L1 Test Stand DAQ

Expected Result

Step 2-1 from LVV-T1060	Description

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

Expected Result

Step 2-2 from LVV-T1060	Description

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

Expected Result

Step 3	Description

Observe that guider calibration products have been produced.

Expected Result

Well-formed calibration frames for the guider sensors.

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

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Eric Bellm

Open LVV-T45 in Jira

4.1.6.1 Verification Elements

None.

4.1.6.2 Test Items

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

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

4.1.6.3 Test Procedure

Step 1	Description
Ingest raw data from L1 Test Stand DAQ.	
	Expected Result

Step 2-1 from LVV-T866	Description
Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.	
	Expected Result
An output dataset including difference images and DIASource and DIAObject measurements.	

Step 2-2 from LVV-T866	Description
Verify that the expected data products have been produced, and that catalogs contain reasonable values for measured quantities of interest.	
	Expected Result

Step 3	Description
Load the Prompt Processing QC reports, and observe that a dynamically updated Data Quality Report has become available at the relevant UI.	
	Expected Result
A Prompt Processing QC report is available via a UI, and contains information about the photometric zeropoint, sky brightness, seeing, PSF, and detection efficiency, and possibly other relevant quantities.	

Step 4	Description
Check that a static report is created and archived in a readily-accessible location.	

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

4.1.7 LVV-T47 - Verify implementation of Prompt Processing Calibration Report Definition

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Eric Bellm

Open LVV-T47 in Jira

4.1.7.1 Verification Elements

None.

4.1.7.2 Test Items

Verify that the DMS produces a Prompt Processing Calibration Report. Specifically check that this report is capable of identifying when aspects of the telescope or camera are changing with time.

4.1.7.3 Test Procedure

Step 1	Description
Identify precursor and simulated calibration datasets on which to run the L1 calibration pipeline.	

Expected Result

Step 2-1 from LVV-T1059	Description
Execute the Daily Calibration Products Update payload. The payload uses raw calibration images and information from the Transformed EFD to generate a subset of Master Calibration Images and Calibration Database entries in the Data Backbone.	

Expected Result

Step 2-2 from LVV-T1059	Description

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

	Expected Result

Step 3	Description

Check that a dynamic report is created that triggers alerts if calibrations go out of range.

	Expected Result

A dynamic report is available via UI to users, and if any out-of-spec changes have occurred, alerts have been issued.

Step 4	Description

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

	Expected Result

An archived version of the calibration report is available and will be retained in a static file format.

4.1.8 LVV-T48 - Verify implementation of Exposure Catalog

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Jim Bosch

Open LVV-T48 in Jira

4.1.8.1 Verification Elements

None.

4.1.8.2 Test Items

Verify that the DMS creates an Exposure Catalog that includes
1. Observation datetime, exposure time
2. Filter

3. Dome, telescope orientation and status
4. Calibration status
5. Airmass and zenith
6. Environmental information
7. Per-sensor information

4.1.8.3 Test Procedure

Step 1	Description

Verify that Exposure Catalogs contain the required elements. At present, the form of the exposure catalog is not defined. This information can be found for a given Butler repo from the metadata, but will ultimately be aggregated into a database/table summarizing available exposures.

Expected Result
A list of the required metadata for a set of exposures is returned and both human- and machine-readable.

4.1.9 LVV-T61 - Verify implementation of Associate Sources to Objects

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Jim Bosch

Open LVV-T61 in Jira

4.1.9.1 Verification Elements

None.

4.1.9.2 Test Items

Verify that each Source record contains an ID that associates it with a best guess at the Object it corresponds to.

4.1.9.3 Test Procedure

Step 1-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 2	Description

Read a dataset via the Butler and extract its source and object catalogs.

Expected Result

Step 3	Description

Verify that sources have objects

Expected Result

Step 4	Description

Verify that objects list sources that seem reasonably near them.

Expected Result

4.1.10 LVV-T65 - Verify implementation of Source Catalog

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Jim Bosch

Open LVV-T65 in Jira

4.1.10.1 Verification Elements

None.

4.1.10.2 Test Items

Verify that all Sources produced by the DRP pipelines contain the entries listed in DMS-REQ-0267.

4.1.10.3 Test Procedure

Step 1	Description
Identify a suitable small dataset to process through the DRP.	
	Expected Result

Step 2-1 from LVV-T1064	Description
Process data with the Data Release Production payload, starting from raw science images and generating science data products, placing them in the Data Backbone.	
	Expected Result

Step 3	Description
Confirm that source catalogs have been produced for single visits and coadds, and that it contains the required measurements.	
	Expected Result
A source catalog containing the measured attributes (and associated errors), including location on the focal plane; a static point-source model fit to world coordinates and flux; a centroid and adaptive moments; and surface brightnesses through elliptical multiple apertures that are concentric, PSF-homogenized, and logarithmically spaced in intensity.	

4.1.11 LVV-T66 - Verify implementation of Forced-Source Catalog

Version	Status	Priority	Verification Type	Owner

1	Defined	Normal	Test	Jim Bosch

Open LVV-T66 in Jira

4.1.11.1 Verification Elements

None.

4.1.11.2 Test Items

Verify that all ForcedSources produced by the DRP pipelines contain fluxes measured on difference and direct single-epoch images, associated uncertainties, an Object ID, and a Visit ID.

4.1.11.3 Test Procedure

Step 1-1 from LVV-T987 Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 2 Description

Retrieve the forced-source catalog from the Butler and verify it to be non-empty.

Expected Result

Step 3	Description

Verify that there exist entries in the forced-photometry table for all coadd objects for the PVIs on which the object should appear.

Expected Result

Step 4	Description

Verify that there exist entries in a forced-photometry table for each image for all DIAObjects.

Expected Result

4.1.12 LVV-T82 - Verify implementation of Tracking Characterization Changes Between Data Releases

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Jim Bosch

Open LVV-T82 in Jira

4.1.12.1 Verification Elements

None.

4.1.12.2 Test Items

Verify that small-area subsets of a DR can be retained when most of that DR is retired, for comparison with future DRs.

4.1.12.3 Test Procedure

Step 1	Description

Prepare a second DRP run -> DPDD with different configuration parameters for this second test Data Release.

Expected Result

Step 2-1 from LVV-T1064 Description

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

Expected Result

Step 3 Description

Stage subset of products from first test Data Release to separate storage.

Expected Result

Step 4 Description

Scientifically compare the results of the subset of that region of sky to those in the second test Data Release comparing the results of the DRP Scientific Verification tests.

Expected Result

Diagnostic plots quantifying the differences between scientific outputs between the first and second test datasets.

4.1.13 LVV-T83 - Verify implementation of Bad Pixel Map

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Robert Lupton

Open LVV-T83 in Jira

4.1.13.1 Verification Elements

None.

4.1.13.2 Test Items

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

4.1.13.3 Test Procedure

Step 1	Description
Interrogate the calibRegistry for the metadata associated with a bad pixel map, where the validity range contains the date of interest.	
	Expected Result
A bad pixel map for the requested date has been returned.	

Step 2	Description
Check that the bad pixel pathologies are encoded as at least 32-bit values, and that the various pathologies are represented by different encoding.	
	Expected Result
Bad pixel values can be decoded to determine their pathologies using their 32-bit values.	

4.1.14 LVV-T85 - Verify implementation of Crosstalk Correction Matrix

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Robert Lupton

Open LVV-T85 in Jira

4.1.14.1 Verification Elements

None.

4.1.14.2 Test Items

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

4.1.14.3 Test Procedure

Step 1	Description

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

Expected Result

Step 2-1 from LVV-T1060	Description

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

Expected Result

Step 2-2 from LVV-T1060	Description

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

Expected Result

Step 3	Description

Confirm that the crosstalk correction matrix is produced and persisted.

Expected Result

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

Step 4	Description

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

Expected Result

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

4.1.15 LVV-T88 - Verify implementation of Calibration Data Products

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Robert Lupton

Open LVV-T88 in Jira

4.1.15.1 Verification Elements

None.

4.1.15.2 Test Items

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

4.1.15.3 Test Procedure

Step 1	Description

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

Expected Result

Step 2-1 from LVV-T1060	Description

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

Expected Result

Step 2-2 from LVV-T1060	Description

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

Expected Result

Step 3	Description

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

Expected Result

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

Step 4	Description

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

Expected Result

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

4.1.16 LVV-T89 - Verify implementation of Calibration Image Provenance

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Robert Lupton

Open LVV-T89 in Jira

4.1.16.1 Verification Elements

None.

4.1.16.2 Test Items

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

4.1.16.3 Test Procedure

Step 1	Description

Ingest an appropriate precursor calibration dataset into a Butler repo.

Expected Result

Step 2-1 from LVV-T1060	Description

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

Expected Result

Step 2-2 from LVV-T1060	Description

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

Expected Result

Step 3	Description

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

Expected Result

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

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

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Kian-Tat Lim

Open LVV-T97 in Jira

4.1.17.1 Verification Elements

None.

4.1.17.2 Test Items

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

4.1.17.3 Test Procedure

Step 1 Description

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

Expected Result

Step 2-1 from LVV-T1064 Description

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

Expected Result

Step 3-1 from LVV-T987 Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 4 Description

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

Expected Result

Multiple datasets resulting from processing of the same input data.

Step 5 Description

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

Expected Result

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

4.1.18 LVV-T98 - Verify implementation of Selection of Datasets

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Kian-Tat Lim

Open LVV-T98 in Jira

4.1.18.1 Verification Elements

None.

4.1.18.2 Test Items

Verify that the DMS can identify and retrieve datasets consisting of logical groupings of Exposures, metadata, provenance, etc., or other groupings that are processed or produced as a logical unit.

4.1.18.3 Test Procedure

Step 1-1 from LVV-T987 Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 2	Description

Ingest data from an appropriate processed dataset.

	Expected Result

Step 3	Description

Observe retrieval of single Processed Visit Image (PVI) with metadata.

	Expected Result

A PVI and its associated metadata.

Step 4	Description

Observe retrieval of multiple PVIs with metadata.

	Expected Result

A set of PVIs and their associated metadata.

Step 5	Description

Observe retrieval of coadd patch with metadata and provenance information.

	Expected Result

An image of coadded data in a patch, along with its metadata and information describing the provenance of the patch constituents.

Step 6	Description

Observe retrieval of subset of rows in each of the above catalogs.

	Expected Result

4.1.19 LVV-T103 - Verify implementation of Generate Data Quality Report Within Specified Time

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Kian-Tat Lim

Open LVV-T103 in Jira

4.1.19.1 Verification Elements

None.

4.1.19.2 Test Items

Verify that the DMS can generate a nightly L1 Data Quality Report within **dqReportComplTime = 4[hour]**, in both human- and machine-readable formats.

4.1.19.3 Test Procedure

Step 1	Description
Execute single-day operations rehearsal	
	Expected Result

Step 2	Description
After **dqReportComplTime = 4[hour]** has passed, confirm (via timestamps) that the data quality report has been generated within **dqReportComplTime = 4[hour],** and that it contains the correct contents.	
	Expected Result
Both human- and machine-readable versions of the L1 Data Quality Report are available with dqReportComplTime.	

4.1.20 LVV-T112 - Verify implementation of Alert Filtering Service

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Eric Bellm

Open LVV-T112 in Jira

4.1.20.1 Verification Elements

None.

4.1.20.2 Test Items

Verify that user-defined filters can be used to generate a basic alert filtering service.

4.1.20.3 Test Procedure

Step 1	Description
Identify a suitable precursor dataset for processing through the Alert Production pipeline.	
	Expected Result

Step 2-1 from LVV-T866	Description
Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.	
	Expected Result
An output dataset including difference images and DIASource and DIAObject measurements.	

Step 2-2 from LVV-T866	Description
Verify that the expected data products have been produced, and that catalogs contain reasonable values for measured quantities of interest.	
	Expected Result

Step 3	Description
Confirm that alerts are generated, and that an Alert Distribution service is making them available via a stream.	
	Expected Result
Via either a UI or API, confirmation that a stream of alerts are available.	

Step 4	Description
Confirm that a UI (or API) exists that allows users to define simple filters. Define a filter, and observe both the full and the filtered alert streams to confirm that the filter has reduced the volume of alerts.	

Expected Result
The user-defined filter has reduced the number of alerts being received relative to the full stream.

4.1.21 LVV-T113 - Verify implementation of Performance Requirements for LSST Alert Filtering Service

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Eric Bellm

Open LVV-T113 in Jira

4.1.21.1 Verification Elements

None.

4.1.21.2 Test Items

Verify that the DMS alert filter service provides sufficient bandwidth for **numBrokerUsers = 100** simultaneously-operating brokers to receive up to **numBrokerAlerts = 20** alerts per visit.

4.1.21.3 Test Procedure

Step 1	Description
Create a simulated alert stream.	

Expected Result

Step 2	Description
Simultaneously execute user-defined alert filters for at least **numBrokerUsers = 100** users, and confirm that the system successfully filters the stream as requested. Confirm that the bandwidth requirement of **numBrokerAlerts = 20** per user was met.	

Expected Result
All of the (simulated) users successfully receive their requested filtered alerts, with **numBrokerAlerts = 20** per user.

4.1.22 LVV-T114 - Verify implementation of Pre-defined alert filters

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Eric Bellm

Open LVV-T114 in Jira

4.1.22.1 Verification Elements

None.

4.1.22.2 Test Items

Verify that users of the Alert Filtering service can use a predefined set of filters.

4.1.22.3 Test Procedure

Step 1	Description

Create a simulated alert stream. Confirm that alerts are generated, and that an Alert Distribution service is making them available.

Expected Result

A stream of alerts that is confirmed to be generated and distributed.

Step 2	Description

Confirm that a UI (or API) exists that presents users some pre-defined filters.

Expected Result

The UI (or API) for accessing alert streams has some pre-defined filters available for users.

Step 3	Description

Select one of the pre-defined filters, and confirm that the results have been properly filtered.

Expected Result

After applying the pre-defined filter, the number of alerts has decreased relative to the raw stream.

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

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Jeffrey Carlin

Open LVV-T124 in Jira

4.1.23.1 Verification Elements

None.

4.1.23.2 Test Items

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

4.1.23.3 Test Procedure

Step 1-1 from LVV-T860	Description

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

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

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

Example Code

source 'path'
setup lsst_distrib

Expected Result

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

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

Step 2 **Description**

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

Expected Result

Step 3 **Description**

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

Expected Result

Step 4-1 from LVV-T987 **Description**

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 5	Description

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

Expected Result

Step 6	Description

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

Expected Result

Step 7	Description

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

Expected Result

Step 8	Description

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

Expected Result

4.1.24 LVV-T131 - Verify implementation of Provide User Interface Services

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Gregory Dubois-Felsmann

Open LVV-T131 in Jira

4.1.24.1 Verification Elements

None.

4.1.24.2 Test Items

Verify the availability and functionality of the broad range of user interface services called for in the requirement, as applied to both Nightly and DRP data. This will primarily be done by verifications performed at the LSST Science Platform level, based on the requirements in LDM-554; however, a high-level set of tests corresponding to the DMS-REQ-0160 requirement are defined below.

4.1.24.3 Environment Needs

4.1.24.3.1 Hardware

As noted in Verification Configuration, the systems required to carry out the tests include both an "inside" test execution platform - the ability to execute test notebooks within the Science Platform Notebook Aspect - and an "outside" test execution platform with connectivity to the Science Platform instance under test that is comparable to that available to offsite science users.

4.1.24.4 Test Procedure

Step 1	Description

Establishment of test coordinates:

Establish sky positions and surrounding regions (e.g., cones or polygons), field sizes, filter bands, and temporal epochs for the tests that are consistent with the known content of the test dataset, whether precursor or LSST commissioning data.

Establishing sky positions should include pre-determining the corresponding LSST "tract and patch" identifiers.

If the plan to not keep all calibrated single-epoch images on disk is still in place at the time of the test, identify for use in the test both images that are, and are not, on disk.

Establish target image boundaries, projections, and pixel scales to be used for resampling tests. Ensure that at least some of these test conditions include coadded image boundaries that cross tract and patch boundaries, and single-epoch image boundaries that cross focal plane raft boundaries.

	Expected Result

Step 2	Description

Butler image access:

From within the Notebook Aspect, verify that coadded images for the identified regions of sky and filter bands are accessible via the Butler. Verify that the same images are available whether obtained by direct reference to the previous established tract/patch identifiers or by the use of LSST stack code for retrieving images based on sky coordinates.
From within the Notebook Aspect, verify that single-epoch raw images for the selected locations and times are available. Verify that calibrated images (PVIs) for the selected locations and times are available; depending on the details of the test dataset, verify that PVIs still on disk can be retrieved immediately.
Verify that lists or tables of image metadata, not just individual images, can be retrieved. E.g., a list of all the single-epoch images covering a selected sky location.

Expected Result

Step 3 Description

Programmatic PVI re-creation:
From within the Notebook Aspect, verify that the recreation on demand of a PVI can be performed. Ideally, this should be done as follows:

- Verify that recreation of a PVI that *is* still available works and that it reproduces the original PVI exactly (except for provenance metadata that must be different) or within the reasonable ability of processing systems to do so (e.g., taking into account that the original calibration and the recreation may have run on different CPU architectures).
- The test conditions should ensure the verification that a recreation was actually performed, i.e., that the still-available PVI was not returned instead.
- Note that it does not appear to be a requirement that *at Butler level* recreation on demand of PVIs is a completely transparent process. If this *is* decided to be a requirement, the test must also verify that it has been satisfied. If it is *not* a requirement, verify that adequate documentation on the PVI-recreation process (e.g., the SuperTasks and configuration to be used) is available.

Expected Result

Step 4 Description

Butler catalog access:
From within the Notebook Aspect, verify that all the catalog data products described in the DPDD can be retrieved for the coordinates selected above via the Butler. (This test should include access to SSObject data, but the details of how such a test would depend on the coordinate selections require additional thought.)

Expected Result

Step 5 Description

LSST-stack-based resampling/reprojection:
Verify the availability of software in the LSST stack, and associated documentation, that permits the resampling of LSST images to different pixel grids and projections.
Exercise this capability for the test conditions selected in Step 1 above.
Perform photometric and astrometric tests on the resulting resampled images to provide evidence that the transformations performed were correct to the accuracy supported by the data.

Expected Result

Step 6 Description

Comment:

The following API Aspect test steps should be carried out on the required "offsite-like" test platform, to ensure that their success does not reflect any privileged access given to processes inside the Data Access Center or other Science Platform instance. However, at least a small sampling of them should *also* be carried out *within* the Science Platform environment, i.e., in the Notebook Aspect, and the results compared.

Expected Result

Step 7 Description

API Aspect image access:

Using IVOA services such as the Registry and ObsTAP, from the "offsite-like" test platform, verify that the existence of the classes of image data products foreseen in the DPDD can be determined.

Verify that ObsTAP and/or SIAv2 can be used to find the same images and lists of images for the established test coordinates that were retrieved via the Butler in Step 2 above.

Verify that the selected images are retrievable from the Web services.

Verify that the retrieved images are identical in their pixel content and metadata.

The tests must include both coadded and single-epoch images.

Expected Result

Step 8 Description

API Aspect image transformations:

Verify that image cutouts and resamplings can be performed via the IVOA SODA service, and that the results are identical to those obtained for the same parameters from the LSST-stack-based tests in Step 5.

(The requirements for supported reprojections, if any, in the SODA service have not been established at the time of writing.)

Expected Result

Step 9 Description

API Aspect catalog data access:

Verify that the IVOA Registry, RegTAP, TAP_SCHEMA, and other relevant mechanisms can be used to discover the existence of all the catalog data products foreseen in the DPDD.

Using the IVOA TAP service, verify that all the catalog data products foreseen in the DPDD can be retrieved for the coordinates determined in Step 1. Verify that their scientific content is the same as when they are retrieved via the Butler.

Expected Result

Step 10	Description

Comment:
The Portal Aspect tests below should be carried out from a web browser on an "offsite-like" test platform, to ensure that no privileged access provided to intra-data-center clients is relied upon.

Expected Result

Step 11	Description

Portal Aspect data browsing:
Verify that the Portal Aspect can be used to discover the existence of all the data products foreseen in the DPDD. Verify that the UI permits locating the data for the coordinates selected in Step 1 by visual means, e.g., by zooming and panning in from an all-sky view.
Verify that the UI permits locating the data by typing in coordinates as well.

Expected Result

Step 12	Description

Portal Aspect image access:
Verify that the Portal Aspect allows both the retrieval of "original" image data, i.e., in its native LSST pixel projection and with full metadata, as well as retrieval of on-demand UI cutouts of coadded image data for selected locations.

Expected Result

Step 13	Description

Portal Aspect catalog query and visualization:
Verify that the Portal Aspect allows graphical querying of DPDD catalog data, both coadded and single-epoch, for selected regions of sky and/or with selected properties, and supports the visualization of the results (including histogramming, scatterplots, time series, table manipulations, and overplotting on image data).
(Note that the Science Platform requirements, LDM-554, lay out a detailed set of requirements on the selection and visualization of catalog data.)

Expected Result

Step 14	Description

Portal Aspect data download:
Verify that data identified and/or visualized in the Portal Aspect can be downloaded to the remote system running the web browser in which the Portal is displayed, as well as to the User Workspace.

Expected Result

4.1.25 LVV-T136 - Verify implementation of Image Data Product Access

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Colin Slater

Open LVV-T136 in Jira

4.1.25.1 Verification Elements

None.

4.1.25.2 Test Items

Verify that available image data products can be listed and retrieved.

4.1.25.3 Test Procedure

Step 1	Description

Details of the Gen3 Butler and ObsTAP tables are still being worked out. The general overview of this test will be to use some combination of the Gen3 Butler and TAP access to the ObsTAP tables to test that the required access is provided.

Expected Result

Verification that the relevant data products and their related tables, metadata, and provenance information are available and readily accessible.

4.1.26 LVV-T140 - Verify implementation of Production Orchestration

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Leanne Guy

Open LVV-T140 in Jira

4.1.26.1 Verification Elements

None.

4.1.26.2 Test Items

Demonstrate use of orchestration software to perform real-time and batch production on LSST compute platform(s).

4.1.26.3 Test Procedure

Step 1	Description
Identify an appropriate precursor dataset.	
	Expected Result

Step 2	Description
Execute a batch processing job using the orchestration system, and confirm (manually and/or via QA tools typically used for HSC reprocessing) that the pipeline executed and produced all expected products (or error logs in cases of failure).	
	Expected Result
Calexp single-visit and coadd images, and associated catalogs, are present in a Butler repository. Logs of the processing are available to be inspected for identification of problems in the processing.	

4.1.27 LVV-T141 - Verify implementation of Production Monitoring

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Robert Gruendl [X]

Open LVV-T141 in Jira

4.1.27.1 Verification Elements

None.

4.1.27.2 Test Items

Demonstrate monitoring capabilities that give real-time view of pipeline execution and production systems usage/load.

4.1.27.3 Predecessors

LVV-T140

4.1.27.4 Test Procedure

Step 1-1 from LVV-T1064	Description

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

	Expected Result

Step 2	Description

While DRP processing is executing, monitor the progress and resource usage of processing.

	Expected Result

Ability to monitor in real-time the orchestrated production processing, including resource usage.

4.1.28 LVV-T150 - Verify implementation of Maintain Archive Publicly Accessible

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Colin Slater

Open LVV-T150 in Jira

4.1.28.1 Verification Elements

None.

4.1.28.2 Test Items

Verify that prior data releases remain accessible.

4.1.28.3 Test Procedure

Step 1	Description
Confirm that at least two data releases (the most recent, and one previous) are accessible to users (and can be queried) from the standard channels.	
	Expected Result
Simple queries return catalog data from the data releases that are available in QSERV.	

Step 2	Description
Confirm that previous data releases are accessible for bulk download (perhaps with significant latency) from tape or other bulk store, and that the downloaded tables contain the expected data products.	
	Expected Result
A download of an entire previous data release from its bulk store.	

4.1.29 LVV-T153 - Verify implementation of Provide Engineering and Facility Database Archive

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Robert Gruendl [X]

Open LVV-T153 in Jira

4.1.29.1 Verification Elements

None.

4.1.29.2 Test Items

Demonstrate Engineering and Facilities Data (images, associated metadata, and observatory environment and control data) are archived and available for public access within **L1PublicT (24 hours)**.

4.1.29.3 Test Procedure

Step 1	Description
Execute a single-day operations rehearsal, ingesting (simulated) OCS commands into the EFD.	
	Expected Result

Step 2	Description
Wait at least **L1PublicT=24** hours, then access the archived EFD. Confirm that the data products are present in the archived EFD after **L1PublicT=24** hours have elapsed.	
	Expected Result
The EFD contains the simulated OCS commands, and they were ingested within **L1PublicT=24** hours of the operations rehearsal.	

Step 3	Description
From the public access portal to the EFD, execute a query and demonstrate that the data are publicly available.	
	Expected Result
A query at the public interface to the EFD successfully executes and returns EFD data.	

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

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Gregory Dubois-Felsmann

Open LVV-T183 in Jira

4.1.30.1 Verification Elements

None.

4.1.30.2 Test Items

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

4.1.30.3 Test Procedure

Step 1	Description
From the Base Site, connect to the (simulated) OCS telemetry stream.	
	Expected Result

Step 2	Description
Send a command to the OCS, and observe that the command has been executed.	
	Expected Result
Confirmation that the OCS command successfully executed.	

Step 3	Description
Extract information from the telemetry being broadcast by the OCS, and ensure that these data are readable.	
	Expected Result
A readable extract from the OCS telemetry stream.	

4.1.31 LVV-T385 - Verify implementation of minimum number of simultaneous retrievals of CCD-sized coadd cutouts

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Test	Leanne Guy

Open LVV-T385 in Jira

4.1.31.1 Verification Elements

None.

4.1.31.2 Test Items

Verify that at least **ccdRetrievalUsers = 20** users can simultaneously retrieve a single CCD-sized coadd cutout using the IVOA SODA protocol.

4.1.31.3 Test Procedure

Step 1	Description
Confirm that CCD-sized cutouts from coadds, also containing mask and variance planes, are available on the SODA server. If none are available, copy an image (or some images) to the server.	
	Expected Result
At least one CCD-sized coadd cutout is available, and is a well-formed image.	

Step 2	Description
Simulate SODA queries by at least **ccdRetrievalUsers = 20** users at the same time.	
	Expected Result

Step 3	Description
Confirm that all simulated users retrieved the desired image(s), and that the returned images are well-formed, with (at least) image, mask, and variance planes.	
	Expected Result
All of the simulated **ccdRetrievalUsers = 20** users retrieved images within the specified time (see related Verification Element and Test Case).	

4.1.32 LVV-T1252 - Verify number of simultaneous alert filter users

Version	Status	Priority	Verification Type	Owner

1	Defined	Normal	Test	Eric Bellm

Open LVV-T1252 in Jira

4.1.32.1 Verification Elements

None.

4.1.32.2 Test Items

Verify that the DMS alert filter service supports **numBrokerUsers = 100** simultaneous brokers.

4.1.32.3 Test Procedure

Step 1	Description

Create a simulated alert stream.

Expected Result

Step 2	Description

Simultaneously execute user-defined alert filters for at least **numBrokerUsers = 100** users, and confirm that the system successfully filters the stream as requested. Confirm that the bandwidth requirement of **numBrokerAlerts = 20** per user was met.Simultaneously execute user-defined alert filters for at least 100 users, and confirm that the system successfully filters the stream as requested.

Expected Result

All of the (simulated) **numBrokerUsers = 100** users successfully receive their requested filtered alerts.

4.1.33 LVV-T1332 - Verify implementation of maximum time for retrieval of CCD-sized coadd cutouts

Version	Status	Priority	Verification Type	Owner

1	Defined	Normal	Test	Leanne Guy

Open LVV-T1332 in Jira

4.1.33.1 Verification Elements

None.

4.1.33.2 Test Items

Verify that at least **ccdRetrievalUsers = 20** users can retrieve CCD-sized coadd cutouts using the IVOA SODA protocol within a maximum retrieval time of **ccdRetrievalTime = 15 seconds**.

4.1.33.3 Test Procedure

Step 1	Description

Confirm that CCD-sized cutouts from coadds, also containing mask and variance planes, are available on the SODA server. If none are available, copy an image (or some images) to the server.

Expected Result

At least one CCD-sized coadd cutout is available, and is a well-formed image.

Step 2	Description

Simulate SODA queries by at least **ccdRetrievalUsers = 20** users at the same time.

Expected Result

Step 3	Description

Monitor the time that each query takes to complete, and confirm that all simulated users retrieved the desired image(s) within **ccdRetrievalTime = 15 seconds.**

Expected Result

All of the simulated **ccdRetrievalUsers = 20** users retrieved images within **ccdRetrievalTime = 15 seconds.**

4.2 Approved Test Cases

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Colin Slater

Open LVV-T28 in Jira

4.2.1.1 Verification Elements

None.

4.2.1.2 Test Items

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

4.2.1.3 Test Procedure

Step 1-1 from LVV-T987 Description

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

Example Code

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

Expected Result

Butler repo available for reading.

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

Step 3	Description
Verify that the single-visit catalog provides measurements in flux units.	
	Expected Result
Confirmation of measurements in catalogs encoded in flux units.	

4.2.2 LVV-T38 - Verify implementation of Processed Visit Images

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Eric Bellm

Open LVV-T38 in Jira

4.2.2.1 Verification Elements

None.

4.2.2.2 Test Items

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

The verification should include confirming that the images have been trimmed of the overscan, and that correction of the instrumental signature (including crosstalk) has been applied properly.

4.2.2.3 Test Procedure

Step 1	Description

Identify suitable precursor datasets containing unprocessed raw images.

Expected Result

Step 2-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 3	Description

Run the initial steps (including instrument signature removal and calibration) of Data Release (or Prompt) Processing on these data. Verify that Processed Visit Images are generated at the correct size and with significant instrumental artifacts removed.

Expected Result

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

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jim Bosch

Open LVV-T39 in Jira

4.2.3.1 Verification Elements

None.

4.2.3.2 Test Items

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

4.2.3.3 Test Procedure

Step 1	Description

Identify a dataset with processed visit images in multiple filters.

Expected Result

Step 2-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 3	Description

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

Expected Result

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

Step 4	Description

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

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

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jim Bosch

Open LVV-T40 in Jira

4.2.4.1 Verification Elements

None.

4.2.4.2 Test Items

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.

4.2.4.3 Test Procedure

Step 1	Description
Identify an appropriate processed dataset for this test.	

Expected Result
A dataset with Processed Visit Images available.

Step 2-1 from LVV-T987	Description
Identify the path to the data repository, which we will refer to as 'DATA/path', then execute the following:	

Example Code

```
from lsst.daf.butler import Butler
```

```
repo = 'Data/path'
collection = 'collection'
butler = Butler(repo, collections=collection)
```

	Expected Result

Butler repo available for reading.

Step 3	Description

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

	Expected Result

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

Step 4	Description

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

	Expected Result

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

Step 5	Description

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

	Expected Result

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

Step 6	Description

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

	Expected Result

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

Step 7	Description

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

	Expected Result

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

Step 8	Description

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

	Expected Result

astrometricAccuracy requirement is met over the entire image.

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jim Bosch

Open LVV-T41 in Jira

4.2.5.1 Verification Elements

None.

4.2.5.2 Test Items

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.

4.2.5.3 Test Procedure

Step 1	Description

Identify a dataset with processed visit images in multiple filters.

	Expected Result

Step 2-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 3	Description

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

Expected Result

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

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jim Bosch

Open LVV-T42 in Jira

4.2.6.1 Verification Elements

None.

4.2.6.2 Test Items

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

4.2.6.3 Test Procedure

Step 1-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 2	Description

Ingest the data from an appropriate processed dataset.

Expected Result

Step 3	Description

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

Expected Result

Step 4	Description

Inspect the calexp image to ensure that

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

Expected Result

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

Step 5	Description

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

Expected Result

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

Step 6	Description

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

Expected Result

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

Step 7	Description

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

Expected Result

4.2.7 LVV-T43 - Verify implementation of Background Model Calculation

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jim Bosch

Open LVV-T43 in Jira

4.2.7.1 Verification Elements

None.

4.2.7.2 Test Items

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.

4.2.7.3 Predecessors

LVV-T15

LVV-T19

4.2.7.4 Test Procedure

Step 1	Description

Identify a dataset with processed visit images in multiple filters.

	Expected Result

Step 2-1 from LVV-T987	Description

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

	Example Code

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

	Expected Result

Butler repo available for reading.

Step 3	Description

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.

Step 4	Description

Confirm that the pixel values of the calexp + calexpBackground are approximately equal to those of the postISRCCD image.

Expected Result
All calexp+calexpBackground images should have pixel values *approximately* equal to those of postISRCCD images. Small differences are expected due to cosmic-ray repair and other similar corrections, but the median should be equal.

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

Version	Status	Priority	Verification Type	Owner
2	Approved	Normal	Test	Jim Bosch

Open LVV-T62 in Jira

4.2.8.1 Verification Elements

None.

4.2.8.2 Test Items

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.

4.2.8.3 Test Procedure

Step 1	Description
Identify a dataset with coadded images in multiple filters.	

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

Step 2-1 from LVV-T987	Description
Identify the path to the data repository, which we will refer to as 'DATA/path', then execute the following:	

Example Code

```
from lsst.daf.butler import Butler
```

```
repo = 'Data/path'
collection = 'collection'
butler = Butler(repo, collections=collection)
```

Expected Result
Butler repo available for reading.

Step 3	Description

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

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

4.2.9 LVV-T74 - Verify implementation of Template Coadds

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Eric Bellm

Open LVV-T74 in Jira

4.2.9.1 Verification Elements

None.

4.2.9.2 Test Items

Verify that the DMS can produce Template Coadds for DIA processing.

4.2.9.3 Test Procedure

Step 1-1 from LVV-T866	Description

Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.

	Expected Result

An output dataset including difference images and DIASource and DIAObject measurements.

Step 1-2 from LVV-T866	Description

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

	Expected Result

Step 2	Description

Confirm that the template coadds have been created and are well-formed.

	Expected Result

4.2.10 LVV-T77 - Verify implementation of Best Seeing Coadds

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jim Bosch

Open LVV-T77 in Jira

4.2.10.1 Verification Elements

None.

4.2.10.2 Test Items

Verify that the DRP pipelines produce a suite of per-band coadds with input images filtered to optimize the size of the effective PSF on the coadd.

4.2.10.3 Test Procedure

Step 1-1 from LVV-T860	Description

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

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

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

Example Code

source 'path'
setup lsst_distrib

Expected Result

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

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

Step 2-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 3	Description

Explicitly create a coadd for a specified seeing range in each filter.

	Expected Result

Step 4	Description

Verify that these coadds exist.

	Expected Result

4.2.11 LVV-T78 - Verify implementation of Persisting Data Products

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Kian-Tat Lim

Open LVV-T78 in Jira

4.2.11.1 Verification Elements

None.

4.2.11.2 Test Items

Verify that per-band deep coadds and best-seeing coadds are present, kept, and available.

4.2.11.3 Test Procedure

Step 1-1 from LVV-T987	Description

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

	Example Code

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

Expected Result

Butler repo available for reading.

Step 2	Description

Identify some single-band deep coadds and retrieve them from the butler

Expected Result

Step 3	Description

Examine the deep coadds and confirm that they are well-formed images

Expected Result

Step 4	Description

Identify some single-band best-seeing coadds and retrieve them from the butler

Expected Result

Step 5	Description

Examine the best-seeing coadds and confirm that they are well-formed images

Expected Result

4.2.12 LVV-T84 - Verify implementation of Bias Residual Image

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T84 in Jira

4.2.12.1 Verification Elements

None.

4.2.12.2 Test Items

Verify that DMS can construct a bias residual image that corrects for temporally-stable bias structures.
Verify that DMS can do this on demand.

4.2.12.3 Test Procedure

Step 1	Description

Identify the location of an appropriate precursor dataset.

Expected Result

Step 2-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 3	Description

Import the standard libraries required for the rest of this test:

Example Code

```
import os
import lsst.afw.display as afwDisplay
from lsst.daf.persistence import Butler
from lsst.ip.isr import IsrTask
```

Expected Result

Step 4	Description

Ingest the dataset from step 1 using the Butler (e.g., following example code below).

Example Code

```
butler = Butler($REPOSITORY_PATH)
raw = butler.get("raw", visit=$VISIT_ID, detector=2)
bias = butler.get("bias", visit=$VISIT_ID, detector=2)
```

Expected Result

Step 5	Description

Display the bias image and inspect that its pixels contain unique values.

Expected Result

A relatively flat image showing the bias level with roughly Poisson noise.

Step 6	Description

Configure and run an Instrument Signature Removal (ISR) task on the raw data. Most corrections are disabled for simplicity, but the bias frame is applied.

Example Code

```
isr_config = IsrTask.ConfigClass()
isr_config.doDark=False
isr_config.doFlat=False
isr_config.doFringe=False
isr_config.doDefect=False
isr_config.doLinearize=False
isr = IsrTask(config=isr_config)
result = isr.run(raw, bias=bias, detectorNum=raw.detector.getId(), camera=obs_lsst.LsstCamImSim.getCamera())
```

Expected Result
A trimmed, bias-corrected image in 'result'.

Step 7	Description
Display the 'result' image and confirm that the bias correction has been performed.	

Expected Result
A displayed image with bias removed (i.e., typical background counts reduced relative to the raw frame).

4.2.13 LVV-T90 - Verify implementation of Dark Current Correction Frame

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T90 in Jira

4.2.13.1 Verification Elements

None.

4.2.13.2 Test Items

Verify that the DMS can produce a dark correction frame calibration product.

4.2.13.3 Test Procedure

Step 1	Description
Identify the path to a dataset containing dark frames (i.e., exposures taken with the shutter closed).	

Expected Result

Step 2	Description
Execute the relevant steps from 'cp_pipe' (the calibration pipeline) to produce dark correction frames.	

Expected Result

Step 3	Description

Inspect the resulting dark correction frame to confirm that it appears as expected.

Expected Result

A well-formed dark correction frame is present and accessible via the Data Butler.

4.2.14 LVV-T91 - Verify implementation of Fringe Correction Frame

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T91 in Jira

4.2.14.1 Verification Elements

None.

4.2.14.2 Test Items

Verify that the DMS can produce an fringe-correction frame calibration product.
Verify that the DMS can determine the effectiveness of the fringe-correction frame and determine how often it should be updated.

4.2.14.3 Test Procedure

Step 1	Description

Execute Test Case LVV-T88, which runs the calibration products pipeline.

Expected Result

Step 2	Description

Examine the fringe-correction frames created by the pipeline to ensure that they are well-formed.

Expected Result

Fringe frame is an lsst.afw.image.Exposure with reasonable pixel values.

Step 3	Description

Apply the fringe correction to a science image and confirm that it has the desired effect.

Expected Result

Images before and after correction have different pixel values.

4.2.15 LVV-T115 - Verify implementation of Calibration Production Processing

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Kian-Tat Lim

Open LVV-T115 in Jira

4.2.15.1 Verification Elements

None.

4.2.15.2 Test Items

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

4.2.15.3 Test Procedure

Step 1	Description

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

Expected Result

Step 2-1 from LVV-T1060 Description

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

Expected Result

Step 2-2 from LVV-T1060 Description

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

Expected Result

Step 3 Description

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

Expected Result

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

4.2.16 LVV-T125 - Verify implementation of Simulated Data

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Robert Lupton

Open LVV-T125 in Jira

4.2.16.1 Verification Elements

None.

4.2.16.2 Test Items

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

This should be verified for simulated stars and fake galaxies. Furthermore, verification should include the following:

- Demonstration that fluxes of simulated stars and galaxies are recovered to within ~10% of their true values,
- Demonstration that artificial sources are recovered to the completeness levels that are required,
- Demonstration that star/galaxy identification is correct for a reasonable fraction of simulated sources.

4.2.16.3 Test Procedure

Step 1	Description
Identify a dataset that has been (or can be readily) processed through single-frame processing and coaddition.	
	Expected Result
The 'calexp' and 'deepCoadd_calexp' images and their associated source catalogs are created.	

Step 2	Description
Roughly determine the coordinates of a bounding box that is contained within the images that were processed.	
	Expected Result
RA, Dec boundaries of a region in which to generate fake sources.	

Step 3	Description
Generate a catalog in the correct format for 'insertFakes' to accept. The catalog should specify positions and magnitudes of stars (and optionally, parameters specifying galaxy shape, if galaxies are also being inserted).	
	Expected Result
An input catalog of fake source positions and magnitudes to be inserted into the images.	

Step 4	Description
Execute 'insertFakes.py' on the repository, specifying the input catalog from the previous step.	
	Expected Result
A repository with images that have fake sources inserted.	

Step 5	Description
Run 'multiBandDriver.py' on the repository, specifying the fake-source repository as the input.	
	Expected Result
'calexp' and coadd images containing the artificial sources and sources catalogs that contain their measurements along with the sources detected in the original run.	

Step 6	Description
Confirm that the injected sources appear in the images and the catalogs.	
	Expected Result
Fake sources and their measured properties are recoverable.	

4.2.17 LVV-T126 - Verify implementation of Image Differencing

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Eric Bellm

Open LVV-T126 in Jira

4.2.17.1 Verification Elements

None.

4.2.17.2 Test Items

Verify that the DMS can perform image differencing from single exposures and coadds.

4.2.17.3 Test Procedure

Step 1	Description
Identify a repository containing data that have been processed through the difference imaging pipeline. (e.g., the HiTS 2015 data that are processed monthly for testing)	

Expected Result

A dataset containing calexps, difference images, and source catalogs (of diaSrcs).

Step 2-1 from LVV-T987 Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 3 Description

Extract a 'calexp', a 'deepDiff_differenceExp', and the 'deepDiff_diaSrc' catalog of measurements.

Expected Result

Well-formed images and catalogs containing the calexp from the visit image and the difference image, and measurements of sources from the difference image.

Step 4 Description

Confirm (by visual inspection) that the difference image is mostly blank sky (i.e., has had a template of the same field subtracted), and that the source catalog contains sources with photometric and astrometric measurements.

Expected Result

A mostly blank image (with perhaps some artifacts due to imperfect subtraction) and a catalog of sources detected/measured from that image.

4.2.18 LVV-T127 - Verify implementation of Provide Source Detection Software

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T127 in Jira

4.2.18.1 Verification Elements

None.

4.2.18.2 Test Items

Verify that the DMS provides source detection software that can be applied to calibrated images, including both difference images and coadds. This will be verified using simulated data, but could also be done by inserting artificial sources into existing datasets.

4.2.18.3 Test Procedure

Step 1	Description
Run DRP and AP processing, including source detection and measurement algorithms, on a small portion of the data from a simulated dataset.	
	Expected Result
Source catalogs containing measurements of all sources detected in the input images.	

Step 2	Description
Confirm that the output repos contain catalogs of source detections. Compare these output catalogs to the original simulated source catalogs, and confirm that a large fraction of the sources within a reasonable signal-to-noise range were recovered.	
	Expected Result
Most sources above a reasonable S/N threshold were detected, and their measured fluxes are reasonably close to the simulated inputs.	

4.2.19 LVV-T129 - Verify implementation of Provide Calibrated Photometry

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T129 in Jira

4.2.19.1 Verification Elements

None.

4.2.19.2 Test Items

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

4.2.19.3 Test Procedure

Step 1-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 2	Description

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

Expected Result

Step 3	Description

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

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

Step 4	Description
Ingest the data products from an appropriate AP processing dataset.	

Expected Result

Step 5	Description
Confirm that AB-calibrated magnitudes and fluxes are available for all measured Sources, DIASources, and Objects. [An enhanced verification could include matching the sources to an external source catalog and comparing the magnitudes to show that they are well-calibrated.]	

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

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Robert Gruendl [X]

Open LVV-T132 in Jira

4.2.20.1 Verification Elements

None.

4.2.20.2 Test Items

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.

4.2.20.3 Test Procedure

Step 1	Description

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

Expected Result

Step 2	Description

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

Expected Result

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

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Robert Lupton

Open LVV-T133 in Jira

4.2.21.1 Verification Elements

None.

4.2.21.2 Test Items

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

4.2.21.3 Test Procedure

Step 1	Description

On the LSST development cluster or notebook aspect, git clone the repo containing the CBP package: `https://github.com/lsst/cbp`

Expected Result

Step 2	Description

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

Expected Result

Step 3	Description

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

Expected Result

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

4.2.22 LVV-T137 - Verify implementation of Data Product Ingest

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Colin Slater

Open LVV-T137 in Jira

4.2.22.1 Verification Elements

None.

4.2.22.2 Test Items

Verify that data products can be ingested.

4.2.22.3 Test Procedure

Step 1	Description

Identify a suitable set of raw data to be run through "mini-DRP" processing.

Expected Result

Step 2-1 from LVV-T1064	Description

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

Expected Result

Step 3-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 4	Description

Confirm that the data products from the DRP processing have been ingested into the Data Backbone.

Expected Result

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

4.2.23 LVV-T144 - Verify implementation of Task Specification

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Kian-Tat Lim

Open LVV-T144 in Jira

4.2.23.1 Verification Elements

None.

4.2.23.2 Test Items

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

4.2.23.3 Test Procedure

Step 1	Description
Inspect software architecture. Verify that there exist Tasks that can be run and configured without re-compilation.	
	Expected Result
Confirmation that the software architecture has allowed for reconfiguring and running Tasks without recompilation.	

Step 2	Description
Verify that tasks can consist of multiple subtasks chained together.	
	Expected Result
Confirmation that the software architecture has allowed for the use of subsets and chains of tasks.	

Step 3	Description
Verify that an example science algorithm can be run through one of these Tasks.	
	Expected Result
Successful Task execution with different configurations, including confirmation that the outputs are different from tasks with altered configurations.	

4.2.24 LVV-T145 - Verify implementation of Task Configuration

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Robert Lupton

Open LVV-T145 in Jira

4.2.24.1 Verification Elements

None.

4.2.24.2 Test Items

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

4.2.24.3 Test Procedure

Step 1	Description
Inspect software design to verify that one can define the configuration for a Task.	
	Expected Result

Step 2	Description
Run a Task with a known invalid configuration. Verify that the error is caught before the science algorithm executes.	
	Expected Result

Step 3	Description
Run a simple task with two different configurations that make a material difference for a Task. E.g., specify a different source detection threshold. Verify that the configuration is different between the two runs through difference in recorded provenance and in results.	
	Expected Result

4.2.25 LVV-T146 - Verify implementation of DMS Initialization Component

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Robert Gruendl [X]

Open LVV-T146 in Jira

4.2.25.1 Verification Elements

None.

4.2.25.2 Test Items

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

4.2.25.3 Test Procedure

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

Step 2	Description
Observe each system and ensure that it has recovered in a properly initialized state.	
	Expected Result
Systems are all active and initialized for their designated purpose.	

4.2.26 LVV-T149 - Verify implementation of Catalog Queries

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Leanne Guy

Open LVV-T149 in Jira

4.2.26.1 Verification Elements

None.

4.2.26.2 Test Items

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

4.2.26.3 Test Procedure

Step 1	Description
Execute a simple query (for example, the one below) and confirm that it returns the expected result.	
	Example Code
SELECT * FROM Object WHERE qserv_areaspec_box(316.582327, −6.839078, 316.653938, −6.781822)	
	Expected Result
A catalog of objects satisfying the specified constraints.	

Step 2	Description
Repeat the query from all available access routes (e.g., an external VO client, internal DM tools on the development cluster, the Science Platform query tool, and from within the Notebook Aspect), confirming in each case that the results are as expected.	
	Expected Result

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Colin Slater

Open LVV-T151 in Jira

4.2.27.1 Verification Elements

None.

4.2.27.2 Test Items

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

4.2.27.3 Test Procedure

Step 1-1 from LVV-T837	Description

Authenticate to the notebook aspect of the Rubin Science Platform (NB-RSP). This is currently at either `https://data.lsst.cloud/nb` (for the interim data facility, or IDF) or `https://usdf-rsp.slac.stanford.edu/nb` (for the US data facility, or USDF).

Expected Result

Redirection to the spawner page of the NB-RSP allowing selection of the containerized science pipelines version and machine flavor.

Step 1-2 from LVV-T837	Description

Spawn a container by:
1) choosing an appropriate science pipelines version: e.g. the latest weekly.
2) choosing an appropriate machine flavor: e.g. medium
3) click “Spawn”

Expected Result

Redirection to the JupyterLab environment served from the chosen container containing the correct science pipelines version.

Step 2-1 from LVV-T838	Description

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

Expected Result

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

Step 2-2 from LVV-T838	Description

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

Expected Result

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

Step 3-1 from LVV-T1207 Description

Execute a query in a notebook to select a small number of stars. In the example code below, we query the Data Preview 0.2 (DP0.2) catalog, then extract the results to an Astropy table.

Example Code

```
CELL 1:

from IPython.display import Markdown as md
from lsst.rsp import get_tap_service, retrieve_query

service = get_tap_service()
md(f'The service endpoint for TAP in this environment is:\n\n &#10145;     {service.baseurl}')

CELL 2:

results = service.search("SELECT coord_ra, coord_dec, g_cModelFlux, r_cModelFlux \
                          FROM dp02_dc2_catalogs.Object \
                          WHERE CONTAINS(POINT('ICRS', coord_ra, coord_dec), \
                          CIRCLE('ICRS', 60.0, -30.0, 0.05)) = 1")
```

Expected Result

Screen output from CELL 1:

The service endpoint for TAP in this environment is:
âž¡ `https://data.lsst.cloud/api/tap`

Example screen output from CELL 2 (may not contain the same 10 entries):

Table length=5533

coord_ra

coord_dec

g_cModelFlux

r_cModelFlux

deg

deg

nJy

nJy

float64

float64

float64

float64

59.9987401

-29.9728812

62.7060123

49.3496319

59.9995813

-29.9743232

166.0433743

394.8261645

59.9989853

-29.9750457

78.9557388

85.2691232

59.9993731

-29.9732406

111.0082072

165.6229656

60.0477786

-29.9736805

68.4818592

49.4783714

60.0400024

-29.9731507

52.0567337

114.2562171

60.0054666

-29.9728639

146.053072

134.1795803

60.00489

-29.9732239

1436.7150639

3606.8163133

60.0469583

-29.9735655

64.8838762

56.5677789

...

...

...

...

60.0053313

-30.0240394

125.6977786

379.8120713

59.9574061

-30.0163726

181.050889

200.8032979

60.0294415

-30.0241709

133.662163

230.8673464

59.9563419

-30.0239843

1551.2308712

4611.0406542

59.9879157

-30.0181116

76.3796313

46.5682713

60.0204061

-30.0228981

174.7738892

304.9991558

60.001638

-30.0183336

43.9593753

46.9695823

59.9861714

-30.0173405

164.6261404

288.8650875

59.9537443

-30.0160515

2228.7204658

5091.2041475

59.9683498

-30.0239539

835.415374

1101.0548649

—

—

Step 4	Description

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

Confirm that non-empty output files appear on disk.

Example Code

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

Expected Result

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

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

Step 5	Description

Check that these files contain the same number of rows:

Example Code

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


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

Expected Result
Print statement produces output “5533 5533 5533”.

Step 6-1 from LVV-T1208	Description

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

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

Expected Result
You will be returned to the RSP landing page: either `https://data.lsst.cloud/nb` (for the interim data facility, or IDF) or `https://usdf-rsp.slac.stanford.edu/nb` (for the US data facility, or USDF). It is now safe to close the browser window.

4.2.28 LVV-T190 - Verify implementation of Base Facility Co-Location with Existing Facility

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T190 in Jira

4.2.28.1 Verification Elements

None.

4.2.28.2 Test Items

Verify that the Base Facility is located at an existing known supported facility.

4.2.28.3 Test Procedure

Step 1	Description
Analyze design	
	Expected Result

4.2.29 LVV-T199 - Verify implementation of Archive Center Co-Location with Existing Facility

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T199 in Jira

4.2.29.1 Verification Elements

None.

4.2.29.2 Test Items

Verify the Archive Center is located at an existing supported facility.

4.2.29.3 Test Procedure

Step 1	Description
Analyze design	
	Expected Result

4.2.30 LVV-T216 - Installation of the Alert Distribution payloads.

Version	Status	Priority	Verification Type	Owner

1	Approved	Normal	Test	Eric Bellm

Open LVV-T216 in Jira

4.2.30.1 Verification Elements

None.

4.2.30.2 Test Items

This test will check:

- That the Alert Distribution payloads are available from documented channels.
- That the Alert Distribution payloads can be installed on LSST Data Facility-managed systems.
- That the Alert Distribution payloads can be executed by LSST Data Facility-managed systems.

4.2.30.3 Environment Needs

4.2.30.3.1 Hardware

This test case shall be executed on the Kubernetes Commons at the LDF.
As discussed in https://dmtn-028.lsst.io/ and https://dmtn-081.lsst.io/, the test machine should have at least 16 cores, 64 GB of memory and access to at least 1.5 TB of shared storage.

4.2.30.4 Test Procedure

Step 1	Description

Download Kafka Docker image from https://github.com/lsst-dm/alert_stream.

Expected Result
Runs without error

Step 2	Description

Change to the alert_stream directory and build the docker image.

```
docker build -t "lsst-kub001:5000/alert_stream"
```

Expected Result
Runs without error

Step 3	Description

Register it with Kubernetes

docker push lsst-kub001:5000/alert_stream

Expected Result
Runs without error

Step 4	Description

From the alert_stream/kubernetes directory, start Kafka and Zookeeper:

```
kubectl create -f zookeeper-service.yaml
kubectl create -f zookeeper-deployment.yaml
kubectl create -f kafka-deployment.yaml
kubectl create -f kafka-service.yaml
```

(use kubectl get pods/services between each command to check status; wait until each is “Running” before starting the next command)

Expected Result
Runs without error

Step 5	Description

Confirm Kafka and Zookeeper are listed when running

kubectl get pods

and

kubectl get services

	Expected Result

Output should be similar to:

```
kubectl get pods
NAME                    READY   STATUS    RESTARTS   AGE
kafka-768ddf5564-xwgvh     1/1     Running   0         31s
zookeeper-f798cc548-mgkpn   1/1     Running   0         1m
```

```
kubectl get services
NAME      TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE
kafka     ClusterIP   10.105.19.124   <none>      9092/TCP   6s
zookeeper   ClusterIP   10.97.110.124   <none>      32181/TCP   2m
```

4.2.31 LVV-T217 - Full Stream Alert Distribution

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Eric Bellm

Open LVV-T217 in Jira

4.2.31.1 Verification Elements

None.

4.2.31.2 Test Items

This test will check that the full stream of LSST alerts can be distributed to end users.

Specifically, this will demonstrate that:

- Serialized alert packets can be loaded into the alert distribution system at LSST-relevant scales (10,000 alerts every 39 seconds);
- Alert packets can be retrieved from the queue system at LSST-relevant scales.

4.2.31.3 Predecessors

LVV-T216

4.2.31.4 Environment Needs

4.2.31.4.1 Software

The Kafka cluster and Zookeeper shall be instantiated according to the procedure described in LVV-T216.

4.2.31.4.2 Hardware

This test case shall be executed on the Kubernetes Commons at the LDF.
As discussed in https://dmtn-028.lsst.io/ and https://dmtn-081.lsst.io/, the test machine should have at least 16 cores, 64 GB of memory and access to at least 1.5 TB of shared storage.

4.2.31.5 Input Specification

Input data: A sample of Avro-formatted alert packets.

4.2.31.6 Output Specification

Multiple Kafka consumers will run and write log files to disk.

The logs will include printing every *Nth* alert to to the log as well as a log summarizing the queue offset.

4.2.31.7 Test Procedure

Step 1-1 from LVV-T866	Description

Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.

Expected Result

An output dataset including difference images and DIASource and DIAObject measurements.

Step 1-2 from LVV-T866	Description

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

Expected Result

Step 2	Description

Start a consumer that monitors the full stream and logs a deserialized version of every Nth packet:

```
kubectl create -f consumerall-deployment.yaml
```

Expected Result

Runs without error

Step 3	Description

```
Start a producer that reads alert packets from disk and loads them into the Kafka queue:
```

```
kubectl create -f sender-deployment.yaml
```

Expected Result

Runs without error

Step 4	Description

Determine the name of the alert sender pod with

kubectl get pods

Examine output log files.

kubectl logs <pod name>

Verify that alerts are being sent within 40 seconds by subtracting the timing measurements.

Expected Result

Similar to

kubectl logs sender-7d6f98586f-nhwfj
visit: 1570. time: 1530588618.0313473
visits finished: 1 time: 1530588653.5614944
visit: 1571. time: 1530588657.0087624
visits finished: 2 time: 1530588692.506188
visit: 1572. time: 1530588696.0051727
visits finished: 3 time: 1530588731.5900314

Step 5	Description

Determine the name of the consumer pod with

kubectl get pods

Examine output log files.

kubectl logs <pod name>

The packet log should show deserialized alert packets with contents matching the input packets.

Expected Result

Similar to {'alertId': 12132024420, 'l1dbId': 71776805594116, 'diaSource': {'diaSourceId': 73499448928374785, 'ccdVisitId': 2020011570, 'diaObjectId': 71776805594116, 'ssO bjectId': None, 'parentDiaSourceId': None, 'midPointTai': 59595.37041, 'filterNa me': 'y', 'ra': 172.24912810036074, 'decl': -80.64214929176521, 'ra_decl_Cov': { 'raSigma': 0.0003428002819418907, 'declSigma': 0.00027273103478364646, 'ra_decl_ Cov': 0.000628734880592674}, 'x': 2979.08837890625, 'y': 3843.328857421875, 'x_y _Cov': {'xSigma': 0.6135467886924744, 'ySigma': 0.77132648229599, 'x_y_Cov': 0.0 007463791407644749}, 'apFlux': None, 'apFluxErr': None, 'snr': 0.366516500711441 04, 'psFlux': 7.698232025177276e-07, 'psRa': None, 'psDecl': None, 'ps_Cov': Non e, 'psLnL': None, 'psChi2': None, 'psNdata': None, 'trailFlux': None, 'trailRa':
etc.

4.2.32 LVV-T218 - Simple Filtering of the LSST Alert Stream

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Eric Bellm

Open LVV-T218 in Jira

4.2.32.1 Verification Elements

None.

4.2.32.2 Test Items

This test will demonstrate the LSST Alert Filtering Service that returns a subset of alerts from the full stream identified by user-provided filters.

Specifically, this will demonstrate that:

- The filtering service can retrieve alerts from the full alert stream and filter them according to their contents;

- The filtered subset can be delivered to science users.

4.2.32.3 Predecessors

LVV-T216
LVV-T217

4.2.32.4 Environment Needs

4.2.32.4.1 Software

The Kafka cluster and Zookeeper shall be instantiated according to the procedure described in LVV-T216.

4.2.32.4.2 Hardware

This test case shall be executed on the Kubernetes Commons at the LDF.
As discussed in https://dmtn-028.lsst.io/ and https://dmtn-081.lsst.io/, the test machine should have at least 16 cores, 64 GB of memory and access to at least 1.5 TB of shared storage.

4.2.32.5 Test Procedure

Step 1-1 from LVV-T216 Description

Download Kafka Docker image from https://github.com/lsst-dm/alert_stream.

Expected Result

Runs without error

Step 1-2 from LVV-T216 Description

Change to the alert_stream directory and build the docker image.

```
docker build -t "lsst-kub001:5000/alert_stream"
```

Expected Result
Runs without error

Step 1-3 from LVV-T216	Description

Register it with Kubernetes

docker push lsst-kub001:5000/alert_stream

Expected Result
Runs without error

Step 1-4 from LVV-T216	Description

From the alert_stream/kubernetes directory, start Kafka and Zookeeper:

```
kubectl create -f zookeeper-service.yaml
kubectl create -f zookeeper-deployment.yaml
kubectl create -f kafka-deployment.yaml
kubectl create -f kafka-service.yaml
```

(use kubectl get pods/services between each command to check status; wait until each is "Running" before starting the next command)

Expected Result
Runs without error

Step 1-5 from LVV-T216	Description

Confirm Kafka and Zookeeper are listed when running

kubectl get pods

and

kubectl get services

	Expected Result

Output should be similar to:

kubectl get pods
NAME READY STATUS RESTARTS AGE
kafka-768ddf5564-xwgvh 1/1 Running 0 31s
zookeeper-f798cc548-mgkpn 1/1 Running 0 1m

kubectl get services
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kafka ClusterIP 10.105.19.124 <none> 9092/TCP 6s
zookeeper ClusterIP 10.97.110.124 <none> 32181/TCP 2m

Step 2	Description

Start 100 consumers that consume the filtered streams and logs a deserialized version of every Nth packet:

```
kubectl create -f consumer1-deployment.yaml
kubectl create -f consumer2-deployment.yaml
kubectl create -f consumer3-deployment.yaml
kubectl create -f consumer4-deployment.yaml
kubectl create -f consumer5-deployment.yaml
kubectl create -f consumer6-deployment.yaml
kubectl create -f consumer7-deployment.yaml
kubectl create -f consumer8-deployment.yaml
kubectl create -f consumer9-deployment.yaml
kubectl create -f consumer10-deployment.yaml
```

	Expected Result

Runs without error

Step 3	Description

Start 5 filter groups:

```
kubectl create -f filterer1-deployment.yaml
kubectl create -f filterer2-deployment.yaml
kubectl create -f filterer3-deployment.yaml
kubectl create -f filterer4-deployment.yaml
kubectl create -f filterer5-deployment.yaml
```

Expected Result
Runs without error

Step 4	Description

Start a producer that reads alert packets from disk and loads them into the Kafka queue:

```
kubectl create -f sender-deployment.yaml
```

Expected Result
Runs without error

Step 5	Description

Determine the name of the alert sender pod with

kubectl get pods

Examine output log files.

kubectl logs <pod name>

Verify that alerts are being sent within 40 seconds by subtracting the timing measurements.

Expected Result
Similar to

kubectl logs sender-7d6f98586f-nhwfj
visit: 1570. time: 1530588618.0313473
visits finished: 1 time: 1530588653.5614944
visit: 1571. time: 1530588657.0087624
visits finished: 2 time: 1530588692.506188
visit: 1572. time: 1530588696.0051727
visits finished: 3 time: 1530588731.5900314

Step 6	Description

Determine the name of the consumer pods with

kubectl get pods

Examine output log files.

kubectl logs <pod name>

The packet log should show deserialized alert packets with contents matching the input packets.

	Expected Result

Similar to

{'alertId': 12132024420, 'l1dbId': 71776805594116, 'diaSource': {'diaSourceId':
73499448928374785, 'ccdVisitId': 2020011570, 'diaObjectId': 71776805594116, 'ssO
bjectId': None, 'parentDiaSourceId': None, 'midPointTai': 59595.37041, 'filterNa
me': 'y', 'ra': 172.24912810036074, 'decl': -80.64214929176521, 'ra_decl_Cov': {
'raSigma': 0.0003428002819418907, 'declSigma': 0.00027273103478364646, 'ra_decl_
Cov': 0.000628734880592674}, 'x': 2979.08837890625, 'y': 3843.328857421875, 'x_y
_Cov': {'xSigma': 0.6135467886924744, 'ySigma': 0.77132648229599, 'x_y_Cov': 0.0
007463791407644749}, 'apFlux': None, 'apFluxErr': None, 'snr': 0.366516500711441
04, 'psFlux': 7.698232025177276e-07, 'psRa': None, 'psDecl': None, 'ps_Cov': Non
e, 'psLnL': None, 'psChi2': None, 'psNdata': None, 'trailFlux': None, 'trailRa':
etc.

4.2.33 LVV-T283 - RAS-00-00: Writing well-formed raw image

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Michelle Butler [X]

Open LVV-T283 in Jira

4.2.33.1 Verification Elements

None.

4.2.33.2 Test Items

This test will check:

- The successful integration of the Pathfinder components with the DM Header Service and the Level 1 Archiver;
- That the raw images are well-formed and meet specifications in change-controlled documents LSE-61;

This Test Case shall be repeated for each of the different cameras (ATScam, LSSTCam) and sensors (Science, Wavefront, and Guider) combination.

4.2.33.3 Predecessors

None.

4.2.33.4 Environment Needs

4.2.33.4.1 Software

- Level 1 software and services needed to create raw image
- LSST Monitoring Service and plugins specific to monitoring Level 1 Test Stand and services

4.2.33.4.2 Hardware

- Level 1 test stand
- Test machine for LSST Monitoring Service

4.2.33.5 Input Specification

None.

4.2.33.6 Output Specification

Raw image(s) that follow specifications defined in change-controlled document LSE-61.

4.2.33.7 Test Procedure

Step 1	Description
Configure system to pull appropriate data from the DAQ emulator	
	Expected Result
A functional DAQ for images to be received from.	

Step 2	Description
Acquire raw data from DAQ readout and DMHS	
	Expected Result
a raw image and a header from the DMHS	

Step 3	Description
Fetch data and reassemble correctly, regardless of CCD/Sensor manufacturer type (two different types will be used)	
	Expected Result
Build the data into a fits file	

Step 4	Description
Check completeness and correctness of the raw images including format, metadata, and image data;	

- Check proper fetch and reassembly of image data from camera DAQ (correct format and data);
- Check proper merge of header service data with image data;
- Check correct insertion of exposure specific data needed in the data file that is not supplied by header service;
- Check minimum required metadata (from requirements document LSE-61) exists in raw image header;

Expected Result
a well formed FITS file with a proper header that has been verified to be correct.

Step 5	Description
Check that the checksum of the file matches the previously calculated value that will be passed on to downstream services	

Expected Result
a MD5sum number generated from the step 4 file.

Step 6	Description
Check confirmation that the data files arrive at their destination intact	

Expected Result
a transfer of the file to the correct location for further retrieval from other services.

Step 7	Description
Check that LSST Monitoring Service showed the appropriate information successfully	

Expected Result
all systems remained green through out the test, and showed all systems up and available.

4.2.34 LVV-T285 - RAS-00-10: Raw images in Observatory Operations Data Service

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Michelle Butler [X]

Open LVV-T285 in Jira

4.2.34.1 Verification Elements

None.

4.2.34.2 Test Items

This test will check:

- The handoff of a raw image from the Level 1 Archiver to the OODS cache manager is successful;
- A recently taken raw image is accessible to the Observatory Operations staff at the base and summit;

This Test Case shall be repeated for each of the different cameras (ATScam, LSSTCam) and sensors (Science, Wavefront, and Guider) combination.

4.2.34.3 Predecessors

LVV-T283

4.2.34.4 Environment Needs

4.2.34.4.1 Software

The following software must be installed:

- Level 1 Test Stand (include software from LVV-T283 - RAS-00-00)
- OODS cache manager
- LSST Monitoring Service and plugins specific to monitoring raw images and OODS
- LSST stack for checking raw images

4.2.34.4.2 Hardware

To complete all tests in a manner which reflects the real system, the following hardware is needed. Note: If not testing inter-machine access, the hardware can be minimized to a single machine outside of the Level 1 Test Stand.

- Level1TestStand(include hardware from LVV-T283 - RAS-00-00)+read/write access to OODS cache disk
- Test Machine for OODS cache manager with read/write access to OODS cache disk
- Test machine for Observatory Operations staff at ”base” that can access OODS cache disk
- Test machine for Observatory Operations staff at ”summit” that can access OODS cache disk
- Test machine for LSST Monitoring Service

Size of cache disk is determined by number of files to be included in the test.

4.2.34.5 Input Specification

4.2.34.6 Output Specification

Raw image(s) that follow format defined in LSE-61;
Database (may be SQLite file) that enables the raw image(s) to be accessed via a “Data Butler”.

4.2.34.7 Test Procedure

Step 1	Description
Initialize all services configuring the Level 1 Archiver Service so that the raw images are to be saved to the OODS	
	Expected Result
all camera and services for images are running and reporting green through the monitoring programs for the services.	

Step 2	Description

Acquire a raw image

Expected Result

Image present in the input folder.

Step 3	Description

The handoff of the raw image from the Level 1 Archiver Service to the test OODS automatically occurs

Expected Result

the raw image with a proper header is written to a file area managed by the OODS

Step 4	Description

For each of the expected raw images, verify that the checksum matches the original Level 1 checksum

Expected Result

checksum of the file is checked against the file for verification that the OODS has the correct file and it matches the original md5sum of the FITS file.

Step 5	Description

Check that LSST Monitoring Service showed the appropriate information successfully

Expected Result

Make sure all camera and OODS systems were available thorughout this test.

4.2.35 LVV-T286 - RAS-00-20: Raw image are part of the permanent record of survey via DBB

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Michelle Butler [X]

Open LVV-T286 in Jira

4.2.35.1 Verification Elements

None.

4.2.35.2 Test Items

This test will check:

- That the handoff of a raw image from the Level 1 Archiver Service to the DBB buffer manager is successful;
- That the raw image is ingested into the Data Backbone successfully;
- That the monitoring of the above items is successful;

This Test Case shall be repeated for each of the different cameras (ATScam, LSSTCam) and sensors (Science, Wavefront, and Guider) combination.

Note: For a complete check of the various aspects of what it means for a raw image to be in the Data Backbone, see the tests for the Data Backbone.

4.2.35.3 Predecessors

LVV-T283

4.2.35.4 Environment Needs

4.2.35.4.1 Software

- Level 1 Test Stand
- DBB buffer manager
- DBB raw image ingestion
- DBB database
- LSST Monitoring Service and plugins specific to monitoring raw images, DBB buffer manager, and DBB

4.2.35.4.2 Hardware

- Level 1 Test Stand (include hardware from LVV-T-283 - RAS-00-00) + read/write access to DBB buffer disk;
- Test Machine for DBB buffer manager with read/write access to DBB buffer disk;
- Test machine for each DBB endpoint with read/write access to DBB disk;
- Test machine for LSST Monitoring Service

Size of buffer disk and DBB disk is determined by number of files to be included in the test.

Note: If not testing inter-machine operability, then the hardware can be minimized to a single machine outside of the Level 1 test stand.

4.2.35.5 Input Specification

None

4.2.35.6 Output Specification

- Raw image(s) are saved to storage and replicated to correct locations with checksums that match original Level 1 checksum;
- Database containing information of the following types: physical, location, science metadata, provenance as specified in LSE-61;
- Both image(s) and database entries replicated correctly;

4.2.35.7 Test Procedure

Step 1	Description
	Initialize all services configuring the Level 1 Archiver Service so that the raw images are to be archived to the DBB
	Expected Result
	all services for the camera images and the DBB services are all running and ready for data.

Step 2	Description
	Acquire a raw image (see LVV-T283 - RAS-00-00)
	Expected Result
	have a raw Fits file with proper header.

Step 3	Description
	After the automatic handoff of the raw image between the Level 1 Archiver Service and the DBB buffer manager, the raw image will automatically be ingested into the Data Backbone
	Expected Result
	the DBB file systems will have the file, and metadata and providence will be recorded in the consolidated DB. The file will also be replicated to mulitple locations for DR.

Step 4	Description
	Check that the raw image is accessible at each DBB endpoint and matches original Level 1 checksum
	Expected Result
	data resides at NCSA DBB end point, and Chile end point and match with the same checksum.

Step 5	Description
	Check that LSST Monitoring Service showed the appropriate information successfully

Expected Result
all related systems remained up during this test.

Step 6	Description
More complete tests of the DBB can be done by running the DBB service tests on the raw image(s). These would check correctness and completeness of the data stored in the database as well as checking that the file has been replicated to all required places	

Expected Result
These would be more tests of when things go wrong to make sure that the DBB is able to continue to work, and not be in the way of taking images from the camera

4.2.36 LVV-T287 - RAS-00-30: Raw Image Archiving Availability, Throughput, Reliability, and Heterogeneity

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Michelle Butler [X]

Open LVV-T287 in Jira

4.2.36.1 Verification Elements

None.

4.2.36.2 Test Items

This test will check:

- Raw Image Archiving meets availability requirements;

- Raw Image Archiving meets throughput requirements;
- Raw Image Archiving meets reliability requirements;
- Raw Image Archiving meets heterogeneity requirements;

This test case need to be completed when more information is available.

4.2.36.3 Test Procedure

Step 1	Description

these will be filled out as the service becomes more known as to what the availablility, throughput, reliability and heterogeneity are.

Expected Result

The archive system will stay up through thick and thin and perform like it's suppose to.

4.2.37 LVV-T362 - Installation of the LSST Science Pipelines Payloads

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	John Swinbank

Open LVV-T362 in Jira

4.2.37.1 Verification Elements

None.

4.2.37.2 Test Items

This test will check that:

- The Alert Production Pipeline payload is available for installation from documented channels;
- The Data Release Production Pipeline payload is available for installation from documented channels;
- The Calibration Products Production Pipeline payload is available for installation from documented channels;
- These payloads can be installed on systems at the LSST Data Facility following available documentation;
- The installed pipeline payloads are capable of successfully executing basic integration tests.

Note that this test assumes packaging of the Science Pipelines software, in which all the above payloads are represented by a single “meta-package”, lsst_distrib.

4.2.37.3 Environment Needs

4.2.37.3.1 Software

Science Pipelines prerequisite software, as documented at https://pipelines.lsst.io/, must be installed on the target system.

4.2.37.3.2 Hardware

This test requires a workstation or equivalent system running an operating system supported by the LSST Science Pipelines.

4.2.37.4 Test Procedure

Step 1	Description

The LSST Science Pipelines, described by the lsst_distrib meta-package, should be installed following the documentation available at https://pipelines.lsst.io/. The suggested Conda environment will be used to ensure that a supported execution environment is available.

	Expected Result

Detailed output will depend on the installation method chosen, but will confirm the successful installation of the Science Pipelines.

Step 2	Description

The lsst_distrib top-level metapackage will be enabled. Assuming that the software has been installed at ${LSST_DIR}:

```
source ${LSST_DIR}/loadLSST.bash
setup lsst_distrib
```

	Expected Result

Nothing is printed. The command

```
eups list -s lsst_distrib
```

may be used to confirm that the correct version of the codebase has been installed.

Step 3	Description

The "LSST Stack Demo" package will be downloaded onto the test system from https://github.com/lsst/pipelines_check/releases. The version corresponding to to the version of the Science Pipelines under test should be chosen.

	Expected Result

Depends on the tool selected by the user for downloading.

Step 4	Description

The stack demo package is uncompressed into a directory ${DEMO_DIR}.

	Expected Result

Depends on options given to the tar command. Should confirm the availability of the stack demo source.

Step 5	Description

The demo package will be executed by following the instructions in its README file.

	Expected Result

Successful execution will result in the string "Ok" being returned.

4.2.38 LVV-T363 - Science Pipelines Release Documentation

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Inspection	John Swinbank

Open LVV-T363 in Jira

4.2.38.1 Verification Elements

None.

4.2.38.2 Test Items

This test will check:

- That a particular Science Pipelines release is adequately described by documentation at the https://pipelines.lsst.io/ site;
- That the Science Pipelines release is accompanied by a characterization report which describes its scientific performance.

4.2.38.3 Environment Needs

4.2.38.3.1 Software

A web browser.

4.2.38.3.2 Hardware

A device with internet access.

4.2.38.4 Test Procedure

Step 1	Description

Load the Science Pipelines website at https://pipelines.lsst.io/.

Expected Result

The website is displayed.

Step 2	Description

Identify documentation for the release under test. This should be clearly labelled on the documentation site.

If the latest release is being tested, the default page loaded when visiting https://pipelines.lsst.io/ should be the documentation required.

If this test is for another release, the site should present clear instructions for changing the edition (or version) of the documentation being examined, and documentation for the release under test should be available.

Expected Result

The documentation for the release under test is displayed.

Step 3	Description

Inspect the documentation to ensure that it refers to the release under test, and that it provides:

- Release notes, describing changes in this release relative to the previous;
- Installation instructions, together with a list of supported platforms and prerequisites;
- Getting started information.

Expected Result

The user is satisfied that the required information is available.

Step 4	Description

Locate the Characterization Metric Report corresponding to this release. It should be linked from the main release documentation.

Expected Result

The user is satisfied that the report is available.

Step 5	Description

Verify that the characterization metric report describes the scientific performance of the release in terms of a selection of performance metrics drawn from high-level requirements documentation (the Science Requirements Document, LPM-17; the LSST

System Requirements, LSE-29; and/or the Observatory System Specifications, LSE-30).

Expected Result
Metric values describing the performance of the release, for example as computed by validate_drp, are described in the report.

4.2.39 LVV-T368 - Loading and processing Camera test data

Version	Status	Priority	Verification Type	Owner
2	Approved	Normal	Test	John Swinbank

Open LVV-T368 in Jira

4.2.39.1 Verification Elements

None.

4.2.39.2 Test Items

This test will check:

- That Camera test data is available for processing in the LSST Data Facility, and accessible through the LSST Science Platform;
- That the Data Management I/O abstraction (the "Data Butler") can load that data into the Science Platform environment;
- That Data Management algorithmic "tasks" can be executed to process that data;
- That results can be displayed in the Firefly display tool.

4.2.39.3 Predecessors

Executing LVV-T374 will satisfy the preconditions for this test, assuming that $REPOSITORY_PATH is set equal to the output location used in LVV-T374.

4.2.39.4 Environment Needs

4.2.39.4.1 Software

The LSST Science Pipelines version w_2018_45 must be available within the Notebook Aspect of the LSST Science Platform.

4.2.39.4.2 Hardware

This test assumes the availability of the Notebook and Portal aspects of the LSST Science Platform, deployed at https://lsst-lspdev.ncsa.illinois.edu.

4.2.39.5 Test Procedure

Step 1	Description

Connect to the Notebook Aspect of the Science Platform following the instructions at https://nb.lsst.io/. Log in, and "spawn" a new machine with image "Weekly 2018_45" and size "small".

Expected Result

The JupyterLab environment appears.

Step 2	Description

Create a terminal session. Use it to set up the LSST tools, then download and build version 5c12b06e6 of obs_lsst:

```
$ source /opt/lsst/software/stack/loadLSST.bash
$ setup lsst_distrib
$ git clone https://github.com/lsst/obs_lsst.git
$ cd obs_lsst
$ git checkout 5c12b06e6
$ setup -k -r .
$ scons
```

Arrange for obs_lsst to automatically be added to the environment when starting a new notebook:

```
$ echo "setup -j -r ~/obs_lsst" >> ~/notebooks/.user_setups
```

Exit the terminal.

Expected Result
No errors are seen during execution of the provided commands.

Step 3	Description

Create a new “LSST” notebook.

Import the standard libraries required for the rest of this test:

```
import os
import lsst.afw.display as afwDisplay
from lsst.daf.persistence import Butler
from lsst.ip.isr import IsrTask
from firefly_client import FireflyClient
from IPython.display import IFrame
```

and execute the cell.

Expected Result
Nothing is printed.

Step 4	Description

Create a Data Butler client, and use it to retrieve the data which will be used for this test.

```
butler = Butler($REPOSITORY_PATH)
raw = butler.get("raw", visit=$VISIT_ID, detector=2)
bias = butler.get("bias", visit=$VISIT_ID, detector=2)
```

Expected Result
Nothing is printed.

Step 5	Description

Initialize the Firefly display system:

```
my_channel = '{}_test_channel'.format(os.environ['USER'])
server = 'https://lsst-lspdev.ncsa.illinois.edu'
ff='{}/firefly/slate.html?__wsch={}'.format(server, my_channel)
IFrame(ff,800,600)
afwDisplay.setDefaultBackend('firefly')
afw_display = afwDisplay.getDisplay(frame=1,
                                    name=my_channel)
```

Click on the link provided after executing the above.

	Expected Result

A Firefly window is shown.

Step 6	Description

Display the raw image data in the Firefly window:

afw_display.mtv(raw)

	Expected Result

Raw image data is displayed.

Step 7	Description

Configure and run an Instrument Signature Removal (ISR) task on the raw data. Most corrections are disabled for simplicity. but the bias frame is applied.

```
isr_config = IsrTask.ConfigClass()
isr_config.doDark=False
isr_config.doFlat=False
isr_config.doFringe=False
isr_config.doDefect=False
isr_config.doAddDistortionModel=False
isr_config.doLinearize=False
isr = IsrTask(config=isr_config)
result = isr.run(raw, bias=bias)
```

	Expected Result

Nothing is printed.

Step 8	Description

Display the corrected image data in the Firefly window:

afw_display.mtv(result.exposure)

	Expected Result

Processed (trimmed, bias-subtracted) image data is displayed.

4.2.40 LVV-T374 - Ingesting Camera test data

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	John Swinbank

Open LVV-T374 in Jira

4.2.40.1 Verification Elements

None.

4.2.40.2 Test Items

This test will check:

- That raw Camera test data is available on a filesystem in the LSST Data Facility;
- That raw Camera test data can be ingested and made available through the Data Management I/O abstraction (the "Data Butler").

4.2.40.3 Environment Needs

4.2.40.3.1 Software

The LSST Science Pipelines version w_2018_45 must be available within the Notebook Aspect of the LSST Science Platform.

4.2.40.3.2 Hardware

This test assumes the availability of the Notebook aspect of the LSST Science Platform, deployed at https://lsst-lspdev.ncsa.illinois.edu.

4.2.40.4 Test Procedure

Step 1	Description

Connect to the Notebook Aspect of the Science Platform following the instructions at https://nb.lsst.io/. Log in, and "spawn" a new machine with image "Weekly 2018_45" and size "large".

Expected Result

The JupyterLab environment appears.

Step 2	Description

Create a terminal session. Use it to set up the LSST tools, then download and build version 5c12b06e6 of obs_lsst:

$ source /opt/lsst/software/stack/loadLSST.bash
$ setup lsst_distrib
$ git clone https://github.com/lsst/obs_lsst.git
$ cd obs_lsst
$ git checkout 5c12b06e6
$ setup -k -r .
$ scons

Expected Result

No errors are seen during execution of the provided commands.

Step 3	Description

Ingest RTM-007 test data by executing the following commands:

OUTPUT_REPO_DIR=$OUTPUT_DATA_DIR
INPUT_DATA_DIR=$INPUT_DATA_DIR
mkdir -p $OUTPUT_REPO_DIR
echo "lsst.obs.lsst.ts8.Ts8Mapper" > $OUTPUT_REPO_DIR/_mapper
ingestImages.py $OUTPUT_REPO_DIR $INPUT_DATA_DIR/*/*.fits
constructBias.py $OUTPUT_REPO_DIR –rerun calibs –id imageType=BIAS –batch-type smp –cores 4
ingestCalibs.py $OUTPUT_REPO_DIR –calibType bias $OUTPUT_REPO_DIR/rerun/calibs/bias/*/*.fits –validity 9999 –output $OUTPUT_REPO_DIR/CALIB –mode=link

Where:

$OUTPUT_DATA_DIR is some location on shared storage to which the user has write permission;
$INPUT_DATA_DIR is defined in the test case description.

Expected Result

Many status messages are logged to screen, and the command exits with status 0.

Step 4	Description

Demonstrate that raw and bias data for visit $VISIT_ID have been made available in the repository. Load a Python interpreter (run â€œpython”) and execute the following:

```
from lsst.daf.persistence import Butler
visit_id = $VISIT_ID
b = Butler($OUTPUT_DATA_DIR)
b.get("raw", visit=visit_id, detector=2)
b.get("bias", visit=visit_id, detector=2)
```

	Expected Result

Each call to b.get() returns an instance of an ExposureF object. Warnings about lack of dark-time or WCS information may be ignored.

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Leanne Guy

Open LVV-T376 in Jira

4.2.41.1 Verification Elements

None.

4.2.41.2 Test Items

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

4.2.41.3 Test Procedure

Step 1-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 2 Description

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

Expected Result

Step 3 Description

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

Expected Result

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

Step 4 Description

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

Expected Result

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

4.2.42 LVV-T377 - Verify Calculation of Photometric Performance Metrics

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Leanne Guy

Open LVV-T377 in Jira

4.2.42.1 Verification Elements

None.

4.2.42.2 Test Items

Verify that the DMS system provides software to calculate photometric performance metrics, and that the algorithms are properly calculating the desired quantities. Note that because the DMS requirement is that the software shall be provided (and not on the actual measured values of the metrics), we verify all of the requirements via a single test case.

4.2.42.3 Test Procedure

Step 1-1 from LVV-T987 Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 2 Description

Point the butler to a simulated dataset containing data in all filters, that is sufficient for the purposes of measuring photometric performance metrics.

Expected Result

Step 3 Description

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

	Expected Result

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

Step 4	Description

Compare measured photometry to known values from input simulated data, and confirm that the output values for all of the photometric performance metrics are as expected.

	Expected Result

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

4.2.43 LVV-T454 - LDM-503-8 Enable LSP viewing of spectrograph data.

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Michelle Gower

Open LVV-T454 in Jira

4.2.43.1 Verification Elements

None.

4.2.43.2 Test Items

- Acquire spectrograph image data, transfer that data to NCSA, ingest data into a Butler (G2 or G3 when available), and enable viewing of data on LSP.

4.2.43.3 Predecessors

LDM-503-4b

4.2.43.4 Environment Needs

4.2.43.4.1 Hardware

ATS storage server system housed with spectrograph. Receiver system at NCSA for data.

4.2.43.5 Test Procedure

Step 1	Description

Have data on the ATS archiver system from the spectrograph.

Expected Result

Well formed files on the ATS system that need to be transferred to NCSA for further analysis

Step 2	Description

A first few iterations is the human runs script to transfer data to NCSA through secure pipeline. after the process is unchanging/solid, a cronjob starts up data "sync" process.

Expected Result

Data is transferred to NCSA, and is located in NCSA file systems.

Step 3	Description

All files transferred have a ButlerG2 (or G3 when ready) ingest process.

Expected Result

files now can be accessed by Butler access methods

Step 4	Description

LSP processes can now view spectrograph generate files

Expected Result

LSP jupyter notebooks can view spectrograph files.

4.2.44 LVV-T1085 - Short Queries Functional Test

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Fritz Mueller

Open LVV-T1085 in Jira

4.2.44.1 Verification Elements

None.

4.2.44.2 Test Items

The objective of this test is to ensure that the short queries are performing as expected and establish a timing baseline benchmark for these types of queries.

4.2.44.3 Test Procedure

Step 1	Description

Execute single object selection:

SELECT * **FROM** Object **WHERE** deepSourceId = 9292041530376264

and record execution time.

Expected Result

Query runs in less than 10 seconds.

Step 2	Description

Execute spatial area selection from Object:

SELECT COUNT(*) FROM Object **WHERE**

qserv_areaspec_box(316.582327, −6.839078, 316.653938, −6.781822) and record execution time.

Expected Result
Query runs in less than 10 seconds.

4.2.45 LVV-T1086 - Full Table Scans Functional Test

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Fritz Mueller

Open LVV-T1086 in Jira

4.2.45.1 Verification Elements

None.

4.2.45.2 Test Items

The objective of this test is to ensure that the full table scan queries are performing as expected and establish a timing baseline benchmark for these types of queries.

4.2.45.3 Test Procedure

Step 1	Description

Execute query:

```
SELECT ra , decl , u_psfFlux , g_psfFlux , r_psfFlux FROM Object
WHERE y_shapeIxx BETWEEN 20 AND 20.1
```

and record execution time and output size.

Expected Result
Query expected to run in less than 1 hour.

Step 2	Description

Execute query:

SELECT COUNT(*) **FROM** Source **WHERE** flux_sinc **BETWEEN** 1 **AND** 1.1

and record the execution time

Expected Result

Query expected to run in less than 12 hours.

Step 3	Description

Execute query:

SELECT COUNT(*) **FROM** ForcedSource **WHERE** psfFlux **BETWEEN** 0.1 **AND** 0.2

and record the execution time

Expected Result

Query expected to run in less than 12 hours.

4.2.46 LVV-T1087 - Full Table Joins Functional Test

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Fritz Mueller

Open LVV-T1087 in Jira

4.2.46.1 Verification Elements

None.

4.2.46.2 Test Items

The objective of this test is to ensure that the full table join queries are performing as expected and establish a timing baseline benchmark for these types of queries.

4.2.46.3 Test Procedure

Step 1	Description

Execute query:

```
SELECT o.deepSourceId, s.objectId, s.id, o.ra, o.decl
FROM Object o, Source s WHERE o.deepSourceId=s.objectId
  AND s . flux_sinc BETWEEN 0.3 AND 0.31
```

and record execution time.

	Expected Result

Query expected to run in less than 12 hours.

Step 2	Description

Execute query:

```
SELECT o.deepSourceId, f.psfFlux FROM Object o, ForcedSource f
WHERE o.deepSourceId=f.deepSourceId
AND f . psfFlux BETWEEN 0.13 AND 0.14
```

and record execution time.

	Expected Result

Query expected to run in less than 12 hours.

4.2.47 LVV-T1088 - Concurrent Scans Scaling Test

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Fritz Mueller

Open LVV-T1088 in Jira

4.2.47.1 Verification Elements

None.

4.2.47.2 Test Items

This test will show that average completion-time of full-scan queries of the Object catalog table grows sub-linearly with respect to the number of simultaneously active full-scan queries, within the limits of machine resource exhaustion.

4.2.47.3 Test Procedure

Step 1	Description
Repeat steps 2 through 5 below, where "pool of interest" is taken first to be "FTSObj" and subsequently "FTSSrc":	
	Expected Result
At end of each pass, a graph indicating scan scaling rate and machine resource exhaustion cutoff.	

Step 2	Description
Inspect and modify the CONCURRENCY and TARGET_RATES dictionaries in the runQueries.py script. Set CONCURRENCY initially to 1 for the query pool of interest, and to 0 for all other query pools. Set TARGET_RATES for the query pool of interest to the yearly value per table in LDM-552, section 2.2.1.	
	Expected Result
rueQueries.py script updated with appropriate values for test iteration	

Step 3	Description
Execute the runQueries.py script and let it run for at least one, but preferably several, query cycles.	
	Expected Result
Test script executes producing log file.	

Step 4	Description
Examine log file output and compile performance statistics to obtain a growth curve point for the pool of interest for the test report.	

Expected Result
Logs indicate either successful test run, providing another growth point for curve, or errors indicating machine resource exhaustion cutoff has been reached.

Step 5	Description
Adjust the CONCURRENCY value for the pool of interest and repeat from step 3 to establish the growth trend and machine resource exhaustion cutoff for the query pool of interest to an acceptable degree of accuracy.	

Expected Result
Average query execution time for full scan queries of each class should be demonstrated to grow sub-linearly in the number of concurrent queries to the limits of machine resource exhaustion.

4.2.48 LVV-T1089 - Load Test

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Fritz Mueller

Open LVV-T1089 in Jira

4.2.48.1 Verification Elements

None.

4.2.48.2 Test Items

This test will check that Qserv is able to meet average query completion time targets per query class under a representative load of simultaneous high and low volume queries while running against an appropriately scaled test catalog.

4.2.48.3 Test Procedure

Step 1	Description
Inspect and modify the CONCURRENCY and TARGET_RATES dictionaries in the runQueries.py script. Set CONCURRENCY and TARGET_RATES for all pools to the yearly value per table in LDM-552, section 2.2.1.	

Expected Result
Script updated with appropriate values.

Step 2	Description
Execute the runQueries.py script and let it run for 24 hours.	

Expected Result
Script runs without error and produces output log.

Step 3	Description
Examine log file output and compile average query execution times per query type; and compare to yearly target values per table in LDM-552, section 2.2.1.	

Expected Result
Average query times per query type equal or less than corresponding yearly target values in LDM-552, section 2.2.1.

4.2.49 LVV-T1090 - Heavy Load Test

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Fritz Mueller

Open LVV-T1090 in Jira

4.2.49.1 Verification Elements

None.

4.2.49.2 Test Items

This test will check that Qserv is able to meet average query completion time targets per query class under a higher than average load of simultaneous high and low volume queries while running against an appropriately scaled test catalog.

4.2.49.3 Test Procedure

Step 1	Description

Inspect and modify the CONCURRENCY and TARGET_RATES dictionaries in the runQueries.py script. Set CONCURRENCY and TARGET_RATES for LV query pool to 2020 value per table in LDM-552, section 2.2.1. Set CONCURRENCY and TARGET_RATES for all other query pools to values in next column over from current year column (or to 2020 values +10% if year is 2020) per table in LDM-552, section 2.2.1.

Expected Result

Script updated with appropriate values.

Step 2	Description

Execute the runQueries.py script and let it run for 24 hrs.

Expected Result

Script runs without error and produces output log.

Step 3	Description

Examine log file output and compile average query execution times per query type.

Expected Result

Average query times per query type equal or less than corresponding yearly target values in LDM-552, section 2.2.1.

4.2.50 LVV-T1168 - Verify Summit - Base Network Integration

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Inspection	Jeff Kantor

Open LVV-T1168 in Jira

4.2.50.1 Verification Elements

None.

4.2.50.2 Test Items

Verify the integration of the summit to base network by demonstrating a sustained and uninterrupted transfer of data between summit and base over 1 day period at or exceeding rates specified in LDM-142. Done in 3 phases in collaboration with equipment/installation vendors (see test procedure).

4.2.50.3 Predecessors

See pre-conditions by phase above.

4.2.50.4 Environment Needs

4.2.50.4.1 Software

perfsonar on DTN.

4.2.50.4.2 Hardware

OTDR, DTN.

4.2.50.5 Input Specification

PMCS DMTC-7400-2330 COMPLETE
By phase:

1. Posts from Cerro Pachon to AURA Gatehouse repaired/improved. Fiber installed on posts from Cerro Pachon to AURA Gatehouse. Fiber installed from AURA Gatehouse to AURA compound in La Serena. OTDR purchased.
2. AURA DWDM installed in caseta on Cerro Pachon and in existing computer room in La Serena. DTN installed in La Serena. DTN loaded with software and test data staged.
3. Base Data Center (BDC) ready for installation of LSST DWDM. Fiber connecting existing computer room to BDC. LSST DWDM equipment installed in Summit Computer Room and BDC.

4.2.50.6 Output Specification

Fiber tested to within acceptable Db. Bandwidth, latency within specifications.

4.2.50.7 Test Procedure

Step 1	Description

Test optical fiber with OTDR:
Installation of fiber optic cables and Optical Time Domain Reflector (OTDR) fiber testing (completed 20170602 REUNA deliverable RD10)

Test Data

OTDR generated optical data

Expected Result

Fiber tested to within acceptable Db.

Step 2	Description

Test AURA DWDM:
Installation of AURA DWDM and Data Transfer Node (DTN) (completed 20171218 DMTR-82)

Test Data

DTN perfSonar generated data

Expected Result

Summit - Base bandwidth and latency within specifications

Step 3	Description

Test LSST DWDM:
Installation of LSST DWDM and Bit Error Rate Tester (BERT) data (completed 20190505 collection-7743, 20191108 DAQ DWDM Connection Tests)

Test Data

BERT generated data

Expected Result

Summit - Base bandwidth, latency, bit error rate within specifications

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Colin Slater

Open LVV-T1232 in Jira

4.2.51.1 Verification Elements

None.

4.2.51.2 Test Items

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

4.2.51.3 Test Procedure

Step 1-1 from LVV-T849	Description

Navigate to the Portal Aspect endpoint. The stable version of the RSP at the interim data facility (IDF) should be used for this test and is currently located at: `https://data.lsst.cloud/`. The Portal Aspect can be reached by clicking on "Portal" in the RSP home page or by navigating directly to `https://data.lsst.cloud/portal/app`.

Expected Result

A credential-entry screen should be displayed.

Step 1-2 from LVV-T849	Description

Enter a valid set of credentials for an LSST user with RSP access on the instance under test.

Expected Result

The Portal Aspect UI should be displayed following authentication.

Step 2	Description

Select query type "ADQL".

Expected Result

Step 3	Description

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

Example Code

```
SELECT coord_ra, coord_dec, g_cModelFlux, r_cModelFlux FROM dp02_dc2_catalogs.Object WHERE CONTAINS(POINT('ICRS', coord_ra, coord_dec), CIR
```

Expected Result

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

Step 4	Description

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

Expected Result

Step 5	Description

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

Expected Result

Step 6	Description

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

Expected Result

Step 7-1 from LVV-T850	Description

Click the "logout" button at the upper right corner of the Portal screen.

Expected Result

Returned to the RSP home page at https://data.lsst.cloud/. When navigating to the portal endpoint, expect to execute the steps in LVV-T849.

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jim Bosch

Open LVV-T1240 in Jira

4.2.52.1 Verification Elements

None.

4.2.52.2 Test Items

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

4.2.52.3 Test Procedure

Step 1	Description

Identify an appropriate processed dataset for this test.

Expected Result

A dataset with Processed Visit Images.

Step 2-1 from LVV-T987	Description

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

Example Code

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

Expected Result
Butler repo available for reading.

Step 3	Description
Select a single visit from the dataset, and extract its calibration data. For a subset of CCDs, check how many astrometric standards contributed to the solution. Confirm that this number is at least **astrometricMinStandards = 5.**	

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

4.2.53 LVV-T1264 - Verify implementation of archiving camera test data

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Robert Gruendl [X]

Open LVV-T1264 in Jira

4.2.53.1 Verification Elements

None.

4.2.53.2 Test Items

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

4.2.53.3 Test Procedure

Step 1	Description
Obtain some data on a camera test stand.	

Expected Result

Step 2	Description

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

Expected Result

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

Step 3	Description

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

Expected Result

Step 4-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 5	Description

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

Expected Result

Camera test stand data that is well-formed.

4.2.54 LVV-T1549 - LDM-503-6 Comcam verification readiness

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Demonstration	Michelle Butler [X]

Open LVV-T1549 in Jira

4.2.54.1 Verification Elements

None.

4.2.54.2 Test Items

Verify that ComCam has all the services running and verified working for retrieving an image from the ComCam DAQ and store it on file systems at the LDF for viewing by RSP.

4.2.54.3 Test Procedure

Step 1	Description
ComCam-DAQ produces an image	
	Test Data
DAQ produces a SAL message that a image has been created	
	Expected Result
in memory file created in DAQ	

Step 2	Description
ComCam-archiver and ComCam-forwarder build image with proper header from ComCam-header service	
	Test Data
Good image file with proper header with all 9 CCDs	
	Expected Result
9 image files all with individual headers and then 1 header for all 9 images too.	

Step 3	Description
ComCam-archiver/forwarder transfers the file to the l1-handoff machine.	

Test Data
l1-handoff machine has image file now on local disk.

Expected Result
image file now found on disk on L1-handoff with hardlinks to 2 different file systems (OODS and DBB) services.

Step 4	Description
OODS service running and ingests the image file into Butler/G3 (or Gen2) and readies the file systems for the commissioning cluster at the Base to be able to mount and see the new files.	

Test Data
Image file ingested to local butler for Base

Expected Result
Image file ingested

Step 5	Description
DBB transfers the file to NCSA thorough the DBB-gateway machines and DTN nodes at the base.	

Expected Result
data file arrives at file systems at NCSA

Step 6	Description
Files are ingested into the butler/G3 at NCSA and moved to file systems that are viewable by the RSP.	

Expected Result
data can be seen and retrieved by RSP.

4.2.55 LVV-T1550 - LDM-503-10 DAQ Validation

Version	Status	Priority	Verification Type	Owner

1	Approved	Normal	Demonstration	Michelle Butler [X]

Open LVV-T1550 in Jira

4.2.55.1 Verification Elements

None.

4.2.55.2 Test Items

Verify that the DAQ can talk to test machines at the BDC through the DWDM network.

4.2.55.3 Predecessors

DAQ network at the base; forwarders and L1 handoff machine must be available to the DAQ COB at the summit, and forwarders and other test machines must be configured and set up on the BDC networks.

4.2.55.4 Test Procedure

Step 1	Description
have DAQ produce image at the summit	
	Expected Result
Image on At-archiver	

Step 2	Description
The forwarder at the BDC should be able to have communication with the DAQ that the image was taken, and be able to see the file.	
	Expected Result
Image available for the forwarder at the base.	

Step 3	Description

Communication between the forwarder and the DAQ are in place with messages being exchanged.

	Expected Result

if messages can be exchanged, the communication has been established.

4.2.56 LVV-T1556 - LDM-503-10B Large Scale CCOB Data Access

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Demonstration	Michelle Butler [X]

Open LVV-T1556 in Jira

4.2.56.1 Verification Elements

None.

4.2.56.2 Test Items

Demonstrate the ability to transfer data from the SLAC test stand or CCOB with 21 rafts from SLAC and ingested at NCSA and make available through an instance of the RSP

4.2.56.3 Test Procedure

Step 1	Description

Have a system at SLAC that has the 21 raft data that needs to be transferred to NCSA, and all accounts and scripts installed on environment that can read that data.

	Test Data

21 rafts of data with proper headers

Expected Result
scripts are able to transfer the data to NCSA though rsync or bbcp.

Step 2	Description
Data is transferred to NCSA and ingested into Butler	
	Test Data
21 rafts of data	
	Expected Result
Data is transferred to NCSA, and can now be see in file systems by the RSP.	

Step 3	Description
using the RSP view the data in the ingested directory	
	Test Data
21 rafts of data with proper headers and available with Butler.get	
	Expected Result
data can be viewed.	

4.2.57 LVV-T1745 - Verify calculation of median relative astrometric measurement error on 20 arcminute scales

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T1745 in Jira

4.2.57.1 Verification Elements

None.

4.2.57.2 Test Items

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.

4.2.57.3 Test Procedure

Step 1	Description

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

Expected Result

A dataset that has been ingested into a Butler repository.

Step 2-1 from LVV-T860	Description

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

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

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

Example Code

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

Expected Result

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

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

Step 3-1 from LVV-T1744	Description

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

Example Code

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

Expected Result

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

Step 4	Description

Confirm that the metric 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.

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T1746 in Jira

4.2.58.1 Verification Elements

None.

4.2.58.2 Test Items

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.**

4.2.58.3 Test Procedure

Step 1	Description

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

Expected Result

A dataset that has been ingested into a Butler repository.

Step 2-1 from LVV-T860	Description

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

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

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

Example Code

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

Expected Result

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

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

Step 3-1 from LVV-T1744	Description

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

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

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

Step 4	Description
Confirm that the metric 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).

4.2.59 LVV-T1747 - Verify calculation of relative astrometric measurement error on 5 arcminute scales

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T1747 in Jira

4.2.59.1 Verification Elements

None.

4.2.59.2 Test Items

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.**

4.2.59.3 Test Procedure

Step 1	Description

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

Expected Result

A dataset that has been ingested into a Butler repository.

Step 2-1 from LVV-T860	Description

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

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

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

Example Code

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

Expected Result

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

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

Step 3-1 from LVV-T1744	Description

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

Example Code

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

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

Step 4	Description
Confirm that the metric 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.

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T1748 in Jira

4.2.60.1 Verification Elements

None.

4.2.60.2 Test Items

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**.

4.2.60.3 Test Procedure

Step 1	Description
Identify a dataset containing at least one field with multiple overlapping visits.	

Expected Result

A dataset that has been ingested into a Butler repository.

Step 2-1 from LVV-T860 Description

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

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

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

Example Code

source 'path'
setup lsst_distrib

Expected Result

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

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

Step 3-1 from LVV-T1744 Description

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

Example Code

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

Expected Result

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

Step 4	Description
Confirm that the metric AA1 has been calculated, and that its values are reasonable.	
	Expected Result
A JSON file (and/or a report generated from that JSON file) demonstrating that AA1 has been calculated.	

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T1749 in Jira

4.2.61.1 Verification Elements

None.

4.2.61.2 Test Items

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.**

4.2.61.3 Test Procedure

Step 1	Description
Identify a dataset containing at least one field with multiple overlapping visits.	
	Expected Result
A dataset that has been ingested into a Butler repository.	

Step 2-1 from LVV-T860 Description

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

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

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

Example Code

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

Expected Result

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

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

Step 3-1 from LVV-T1744 Description

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

Example Code

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

Expected Result

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

Step 4 Description

Confirm that the metric 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).

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T1750 in Jira

4.2.62.1 Verification Elements

None.

4.2.62.2 Test Items

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.**

4.2.62.3 Test Procedure

Step 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.

Step 2-1 from LVV-T860	Description
The 'path' that you will use depends on where you are running the science pipelines. Options:	

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

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

Example Code

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

Expected Result

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

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

Step 3-1 from LVV-T1744 Description

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

Example Code

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

Expected Result

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

Step 4 Description

Confirm that the metric 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).

4.2.63 LVV-T1751 - Verify calculation of median relative astrometric measurement error on 200 arcminute scales

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T1751 in Jira

4.2.63.1 Verification Elements

None.

4.2.63.2 Test Items

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.

4.2.63.3 Test Procedure

Step 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.

Step 2-1 from LVV-T860	Description

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

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

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

Example Code

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

Expected Result

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

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

Step 3-1 from LVV-T1744 Description

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

Example Code

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

Expected Result

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

Step 4 Description

Confirm that the metric 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.

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T1752 in Jira

4.2.64.1 Verification Elements

None.

4.2.64.2 Test Items

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.**

4.2.64.3 Test Procedure

Step 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.

Step 2-1 from LVV-T860	Description

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

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

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

Example Code

```
source 'path'
```

setup lsst_distrib

Expected Result

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

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

Step 3-1 from LVV-T1744 Description

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

Example Code

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

Expected Result

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

Step 4 Description

Confirm that the metric 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).

4.2.65 LVV-T1753 - Verify calculation of RMS difference of separations relative to r-band

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T1753 in Jira

4.2.65.1 Verification Elements

None.

4.2.65.2 Test Items

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.**

4.2.65.3 Test Procedure

Step 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.

Step 2-1 from LVV-T860 Description

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

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

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

Example Code

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

Expected Result

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

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

Step 3-1 from LVV-T1744	Description

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

Example Code

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

Expected Result

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

Step 4	Description

Confirm that the metric 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.

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T1754 in Jira

4.2.66.1 Verification Elements

None.

4.2.66.2 Test Items

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

4.2.66.3 Test Procedure

Step 1 Description

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

Expected Result

A dataset that has been ingested into a Butler repository.

Step 2-1 from LVV-T860 Description

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

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

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

Example Code

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

Expected Result

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

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

Step 3-1 from LVV-T1744 Description

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

Example Code

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

Expected Result

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

Step 4	Description

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

Expected Result

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

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T1755 in Jira

4.2.67.1 Verification Elements

None.

4.2.67.2 Test Items

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].**

4.2.67.3 Test Procedure

Step 1	Description

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

Expected Result

A dataset that has been ingested into a Butler repository.

Step 2-1 from LVV-T860	Description

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

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

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

Example Code

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

Expected Result

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

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

Step 3-1 from LVV-T1744	Description

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

Example Code

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

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

Step 4	Description
Confirm that the metric TE1 has been calculated, and that its values are reasonable.	

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

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T1756 in Jira

4.2.68.1 Verification Elements

None.

4.2.68.2 Test Items

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

4.2.68.3 Test Procedure

Step 1	Description
Identify a dataset containing at least one field in each of the u, z, and y filters with multiple overlapping visits.	

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

Step 2-1 from LVV-T1744	Description

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

Example Code

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

Expected Result

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

Step 3	Description

Confirm that the metric PA1uzy has been calculated, and that its values are reasonable.

Expected Result

A JSON file (and/or a report generated from that JSON file) demonstrating that PA1uzy has been calculated.

4.2.69 LVV-T1757 - Verify calculation of photometric repeatability in gri filters

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T1757 in Jira

4.2.69.1 Verification Elements

None.

4.2.69.2 Test Items

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

4.2.69.3 Test Procedure

Step 1	Description

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

Expected Result

A dataset that has been ingested into a Butler repository.

Step 2-1 from LVV-T1744	Description

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

Example Code

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

Expected Result

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

Step 3	Description

Confirm that the metric PA1gri has been calculated, and that its values are reasonable.

Expected Result

A JSON file (and/or a report generated from that JSON file) demonstrating that PA1gri has been calculated.

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T1758 in Jira

4.2.70.1 Verification Elements

None.

4.2.70.2 Test Items

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

4.2.70.3 Test Procedure

Step 1 Description

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

Expected Result

A dataset that has been ingested into a Butler repository.

Step 2-1 from LVV-T860 Description

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

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

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

Example Code

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

Expected Result

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

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

Step 3-1 from LVV-T1744 Description

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

Example Code

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

Expected Result

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

Step 4 Description

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

Expected Result

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

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T1759 in Jira

4.2.71.1 Verification Elements

None.

4.2.71.2 Test Items

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

4.2.71.3 Test Procedure

Step 1	Description

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

Expected Result

A dataset that has been ingested into a Butler repository.

Step 2-1 from LVV-T860	Description

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

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

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

Example Code

```
source ‘path‘
setup lsst_distrib
```

Expected Result

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

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

Step 3-1 from LVV-T1744	Description

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

	Example Code

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

	Expected Result

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

Step 4	Description

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

	Expected Result

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

4.2.72 LVV-T1830 - Verify Implementation of Scientific Visualization of Camera Image Data

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Inspection	Jeffrey Carlin

Open LVV-T1830 in Jira

4.2.72.1 Verification Elements

None.

4.2.72.2 Test Items

Verify that all scientific visualization of camera image data uses the coordinate systems defined in LSE-349.

4.2.72.3 Test Procedure

Step 1	Description

Identify an image containing bright saturated stars. Load this image into an image viewer such as Firefly or DS9.

Expected Result

Image with bright stars is displayed.

Step 2	Description

Confirm that each of the following is true:

- the XY coordinate origin is at the lower left,
- the x-coordinate increases left-to-right, and the y-coordinate increases bottom-to-top
- bleed trails of saturated stars are vertical (i.e., the parallel transfer direction is oriented vertically)
- the sky orientation places east 90 degrees counter-clockwise from north

Expected Result

Via coordinate grid overlays or similar, an image is demonstrated to meet the necessary conditions.

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T1946 in Jira

4.2.73.1 Verification Elements

None.

4.2.73.2 Test Items

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

4.2.73.3 Test Procedure

Step 1-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 2	Description

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

Expected Result

Step 3	Description

Verify that the coadd catalog provides measurements in flux units.

Expected Result

Confirmation of measurements in catalogs encoded in flux units.

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

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T1947 in Jira

4.2.74.1 Verification Elements

None.

4.2.74.2 Test Items

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

4.2.74.3 Test Procedure

Step 1-1 from LVV-T987 | Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 2 | Description

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

Expected Result

Step 3 | Description

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

Expected Result

Confirmation of measurements in catalogs encoded in flux units.

4.2.75 LVV-T2202 - Verify that the of zero-point error outlier limit threshold (PA4) can be applied.

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Leanne Guy

Open LVV-T2202 in Jira

4.2.75.1 Verification Elements

None.

4.2.75.2 Test Items

Verify that the DMS has provided the code to apply the zero-point error outlier limit threshold (PA4) to computed values of metrics.

4.2.75.3 Test Procedure

Step 1	Description

Inspect the PF2 pipeline code to see if the PA4 threshold has been specified.

Example Code

```
config.measure.threshPA4 = 15.0
```

Expected Result

The PA4 threshold is specified as part of the PA4 pipeline

Step 2-1 from LVV-T1744	Description

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

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

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

Step 3	Description
Confirm that the PA4 threshold was applied to the assessment of the computed metric PF2	

Expected Result
The dataset has been ingested into a Gen3 Butler repository and is accessible

4.3 Draft Test Cases

4.3.1 LVV-T23 - Verify implementation of Storing Approximations of Per-pixel Metadata

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Simon Krughoff

Open LVV-T23 in Jira

4.3.1.1 Verification Elements

None.

4.3.1.2 Test Items

Test Items

Show that the compressed form depth and mask maps adequately represents the exact version of the same information.

4.3.1.3 Test Procedure

Step 1-1 from LVV-T860 Description

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

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

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

Example Code

source 'path'
setup lsst_distrib

Expected Result

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

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

Step 2-1 from LVV-T987 Description

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

Example Code

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

Expected Result
Butler repo available for reading.

Step 3	Description

For each of the expected data products types (listed in Test Items section 4.3.2) and each of the expected units (PVIs, coadds, etc), retrieve the data product from the Butler and verify that it is non-empty.

Expected Result

Step 4	Description

Create the coadd pixel level depth map for the HSC PDR dataset.

Expected Result

Step 5	Description

Generate compressed representation of the pixel level depth map.

Expected Result

Step 6	Description

Create the coadd pixel level mask map for the HSC PDR dataset.

Expected Result

Step 7	Description

Generate compressed representation of the mask map.

Expected Result

Step 8	Description

Sample randomly from both the pixel level and compressed depth maps. Compare the distribution of depths sampled from the pixel level depth map to that sampled from the compressed representation.

Expected Result

Step 9	Description

Divide the mask planes into two groups: INFO and BAD. BAD flags are any that would cause a particular pixel to be excluded from processing: e.g. EDGE, SAT, BAD. Sample masks from both the pixel level mask map and the compressed mask map.

For each sample, compute sum(mask_pixel xor mask_compressed). Produce the distribution of the number of bits that differ between the samples.

Repeat for both the INFO flags and the BAD flags.

Expected Result

4.3.2 LVV-T24 - Verify implementation of Computing Derived Quantities

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Melissa Graham

Open LVV-T24 in Jira

4.3.2.1 Verification Elements

None.

4.3.2.2 Test Items

To confirm that common derived quantities (apparent magnitude, FWHM in arcsec, ellipticity) are available to an end-user by, e.g., ensuring a color-color diagram is easy to construction, fitting functions to derived data, or generating other common scientific derivatives.

4.3.2.3 Test Procedure

Step 1-1 from LVV-T860	Description

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

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

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

Example Code

source 'path'
setup lsst_distrib

Expected Result

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

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

Step 2-1 from LVV-T987 Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 3 Description

For each of the expected data product types (listed in Test Items section 4.3.2) and each of the expected units (PVIs, coadds, etc), retrieve the data product from the Butler and verify it to be non-empty.

Expected Result

Step 4	Description

Load into DPDD+Science Platform

	Expected Result

Step 5	Description

Constructing color-color diagram and fitting stellar locus in Science Platform.

	Expected Result

Step 6	Description

Invite three members of commissioning team to create color-color diagram from coadd catalogs based on merged coadd reference catalog.

	Expected Result

4.3.3 LVV-T25 - Verify implementation of Denormalizing Database Tables

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Colin Slater

Open LVV-T25 in Jira

4.3.3.1 Verification Elements

None.

4.3.3.2 Test Items

Verify that commonly useful views of data are easy to obtain through the Science Platform.

4.3.3.3 Test Procedure

Step 1	Description

Connect to the Science Platform's portal query interface.

	Expected Result

Step 2	Description

List the available views in the database.

	Expected Result

Step 3	Description

Take 20 sampled queries and determine which are easily done on views and which require complicated joins. Discuss the complicated ones and determine if any could be simplified by adding additional views.

	Expected Result

4.3.4 LVV-T26 - Verify implementation of Maximum Likelihood Values and Covariances

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jim Bosch

Open LVV-T26 in Jira

4.3.4.1 Verification Elements

None.

4.3.4.2 Test Items

- Check that all measurements in source and object schemas include columns containing uncertainties, including covariances between jointly-measured quantities.

- Check that all model-fit measurements in source and object schemas include columns that report goodness-of-fit.
- Check that most sources and objects with successful measurements report finite uncertainty values for those measurements.
- Check that most sources and objects with successful model-fit measurements report finite goodness-of-fit values.

4.3.4.3 Test Procedure

Step 1-1 from LVV-T860	Description

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

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

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

Example Code

```
source ‘path‘
setup lsst_distrib
```

Expected Result

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

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

Step 2-1 from LVV-T987	Description

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

Example Code

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

Expected Result
Butler repo available for reading.

Step 3	Description

For each of the expected data product types (listed in Test Items section 4.3.2) and each of the expected units (PVIs, coadds, etc), retrieve the data product from the Butler and verify it to be non-empty.

Expected Result

Step 4	Description

Verify that maximum likelihood and covariant quantities are provided. Test and manually inspect that they are reasonable (finite, appropriately normed).

Expected Result

4.3.5 LVV-T27 - Verify implementation of Data Availability

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Gregory Dubois-Felsmann

Open LVV-T27 in Jira

4.3.5.1 Verification Elements

None.

4.3.5.2 Test Items

Determine if all required categories of raw data (specifically enumerated: raw exposures, calibration frames, telemetry, configuration metadata) can be located through the Science Platform and are available for download. Verify through (1) administrative review; (2) checking with precursor data; (3) checking on early data feeds from the Summit such as from AuxTel and ComCam.

4.3.5.3 Test Procedure

Step 1	Description
Invite two reviewers to review that plan that seems reasonable to expect the archiving and provision of raw data	
	Expected Result

Step 2	Description
Pass a set of HSC data through (equal in size to the first public data release) the data backbone through ingest and provide interface	
	Expected Result

Step 3	Description
Track the ingestion of AuxTel data during one month in 2018-2019 and verify delivery and test download.	
	Expected Result

4.3.6 LVV-T35 - Verify implementation of Nightly Data Accessible Within 24 hrs

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T35 in Jira

4.3.6.1 Verification Elements

None.

4.3.6.2 Test Items

Test Items

Verify that
1. Alerts are available within OTT1
2. Level 1 Data Products are available within L1PublicT
3. Solar System Object orbits are available within L1PublicT of the updated calculations completion on the following night.

4.3.6.3 Test Procedure

Step 1-1 from LVV-T860	Description

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

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

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

Example Code

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

Expected Result

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

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

Step 2-1 from LVV-T866	Description

Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.

Expected Result

An output dataset including difference images and DIASource and DIAObject measurements.

Step 2-2 from LVV-T866	Description

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

Expected Result

Step 3	Description

Time processing of data starting from (pre-ingested) raw files until an alert is available for distribution; verify that this time is less than OTT1.

Expected Result

Step 4	Description

Time processing of data starting from (pre-ingested) raw files until the required data products are available in the Science Platform. Verify that this time is less than L1PublicT.

Expected Result

Step 5	Description

Run MOPS on 1 night equivalent of LSST observing worth of precursor data and verify that Solar System Object orbits can be updated within 24 hours.

Expected Result

Step 6	Description

Record time between completion of MOPS processing and availability of the updated SSObject catalogue through the Science Platform; verify this time is less than L1PublicT.

Expected Result

4.3.7 LVV-T36 - Verify implementation of Difference Exposures

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T36 in Jira

4.3.7.1 Verification Elements

None.

4.3.7.2 Test Items

Verify successful creation of a

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

4.3.7.3 Test Procedure

Step 1-1 from LVV-T860	Description

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

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

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

Example Code

source 'path'
setup lsst_distrib

Expected Result

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

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

Step 2-1 from LVV-T866 Description

Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.

Expected Result

An output dataset including difference images and DIASource and DIAObject measurements.

Step 2-2 from LVV-T866 Description

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

Expected Result

Step 3 Description

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

Expected Result

4.3.8 LVV-T37 - Verify implementation of Difference Exposure Attributes

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T37 in Jira

4.3.8.1 Verification Elements

None.

4.3.8.2 Test Items

Verify that for each Difference Exposure the DMS stores
1. The identify of the input exposures and related provenance information
2. Metadata attributes of the subtraction, including the PSF-matching kernel used.

4.3.8.3 Test Procedure

Step 1-1 from LVV-T860	Description

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

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

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

Example Code

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

Expected Result

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

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

Step 2-1 from LVV-T866	Description

Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.

Expected Result

An output dataset including difference images and DIASource and DIAObject measurements.

Step 2-2 from LVV-T866	Description

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

Expected Result

Step 3	Description

For each of HSC PDR and DECAM HiTS data: set up three different templates and run subtractions on 10 different images from at least two different filters. Verify that we can recover the provenance information about which template was used for each subtraction, which input images were used for that template, and that we can successfull extract the PSF matching kernel.

Expected Result

4.3.9 LVV-T44 - Verify implementation of Documenting Image Characterization

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jim Bosch

Open LVV-T44 in Jira

4.3.9.1 Verification Elements

None.

4.3.9.2 Test Items

Verify that the persisted format for Processed Visit Images and associated instrument-signature-removal data products is documented.

4.3.9.3 Test Procedure

Step 1	Description
Delegate to Alert Production	
	Expected Result

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

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T46 in Jira

4.3.10.1 Verification Elements

None.

4.3.10.2 Test Items

Verify that the DMS produces a Prompt Processing Performance Report. Specifically check that the number of observations that describe each of the following:
1. Successfully processed, recoverable failures, unrecoverable failures.
2. Archived
3. Result in science.

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

4.3.10.3 Test Procedure

Step 1	Description

Execute single-day operations rehearsal, observe report

	Expected Result

4.3.11 LVV-T49 - Verify implementation of DIASource Catalog

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T49 in Jira

4.3.11.1 Verification Elements

None.

4.3.11.2 Test Items

Verify that the DMS produces a Source catalog from Difference Exposures with the required attributes.

4.3.11.3 Test Procedure

Step 1-1 from LVV-T860	Description

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

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

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

Example Code

source 'path'
setup lsst_distrib

Expected Result

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

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

Step 2-1 from LVV-T866 Description

Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.

Expected Result

An output dataset including difference images and DIASource and DIAObject measurements.

Step 2-2 from LVV-T866 Description

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

Expected Result

Step 3-1 from LVV-T987 Description

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

Example Code

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

Expected Result
Butler repo available for reading.

Step 4	Description
Verify that products are produced for DIASource catalog	

Expected Result

4.3.12 LVV-T50 - Verify implementation of Faint DIASource Measurements

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T50 in Jira

4.3.12.1 Verification Elements

None.

4.3.12.2 Test Items

Verify that the DMS can produces DIASources measurements for sources below the nominal S/N cutoff that satisfy additional criteria.

4.3.12.3 Test Procedure

Step 1-1 from LVV-T860	Description

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

- local (newinstall.sh - based install):[path_to_installation]/loadLSST.bash

- development cluster ("lsst-dev"): /software/lsstsw/stack/loadLSST.bash
- LSP Notebook aspect (from a terminal): /opt/lsst/software/stack/loadLSST.bash

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

Example Code

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

Expected Result

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

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

Step 2-1 from LVV-T866 Description

Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.

Expected Result

An output dataset including difference images and DIASource and DIAObject measurements.

Step 2-2 from LVV-T866 Description

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

Expected Result

Step 3 Description

As an example of selecting with constrains, Re-run source detection as an afterburner to select isolated sources (defined as more than 2 arcseconds away from any other objects in the single-image-depth catalog) that are fainter than the fiducial transSNR cut.

Expected Result

4.3.13 LVV-T51 - Verify implementation of DIAObject Catalog

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T51 in Jira

4.3.13.1 Verification Elements

None.

4.3.13.2 Test Items

Verify that the DIAObject includes a unique ID, identifiers for nearest stars and nearest galaxies, and probability of matching to static Object.

4.3.13.3 Test Procedure

Step 1-1 from LVV-T866	Description

Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.

Expected Result

An output dataset including difference images and DIASource and DIAObject measurements.

Step 1-2 from LVV-T866	Description

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

Expected Result

Step 2-1 from LVV-T987	Description

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

Example Code

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

Expected Result
Butler repo available for reading.

Step 3	Description
Verify that DIAObjects have diaNearbyObjMaxStar and diaNearbyObjMaxGalaxies that point to the Object catalog and are within dianNearbyObjRadius; the probability of association; and the required DIAObject properties.	

Expected Result

4.3.14 LVV-T52 - Verify implementation of DIAObject Attributes

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T52 in Jira

4.3.14.1 Verification Elements

None.

4.3.14.2 Test Items

Verify that the DMS provides summary attributes for each DIAObject, including periodicity measures.

4.3.14.3 Test Procedure

Step 1-1 from LVV-T866	Description

Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.

Expected Result

An output dataset including difference images and DIASource and DIAObject measurements.

Step 1-2 from LVV-T866	Description

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

Expected Result

Step 2-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 3	Description

Confirm that the DIAObjects include summary attributes as specified.

Expected Result

4.3.15 LVV-T53 - Verify implementation of SSObject Catalog

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T53 in Jira

4.3.15.1 Verification Elements

None.

4.3.15.2 Test Items

Verify that the DMS produces a catalog of Solar System Objects identify from Moving Object Processing.
Verify that the SSObject catalog includes orbital elements and additional related quanitites.

4.3.15.3 Test Procedure

Step 1-1 from LVV-T866 Description

Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.

Expected Result

An output dataset including difference images and DIASource and DIAObject measurements.

Step 1-2 from LVV-T866 Description

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

Expected Result

Step 2-1 from LVV-T901 Description

Perform the steps of Moving Object Pipeline (MOPS) processing on newly detected DIASources, and generate Solar System data products including Solar System objects with associated Keplerian orbits, errors, and detected DIASources. This includes running processes to link DIASource detections within a night (called tracklets), to link these tracklets across multiple nights (into tracks), to fit the tracks with an orbital model to identify those tracks that are consistent with an asteroid orbit, to match these new orbits with existing SSObjects, and to update the SSObject table.

Expected Result

An output dataset consisting of an updated SSObject database with SSObjects both added and pruned as the orbital fits have been refined, and an updated DIASource database with DIASources assigned and unassigned to SSObjects.

Step 2-2 from LVV-T901	Description

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

Expected Result

Step 3-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 4	Description

Inspect SSObject catalog and verify the presence of the required elements (LVV-104).

Expected Result

4.3.16 LVV-T54 - Verify implementation of Alert Content

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T54 in Jira

4.3.16.1 Verification Elements

None.

4.3.16.2 Test Items

Verify that the DMS creates an Alert for each detected DIASource
Verify that this Alert is broadcasted using community protocols
Verify that the context of the Alert packet match requirements.

4.3.16.3 Test Procedure

Step 1-1 from LVV-T866	Description
Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.	
	Expected Result
An output dataset including difference images and DIASource and DIAObject measurements.	

Step 1-2 from LVV-T866	Description
Verify that the expected data products have been produced, and that catalogs contain reasonable values for measured quantities of interest.	
	Expected Result

Step 2	Description
Examine the serialized alert packets to confirm the presence of the required elements (LVV-105).	
	Expected Result

4.3.17 LVV-T55 - Verify implementation of DIAForcedSource Catalog

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T55 in Jira

4.3.17.1 Verification Elements

None.

4.3.17.2 Test Items

Verify that the DMS produces a DIAForcedSource Catalog and that the catalog contains measured fluxes for DIAObjects.

4.3.17.3 Test Procedure

Step 1-1 from LVV-T866	Description

Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.

Expected Result

An output dataset including difference images and DIASource and DIAObject measurements.

Step 1-2 from LVV-T866	Description

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

Expected Result

Step 2-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 3	Description

Confirm that the DIAForcedSource catalog contains measurements for each source.

	Expected Result

4.3.18 LVV-T56 - Verify implementation of Characterizing Variability

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T56 in Jira

4.3.18.1 Verification Elements

None.

4.3.18.2 Test Items

Verify that the variability characterization in the DIAObject catalog includes data collected within previous “diaCharacterizationCutoff” period of time.

4.3.18.3 Test Procedure

Step 1-1 from LVV-T866	Description

Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.

	Expected Result

An output dataset including difference images and DIASource and DIAObject measurements.

Step 1-2 from LVV-T866	Description

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

Expected Result

Step 2	Description

Verify that the issued alerts contain measurements during the diaCharacterizationCutoff.

Expected Result

4.3.19 LVV-T57 - Verify implementation of Calculating SSObject Parameters

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T57 in Jira

4.3.19.1 Verification Elements

None.

4.3.19.2 Test Items

Verify that the DMS database provides functions to compute phase angles and magnitudes in LSST bands for every SSObject.

4.3.19.3 Test Procedure

Step 1-1 from LVV-T866	Description

Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.

Expected Result

An output dataset including difference images and DIASource and DIAObject measurements.

Step 1-2 from LVV-T866	Description

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

	Expected Result

Step 2-1 from LVV-T901	Description

Perform the steps of Moving Object Pipeline (MOPS) processing on newly detected DIASources, and generate Solar System data products including Solar System objects with associated Keplerian orbits, errors, and detected DIASources. This includes running processes to link DIASource detections within a night (called tracklets), to link these tracklets across multiple nights (into tracks), to fit the tracks with an orbital model to identify those tracks that are consistent with an asteroid orbit, to match these new orbits with existing SSObjects, and to update the SSObject table.

	Expected Result

An output dataset consisting of an updated SSObject database with SSObjects both added and pruned as the orbital fits have been refined, and an updated DIASource database with DIASources assigned and unassigned to SSObjects.

Step 2-2 from LVV-T901	Description

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

	Expected Result

Step 3	Description

Computer the phase angle, reduced and absolute asteroid magnitudes for objects identified in SSObject Catalog

	Expected Result

4.3.20 LVV-T58 - Verify implementation of Matching DIASources to Objects

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T58 in Jira

4.3.20.1 Verification Elements

None.

4.3.20.2 Test Items

Verify that a cross-match table is available between DIASources and Objects.

4.3.20.3 Test Procedure

Step 1-1 from LVV-T866 Description

Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.

Expected Result

An output dataset including difference images and DIASource and DIAObject measurements.

Step 1-2 from LVV-T866 Description

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

Expected Result

Step 2-1 from LVV-T987 Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 3 Description

Verify that a cross-match table between the Prompt DIASources and DRP Objects is available.

Expected Result

4.3.21 LVV-T59 - Verify implementation of Regenerating L1 Data Products During Data Release Processing

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Kian-Tat Lim

Open LVV-T59 in Jira

4.3.21.1 Verification Elements

None.

4.3.21.2 Test Items

Verify that the Prompt Processing data products are regenerated during DRP.

4.3.21.3 Test Procedure

Step 1	Description

Execute DRP

Expected Result

Step 2	Description

Observe production of difference image data products

Expected Result

4.3.22 LVV-T60 - Verify implementation of Publishing predicted visit schedule

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T60 in Jira

4.3.22.1 Verification Elements

None.

4.3.22.2 Test Items

Verify that a predict-visit schedule can be published by the OCS.

4.3.22.3 Test Procedure

Step 1	Description
	Expected Result

4.3.23 LVV-T63 - Verify implementation of Produce Images for EPO

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Gregory Dubois-Felsmann

Open LVV-T63 in Jira

4.3.23.1 Verification Elements

None.

4.3.23.2 Test Items

This test will verify that the DRP pipelines produce the image data products called out in LSE-131. Currently this is limited to a color all-sky HiPS map. This will be verified (1) by inspection of pipeline configurations and (2) in operations rehearsals on precursor data. The production of a usable HiPS map will be verified by browsing it with community tools.

4.3.23.3 Test Procedure

Step 1-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 2	Description

For each of the expected data product types needed for creation of HiPS images, retrieve the data product from the Butler and verify it to be non-empty.

Expected Result

Step 3	Description

Verify that a HiPS image map covering the LSST survey area, with a limiting depth yielding 1 arcsecond resolution, has been produced matching the color prescriptions provided by EPO (in updates to LSE-131 which are expected to be made "once ComCam data is available").

Expected Result

Step 4	Description

Place the image map in a location accessible to a Firefly and an Aladin Lite client, ideally with the client running in the EPO data systems environment.

	Expected Result

Step 5	Description

Use Firefly to manually explore the image map at the largest scales to verify coverage of the entire sky. Sample in various locations to confirm the 1 arcsecond maximum depth.
Confirm using Aladin Lite that the format of the image map is supported by this common community tool.

	Expected Result

Step 6	Description

Verify programmatically, perhaps both by sampling a variety of locations, and by counting the tiles created at the 1-arcsecond-resolution depth, that the map is complete and meets its specifications.

	Expected Result

Step 7	Description

Apply an IVOA-community HiPS service validation tool, if available, to the service location.

	Expected Result

Step 8	Description

Verify that the HiPS map created is in a location accessible to the EPO data systems.

	Expected Result

4.3.24 LVV-T64 - Verify implementation of Coadded Image Provenance

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jim Bosch

Open LVV-T64 in Jira

4.3.24.1 Verification Elements

None.

4.3.24.2 Test Items

Verify that all coadd data products produced by the DRP pipelines are associated with provenance information that includes the set of input epochs contributing to that coadd as well as any additional information needed to exactly produce that coadd.

4.3.24.3 Test Procedure

Step 1-1 from LVV-T860 Description

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

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

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

Example Code

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

Expected Result

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

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

Step 2-1 from LVV-T987 Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 3	Description

For each of the expected data product types and each of the expected units (PVIs, coadds, etc), retrieve the data product from the Butler and verify it to be non-empty.

Expected Result

Step 4	Description

Query and verify provenance of input images, and software versions that went into producing stack.

Expected Result

Step 5	Description

Test re-generating 10 different coadds tract+patches based on the provenance image given

Expected Result

4.3.25 LVV-T67 - Verify implementation of Object Catalog

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jim Bosch

Open LVV-T67 in Jira

4.3.25.1 Verification Elements

None.

4.3.25.2 Test Items

Verify that the DRP pipelines produce an Object catalog derived from detections made on both coadded images and difference images and measurements performed on coadds and possibly overlapping single-epoch images.

4.3.25.3 Test Procedure

Step 1	Description

load LSST DM Stack

	Expected Result

Step 2	Description

Run the single-frame processing and self-calibration steps of the DRP pipeline.

	Expected Result

Step 3	Description

Insert simulated sources into all single-frame images, including:

- static objects (e.g. galaxies), including some too faint to be detectable in single-epoch images;
- objects with static positions that are sufficiently bright and variable that they should be detectable in single-epoch difference images;
- transient objects that appear in only a few epochs;
- stars with significant proper motions and parallaxes, some below the single-epoch detection limit
- simulated solar system objects with orbits that can be constrained from just the epochs in the test dataset

	Expected Result

Step 4	Description

Run all remaining DRP pipeline steps.

Expected Result

Step 5	Description

Load data into DRP database

Expected Result

Step 6	Description

Verify that the injected simulated objects are recovered at a rate consistent with their S/N *when not blended with each other or real objects*, and that flags indicating how each Object was detected are consistent with their properties:

- static objects should be detected in coadds only (not difference images)
- static-position/variable-flux objects should be detected in coadds and possibly difference images
- transient objects should be detected in difference images only
- stars with significant proper motions may be detected in either coadds or difference images
- solar system objects should be detected in difference images only.

Expected Result

4.3.26 LVV-T68 - Verify implementation of Provide Photometric Redshifts of Galaxies

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jim Bosch

Open LVV-T68 in Jira

4.3.26.1 Verification Elements

None.

4.3.26.2 Test Items

Verify that Object catalogs produced by the DRP Pipeline include photometric redshift information.

4.3.26.3 Test Procedure

Step 1	Description
Run DRP processing steps through (at least) final galaxy photometry measurements.	
	Expected Result

Step 2	Description
Train photometric redshift algorithm(s) on spectroscopic and high-accuracy photometric redshift catalogs.	
	Expected Result

Step 3	Description
Estimate photometric redshifts for all Objects generated by DRP processing.	
	Expected Result

Step 4	Description
Load into DRP Database	
	Expected Result

Step 5	Description
Inspect database to verify that photometric redshifts are present for all objects	
	Expected Result

4.3.27 LVV-T69 - Verify implementation of Object Characterization

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jim Bosch

Open LVV-T69 in Jira

4.3.27.1 Verification Elements

None.

4.3.27.2 Test Items

Verify that Object catalogs produced by the DRP pipeline include all measurements listed in DMS-REQ-0276: a point-source model fit, a bulge-disk model fit, standard colors, a centroid, adap- tive moments, Petrosian and Kron fluxes, surface brightness at multiple apertures, proper motion and parallax, and a variability characterization.

4.3.27.3 Test Procedure

Step 1	Description
Precursor data, execute DRP, load results, observe catalog contents	
	Expected Result

4.3.28 LVV-T71 - Verify implementation of Detecting extended low surface brightness objects

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jim Bosch

Open LVV-T71 in Jira

4.3.28.1 Verification Elements

None.

4.3.28.2 Test Items

Verify that low-surface brightness objects (including those whose PSF S/N is lower than the detection threshold) are detected in coadds.

4.3.28.3 Test Procedure

Step 1	Description
load LSST DM Stack	
	Expected Result

Step 2	Description
Run the single-frame processing and self-calibration steps of the DRP pipeline.	
	Expected Result

Step 3	Description
Insert simulated low-surface-brightness galaxies (with exponential profiles) consistently into all calibrated single-epoch images.	
	Expected Result

Step 4	Description
Run all remaining DRP pipeline steps.	
	Expected Result

Step 5	Description
Load data into DRP database	
	Expected Result

Step 6	Description
Verify that the injected simulated objects are recovered at a rate consistent with their S/N and true profile *when not blended with each other or real objects.*	

Expected Result

4.3.29 LVV-T72 - Verify implementation of Coadd Image Method Constraints

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jim Bosch

Open LVV-T72 in Jira

4.3.29.1 Verification Elements

None.

4.3.29.2 Test Items

Verify the implementation of how Coadd images are created.

4.3.29.3 Test Procedure

Step 1 Description

Identify a dataset that has been processed to create coadd images.

Expected Result

Step 2-1 from LVV-T987 Description

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

Example Code

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

Expected Result
Butler repo available for reading.

Step 3	Description
Retrieve the coadds in the dataset and verify that they are non-empty.	

Expected Result

Step 4	Description
Verify that coadds were created following specification	

Expected Result

4.3.30 LVV-T73 - Verify implementation of Deep Detection Coadds

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jim Bosch

Open LVV-T73 in Jira

4.3.30.1 Verification Elements

None.

4.3.30.2 Test Items

Verify that the DRP pipelines produce a suite of per-band coadded images that are optimized for depth.

4.3.30.3 Test Procedure

Step 1-1 from LVV-T987	Description
Identify the path to the data repository, which we will refer to as 'DATA/path', then execute the following:	

Example Code

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

Expected Result

Butler repo available for reading.

Step 2	Description

Verify through inspection that per-filter coadds exist for each tract+patch possible

Expected Result

Step 3	Description

Verify through inspection that the images used to generate those coadds met specified conditions

Expected Result

Step 4	Description

Visually inspect a subset of the coadds to verify that they visually appear reasonable and to be from good quality data.

Expected Result

4.3.31 LVV-T75 - Verify implementation of Multi-band Coadds

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jim Bosch

Open LVV-T75 in Jira

4.3.31.1 Verification Elements

None.

4.3.31.2 Test Items

Verify that the DRP pipelines produce multi-band coadds for detection purposes.

4.3.31.3 Test Procedure

Step 1-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 2	Description

Verify that deep detection coadds exist based on all filters.

Expected Result

4.3.32 LVV-T76 - Verify implementation of All-Sky Visualization of Data Releases

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Simon Krughoff

Open LVV-T76 in Jira

4.3.32.1 Verification Elements

None.

4.3.32.2 Test Items

Show that it's possible to produce large area visualizations from Data Release data products.

4.3.32.3 Test Procedure

Step 1-1 from LVV-T987 Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 2 Description

Run all sky tile generation task to produce the data products necessary for serving the all sky visualization.

Expected Result

Step 3 Description

Manually perform, and log (including timing where applicable), the following steps against that all sky visualization application. At all steps take special care to note any missing or un-rendered image tiles:

1. Navigate to the all sky viewer and log the URL, browser and version.
2. Zoom to native pixel display (1 image pixel per display pixel)
3. Zoom to fit the full PDR footprint
4. Zoom to 1/4x native resolution
5. Pan to eastern edge of the footprint.
6. Pan to western edge of the footprint.

7. Navigate to the middle of the footprint.
8. Zoom to max magnification

Expected Result

4.3.33 LVV-T79 - Verify implementation of PSF-Matched Coadds

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jim Bosch

Open LVV-T79 in Jira

4.3.33.1 Verification Elements

None.

4.3.33.2 Test Items

Verify that the DRP pipelines produce PSF matched coadds.

4.3.33.3 Test Procedure

Step 1-1 from LVV-T987 Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 2	Description

Verify that PSF-matched coadds were created.

	Expected Result

4.3.34 LVV-T80 - Verify implementation of Detecting faint variable objects

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Melissa Graham

Open LVV-T80 in Jira

4.3.34.1 Verification Elements

None.

4.3.34.2 Test Items

To verify that the Data Release Production pipeline will be able to detect faint sources with long-term variability (e.g., quasars, proper motion stars) via, e.g., shorter timescale coadds (month to a few months).

4.3.34.3 Test Procedure

Step 1-1 from LVV-T866	Description

Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.

	Expected Result

An output dataset including difference images and DIASource and DIAObject measurements.

Step 1-2 from LVV-T866	Description

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

Expected Result

Step 2-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 3	Description

Identify 100 objects from Gaia with proper motions high enough to have detectably moved during HSC observations.

Expected Result

Step 4	Description

Measure reported proper motion of these objects in DM Stack processing. Verify that it is consistent with Gaia objects.

Expected Result

Step 5	Description

Identify 100 quasars from color-space or existing extragalactic spectroscopic catalog.

Expected Result

Step 6	Description

Measure lightcurves of these quasars. Determine if structure function is reasonable (may require at least a year to determine if the structure function of 100 quasars is "reasonable").

Expected Result

Step 7	Description

(Alternative: if faint variable source can be injected into the input data, test to see if they are recovered).

	Expected Result

(This Alternative would enable us not only to tell if faint variable objects are detected, but exactly which kinds, how faint, and with what efficiency.)

4.3.35 LVV-T81 - Verify implementation of Targeted Coadds

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jim Bosch

Open LVV-T81 in Jira

4.3.35.1 Verification Elements

None.

4.3.35.2 Test Items

Verify that small sections of any coadd produced by the DRP pipelines can be retained, even if the full coadd is not.

4.3.35.3 Test Procedure

Step 1	Description

Remove DR from disk

	Expected Result

Step 2	Description

Observe retention of designated coadd sections

	Expected Result

Step 3	Description

Observe accessibility of designated coadd sections via simulated DAC LSP instance

	Expected Result

4.3.36 LVV-T86 - Verify implementation of Illumination Correction Frame

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Lupton

Open LVV-T86 in Jira

4.3.36.1 Verification Elements

None.

4.3.36.2 Test Items

Verify that the DMS can produce an illumination correction frame calibration product. Verify that the DMS can determine the effectiveness of an illumination correction and determine how often it should be updated.

4.3.36.3 Test Procedure

Step 1	Description

Delegate to CPP

	Expected Result

4.3.37 LVV-T87 - Verify implementation of Monochromatic Flatfield Data Cube

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Lupton

Open LVV-T87 in Jira

4.3.37.1 Verification Elements

None.

4.3.37.2 Test Items

Verify that the DMS can generate a calibration image/cube that corrects for pixel-to-pixel wavelength-dependent detector response.
Verify that the DMS can measure the effectiveness of this monochromatic flatfield data cube.

4.3.37.3 Test Procedure

Step 1	Description
Delegate to CPP	
	Expected Result

4.3.38 LVV-T92 - Verify implementation of Processing of Data From Special Programs

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Melissa Graham

Open LVV-T92 in Jira

4.3.38.1 Verification Elements

None.

4.3.38.2 Test Items

For a simulated night of observing that includes some special program observations, show that the SP observations are reduced using their designated reconfigured pipelines (i.e., that the image metadata is sufficient to trigger the processing and include all other relevant images in the processing).

4.3.38.3 Test Procedure

Step 1	Description

(1) Special Programs data that can be processed by the Prompt pipeline (i.e., standard visits).
Check that all images with the header keyword for SP were processed by the Prompt pipeline. Check that the Prompt pipeline's data products – DIASource, DIAObject catalogs and the Alerts – contain items flagged with their origin as that SP.

	Expected Result

Step 2	Description

(2) Special Programs data that requires 'real-time' (~24) processing with a reconfigured pipeline (e.g., DDF imaging sequence)
Check that all images with the header keywords for a given SP were processed by their reconfigured pipeline. Check that the pipeline's data products have been updated, and passed their QA.

	Expected Result

Step 3	Description

(3) Special Programs data that can (should) be processed by the Data Release pipeline (e.g., North Ecliptic Spur standard visits).
SP data would be added manually to the DRP processing. Check that the DRP's data products – Source, Object, CoAdds – contain items flagged as originating in that SP.

	Expected Result

4.3.39 LVV-T93 - Verify implementation of Level 1 Processing of Special Programs Data

Version	Status	Priority	Verification Type	Owner

1	Draft	Normal	Test	Melissa Graham

Open LVV-T93 in Jira

4.3.39.1 Verification Elements

None.

4.3.39.2 Test Items

Execute multi-day operations rehearsal. Observe whether Prompt Processing data products generated in time and confirm whether processing has completed before the start of the next simulated night.

4.3.39.3 Test Procedure

Step 1	Description

If imaging data for a Special Program that requires processing with the Prompt pipeline was obtained the previous night, check that there exist DIASources/Objects/Alerts with flags that they originated from the Special Program.

Expected Result

Step 2	Description

If imaging data for a Special Program that requires prompt processing with a reconfigured pipeline was obtained the previous night, check that the relevant data products have been updated.

Expected Result

4.3.40 LVV-T94 - Verify implementation of Special Programs Database

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Melissa Graham

Open LVV-T94 in Jira

4.3.40.1 Verification Elements

None.

4.3.40.2 Test Items

To confirm that data products from Special Programs are based solely on images obtained as part of SP via, e.g., metadata queries. To confirm that the SP data products can be joined to Prompt and DRP products by attempting to do so via, e.g., coordinate table joins, and attempting to e.g., find the faint counterparts in a Deep Drilling stack to variables with no Object detections in the DRP coadds.

4.3.40.3 Test Procedure

Step 1	Description

SP data product: DDF DIAObjects catalog
Non-SP data product: WFD DIAObjects catalog
Test: join the two catalogs by coordinate (e.g., to get a longer time baseline for variable stars in the DDF)

	Expected Result

Step 2	Description

SP data product: DDF Objects catalog
Non-SP data product: WFD DIAObjects catalog
Test: join the two catalogs by coordinate to identify faint host galaxies of transients found in WFD

	Expected Result

4.3.41 LVV-T95 - Verify implementation of Constraints on Level 1 Special Program Products Generation

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Melissa Graham

Open LVV-T95 in Jira

4.3.41.1 Verification Elements

None.

4.3.41.2 Test Items

Execute single-day operations rehearsal. Observe Prompt Processing data products generated in time. Confirm that data from Special Programs is processed with the same latency as required for main survey data: release of public data within L1publicT and Alerts within OTT1.

4.3.41.3 Test Procedure

Step 1-1 from LVV-T866	Description

Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.

Expected Result

An output dataset including difference images and DIASource and DIAObject measurements.

Step 1-2 from LVV-T866	Description

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

Expected Result

Step 2	Description

Confirm that Special Program prompt data products have been generated within 24 hours.

Expected Result

4.3.42 LVV-T96 - Verify implementation of Query Repeatability

Version	Status	Priority	Verification Type	Owner

1	Draft	Normal	Test	Colin Slater

Open LVV-T96 in Jira

4.3.42.1 Verification Elements

None.

4.3.42.2 Test Items

Verify that prior queries can be rerun with identical results, or with new additional data for live (Alert Production) databases.

4.3.42.3 Test Procedure

Step 1	Description
Select and download (deterministic) random subsample of records from Data Release Object and Source tables.	
	Expected Result

Step 2	Description
Select and download random subsample of PPDB DIAObject and DIASource tables.	
	Expected Result

Step 3	Description
As appropriate, wait for some amount of non-trivial database usage to occur, such as Prompt Processing ingestion or ingestion of other DRP database tables.	
	Expected Result

Step 4	Description
Re-run the queries in steps 1 and 2 and verify that the resulting data are identical.	

Expected Result

4.3.43 LVV-T99 - Verify implementation of Processing of Datasets

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Kian-Tat Lim

Open LVV-T99 in Jira

4.3.43.1 Verification Elements

None.

4.3.43.2 Test Items

Execute AP and DRP, simulate failures, observe correct processing

4.3.43.3 Test Procedure

Step 1	Description

Execute AP and DRP

Expected Result

Step 2	Description

Simulate failures

Expected Result

Step 3	Description

Observe correct processing

Expected Result

4.3.44 LVV-T100 - Verify implementation of Transparent Data Access

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Kian-Tat Lim

Open LVV-T100 in Jira

4.3.44.1 Verification Elements

None.

4.3.44.2 Test Items

Test Items

Observe dataset retrieval from multiple LSP instances

4.3.44.3 Test Procedure

Step 1	Description

Observe dataset retrieval from multiple LSP instances

Expected Result

4.3.45 LVV-T101 - Verify implementation of Transient Alert Distribution

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Kian-Tat Lim

Open LVV-T101 in Jira

4.3.45.1 Verification Elements

None.

4.3.45.2 Test Items

Precursor or simulated data, execute AP, observe distribution to simulated clients using standard protocols

4.3.45.3 Test Procedure

Step 1	Description
Execute AP	
	Expected Result

Step 2	Description
Observe distribution to simulated clients using standard protocols	
	Expected Result

4.3.46 LVV-T102 - Verify implementation of Solar System Objects Available Within Specified Time

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Kian-Tat Lim

Open LVV-T102 in Jira

4.3.46.1 Verification Elements

None.

4.3.46.2 Test Items

Execute single-day operations rehearsal, observe that data products for Solar System Objects are generated in time

4.3.46.3 Test Procedure

Step 1	Description
Execute single-day operations rehearsal	
	Expected Result

Step 2	Description
Observe data products generated in time	
	Expected Result

4.3.47 LVV-T104 - Verify implementation of Generate DMS Performance Report Within Specified Time

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Kian-Tat Lim

Open LVV-T104 in Jira

4.3.47.1 Verification Elements

None.

4.3.47.2 Test Items

Verify that the DMS can generate a nightly Perfomance Report within perfReportComplTime

4.3.47.3 Test Procedure

Step 1	Description
Execute single-day operations rehearsal	
	Expected Result

Step 2	Description
Observe performance report is generated on time and with correct contents	
	Expected Result

4.3.48 LVV-T105 - Verify implementation of Generate Calibration Report Within Specified Time

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Kian-Tat Lim

Open LVV-T105 in Jira

4.3.48.1 Verification Elements

None.

4.3.48.2 Test Items

Verify that the DMS can generate a night Calibration Report in both human-readable and machine-parseable forms.

4.3.48.3 Test Procedure

Step 1	Description
Execute single-day operations rehearsal	
	Expected Result

Step 2	Description
Observe calibration report is generated on time and with correct contents	
	Expected Result

4.3.49 LVV-T106 - Verify implementation of Calibration Images Available Within Specified Time

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Kian-Tat Lim

Open LVV-T106 in Jira

4.3.49.1 Verification Elements

None.

4.3.49.2 Test Items

Execute single-day operations rehearsal, observe data products generated

4.3.49.3 Test Procedure

Step 1	Description

Identify a dataset of raw calibration exposures containing at least **nCalExpProc = 25** exposures. (If it contains more than 25 exposures, use only 25 for the test.)

Expected Result

Step 2-1 from LVV-T1059	Description

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

Expected Result

Step 2-2 from LVV-T1059	Description

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

Expected Result

Step 3	Description

Confirm that the processing completed successfully within **calProcTime = 1200 seconds.**

Expected Result

Calibration products resulting from processed raw calibration exposures are present within calProcTime, and are well-formed images.

4.3.50 LVV-T107 - Verify implementation of Level-1 Production Completeness

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T107 in Jira

4.3.50.1 Verification Elements

None.

4.3.50.2 Test Items

Verify that the DMS successfully processes all images of sufficiently quality for processing are eventually processed even after connectivity failures.

4.3.50.3 Predecessors

LVV-T284

4.3.50.4 Test Procedure

Step 1	Description
Ingest raw data while simulating failures and outages, observe eventual recovery	
	Expected Result

4.3.51 LVV-T108 - Verify implementation of Level 1 Source Association

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T108 in Jira

4.3.51.1 Verification Elements

None.

4.3.51.2 Test Items

Verify that the DMS associates DIASources into a DIAObject or SSObject.

4.3.51.3 Test Procedure

Step 1	Description
Delegate to AP	
	Expected Result

4.3.52 LVV-T109 - Verify implementation of SSObject Precovery

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T109 in Jira

4.3.52.1 Verification Elements

None.

4.3.52.2 Test Items

Verify that the DMS associates additional DIAObjects (both forward and back in time) with objects classified as SSObjects.

4.3.52.3 Test Procedure

Step 1	Description
Delegate to AP	
	Expected Result

4.3.53 LVV-T110 - Verify implementation of DIASource Precovery

Version	Status	Priority	Verification Type	Owner

1	Draft	Normal	Test	Eric Bellm

Open LVV-T110 in Jira

4.3.53.1 Verification Elements

None.

4.3.53.2 Test Items

Verify that DMS performs forced photometry for new DIAObjects at all available images within the precoveryWindow.

4.3.53.3 Test Procedure

Step 1	Description
Execute single-day operations rehearsal, observe data products generated in time	
Expected Result	

4.3.54 LVV-T111 - Verify implementation of Use of External Orbit Catalogs

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T111 in Jira

4.3.54.1 Verification Elements

None.

4.3.54.2 Test Items

Verify that the DMS can make use of external catalogs to improve identification of SSObjects.

4.3.54.3 Test Procedure

Step 1	Description
Delegate to AP	
	Expected Result

4.3.55 LVV-T116 - Verify implementation of Associating Objects across data releases

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Kian-Tat Lim

Open LVV-T116 in Jira

4.3.55.1 Verification Elements

None.

4.3.55.2 Test Items

Load DR, observe queryable association

4.3.55.3 Test Procedure

Step 1	Description
Load DR	

	Expected Result

Step 2	Description

Observe queryable association

	Expected Result

4.3.56 LVV-T117 - Verify implementation of DAC resource allocation for Level 3 processing

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Colin Slater

Open LVV-T117 in Jira

4.3.56.1 Verification Elements

None.

4.3.56.2 Test Items

Verify that compute time and storage space allocations can be granted to science users.

4.3.56.3 Test Procedure

Step 1	Description

Create a test user account for the Rubin Science Platform.

	Expected Result

Step 2	Description

Set the RSP resource allocations for the test user to very low values.

	Expected Result

Step 3	Description

Initiate example batch jobs and notebook sessions that will exceed the specified resource limits.

	Expected Result

Quota error.

Step 4	Description

Transfer sufficient data volumes into the user workspace and MyDB tables that would exceed the resource quotas.

	Expected Result

Quota error.

Step 5	Description

Reset the user resource quotas to normal values.

	Expected Result

Step 6	Description

Initiate the same example batch jobs and notebook sessions that previously caused an error.

	Expected Result

Successful notebook and batch job execution.

Step 7	Description

Transfer the same data volumes into the user workspace and MyDB tables that previously caused an error.

	Expected Result

Successful data transfer.

4.3.57 LVV-T118 - Verify implementation of Level 3 Data Product Self Consistency

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Colin Slater

Open LVV-T118 in Jira

4.3.57.1 Verification Elements

None.

4.3.57.2 Test Items

Verify that user-driven Level 3 processing is conducted on consistent sets of input data.

4.3.57.3 Test Procedure

Step 1	Description
Execute representative processing on DR in PDAC, observe consistency	
	Expected Result

4.3.58 LVV-T119 - Verify implementation of Provenance for Level 3 processing at DACs

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Colin Slater

Open LVV-T119 in Jira

4.3.58.1 Verification Elements

None.

4.3.58.2 Test Items

Verify that provenance information is recorded and accessible for user-generated Level 3 products.

4.3.58.3 Test Procedure

Step 1	Description
Execute representative processing on DR in PDAC, observe provenance recording	
	Expected Result

4.3.59 LVV-T120 - Verify implementation of Software framework for Level 3 catalog processing

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Colin Slater

Open LVV-T120 in Jira

4.3.59.1 Verification Elements

None.

4.3.59.2 Test Items

Verify that user-driven Level 3 processing can be consistently applied to all records in a catalog.

4.3.59.3 Test Procedure

Step 1	Description
Execute representative processing on DR in PDAC, observe recognition of and recovery from failures	
	Expected Result

4.3.60 LVV-T121 - Verify implementation of Software framework for Level 3 image processing

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Colin Slater

Open LVV-T121 in Jira

4.3.60.1 Verification Elements

None.

4.3.60.2 Test Items

Verify that user-specified Level 3 processing can be applied to the desired set of images.

4.3.60.3 Test Procedure

Step 1	Description
Execute representative processing on DR in PDAC, observe recognition of and recovery from failures	
	Expected Result

4.3.61 LVV-T122 - Verify implementation of Level 3 Data Import

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Colin Slater

Open LVV-T122 in Jira

4.3.61.1 Verification Elements

None.

4.3.61.2 Test Items

Verify that the Science Platform can ingest data from community-standard file formats.

4.3.61.3 Test Procedure

Step 1	Description
Use the Science Platform catalog upload tool to ingest a small example FITS table.	
	Expected Result

Step 2	Description
Use the Science Platform catalog upload tool to ingest a small example CSV table.	
	Expected Result

Step 3	Description
Use the Science Platform catalog upload tool to ingest a large FITS table that needs to be spatially-sharded in the database.	
	Expected Result

Step 4	Description
Perform example queries on each of the three tables to verify that all data is present.	
	Expected Result
Data returned in the queries is identical to the data uploaded.	

4.3.62 LVV-T123 - Verify implementation of Access Controls of Level 3 Data Products

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T123 in Jira

4.3.62.1 Verification Elements

None.

4.3.62.2 Test Items

This test touches upon the interface between the following areas: IT Security, Identity Management, LSP Portal, and Parallel Distributed Database. The purpose is to show that access to user generated data products (previously Level 3) can have a variety of access restrictions varying from single-user, a list, a named group, or open access.

4.3.62.3 Test Procedure

Step 1	Description
Configure representative access controls in PDAC, observe proper restrictions	
	Expected Result

4.3.63 LVV-T128 - Verify implementation Provide Astrometric Model

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Colin Slater

Open LVV-T128 in Jira

4.3.63.1 Verification Elements

None.

4.3.63.2 Test Items

Verify that an astrometric model is available for Objects and DIAObjects.

4.3.63.3 Test Procedure

Step 1	Description
Delegate to AP and DRP	
	Expected Result

4.3.64 LVV-T130 - Verify implementation of Enable a Range of Shape Measurement Approaches

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Colin Slater

Open LVV-T130 in Jira

4.3.64.1 Verification Elements

None.

4.3.64.2 Test Items

Verify that multiple shape measurement algorithms can be used.

4.3.64.3 Test Procedure

Step 1	Description
Delegate to AP and DRP	

Expected Result

4.3.65 LVV-T134 - Verify implementation of Provide Image Access Services

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Inspection	Gregory Dubois-Felsmann

Open LVV-T134 in Jira

4.3.65.1 Verification Elements

None.

4.3.65.2 Test Items

Verify that images can be identified and that images and image cut-outs can be retrieved using the network interfaces - primarily IVOA standards-based - and Python APIs provided for image access by science users.

4.3.65.3 Test Procedure

Step 1	Description

Inspect that the following test cases have been executed and passed: LVV-T803, LVV-T810, LVV-T811, LVV-T812.

The requirement is fully satisfied by lower-level LSP test cases.

Expected Result

Test cases LVV-T803, LVV-T810, LVV-T811, LVV-T812 passed without blocking issues.

4.3.66 LVV-T138 - Verify implementation of Bulk Download Service

Version	Status	Priority	Verification Type	Owner

1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T138 in Jira

4.3.66.1 Verification Elements

None.

4.3.66.2 Test Items

Bulk Download

4.3.66.3 Test Procedure

Step 1	Description
Setup large transfer request and examine the data transfer rates achieved.	
	Expected Result

Step 2	Description
Test should be repeated while observing in firehose mode (with LSSTCam) during science verification to ensure that bulk transfer does not compromise normal nightly operations.	
	Expected Result

4.3.67 LVV-T142 - Verify implementation of Production Fault Tolerance

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T142 in Jira

4.3.67.1 Verification Elements

None.

4.3.67.2 Test Items

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

4.3.67.3 Test Procedure

Step 1	Description
Execute AP and DRP, simulate failures, observe correct processing	
	Expected Result

4.3.68 LVV-T147 - Verify implementation of Control of Level-1 Production

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T147 in Jira

4.3.68.1 Verification Elements

None.

4.3.68.2 Test Items

Demonstrate that the DMS can control all Prompt Processing across DMS facilities.

4.3.68.3 Test Procedure

Step 1	Description
Observe existence and capability of Prompt DMCS	
	Expected Result

4.3.69 LVV-T148 - Verify implementation of Unique Processing Coverage

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Colin Slater

Open LVV-T148 in Jira

4.3.69.1 Verification Elements

None.

4.3.69.2 Test Items

Verify that a user-specified criterion can be used to process each record in a table exactly once.

4.3.69.3 Test Procedure

Step 1	Description
Execute representative processing, observe lack of duplicates or missing rows even in the presence of failures	

	Expected Result

4.3.70 LVV-T152 - Verify implementation of Keep Historical Alert Archive

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T152 in Jira

4.3.70.1 Verification Elements

None.

4.3.70.2 Test Items

Verify that the DMS preserves and makes accessible an Alert Archive for reference and for false alert analyses

4.3.70.3 Test Procedure

Step 1	Description
Simulated alert stream, load Alert DB, observe access to Alert DB	
	Expected Result

4.3.71 LVV-T154 - Verify implementation of Raw Data Archiving Reliability

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Colin Slater

Open LVV-T154 in Jira

4.3.71.1 Verification Elements

None.

4.3.71.2 Test Items

Verify that raw images are reliably archived.

4.3.71.3 Test Procedure

Step 1	Description
Analyze sources of loss or corruption after mitigation to compute estimated reliability	
	Expected Result

4.3.72 LVV-T155 - Verify implementation of Un-Archived Data Product Cache

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T155 in Jira

4.3.72.1 Verification Elements

None.

4.3.72.2 Test Items

Demonstrate that the DMS provides low-latency storage for at least I1CacheLifetime (30 days) to keep prompt processing pre-covery images on hand.

4.3.72.3 Test Procedure

Step 1	Description
Delegate to DBB	
	Expected Result

4.3.73 LVV-T156 - Verify implementation of Regenerate Un-archived Data Products

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Simon Krughoff

Open LVV-T156 in Jira

4.3.73.1 Verification Elements

None.

4.3.73.2 Test Items

Not all of the ancillary data products produced by a data release will be archived permanently. These ancillary products have been promised as accessible to the community. Show that these products can be produced from an archived data release after the fact.

4.3.73.3 Test Procedure

Step 1	Description
Run a small DRP processing job and download unarchived data products.	
	Expected Result

Step 2	Description

Wait for (or force) a processing stack change so that the subsequent re-processing will be forced to use an older software build.

Expected Result

Step 3	Description

Using provenance information from the products in Step 1, request a re-processing and compare results with previously unarchived products.

Expected Result

4.3.74 LVV-T157 - Verify implementation Level 1 Data Product Access

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Colin Slater

Open LVV-T157 in Jira

4.3.74.1 Verification Elements

None.

4.3.74.2 Test Items

Verify that Level 1 Data Products are accessible by science users.

4.3.74.3 Test Procedure

Step 1	Description

Delegate to LSP

Expected Result

4.3.75 LVV-T158 - Verify implementation Level 1 and 2 Catalog Access

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Colin Slater

Open LVV-T158 in Jira

4.3.75.1 Verification Elements

None.

4.3.75.2 Test Items

Verify that Data Release Products are accessible by science users.

4.3.75.3 Test Procedure

Step 1	Description
Delegate to LSP	
	Expected Result

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

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Simon Krughoff

Open LVV-T159 in Jira

4.3.76.1 Verification Elements

None.

4.3.76.2 Test Items

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

4.3.76.3 Test Procedure

Step 1	Description
Delegate to LSP	
	Expected Result

4.3.77 LVV-T160 - Verify implementation of Providing a Precovery Service

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Gregory Dubois-Felsmann

Open LVV-T160 in Jira

4.3.77.1 Verification Elements

None.

4.3.77.2 Test Items

Verify that a technical capability to perform user-directed precovery analyses on difference images exists and that it is exposed through the LSST Science Platform. Verified by testing against precursor datasets.

(Involves: LSP Portal, MOPS and Forced Photometry)

4.3.77.3 Test Procedure

Step 1	Description

Run Precovery within follow-on Alert Production (i.e. daily post-processing on 30 day store).

	Expected Result

Step 2	Description

Within Science Platform, initiate request to perform precovery for a list of sources over same period (and longer). Include among the sources for precovery quasars from LVV-T80.

	Expected Result

Step 3	Description

Examine the results. Compare the results for the period where there is overlap with precovery run... and quasar photometry with those from LVV-T80 to verify user service performs as production services.

	Expected Result

4.3.78 LVV-T161 - Verify implementation of Logging of catalog queries

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T161 in Jira

4.3.78.1 Verification Elements

None.

4.3.78.2 Test Items

Demonstrate logging of queries of LSST databases. Logged queries are globally available to DB administrators but otherwise private excepting the user that made the query.

4.3.78.3 Test Procedure

Step 1	Description
Delegate to LSP	
	Expected Result

4.3.79 LVV-T162 - Verify implementation of Access to Previous Data Releases

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Gregory Dubois-Felsmann

Open LVV-T162 in Jira

4.3.79.1 Verification Elements

None.

4.3.79.2 Test Items

Verify this high-level requirement, which states that the other data access requirements, for images and catalogs, all must be satisfied for multiple data releases. Verified by inspection, i.e., by determining that the data access system components, from middleware through APIs to user interfaces, are designed to support data from multiple releases, as well as by direct testing using a synthetic test environment containing multiple releases.
(Involves: Data Backbone, Managed Database, LSP Portal, LSP JupyterLab, LSP Web APIs, Parallel Distributed Database)

4.3.79.3 Test Procedure

Step 1	Description
From Science Platform initiate request for image and catalog products from one of the two release sets.	
	Expected Result

Step 2	Description
From Science Platform re-issue the same request but specifying the alternate/earlier release set.	
	Expected Result

Step 3	Description
Compare results and identify differences that are germaine to the relevant Data Release Sets are found.	
	Expected Result

4.3.80 LVV-T163 - Verify implementation of Data Access Services

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T163 in Jira

4.3.80.1 Verification Elements

None.

4.3.80.2 Test Items

Demonstrate that Data Access Services are capable of scaling to serve data from nDRTot (11) data releases over a surveyYears (10) year survey.

4.3.80.3 Test Procedure

Step 1	Description
Delegate to LSP	
	Expected Result

4.3.81 LVV-T164 - Verify implementation of Operations Subsets

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T164 in Jira

4.3.81.1 Verification Elements

None.

4.3.81.2 Test Items

Demonstrate that Data Access Services are designed such that subsets of a Data Release may be retained and served (made available) after a Data Release has been superseded. (Data Backbone, Managed Database, LSP Portal, LSP JupyterLab, LSP Web APIs, Parallel Distributed Database)

4.3.81.3 Test Procedure

Step 1	Description
Delegate to LSP	
	Expected Result

4.3.82 LVV-T165 - Verify implementation of Subsets Support

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Lupton

Open LVV-T165 in Jira

4.3.82.1 Verification Elements

None.

4.3.82.2 Test Items

Verify that the DMS can provide designated subsets of previous Data Releases.

4.3.82.3 Test Procedure

Step 1	Description
Delegate to LSP	
	Expected Result

4.3.83 LVV-T166 - Verify implementation of Access Services Performance

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T166 in Jira

4.3.83.1 Verification Elements

None.

4.3.83.2 Test Items

Demonstrate monitoring of Data Access Services that give real and long-time views of system performance and usage.

4.3.83.3 Test Procedure

Step 1	Description
Delegate to LSP	
	Expected Result

4.3.84 LVV-T167 - Verify Capability to serve older Data Releases at Full Performance

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T167 in Jira

4.3.84.1 Verification Elements

None.

4.3.84.2 Test Items

Verify that implementation of the data access services do not preclude serving all older Data Releases with the same performance requirements as current Data Releases. Note that it is an operational consideration whether sufficient compute and storage resources would actually be provisioned to meet those requirements.

4.3.84.3 Test Procedure

Step 1	Description
Delegate to LSP	
	Expected Result

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

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T168 in Jira

4.3.85.1 Verification Elements

None.

4.3.85.2 Test Items

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

4.3.85.3 Test Procedure

Step 1	Description
Delegate to LSP	
	Expected Result

4.3.86 LVV-T169 - Verify implementation of Older Release Behavior

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Gregory Dubois-Felsmann

Open LVV-T169 in Jira

4.3.86.1 Verification Elements

None.

4.3.86.2 Test Items

Verify that the components of the data access system are technically capable of handling data releases beyond the two for which full services are required. DMS-REQ-0364 requires that up to 11 be supported. Verified by inspection, i.e., by determination that the system design and implementation contain the necessary features to support this number of releases, and by direct test in a synthetic test environment with multiple releases.
(Involves: Data Backbone, Managed Database, LSP Portal, LSP JupyterLab, LSP Web APIs, Parallel Distributed Database)

4.3.86.3 Test Procedure

Step 1	Description
Delegate to LSP	
	Expected Result

4.3.87 LVV-T170 - Verify implementation of Query Availability

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Colin Slater

Open LVV-T170 in Jira

4.3.87.1 Verification Elements

None.

4.3.87.2 Test Items

Verify that queries continue to be successfully executable over time.

4.3.87.3 Test Procedure

Step 1	Description
Delegate to LSP	
	Expected Result

4.3.88 LVV-T171 - Verify implementation of Pipeline Availability

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T171 in Jira

4.3.88.1 Verification Elements

None.

4.3.88.2 Test Items

Demonstrate that Data Management System pipelines are available for use without disruptions of greater than productionMaxDowntime (24 hours). This requires a regimented change control process and testing infrastructure for all pipelines and their underlying software services, and regimented management and monitoring of compute and networking resources. The list of services covered by this test include: Image and EFD Archiving, Prompt Processing, OCS Driven Batch, Telemetry Gateway, Alert Distribution, Alert Filtering, Batch Production, Data Backbone, Compute/Storage/LAN, Inter-Site Networks, and Service Management and Monitoring.

4.3.88.3 Test Procedure

Step 1	Description
Analyze sources of downtime after mitigation to compute estimated reliability; observe unscheduled downtime of developer, integration, and pre-production systems	
	Expected Result

Step 2	Description
Check that disruptions do not exceed the productionMaxDowntime (24 hours)	
	Expected Result

4.3.89 LVV-T172 - Verify implementation of Optimization of Cost, Reliability and Availability

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T172 in Jira

4.3.89.1 Verification Elements

None.

4.3.89.2 Test Items

In matters of cost, system reliability (functioning properly at a given time) has precedence over system availability (ability to use the system at a given time). The optimization may be outside the realm of direct testing as it is more of a system provisioning guideline but on its face it demands that the Data Management System include failure reporting, regimented change control, acceptance testing, maintenance and monitoring.

4.3.89.3 Test Procedure

Step 1	Description
Analyze resource management policy	
	Expected Result

4.3.90 LVV-T173 - Verify implementation of Pipeline Throughput

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T173 in Jira

4.3.90.1 Verification Elements

None.

4.3.90.2 Test Items

Demonstrate that the Alert Production Pipeline is capable of processing nRawExpNightMax (2800) science exposures within a (24-nightDurationMax) 12 hour period and issue alerts in offline batch mode.

4.3.90.3 Test Procedure

Step 1	Description

Execute single-day operations rehearsal, observe data products generated in time

	Expected Result

4.3.91 LVV-T174 - Verify implementation of Re-processing Capacity

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T174 in Jira

4.3.91.1 Verification Elements

None.

4.3.91.2 Test Items

Verify that the DMS has sufficient processing, storage, and network to reprocess all data within "drProcessingPeriod" (1 year) while maintaining full Prompt Processing capability.

4.3.91.3 Test Procedure

Step 1	Description

Analyze sizing model; execute DRP, observe scaling

	Expected Result

4.3.92 LVV-T175 - Verify implementation of Temporary Storage for Communications Links

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T175 in Jira

4.3.92.1 Verification Elements

None.

4.3.92.2 Test Items

Demonstrate that storage capacity is present and usable to prevent data loss if networking is interrupted between summit and base, base and archive, or archive and DAC. The requirement is to have storage necessary to hold tempStorageRelMTTR (200%) of the expected raw data that would arrive during the Mean Time to Repair (summToBaseNetMTTR = 24 hours, baseToArchNetMTTR = 48 hours, archToDacNetMTTR = 48 hours). This scale is further set by nCalibExpDay + nRawExpNightMax = 450 + 2800 = 3250 exposures/day.

4.3.92.3 Test Procedure

Step 1	Description
Analyze sizing model and network/storage design	
	Expected Result

4.3.93 LVV-T176 - Verify implementation of Infrastructure Sizing for "catching up"

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T176 in Jira

4.3.93.1 Verification Elements

None.

4.3.93.2 Test Items

Demonstrate Data Management System has sufficient excess capacity (compute infrastructure) to process one night's data (2800 exposures) within 24 hours while also maintaining nightly Alert Production (note this is very similar to LVV-T173).

4.3.93.3 Test Procedure

Step 1	Description
Execute single-day operations rehearsal including catch-up after failure, observe data products generated in time	
	Expected Result

4.3.94 LVV-T177 - Verify implementation of Incorporate Fault-Tolerance

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T177 in Jira

4.3.94.1 Verification Elements

None.

4.3.94.2 Test Items

Demonstrate that Data Management Systems have features that prevent data loss. Includes: MD5SUM/checksum verification for data transfer; RAID to eliminate single-point disk failures; multi-site and tape for disaster recovery of raw data; multiple site (and tape?) for backup/recovery of Data Release products; DB transaction logging and backup to maintain DB integrity. (Note: storage to prevent loss in case of networking failures is covered in LVV-T175).

4.3.94.3 Test Procedure

Step 1	Description
Analyze design; execute single-day operations rehearsal including failures, observe recovery without loss of data	
	Expected Result

4.3.95 LVV-T178 - Verify implementation of Incorporate Autonomics

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T178 in Jira

4.3.95.1 Verification Elements

None.

4.3.95.2 Test Items

Demonstrate that production systems monitor and report faults. Where possible fault mitigation can include re-start, re-submission, or return of partial products for triage.

4.3.95.3 Test Procedure

Step 1	Description

Analyze design; execute single-day operations rehearsal including failures, observe automated recovery and continuation of processing

Expected Result

4.3.96 LVV-T179 - Verify implementation of Compute Platform Heterogeneity

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T179 in Jira

4.3.96.1 Verification Elements

None.

4.3.96.2 Test Items

Demonstrate that production results are the same (within machine accuracy) when production occurs on different platforms (OS, kernel, hardware provisioning).

4.3.96.3 Test Procedure

Step 1	Description

Configure heterogeneous cluster, execute AP+DRP+LSP, observe correct functioning

Expected Result

4.3.97 LVV-T180 - Verify implementation of Data Management Unscheduled Downtime

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T180 in Jira

4.3.97.1 Verification Elements

None.

4.3.97.2 Test Items

This applies only to downtime that would prevent the collection of survey data. Verification means that analysis has occurred to identify likely hardware failures that would prevent survey operations and that mitigations that minimize the downtime to less than DMDowntime (1 day/year) are in place. Known systems that fall in this category include: Image and EFD Archiving, Observatory Operations Data, Telemetry Gateway, Data Backbone, Managed Database, Inter-Site Networks, and Service Management and Monitoring.

4.3.97.3 Test Procedure

Step 1	Description
Analyze likely hardware failures with mitigations to compute estimated unplanned downtime	
	Expected Result

4.3.98 LVV-T181 - Verify Base Voice Over IP (VOIP)

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeff Kantor

Open LVV-T181 in Jira

4.3.98.1 Verification Elements

None.

4.3.98.2 Test Items

Verify as-built VOIP at the Base Facility is operational and performs as expected (i.e. sufficient number of extensions allocated properly, no frequent drop-outs, no frequent jaggies on video, etc.) on both voice calls and videoconferening.

4.3.98.3 Predecessors

PMCS DLP-465 Complete
PMCS IT-702 Complete

4.3.98.4 Environment Needs

4.3.98.4.1 Software

See pre-conditions.

4.3.98.4.2 Hardware

See pre-conditions.

4.3.98.5 Test Procedure

Step 1	Description
	Test voice calls over VOIP system from Base Facility to locations in Base and to other Rubin Observatory facilities.
	Expected Result
	As-built VOIP at the Base Facility is operational and performs as expected (i.e. sufficient number of extensions allocated properly, no frequent drop-outs, etc.).

Step 2	Description

Test video conferences over system from Base Facility to locations in Base and to other Rubin Observatory facilities.

Expected Result

Verify (a) plannned and (b) as-built VOIP at the Base Facility is operational and performs as expected (i.e. no frequent drop-outs, no frequent audio glitches, no frequent jaggies on video, etc.).

4.3.99 LVV-T182 - Verify implementation of Prefer Computing and Storage Down

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T182 in Jira

4.3.99.1 Verification Elements

None.

4.3.99.2 Test Items

Only build compute or storage facilities at the summit that are justified by operational need or to prevent loss of data during networking downtimes.

4.3.99.3 Test Procedure

Step 1	Description

Analyze design and allocation of resources at the summit, base station and data facilities.

Expected Result

Step 2	Description

Confirm that allocation of resources at summit is only what is needed and no more.

	Expected Result

4.3.100 LVV-T185 - Verify implementation of Summit to Base Network Availability

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Inspection	Jeff Kantor

Open LVV-T185 in Jira

4.3.100.1 Verification Elements

None.

4.3.100.2 Test Items

Verify the availability of Summit to Base Network by demonstrating that the mean time between failures is less than summToBaseNetMTBF (90 days) over 1 year.

4.3.100.3 Predecessors

See pre-conditions.

4.3.100.4 Environment Needs

4.3.100.4.1 Software

See pre-conditions.

4.3.100.4.2 Hardware

See pre-conditions.

4.3.100.5 Test Procedure

Step 1	Description

Monitor summit to base networking for at least 1 week

Test Data

LATISS, ComCAM, and/or Full Camera data.

Expected Result

Summit - base network is operational for 1 week and monitoring data is collected.

Step 2	Description

Extrapolate annual availability, compare with at least 6 months of historical data on the link.

Test Data

Historical and current logs

Expected Result

The mean time between failures (MTBF) is projected to be less than summToBaseNetMTBF (90 days) over 1 year.

4.3.101 LVV-T186 - Verify implementation of Summit to Base Network Reliability

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Demonstration	Jeff Kantor

Open LVV-T186 in Jira

4.3.101.1 Verification Elements

None.

4.3.101.2 Test Items

Verify the reliability of the summit to base network by demonstrating reconnection and recovery to transfer of data at or exceeding rates specified in LDM-142 following a cut in network connection, within MTTR specification. The network operator will provide MTTR data on links during commissioning and operations.

4.3.101.3 Predecessors

See pre-conditions.

4.3.101.4 Environment Needs

4.3.101.4.1 Software

See pre-conditions.

4.3.101.4.2 Hardware

See pre-conditions.

4.3.101.5 Test Procedure

Step 1	Description
Disconnect fiber cable at an endpoint location on the base side of the Summit - Base fiber.	
	Test Data

- LATISS, ComCAM, or LSSTCam data

Expected Result
Fiber is disconnected and the fault is detected by the network monitoring system.

Step 2	Description
Measure the cable with the OTDR to locate the distance from the end point. Diagnose that it is a break.	

Test Data
NA

Expected Result
OTDR shows the fiber is disconnected (break).

Step 3	Description

Elapse time to simulate the following:

- Go to the most inaccessible place which would mean carrying all the tools/splicer/generator/tent equipment some metres.
- Erect a tent to make the splice
- Start the generator
- Do a splice on some random piece of cable
- At an end point measure the cable again to ensure it is break free.
- Take down and reinstall an isolated pole (not in the actual fiber path)
- Put the cable on the pole.

Test Data
NA

Expected Result
Wall clock advances by 24 hours.

Step 4	Description
Clean fiber connections. Restore connection (e.g. reconnect cable). Cycle equipment as necessary to confirm fiber is connected.	

Test Data
NA

Expected Result
Network recovers and resumes sending data.

Step 5	Description
Measure with OTDR to ensure back to normal state.	

Test Data

NA

Expected Result

OTDR indicates normal state.

4.3.102 LVV-T187 - Verify implementation of Summit to Base Network Secondary Link

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeff Kantor

Open LVV-T187 in Jira

4.3.102.1 Verification Elements

None.

4.3.102.2 Test Items

Verify automated fail-over from primary to secondary equipment in Rubin Observatory DWDM on simulated failure of primary. Verify bandwidth sufficiency on secondary. Verify automated recovery to primary equipment on simulated restoration of primary. Repeat for failure of Rubin Observatory fiber and fail-over to AURA fiber and DWDM. Demonstrate use of secondary in “catch-up” mode.

4.3.102.3 Predecessors

See pre-conditions.

4.3.102.4 Environment Needs

4.3.102.4.1 Software

See pre-conditions.

4.3.102.4.2 Hardware

See pre-conditions.

4.3.102.5 Test Procedure

Step 1	Description
Transfer data between summit and base on primary equipment (LSST Summit - Base) over uninterrupted 1 day period.	
	Test Data
LATISS, ComCAM, or LSSTCAM data.	
	Expected Result
Normal operations.	

Step 2	Description
Simulate equipment outage by disconnecting power card from primary DWDM equipment on base side of Summit - Base Fiber.	
	Test Data
NA	
	Expected Result
Network fails over to secondary equipment in <=60s.	

Step 3	Description
Transfer data between summit and base over secondary equipment uninterrupted 1 day period while monitoring network.	
	Test Data
NA	
	Expected Result
Verify that secondary equipment is capable of transferring 1 night of raw data (nCalibExpDay + nRawExpNightMax = 450 + 2800 = 3250 exposures) within summToBaseNet2TransMax (72 hours), i.e. at or exceeding rates specified in LDM-142.	

Step 4	Description
Restore primary equipment (i.e. reconnect power card to primary equipment.)	

Test Data
NA

Expected Result
Network recovers to primary in <= 60s.

Step 5	Description
Simulate fiber outage by disconnecting fiber from primary DWDM equipment on base side of Summit - Base Fiber.	

Test Data
NA

Expected Result
Network fails over to AURA DWDM and fiber.

Step 6	Description
Transfer data between summit and base over AURA fiber and equipment uninterrupted 1 day period while monitoring network.	

Test Data
LATISS, ComCAM, or FullCAM data.

Expected Result
Verify that AURA fiber and equipment is capable of transferring 1 night of raw data (nCalibExpDay + nRawExpNightMax = 450 + 2800 = 3250 exposures) within summToBaseNet2TransMax (72 hours), i.e. at or exceeding rates specified in LDM-142.

Step 7	Description
Restore primary fiber (i.e. reconnect fiber to Rubin Observatory DWDM equipment.)	

Expected Result
Network recovers to Rubin Observatory fiber and DWDM.

Step 8	Description
Demonstrate use of secondary in “catch-up” mode.	

Test Data
DAQ data buffer full of images and associated meta-data

Expected Result
Images from DAQ buffer and associated metadata are retrievable over secondary path while current observing data is being transferred over primary path.

4.3.103 LVV-T188 - Verify implementation of Summit to Base Network Ownership and Operation

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Inspection	Jeff Kantor

Open LVV-T188 in Jira

4.3.103.1 Verification Elements

None.

4.3.103.2 Test Items

Verify Summit to Base Network Ownership and Operation by LSST and/or the operations entity by inspection of construction and operations contracts and Indefeasible Rights.

4.3.103.3 Predecessors

PMCS DMTC-7400-2140, -2240, -2330 Complete

4.3.103.4 Environment Needs

4.3.103.4.1 Software

None

4.3.103.4.2 Hardware

None

4.3.103.5 Test Procedure

Step 1	Description

Examine contracts with REUNA and telefonica for fiber ownership and maintenance terms.

Expected Result

Rubin Observatory is owner of fibers on AURA property and Summit - Base DWDM and has 15-year IRU for use of fibers on all segments. REUNA is owner of LS - SCL DWDM on AURA property and in Santiago, and is operator on all fibers and DWDM. Telefonica is contracted to maintain fibers not on AURA property.

4.3.104 LVV-T189 - Verify implementation of Base Facility Infrastructure

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T189 in Jira

4.3.104.1 Verification Elements

None.

4.3.104.2 Test Items

Verify that the (a) planned infrastructure and (b) as-built infrastructure for the Base Facility satisfies the needs for data transfer and buffering, a copy of the Archive Facility, and support for Commissioning.

4.3.104.3 Test Procedure

Step 1	Description

Analyze design and sizing model

Expected Result

4.3.105 LVV-T191 - Verify implementation of Commissioning Cluster

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T191 in Jira

4.3.105.1 Verification Elements

None.

4.3.105.2 Test Items

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

4.3.105.3 Test Procedure

Step 1	Description
Analyze design and budget	
	Expected Result

4.3.106 LVV-T192 - Verify implementation of Base Wireless LAN (WiFi)

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeff Kantor

Open LVV-T192 in Jira

4.3.106.1 Verification Elements

None.

4.3.106.2 Test Items

Verify as-built wireless network at the Base Facility supports minBaseWiFi bandwidth (1000 Mbs).

4.3.106.3 Predecessors

PMCS DLP-465 Complete.

4.3.106.4 Environment Needs

4.3.106.4.1 Software

See pre-conditions.

4.3.106.4.2 Hardware

Desktop with WiFi NIC, email reader, internet browser.

4.3.106.5 Test Procedure

Step 1	Description
Test internet web browsing and file download, email at summit and base over wireless.	
	Test Data
NA	
	Expected Result
Verify as-built wireless network at the Base Facility supports minBaseWiFi bandwidth (1000 Mbs). Verify wireless signal strength meets or exceeds typical, and average and peak bandwidths meet or exceed minBaseWiFI bandwidth.	

4.3.107 LVV-T193 - Verify implementation of Base to Archive Network

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeff Kantor

Open LVV-T193 in Jira

4.3.107.1 Verification Elements

None.

4.3.107.2 Test Items

Verify that the data acquired by a DAQ can be transferred within the required time, i.e. verify that link is capable of transferring image for prompt processing in oArchiveMaxTransferTime = 5[second], i.e. at or exceeding rates specified in LDM-142.

4.3.107.3 Predecessors

PMCS DM-Net-5 Complete

4.3.107.4 Environment Needs

4.3.107.4.1 Software

See pre-conditions.

4.3.107.4.2 Hardware

See pre-conditions.

4.3.107.5 Test Procedure

Step 1	Description

Transfer data between base and archive while monitoring the network over uninterrupted 1 day period (with repeated transfers on normal observing cadence).

Test Data

LATISS, ComCAM, or FullCAM data.

Expected Result

Data transfers occur without significant delay or frequent latency spikes.

Step 2	Description

Analyze the network logs and monitoring system to determine average and peak latency and packet loss statistics.

Expected Result

Data can be transferred within the required time, i.e. verify that link is capable of transferring image for prompt processing in oArchiveMaxTransferTime = 5[second]. Verify transfer of data at or exceeding rates specified in LDM-142 at least 98% of the time.

4.3.108 LVV-T194 - Verify implementation of Base to Archive Network Availability

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeff Kantor

Open LVV-T194 in Jira

4.3.108.1 Verification Elements

None.

4.3.108.2 Test Items

Verify the availability of the Base to Archive Network communications by demonstrating that it meets or exceeds a mean time between failures, measured over a 1-yr period of MTBF > baseToArchNetMTBF (180[day])

4.3.108.3 Predecessors

PMCS DMTC-7400-2130 Complete

4.3.108.4 Test Procedure

Step 1	Description

Transfer data between base and archive over uninterrupted 1 week period.

Test Data

LATISS, ComCAM, or FullCAM data.

Expected Result

Data is successfully transferred during the entire week.

Step 2	Description

Analyze monitoring/performance data, compare to historical data, and extrapolate to a full year, average and peak throughput and latency.

Test Data

NA

Expected Result

Extrapolated network availability meets baseToArchNetMTBF = 180[day]. Note that this is for complete loss of transfer service (all paths), not a single path failure with successful fail-over.

4.3.109 LVV-T195 - Verify implementation of Base to Archive Network Reliability

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeff Kantor

Open LVV-T195 in Jira

4.3.109.1 Verification Elements

None.

4.3.109.2 Test Items

Verify Base to Archive Network Reliability by demonstrating that the network can recover from outages within baseToArchNetMTTR = 48[hour].

4.3.109.3 Predecessors

PMCS DM-NET-5 Complete

4.3.109.4 Environment Needs

4.3.109.4.1 Software

See pre-conditions.

4.3.109.4.2 Hardware

See pre-conditions.

4.3.109.5 Test Procedure

Step 1	Description
Disconnect primary fiber on base side of Base - Archive network.	
	Test Data
LATISS, ComCAM, or FullCAM data.	
	Expected Result
Network fails over to secondary path.	

Step 2	Description
Simulate diagnosis and repair by elapsed time.	

Test Data
NA

Expected Result
Wall clock advances by 48 hours. Data is successfully transferred over secondary path.

Step 3	Description
Reconnect primary fiber on base side of Base - Archive network.	

Test Data
NA

Expected Result
Network recovers to primary path.

Step 4	Description
Analyze fail-over and recovery times. Compare to historical data and extrapolate to MTTR.	

Expected Result
Verify recovery can occur within baseToArchNetMTTR = 48[hour]. Demonstrate reconnection and recovery to transfer of data at or exceeding rates specified in LDM-142.

4.3.110 LVV-T196 - Verify implementation of Base to Archive Network Secondary Link

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeff Kantor

Open LVV-T196 in Jira

4.3.110.1 Verification Elements

None.

4.3.110.2 Test Items

Verify Base to Archive Network Secondary Link failover and capacity, and subsequent recovery primary. Demonstrate the use of the secondary path in "catch-up" mode.

4.3.110.3 Predecessors

PMCS DM-NET-5 Complete
PMCS DMTC-8000-0990 Complete
PMCS DMTC-8100-2130 Complete
PMCS DMTC-8100-2530 Complete
PMCS DMTC-8200-0600 Complete

4.3.110.4 Environment Needs

4.3.110.4.1 Software

See pre-conditions.

4.3.110.4.2 Hardware

See pre-conditions.

4.3.110.5 Test Procedure

Step 1	Description
Transfer data between base and archive on primary links over uninterrupted 1 day period.	
	Test Data
LATISS, ComCAM, or FullCAM data.	
	Expected Result
Data is successfully transferred over primary link at or exceeding rates specified in LDM-142 throughout period.	

Step 2	Description

Simulate outage by disconnecting fiber on primary fiber on Base side of Base - Archive Network.

Test Data

NA

Expected Result

Network fails over to secondary links in <=60s

Step 3	Description

Transfer data between base and archive over secondary equipment uninterrupted 1 day period.

Test Data

LATISS, ComCAM, or FullCAM data.

Expected Result

Data is successfully transferred over secondary link at or exceeding rates specified in LDM-142 throughout period.

Step 4	Description

Restore connection on primary link by reconnecting fiber.

Test Data

NA

Expected Result

Network recovers to primary.

Step 5	Description

Demonstrate use of secondary in catch-up mode.

Test Data

DAQ buffer full of images and associated metadata.

Expected Result

Images from DAQ buffer and associated metadata are retrievable over secondary path while current observing data is being transferred over primary path.

4.3.111 LVV-T197 - Verify implementation of Archive Center

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T197 in Jira

4.3.111.1 Verification Elements

None.

4.3.111.2 Test Items

Verify that the Archive Center is sufficiently provisioned to support prompt processing, DRP, and data access needs.

4.3.111.3 Test Procedure

Step 1	Description
Analyze design and sizing model	
	Expected Result

4.3.112 LVV-T198 - Verify implementation of Archive Center Disaster Recovery

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T198 in Jira

4.3.112.1 Verification Elements

None.

4.3.112.2 Test Items

Verify disaster recovery plan for Archive Center.

4.3.112.3 Test Procedure

Step 1	Description
Analyze design; simulate storage failure, observe restore from disaster recovery	
	Expected Result

4.3.113 LVV-T200 - Verify implementation of Archive to Data Access Center Network

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeff Kantor

Open LVV-T200 in Jira

4.3.113.1 Verification Elements

None.

4.3.113.2 Test Items

Verify archiving of data to Data Access Center Network at or exceeding rates specified in LDM-142, i.e at archToDacBandwidth = 10000[megabit per second].

4.3.113.3 Predecessors

PMCS DMTC-8100-2550 Complete

4.3.113.4 Environment Needs

4.3.113.4.1 Software

See pre-conditions.

4.3.113.4.2 Hardware

See pre-conditions.

4.3.113.5 Test Procedure

Step 1	Description
Transfer data from Data Facility to US and Chilean DACs over an uninterrupted 1 week period.	
	Test Data
Data Release	
	Expected Result
Data transfers without significant failures or extended latency spikes	

Step 2	Description
Analyze network logs and compare with historical data on the links.	
	Test Data
NA	
	Expected Result
The networks can transfer data at archToDacBandwidth = 10000[megabit per second], i.e. at or exceeding rates specified in LDM-142.	

4.3.114 LVV-T201 - Verify implementation of Archive to Data Access Center Network Availability

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeff Kantor

Open LVV-T201 in Jira

4.3.114.1 Verification Elements

None.

4.3.114.2 Test Items

Verify availability of archiving to Data Access Center Network using test and historical data of or exceeding archToDacNetMTBF= 180[day].

4.3.114.3 Predecessors

PMCS DMTC-8100-2550 Complete

4.3.114.4 Environment Needs

4.3.114.4.1 Software

See pre-conditions.

4.3.114.4.2 Hardware

See pre-conditions.

4.3.114.5 Test Procedure

Step 1	Description
Transfer data between archive and DACs over uninterrupted 1 week period.	
	Test Data
Data Release or petabyte-scale test data set	
	Expected Result
Data transfers without failures or extended latency spikes	

Step 2	Description
Analyze test data and compare to historical data. Extrapolate to 1 year testimate of MTBF.	
	Test Data
NA	
	Expected Result
Networks can meet archToDacNetMTBF = 180[day] at or exceeding rates specified in LDM-142.	

4.3.115 LVV-T202 - Verify implementation of Archive to Data Access Center Network Reliability

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeff Kantor

Open LVV-T202 in Jira

4.3.115.1 Verification Elements

None.

4.3.115.2 Test Items

Verify the reliability of Archive to Data Access Center Network by demonstrating successful failover and capacity to the secondary part and subsequent recovery to primary within or exceeding chToDacNetMTTR = 48[hour].

4.3.115.3 Predecessors

PMCS DMTC-8100-2550 Complete

4.3.115.4 Environment Needs

4.3.115.4.1 Software

See pre-conditions.

4.3.115.4.2 Hardware

See pre-conditions.

4.3.115.5 Test Procedure

Step 1	Description
Simulate failure on primary paths by disconnecting fiber at an endpoint location in the archive on the Archive - DACs network.	
	Test Data
NA	
	Expected Result
Networks fail over to secondary paths.	

Step 2	Description
Monitor transfers on secondary paths for 1 day.	
	Expected Result
Transfers occur without extended failures or extended latency spikes. Data transfers on secondary at rates at or above those specified in LDM-142.	

Step 3	Description
Simulate repair and recovery period by leaving primary fiber disconnected for at least 1 day, then reconnecting primary fiber.	
	Test Data
NA	
	Expected Result
Wall clock advances by 1 day. Network recovers to primary path. Verify entire process meets chToDacNetMTTR = 48[hour].	

4.3.116 LVV-T203 - Verify implementation of Archive to Data Access Center Network Secondary Link

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Kian-Tat Lim

Open LVV-T203 in Jira

4.3.116.1 Verification Elements

None.

4.3.116.2 Test Items

Verify the Archive to Data Access Center Network via Secondary Link by simulating a failure on the primary path and capacity on the secondary path.

4.3.116.3 Predecessors

PMCS DMTC-8100-2550 Complete

4.3.116.4 Environment Needs

4.3.116.4.1 Software

See pre-conditions.

4.3.116.4.2 Hardware

See pre-conditions.

4.3.116.5 Test Procedure

Step 1	Description

Transfer data between Archive and DACs on primary path over uninterrupted 1 week period.

Test Data

Data Release or other petabyte-scale test data set.

Expected Result

Data transfers without failures or extended latency spikes, at or exceeding rates specified in LDM-142 throughout fail-over period.

Step 2	Description

Simulate outage on primary path by disconnecting fiber on primary on Archive side of Archive - DACs networks.

Test Data

NA

Expected Result

Network fails over to secondary links in <= 60s.

Step 3	Description

Transfer data between base and archive over secondary equipment uninterrupted 1 day period.

Test Data

Data Release or other petabyte-scale test data set.

Expected Result

Data transfers without failures or extended latency spikes, at or exceeding rates specified in LDM-142 throughout fail-over period.

Step 4	Description

Restore connection on primary link (reconnect fiber).

Test Data

NA

Expected Result

Network recovers to primary in <= 60s.

4.3.117 LVV-T204 - Verify implementation of Access to catalogs for external Level 3 processing

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Kian-Tat Lim

Open LVV-T204 in Jira

4.3.117.1 Verification Elements

None.

4.3.117.2 Test Items

Verify that catalog export, and maintenance/validation tools for Level 3 products to outside of the Data Access Centers.

4.3.117.3 Test Procedure

Step 1	Description

Execute bulk distribution of DRP catalogs

Expected Result

Step 2	Description
Observe correct transfer and use of maintenance/validation tools	
	Expected Result

4.3.118 LVV-T205 - Verify implementation of Access to input catalogs for DAC-based Level 3 processing

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Robert Gruendl [X]

Open LVV-T205 in Jira

4.3.118.1 Verification Elements

None.

4.3.118.2 Test Items

Verify that data products are available at the Data Access Centers for use in Level 3 processing.

4.3.118.3 Test Procedure

Step 1	Description
Load Prompt and DR catalogs into PDAC, observe access via Rubin Science Platform	
	Expected Result

4.3.119 LVV-T206 - Verify implementation of Federation with external catalogs

Version	Status	Priority	Verification Type	Owner

1	Draft	Normal	Test	Colin Slater

Open LVV-T206 in Jira

4.3.119.1 Verification Elements

None.

4.3.119.2 Test Items

Verify that LSST-produced data can be combined with external datasets.

4.3.119.3 Test Procedure

Step 1	Description
Load external catalog into PDAC (using VO if possible), observe federation with other catalogs via Rubin Science Platform (RSP)	
	Expected Result

4.3.120 LVV-T207 - Verify implementation of Access to images for external Level 3 processing

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Kian-Tat Lim

Open LVV-T207 in Jira

4.3.120.1 Verification Elements

None.

4.3.120.2 Test Items

Verify that bulk distribution of images, and accompanying maintenance/validation tools for Level 3 image products to outside of the Data Access Centers.

4.3.120.3 Test Procedure

Step 1	Description
Execute bulk distribution of DRP images	
	Expected Result

Step 2	Description
Observe correct transfer and use of maintenance/validation tools	
	Expected Result

4.3.121 LVV-T208 - Verify implementation of Access to input images for DAC-based Level 3 processing

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Kian-Tat Lim

Open LVV-T208 in Jira

4.3.121.1 Verification Elements

None.

4.3.121.2 Test Items

Verify that prompt processing and DRP products are available at the DACs for Level 3 processing at the DACs.

4.3.121.3 Test Procedure

Step 1	Description
Load Prompt and DR images into PDAC	
	Expected Result

Step 2	Description
Observe access via RSP	
	Expected Result

4.3.122 LVV-T209 - Verify implementation of Data Access Centers

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Analysis	Kian-Tat Lim

Open LVV-T209 in Jira

4.3.122.1 Verification Elements

None.

4.3.122.2 Test Items

Verify that the Data Access Centers are provisioned with computing resources necessary to support end-user access to LSST Data Products.

4.3.122.3 Test Procedure

Step 1	Description
Analyze design	
	Expected Result

4.3.123 LVV-T210 - Verify implementation of Data Access Center Simultaneous Connections

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Kian-Tat Lim

Open LVV-T210 in Jira

4.3.123.1 Verification Elements

None.

4.3.123.2 Test Items

Verify that the each DAC can support at least dacMinConnections simultaneously

4.3.123.3 Test Procedure

Step 1	Description
Simulate data access to PDAC	
	Expected Result

Step 2	Description
Observe scaling	

Expected Result

4.3.124 LVV-T211 - Verify implementation of Data Access Center Geographical Distribution

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Analysis	Kian-Tat Lim

Open LVV-T211 in Jira

4.3.124.1 Verification Elements

None.

4.3.124.2 Test Items

Verify that the DACs are geographically distributed to provide low-latency access to data-rights community.

4.3.124.3 Test Procedure

Step 1	Description

Analyze design

Expected Result

4.3.125 LVV-T212 - Verify implementation of No Limit on Data Access Centers

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Colin Slater

Open LVV-T212 in Jira

4.3.125.1 Verification Elements

None.

4.3.125.2 Test Items

Verify that additional Data Access Centers can be set up.

4.3.125.3 Test Procedure

Step 1	Description
Analyze design; instantiate and load simulated DAC, observe correct functioning	
	Expected Result

4.3.126 LVV-T284 - RAS-00-05: (LDM-503-8b) Writing data from CCOB to the DBB for further data processing

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Michelle Butler [X]

Open LVV-T284 in Jira

4.3.126.1 Verification Elements

None.

4.3.126.2 Test Items

This test will check:

- The successful integration of the DAQ archiver components with the CCOB
- That the file can then be ingested into the DBB and be retrieved for further analysis

4.3.126.3 Predecessors

None.

4.3.126.4 Environment Needs

4.3.126.4.1 Software

- CCOB device and the software to produce a file to be transferred and kept
- DBB software to produce a retrieval file for further processing

4.3.126.4.2 Hardware

- CCOB
- Test machine for LSST Monitoring Service
- consolidate DB
- DBB ingest file system
- DBB output file system
- data transfer protocol to move data from CCOB file systems to DBB ingest file system

4.3.126.5 Input Specification

None.

4.3.126.6 Output Specification

- CCOB (raw image) files that follow specifications;
- DBB files that follow specifications;
- CCOB device directs a human to where a file is wanted to be stored in the DBB;
- Transfer the file to the DBB ingest area;

4.3.126.7 Test Procedure

Step 1	Description
CCOB device directs a human to where a raw file is wanted to be stored in the DBB	
	Expected Result
A file with a unique file name is in a file system somewhere, and the data is then transferred to NCSA.	

Step 2	Description
Move the data from the transferred directory into the DBB foreign file ingest file system.	
	Expected Result
A command is executed by a human with a file name and path to the file wanted to be stored in the DBB. The file is transferred to NCSA's DBB ingest area.	

Step 3	Description
Have data inspected by scientist for managing that all data was transferred.	
	Expected Result
a specific Okay to move forward; or something is broke.	

Step 4	Description
The DBB is notified of a new file being in the ingest area, and the DBB ingest is run manually to ingest the CCOB file.	
	Expected Result
The DBB puts the resulting file into the DBB file systems depending on what type of file it is. The DB is updated with metadata and providence of the file to be kept. The resulting file system is queryable by the LSP to find the CCOB raw image.	

Step 5	Description
The LSP can review and use the CCOB raw data file that was stored originally somewhere else such as slac	
	Expected Result
LSP has the ability to find the file and view/use it.	

4.3.127 LVV-T1097 - Verify Summit Facility Network Implementation

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeff Kantor

Open LVV-T1097 in Jira

4.3.127.1 Verification Elements

None.

4.3.127.2 Test Items

Verify that data acquired by a AuxTel DAQ can be transferred to Summit DWDM and loaded in the EFD without problems.

4.3.127.3 Predecessors

PMCS DMTC-7400-2400 Complete
PMCS T&SC-2600-1545 Complete

4.3.127.4 Environment Needs

4.3.127.4.1 Software

See pre-conditions

4.3.127.4.2 Hardware

See pre-conditions.

4.3.127.5 Test Procedure

Step 1	Description
Verify the pre-conditions have been satisfied	
	Test Data
NA	
	Expected Result
Pre-conditions are satisfied.	

Step 2	Description
Control the AuxTel through a night of Observing. While observing, read out LATISS data and transfer to Rubin Observatory Summit DWDM while monitoring latency.	
	Test Data
LATISS images and metadata	
	Expected Result
Data is fed to DWDM without delays or errors.	

Step 3	Description
Verify that data acquired by a AuxTel DAQ can be transferred and loaded in EFD without problems.	
	Test Data
LATISS images and metadata	
	Expected Result
Examine the EFD to ensure that the data has been loaded properly.	

4.3.128 LVV-T1250 - Verify implementation of minimum number of simultaneous DM EFD query users

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1250 in Jira

4.3.128.1 Verification Elements

None.

4.3.128.2 Test Items

Verify that the DM EFD can support **dmEfdQueryUsers = 5** simultaneous queries. The additional requirement that each query must last no more than **dmEfdQueryTime = 10 seconds** will be verified separately in LVV-T1251, but these must be satisfied together.

4.3.128.3 Test Procedure

Step 1	Description

Send multiple (at least 5) simultaneous queries to the DM EFD.

Expected Result

Step 2	Description

Confirm that (a) the queries executed successfully, and that (b) they return reasonable results.

Expected Result

Step 3	Description

Repeat the above steps for different queries, and different numbers of simultaneous queries, to confirm that the expected performance is met regardless of the query being executed.

Expected Result

4.3.129 LVV-T1251 - Verify implementation of maximum time to retrieve DM EFD query results

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1251 in Jira

4.3.129.1 Verification Elements

None.

4.3.129.2 Test Items

Verify that the DM EFD can support **dmEfdQueryUsers = 5** simultaneous queries, with each query must executing in no more than **dmEfdQueryTime = 10 seconds.** The requirement on at least 5 simultaneous queries will be verified separately in LVV-T1250, but these must be satisfied together.

4.3.129.3 Test Procedure

Step 1	Description

Send multiple (at least 5) simultaneous queries to the DM EFD.

Expected Result

Step 2	Description

Confirm that (a) the queries executed successfully, and that (b) they return reasonable results. Check that the time of execution for all queries was less than 10 seconds.

Expected Result

Step 3	Description

Repeat the above steps for different queries, and different numbers of simultaneous queries, to confirm that the expected performance is met regardless of the query being executed.

Expected Result

4.3.130 LVV-T1276 - Verify implementation of latency of reporting optical transients

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T1276 in Jira

4.3.130.1 Verification Elements

None.

4.3.130.2 Test Items

Verify that alerts are generated for optical transients within **OTT1 = 1 minute** of the completion of the readout of the last image.

4.3.130.3 Test Procedure

Step 1	Description

Identify a precursor dataset containing raw images (and templates), that is suitable for testing the Alert Production.

Expected Result

Step 2-1 from LVV-T866	Description

Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.

Expected Result
An output dataset including difference images and DIASource and DIAObject measurements.

Step 2-2 from LVV-T866	Description
Verify that the expected data products have been produced, and that catalogs contain reasonable values for measured quantities of interest.	

Expected Result

Step 3	Description
Time processing of data starting from (pre-ingested) raw files until an alert is available for distribution; verify that this time is less than OTT1.	

Expected Result
Alerts are received via the alert stream within OTT1=1 minute from the time the Alert Production payload was executed.

4.3.131 LVV-T1277 - Verify processing of maximum number of calibration exposures

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Kian-Tat Lim

Open LVV-T1277 in Jira

4.3.131.1 Verification Elements

None.

4.3.131.2 Test Items

Verify that as many as **nCalExpProc = 25** calibration exposures can be processed together within time calProcTime.

4.3.131.3 Test Procedure

Step 1	Description

Identify a dataset of raw calibration exposures containing at least **nCalExpProc = 25** exposures. (If it contains more than 25 exposures, use only 25 for the test.)

Expected Result

Step 2-1 from LVV-T1059	Description

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

Expected Result

Step 2-2 from LVV-T1059	Description

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

Expected Result

Step 3	Description

Confirm that the processing completed successfully within **calProcTime = 1200 seconds.**

Expected Result

Calibration products resulting from processed raw calibration exposures are present within calProcTime, and are well-formed images.

Step 4	Description

Perform the test again with *more than* nCalExpProc = 25 images, and confirm that the processing completes within **calProcTime = 1200 seconds.**

Expected Result

Calibration products resulting from processed raw calibration exposures are present within calProcTime, and are well-formed images. (To verify that the test with 25 images was not at the limits of what the software can handle – should be able to exceed that bare minimum.)

4.3.132 LVV-T1524 - Verify Implementation of Exporting MOCs as FITS

Version	Status	Priority	Verification Type	Owner

1	Draft	Normal	Demonstration	Jeffrey Carlin

Open LVV-T1524 in Jira

4.3.132.1 Verification Elements

None.

4.3.132.2 Test Items

Verify that the Data Management system provides a means for exporting the LSST-generated MOCs in the FITS serialization form defined in the IVOA MOC Recommendation.

4.3.132.3 Test Procedure

Step 1	Description
	Expected Result

4.3.133 LVV-T1525 - Verify Implementation of Linkage Between HiPS Maps and Coadded Images

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Demonstration	Jeffrey Carlin

Open LVV-T1525 in Jira

4.3.133.1 Verification Elements

None.

4.3.133.2 Test Items

Verify that the HiPS maps produced by the Data Management system provide for straightforward linkage from the HiPS data to the underlying LSST coadded images, and that this has been implemented using a mechanism supported by both the LSST Science Platform and by community tools.

4.3.133.3 Test Procedure

Step 1	Description
	Expected Result

4.3.134 LVV-T1526 - Verify Availability of Secure and Authenticated HiPS Service

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Demonstration	Jeffrey Carlin

Open LVV-T1526 in Jira

4.3.134.1 Verification Elements

None.

4.3.134.2 Test Items

Verify that the Data Management system includes a secure and authenticated Internet endpoint for an IVOA-compliant HiPS service. Confirm that this service is advertised via Registry as well as in the HiPS community mechanism operated by CDS, or whatever equivalent mechanism may exist in the LSST operations era.

4.3.134.3 Test Procedure

Step 1	Description
	Expected Result

4.3.135 LVV-T1527 - Verify Support for HiPS Visualization

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Demonstration	Leanne Guy

Open LVV-T1527 in Jira

4.3.135.1 Verification Elements

None.

4.3.135.2 Test Items

Verify that the LSST Science Platform supports the visualization of LSST-generated HiPS image maps as well as other HiPS maps which satisfy the IVOA HiPS Recommendation. Also verify that integrated behavior is available, such as the overplotting of catalog entries, comparable to that provided for individual source images (e.g., PVIs and coadd tiles).

4.3.135.3 Test Procedure

Step 1	Description
	Expected Result

4.3.136 LVV-T1528 - Verify Visualization of MOCs via Science Platform

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Demonstration	Jeffrey Carlin

Open LVV-T1528 in Jira

4.3.136.1 Verification Elements

None.

4.3.136.2 Test Items

Verify that the LSST Science Platform supports the visualization of the LSST-generated MOCs as well as other MOCs which satisfy the IVOA MOC Recommendation.

4.3.136.3 Test Procedure

Step 1	Description
	Expected Result

4.3.137 LVV-T1529 - Verify Production of All-Sky HiPS Map

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Demonstration	Leanne Guy

Open LVV-T1529 in Jira

4.3.137.1 Verification Elements

None.

4.3.137.2 Test Items

Verify that Data Release Production includes the production of an all-sky image map for the existing coadded image area in each filter band, and at least one pre-defined all-sky color image map, following the IVOA HiPS Recommendation.

4.3.137.3 Test Procedure

Step 1	Description
	Expected Result

4.3.138 LVV-T1530 - Verify Production of Multi-Order Coverage Maps for Survey Data

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Demonstration	Jeffrey Carlin

Open LVV-T1530 in Jira

4.3.138.1 Verification Elements

None.

4.3.138.2 Test Items

Verify that Data Release Production includes the production of Multi-Order Coverage maps for the survey data, conformant with the IVOA MOC recommendation. Confirm that separate MOC are produced for each filter band for the main survey, and additional MOCs are produced to represent special-programs datasets and other collections of on-sky data.

4.3.138.3 Test Procedure

Step 1	Description
	Expected Result

4.3.139 LVV-T1560 - Verify archiving of processing provenance

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Inspection	Jeffrey Carlin

Open LVV-T1560 in Jira

4.3.139.1 Verification Elements

None.

4.3.139.2 Test Items

Verify that provenance information related to data processing, including relevant data from other subsystems, has been archived.

4.3.139.3 Test Procedure

Step 1	Description
	Expected Result

4.3.140 LVV-T1561 - Verify provenance availability to science users

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Inspection	Jeffrey Carlin

Open LVV-T1561 in Jira

4.3.140.1 Verification Elements

None.

4.3.140.2 Test Items

Verify that archived provenance data is available to science users together with the associated science data products.

4.3.140.3 Test Procedure

Step 1	Description
	Expected Result

4.3.141 LVV-T1562 - Verify availability of re-run tools

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Demonstration	Jeffrey Carlin

Open LVV-T1562 in Jira

4.3.141.1 Verification Elements

None.

4.3.141.2 Test Items

Verify that tools are provided to use the archived provenance data to re-run a data processing operation under the same conditions (including LSST software version, its configuration parameters, and supporting data such as calibration frames) as a previous run of that operation.

4.3.141.3 Test Procedure

Step 1	Description
	Expected Result

4.3.142 LVV-T1563 - Verify re-run on different system produces the same results

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Demonstration	Jeffrey Carlin

Open LVV-T1563 in Jira

4.3.142.1 Verification Elements

None.

4.3.142.2 Test Items

Verify that tools are provided to use the archived provenance data to re-run a data processing operation on different systems, and that the results produced are the same to the extent computationally feasible.

4.3.142.3 Test Procedure

Step 1	Description
	Expected Result

4.3.143 LVV-T1564 - Verify re-run on similar system produces the same results

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Demonstration	Jeffrey Carlin

Open LVV-T1564 in Jira

4.3.143.1 Verification Elements

None.

4.3.143.2 Test Items

Verify that a provenance-based re-run that is run on the same system, or a system with identically configured hardware and system software, produces the same results.

4.3.143.3 Test Procedure

Step 1	Description
	Expected Result

4.3.144 LVV-T1612 - Verify Summit - Base Network Integration (System Level)

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Inspection	Jeff Kantor

Open LVV-T1612 in Jira

4.3.144.1 Verification Elements

None.

4.3.144.2 Test Items

Verify ISO Layer 3 full (22 x 10 Gbps ethernet ports on DAQ side with test data from DAQ test stand, AURA, Camera DAQ team do test). Demonstrate transfer of data at or exceeding rates specified in LDM-142.

4.3.144.3 Predecessors

See pre-conditions.

4.3.144.4 Environment Needs

4.3.144.4.1 Software

See pre-conditions.

4.3.144.4.2 Hardware

See pre-conditions.

4.3.144.5 Test Procedure

Step 1	Description

Verify Pre-conditions are satisfied.

Test Data
NA

Expected Result
Pre-conditions are satisfied.

Step 2	Description
Transfer data between summit and base over uninterrupted 1 day period. Monitor transfer of data at or exceeding rates specified in LDM-142.	

Test Data
DAQ pre-loaded data

Expected Result
Data transfers at or exceeding rates specified in LDM-142.

4.3.145 LVV-T1831 - Verify Implementation of Data Management Nightly Reporting

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Demonstration	Jeffrey Carlin

Open LVV-T1831 in Jira

4.3.145.1 Verification Elements

None.

4.3.145.2 Test Items

Verify that the LSST Data Management subsystem produces a searchable - interactive nightly report(s), from information published in the EFD by each subsystem, summarizing performance and behavior over a user defined period of time (e.g. the previous 24 hours).

4.3.145.3 Test Procedure

Step 1	Description
	Expected Result

4.3.146 LVV-T1836 - Verify calculation of resolved-to-unresolved flux ratio errors

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1836 in Jira

4.3.146.1 Verification Elements

None.

4.3.146.2 Test Items

Verify that the DM system has provided code to assess whether the maximum RMS of the ratio of the error in integrated flux measurement between bright, isolated, resolved sources less than 10 arcsec in diameter and bright, isolated unresolved point sources is less than **ResSource = 2**.

4.3.146.3 Test Procedure

Step 1	Description
	Expected Result

4.3.147 LVV-T1837 - Verify calculation of band-to-band color zero-point accuracy

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1837 in Jira

4.3.147.1 Verification Elements

None.

4.3.147.2 Test Items

Verify that the DM system provides code to assess whether the accuracy of absolute band-to-band color zero-points for all colors constructed from any filter pair, excluding the u-band, is less than **PA5 = 5 millimagnitudes**.

4.3.147.3 Test Procedure

Step 1	Description
	Expected Result

4.3.148 LVV-T1838 - Verify calculation of image fraction affected by ghosts

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1838 in Jira

4.3.148.1 Verification Elements

None.

4.3.148.2 Test Items

Verify that the DM system provides code to assess whether the percentage of image area that has ghosts with surface brightness gradient amplitude of more than 1/3 of the sky noise over 1 arcsec is less than **GhostAF = 1 percent**.

4.3.148.3 Test Procedure

Step 1	Description
	Expected Result

4.3.149 LVV-T1839 - Verify calculation of RMS width of photometric zeropoint

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1839 in Jira

4.3.149.1 Verification Elements

None.

4.3.149.2 Test Items

Verify that the DM system provides code to assess whether the RMS width of the internal photometric zero-point (precision of system uniformity across the sky) for all bands except u-band is less than **PA3 = 10 millimagnitudes**.

4.3.149.3 Test Procedure

Step 1	Description
	Expected Result

4.3.150 LVV-T1840 - Verify calculation of sky brightness precision

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1840 in Jira

4.3.150.1 Verification Elements

None.

4.3.150.2 Test Items

Verify that the DM system provides software to assess whether the maximum error in the precision of the sky brightness determination is less than **SBPrec = 1 percent.**

4.3.150.3 Test Procedure

Step 1	Description
	Expected Result

4.3.151 LVV-T1841 - Verify calculation of scientifically unusable pixel fraction

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1841 in Jira

4.3.151.1 Verification Elements

None.

4.3.151.2 Test Items

Verify that the DM system provides software to assess whether the maximum fraction of pixels scientifically unusable per sensor out of the total allowable fraction of sensors meeting this performance is less than **PixFrac = 1 percent**.

4.3.151.3 Test Procedure

Step 1	Description
	Expected Result

4.3.152 LVV-T1842 - Verify calculation of zeropoint error fraction exceeding the outlier limit

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1842 in Jira

4.3.152.1 Verification Elements

None.

4.3.152.2 Test Items

Verify that the DM system provides software to calculate the fraction of zeropoint errors that exceed the zero point error outlier limit, and confirm that it is less than **PF2 = 10 percent.**

4.3.152.3 Test Procedure

Step 1	Description
	Expected Result

4.3.153 LVV-T1843 - Verify calculation of significance of imperfect crosstalk corrections

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1843 in Jira

4.3.153.1 Verification Elements

None.

4.3.153.2 Test Items

Verify that the DM system provides software to assess whether the maximum local significance integrated over the PSF of imperfect crosstalk corrections is less than **Xtalk = 3 sigma**.

4.3.153.3 Test Procedure

Step 1	Description

	Expected Result

4.3.154 LVV-T1844 - Verify calculation of u-band photometric zero-point RMS

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1844 in Jira

4.3.154.1 Verification Elements

None.

4.3.154.2 Test Items

Verify that the DM system provides software to assess whether the RMS width of internal photometric zero-point (precision of system uniformity across the sky) in the u-band is less than **PA3u = 20 millimagnitudes**.

4.3.154.3 Test Procedure

Step 1	Description
	Expected Result

4.3.155 LVV-T1845 - Verify accuracy of photometric transformation to physical scale

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1845 in Jira

4.3.155.1 Verification Elements

None.

4.3.155.2 Test Items

Verify that the DM system provides software to assess whether the accuracy of the transformation of internal LSST photometry to a physical scale (e.g. AB magnitudes) is less than **PA6 = 10 millimagnitudes**.

4.3.155.3 Test Procedure

Step 1	Description
	Expected Result

4.3.156 LVV-T1846 - Verify calculation of band-to-band color zero-point accuracy including u-band

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1846 in Jira

4.3.156.1 Verification Elements

None.

4.3.156.2 Test Items

Verify that the DM system provides software to assess whether the accuracy of absolute band-to-band color zero-points for all colors constructed from any filter pair, including the u-band, is less than **PA5u = 10 millimagnitudes**.

4.3.156.3 Test Procedure

Step 1	Description
	Expected Result

4.3.157 LVV-T1847 - Verify calculation of sensor fraction with unusable pixels

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1847 in Jira

4.3.157.1 Verification Elements

None.

4.3.157.2 Test Items

Verify that the DM system provides software to assess whether the maximum allowable fraction of sensors with **PixFrac > 1** percent scientifically unusable pixels is less than **SensorFraction = 15 percent.**

4.3.157.3 Test Procedure

Step 1	Description
	Expected Result

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

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1862 in Jira

4.3.158.1 Verification Elements

None.

4.3.158.2 Test Items

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

4.3.158.3 Predecessors

Execution of LVV-T90.

4.3.158.4 Test Procedure

Step 1	Description

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

Expected Result

Step 2-1 from LVV-T1060	Description

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

Expected Result

Step 2-2 from LVV-T1060	Description

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

	Expected Result

Step 3	Description

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

	Expected Result

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

4.3.159 LVV-T1863 - Verify ability to process Special Programs data alongside normal processing

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1863 in Jira

4.3.159.1 Verification Elements

None.

4.3.159.2 Test Items

Verify that Special Programs data can be processed alongside either prompt-products or data-release processing with little or no extra effort by DM staff.

4.3.159.3 Test Procedure

Step 1	Description

	Expected Result

4.3.160 LVV-T1865 - Verify implementation of time to L1 public release for Special Programs

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1865 in Jira

4.3.160.1 Verification Elements

None.

4.3.160.2 Test Items

Verify that data from Special Programs are made available via public release within **L1PublicT = 24[hour]** from the acquisition of science data.

4.3.160.3 Test Procedure

Step 1	Description
	Expected Result

4.3.161 LVV-T1866 - Verify latency of reporting optical transients from Special Programs

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1866 in Jira

4.3.161.1 Verification Elements

None.

4.3.161.2 Test Items

Verify that optical transients (Level 1 data products) are reported within OTT1 = 1 minute of last image readout for Special Programs.

4.3.161.3 Test Procedure

Step 1	Description
	Expected Result

4.3.162 LVV-T1867 - Verify implementation of at least numStreams alert streams supported

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1867 in Jira

4.3.162.1 Verification Elements

None.

4.3.162.2 Test Items

Verify that the LSST system supports the transmission of at least **numStreams=5** full alert streams out of the alert distribution system within **OTT1=1 minute**.

4.3.162.3 Test Procedure

Step 1	Description
	Expected Result

4.3.163 LVV-T1868 - Verify implementation of alert streams distributed within latency limit

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1868 in Jira

4.3.163.1 Verification Elements

None.

4.3.163.2 Test Items

Verify that the LSST system supports the transmission of full alert streams out of the alert distribution system within **OTT1=1 minute**.

4.3.163.3 Test Procedure

Step 1	Description
	Expected Result

4.3.164 LVV-T2091 - Verify Fraction of Alerts Transmitted Within Latency Threshold

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T2091 in Jira

4.3.164.1 Verification Elements

None.

4.3.164.2 Test Items

Verify that at least **OTR1 = 98[percent]** of detectable alerts are actually transmitted within latency **OTT1 = 1[minute]**.

4.3.164.3 Test Procedure

Step 1	Description
	Expected Result

4.3.165 LVV-T2092 - Verify Meeting Threshold for Max Fraction of Visits With Failed Alerts

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T2092 in Jira

4.3.165.1 Verification Elements

None.

4.3.165.2 Test Items

Verify that no more than **sciVisitAlertFailure = 0.1[percent]** of visits fail to generate or distribute alerts.

4.3.165.3 Test Procedure

Step 1	Description
	Expected Result

4.3.166 LVV-T2093 - Verify Latency of Reporting Transients

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T2093 in Jira

4.3.166.1 Verification Elements

None.

4.3.166.2 Test Items

Verify that transients are reported within **OTT1 = 1[minute]** following the completion of readout of the last image of a visit. At least **OTR1 = 98[percent]** of the alerts should be transmitted within this latency period.

4.3.166.3 Test Procedure

Step 1	Description
	Expected Result

4.3.167 LVV-T2094 - Verify Peak Number of Alerts Per Standard Visit

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T2094 in Jira

4.3.167.1 Verification Elements

None.

4.3.167.2 Test Items

Verify that the instantaneous peak number of alerts per standard visit does not exceed **nAlertVisitPeak = 40000[integer]**.

4.3.167.3 Test Procedure

Step 1	Description
	Expected Result

4.3.168 LVV-T2095 - Verify Max Fraction of Visits With Alert Delays

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T2095 in Jira

4.3.168.1 Verification Elements

None.

4.3.168.2 Test Items

Verify that no more than **sciVisitAlertDelay = 1[percent]** of science visits have less than **OTR1 = 98[percent]** of the alerts distributed within **OTT1 = 1[minute]**.

4.3.168.3 Test Procedure

Step 1	Description
	Expected Result

4.3.169 LVV-T2096 - Verify Handling of Peak Number of Alerts

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T2096 in Jira

4.3.169.1 Verification Elements

None.

4.3.169.2 Test Items

Verify that the system can identify and distribute at least **nAlertVisitPeak = 40000[integer]** alerts per standard visit.

4.3.169.3 Test Procedure

Step 1	Description
	Expected Result

4.3.170 LVV-T2097 - Verify Handling of Average Number of Alerts

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Eric Bellm

Open LVV-T2097 in Jira

4.3.170.1 Verification Elements

None.

4.3.170.2 Test Items

Verify that the system can identify and distribute an average of **nAlertVisitAvg = 10000[integer]** alerts per standard visit over a given night.

4.3.170.3 Test Procedure

Step 1	Description

	Expected Result

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

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Inspection	Leanne Guy

Open LVV-T2176 in Jira

4.3.171.1 Verification Elements

None.

4.3.171.2 Test Items

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

4.3.171.3 Test Procedure

Step 1	Description

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

	Expected Result

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

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Inspection	Leanne Guy

Open LVV-T2177 in Jira

4.3.172.1 Verification Elements

None.

4.3.172.2 Test Items

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

4.3.172.3 Test Procedure

Step 1	Description
Check that the correct value for the TE3 threshold has been encoded in the faro package.	
	Expected Result

4.3.173 LVV-T2297 - Verify implementation of Science Data Archive

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Demonstration	Leanne Guy

Open LVV-T2297 in Jira

4.3.173.1 Verification Elements

None.

4.3.173.2 Test Items

Verify that a Science Data Archive has been created and that all LSST public data products have been archived together with the raw data necessary to reproduce them. Verify that the archive is scalable to the data from the full survey and all Data Releases.

This requirement will be verified by analysis. Verification must demonstrate that we have a a written plan for how data will be archived and that the storage systems needed exist. The plan should include details on recovery. This is needed before commissioning to support commissioning data taking.

4.3.173.3 Test Procedure

Step 1	Description
Check that all LSST public data products have been archived at the Science Data Archive	
	Expected Result

Step 2	Description
Test that the the public data products can be reproduced from the raw data stored at the archive.	
	Expected Result

Step 3	Description
Test that the archive is scalable to the full survy data volume.	
	Expected Result

4.3.174 LVV-T2302 - Verify the minimum number of simultaneous users retrieving a set of postage stamp images

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Leanne Guy

Open LVV-T2302 in Jira

4.3.174.1 Verification Elements

None.

4.3.174.2 Test Items

Verify that the DMS can support at least postageStampRetrievalUsers = 10 simuylataneous users retrieving a set of postage stamp images in postageStampRetrievalTime = 10sec.Â

4.3.174.3 Test Procedure

Step 1	Description

Execute ten independent processes simultaneously to retrieve a set of postage stamp images. Time the execution until all complete

Expected Result

Step 2	Description

Verify that the time to execute all processes is within specification

Expected Result

4.3.175 LVV-T2303 - Verify Image Archive

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Leanne Guy

Open LVV-T2303 in Jira

4.3.175.1 Verification Elements

None.

4.3.175.2 Test Items

Verify that all image Data Products produced by the DMS (Processed Science Exposures, Calibration Exposures, Coadded Exposures) are either archived, or be capable of being recreated on-demand from inputs and processing provenance.

4.3.175.3 Test Procedure

Step 1	Description
	Expected Result

4.3.176 LVV-T2304 - Verify maximum number of stars associated with a DIASource.

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Leanne Guy

Open LVV-T2304 in Jira

4.3.176.1 Verification Elements

None.

4.3.176.2 Test Items

Verify the maximum number of stars associated with a DIASource does not exceed the maximum of diaNearbyObjMaxStar=3

4.3.176.3 Test Procedure

Step 1-1 from LVV-T866	Description

Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.

Expected Result

An output dataset including difference images and DIASource and DIAObject measurements.

Step 1-2 from LVV-T866	Description

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

Expected Result

Step 2-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 3	Description

Verify that DIAObjects have no more than diaNearbyObjMaxStar that point to the Object catalog

Expected Result

4.3.177 LVV-T2305 - Verify radius considered nearby

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Leanne Guy

Open LVV-T2305 in Jira

4.3.177.1 Verification Elements

None.

4.3.177.2 Test Items

Verify that the radius within which an Object is considered to be near, and possibly coincident with, the DIASource is not greater that the maximum spcification of diaNearbyObjRadius = 6 arcsec.Â

4.3.177.3 Test Procedure

Step 1-1 from LVV-T866	Description

Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.

Expected Result

An output dataset including difference images and DIASource and DIAObject measurements.

Step 1-2 from LVV-T866	Description

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

Expected Result

Step 2-1 from LVV-T987	Description

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

Example Code

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

Expected Result

Butler repo available for reading.

Step 3	Description

Verify that all stars adn galaxies associated with DIAObjects are within dianNearbyObjRadius.

Expected Result

4.3.178 LVV-T2328 - Verify regeneration of un-archived Data Products (Services)_1

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Leanne Guy

Open LVV-T2328 in Jira

4.3.178.1 Verification Elements

None.

4.3.178.2 Test Items

Verify that unarchived Level 1 and Level 2 data products can be regenerated on-demand using and IVOA-standards based service that usese archived inputs and provenance data.Â

4.3.178.3 Test Procedure

Step 1	Description
	Expected Result

4.3.179 LVV-T2329 - Verify the archiving of ancilliary data

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Leanne Guy

Open LVV-T2329 in Jira

4.3.179.1 Verification Elements

None.

4.3.179.2 Test Items

Verufy that the Science Data Archive Â contains all necessary engineering and calibration data for the full understanding of the performance and operation of the Observatory.

4.3.179.3 Test Procedure

Step 1	Description
	Expected Result

4.3.180 LVV-T2330 - Verify that the data processing infrastructure for user computing exists

Version	Status	Priority	Verification Type	Owner

1	Draft	Normal	Test	Leanne Guy

Open LVV-T2330 in Jira

4.3.180.1 Verification Elements

None.

4.3.180.2 Test Items

Verify that at least **userComputingFraction** of the total capability of the DMS is provided for user-dedicated processing and user-dedicated storage, including for the generation of Level 3 data products.

4.3.180.3 Test Procedure

Step 1	Description
	Expected Result

4.3.181 LVV-T2331 - Verify the number of precovery serivce connections

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Leanne Guy

Open LVV-T2331 in Jira

4.3.181.1 Verification Elements

None.

4.3.181.2 Test Items

Verify that the minimum number of precovery service connections that can be supported per hour is precoveryServePeakUsers

4.3.181.3 Test Procedure

Step 1	Description
	Expected Result

4.3.182 LVV-T2332 - Verify the time to retrieve results from a query of the prompt products database

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Leanne Guy

Open LVV-T2332 in Jira

4.3.182.1 Verification Elements

None.

4.3.182.2 Test Items

Verify that the maximum time allowed for retrieving results of a query of the Prompt Products Database is no more that l1QueryTime (10 seconds)

4.3.182.3 Test Procedure

Step 1	Description

	Expected Result

4.3.183 LVV-T2333 - Verify the minimum number of simultaneous users querying the prompt products database.

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Leanne Guy

Open LVV-T2333 in Jira

4.3.183.1 Verification Elements

None.

4.3.183.2 Test Items

Verify that the minimum number of simultaneous users querying the prompt products database that can be supported is **1QueryUsers = 20[integer]** , assuming that the query lasts no more than **l1QueryTime**.

4.3.183.3 Test Procedure

Step 1	Description
	Expected Result

4.3.184 LVV-T2334 - Verify implementation of processed visit images - snaps

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Leanne Guy

Open LVV-T2334 in Jira

4.3.184.1 Verification Elements

None.

4.3.184.2 Test Items

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

The verification should include confirming that the images have been trimmed of the overscan, and that correction of the instrumental signature (including crosstalk) has been applied properly.

This test specifically tests the combination of snaps.Â

4.3.184.3 Test Procedure

Step 1	Description
	Expected Result

4.3.185 LVV-T2692 - Verify implementation of Image Metadata Access

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T2692 in Jira

4.3.185.1 Verification Elements

None.

4.3.185.2 Test Items

Verify that available image data products' metadata can be listed and retrieved.

4.3.185.3 Test Procedure

Step 1	Description
	Expected Result

4.3.186 LVV-T2693 - Verify implementation of Image Provenance Access

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T2693 in Jira

4.3.186.1 Verification Elements

None.

4.3.186.2 Test Items

Verify that available image data products' provenance information can be listed and retrieved.

4.3.186.3 Test Procedure

Step 1	Description
	Expected Result

4.3.187 LVV-T2694 - Verify implementation of File Data Product Access

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T2694 in Jira

4.3.187.1 Verification Elements

None.

4.3.187.2 Test Items

Verify that available file data products can be listed and retrieved.

4.3.187.3 Test Procedure

Step 1	Description
	Expected Result

4.3.188 LVV-T2695 - Verify implementation of file metadata access

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T2695 in Jira

4.3.188.1 Verification Elements

None.

4.3.188.2 Test Items

Verify that available file data products' metadata can be listed and retrieved.

4.3.188.3 Test Procedure

Step 1	Description
	Expected Result

4.3.189 LVV-T2696 - Verify implementation of file provenance access

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T2696 in Jira

4.3.189.1 Verification Elements

None.

4.3.189.2 Test Items

Verify that available file data products' provenance can be listed and retrieved.

4.3.189.3 Test Procedure

Step 1	Description
	Expected Result

4.3.190 LVV-T2697 - Verify implementation of Catalog Data Product Access

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T2697 in Jira

4.3.190.1 Verification Elements

None.

4.3.190.2 Test Items

Verify that available catalog data products can be listed and retrieved.

4.3.190.3 Test Procedure

Step 1	Description
	Expected Result

4.3.191 LVV-T2698 - Verify implementation of Catalog Metadata Access

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T2698 in Jira

4.3.191.1 Verification Elements

None.

4.3.191.2 Test Items

Verify that available catalog data products' metadata can be listed and retrieved.

4.3.191.3 Test Procedure

Step 1	Description
	Expected Result

4.3.192 LVV-T2699 - Verify implementation of Catalog Provenance Access

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T2699 in Jira

4.3.192.1 Verification Elements

None.

4.3.192.2 Test Items

Verify that available catalog data products' provenance can be listed and retrieved.

4.3.192.3 Test Procedure

Step 1	Description
	Expected Result

4.3.193 LVV-T2700 - Verify Result latency for high-volume complex queries

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Demonstration	Colin Slater

Open LVV-T2700 in Jira

4.3.193.1 Verification Elements

None.

4.3.193.2 Test Items

Verify that complex high-volume queries that involve full-sky spatial and temporal correlations can Â be answered in less than **hvComplexQueryTime**.

4.3.193.3 Test Procedure

Step 1	Description
	Expected Result

4.3.194 LVV-T2724 - Verify Result latency for high-volume full-sky queries on the Object table

Version	Status	Priority	Verification Type	Owner

1	Draft	Normal	Test	Colin Slater

Open LVV-T2724 in Jira

4.3.194.1 Verification Elements

None.

4.3.194.2 Test Items

Verify latency against a full scale, e.g DR1 sized Object catalog

4.3.194.3 Test Procedure

Step 1	Description
	Expected Result

5 Reusable Test Cases

Test cases in this section are made up of commonly encountered steps that have been factored out into modular, reusable scripts. These test cases are meant solely for the building of actual tests used for verification, to be inserted in test scripts via the "Call to Test" functionality in Jira/ATM. They streamline the process of writing test scripts by providing pre-designed steps, while also ensuring homogeneity throughout the test suite. These reusable modules are not themselves verifying requirements. Also, these test cases shall not call other reusable test cases in their script.

5.1 LVV-T216 - Installation of the Alert Distribution payloads.

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Eric Bellm

Open LVV-T216 in Jira

5.1.0.1 Test Items

This test will check:

- That the Alert Distribution payloads are available from documented channels.
- That the Alert Distribution payloads can be installed on LSST Data Facility-managed systems.
- That the Alert Distribution payloads can be executed by LSST Data Facility-managed systems.

5.1.0.2 Environment Needs

5.1.0.2.1 Hardware

This test case shall be executed on the Kubernetes Commons at the LDF.

As discussed in https://dmtn-028.lsst.io/ and https://dmtn-081.lsst.io/, the test machine should have at least 16 cores, 64 GB of memory and access to at least 1.5 TB of shared storage.

5.1.0.3 Test Procedure

Step 1	Description

Download Kafka Docker image from https://github.com/lsst-dm/alert_stream.

Expected Result

Runs without error

Step 2	Description

Change to the alert_stream directory and build the docker image.

```
docker build -t "lsst-kub001:5000/alert_stream"
```

Expected Result

Runs without error

Step 3	Description

Register it with Kubernetes

docker push lsst-kub001:5000/alert_stream

Expected Result

Runs without error

Step 4	Description

From the alert_stream/kubernetes directory, start Kafka and Zookeeper:

```
kubectl create -f zookeeper-service.yaml
kubectl create -f zookeeper-deployment.yaml
kubectl create -f kafka-deployment.yaml
kubectl create -f kafka-service.yaml
```

(use kubectl get pods/services between each command to check status; wait until each is “Running” before starting the next command)

Expected Result
Runs without error

Step 5	Description

Confirm Kafka and Zookeeper are listed when running

kubectl get pods

and

kubectl get services

Expected Result

Output should be similar to:

kubectl get pods
NAME READY STATUS RESTARTS AGE
kafka-768ddf5564-xwgvh 1/1 Running 0 31s
zookeeper-f798cc548-mgkpn 1/1 Running 0 1m

kubectl get services
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kafka ClusterIP 10.105.19.124 <none> 9092/TCP 6s
zookeeper ClusterIP 10.97.110.124 <none> 32181/TCP 2m

5.2 LVV-T837 - Authenticate to Notebook Aspect

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Jeffrey Carlin

Open LVV-T837 in Jira

5.2.0.1 Test Items

Not specifically a test – modular script to be used in multiple other Test Scripts.

5.2.0.2 Input Specification

Must have a user account on the LSP.

5.2.0.3 Test Procedure

Step 1	Description

Authenticate to the notebook aspect of the Rubin Science Platform (NB-RSP). This is currently at either `https://data.lsst.cloud/nb` (for the interim data facility, or IDF) or `https://usdf-rsp.slac.stanford.edu/nb` (for the US data facility, or USDF).

Expected Result

Redirection to the spawner page of the NB-RSP allowing selection of the containerized science pipelines version and machine flavor.

Step 2	Description

Spawn a container by:
1) choosing an appropriate science pipelines version: e.g. the latest weekly.
2) choosing an appropriate machine flavor: e.g. medium
3) click "Spawn"

Expected Result

Redirection to the JupyterLab environment served from the chosen container containing the correct science pipelines version.

5.3 LVV-T838 - Access an empty notebook in the Notebook Aspect

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Simon Krughoff

Open LVV-T838 in Jira

5.3.0.1 Test Items

The steps here cover just those necessary to gain access to an empty notebook after authentication is complete.

5.3.0.2 Input Specification

Authentication to the Notebook aspect.

5.3.0.3 Test Procedure

Step 1	Description

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

Expected Result

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

Step 2	Description

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

Expected Result

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

5.4 LVV-T849 - Authenticate to the Portal Aspect of the RSP

Version	Status	Priority	Verification Type	Owner
4	Approved	Normal	Test	Simon Krughoff

Open LVV-T849 in Jira

5.4.0.1 Test Items

Obtain an authenticated session in the Portal Aspect of the Rubin Science Platform

5.4.0.2 Test Procedure

Step 1	Description

Navigate to the Portal Aspect endpoint. The stable version of the RSP at the interim data facility (IDF) should be used for this test and is currently located at: `https://data.lsst.cloud/`. The Portal Aspect can be reached by clicking on "Portal" in the RSP home page or by navigating directly to `https://data.lsst.cloud/portal/app`.

Expected Result

A credential-entry screen should be displayed.

Step 2	Description

Enter a valid set of credentials for an LSST user with RSP access on the instance under test.

Expected Result

The Portal Aspect UI should be displayed following authentication.

5.5 LVV-T850 - Log out of the portal aspect of the RSP

Version	Status	Priority	Verification Type	Owner
1	Approved	Normal	Test	Simon Krughoff

Open LVV-T850 in Jira

5.5.0.1 Test Items

Leave the portal aspect of the Rubin Science Platform in a clean state

5.5.0.2 Test Procedure

Step 1	Description

Click the "logout" button at the upper right corner of the Portal screen.

Expected Result

Returned to the RSP home page at https://data.lsst.cloud/. When navigating to the portal endpoint, expect to execute the steps in LVV-T849.

5.6 LVV-T860 - Initialize science pipelines

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T860 in Jira

5.6.0.1 Test Items

Initialize the science pipelines software for use.

5.6.0.2 Input Specification

An installed software stack, either locally, on 'lsst-dev', or through the Notebook aspect.

5.6.0.3 Test Procedure

Step 1	Description

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

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

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

Example Code

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

Expected Result

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

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

5.7 LVV-T866 - Run Alert Production Payload

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T866 in Jira

5.7.0.1 Test Items

Execute Alert Production payload on a dataset. Generate all (or a subset of) Prompt science data products including Alerts (with the exception of Solar System object orbits) and load them into the Data Backbone and Prompt Products Database.

5.7.0.2 Test Procedure

Step 1	Description
Perform the steps of Alert Production (including, but not necessarily limited to, single frame processing, ISR, source detection/measurement, PSF estimation, photometric and astrometric calibration, difference imaging, DIASource detection/measurement, source association). During Operations, it is presumed that these are automated for a given dataset.	
	Expected Result
An output dataset including difference images and DIASource and DIAObject measurements.	

Step 2	Description
Verify that the expected data products have been produced, and that catalogs contain reasonable values for measured quantities of interest.	
	Expected Result

5.8 LVV-T901 - Run MOPS payload

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T901 in Jira

5.8.0.1 Test Items

Run MOPS payload on a dataset (for example, one night's data). Generate entries in the MOPS Database and the Prompt Products Database, including Solar System Object records, measurements, and orbits. Perform precovery forced photometry of transients.

5.8.0.2 Predecessors

Uses results loaded into Prompt Products database and Data Backbone services in LVV-T866.

5.8.0.3 Test Procedure

Step 1	Description

Perform the steps of Moving Object Pipeline (MOPS) processing on newly detected DIASources, and generate Solar System data products including Solar System objects with associated Keplerian orbits, errors, and detected DIASources. This includes running processes to link DIASource detections within a night (called tracklets), to link these tracklets across multiple nights (into tracks), to fit the tracks with an orbital model to identify those tracks that are consistent with an asteroid orbit, to match these new orbits with existing SSObjects, and to update the SSObject table.

Expected Result

An output dataset consisting of an updated SSObject database with SSObjects both added and pruned as the orbital fits have been refined, and an updated DIASource database with DIASources assigned and unassigned to SSObjects.

Step 2	Description

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

Expected Result

5.9 LVV-T987 - Instantiate the Butler for reading data

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T987 in Jira

5.9.0.1 Test Items

Create a Butler client to read data from an input repository.

5.9.0.2 Input Specification

LVV-T860 must be executed to initialize the science pipelines.

5.9.0.3 Test Procedure

Step 1	Description

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

Example Code

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

Expected Result
Butler repo available for reading.

5.10 LVV-T1059 - Run Daily Calibration Products Update Payload

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1059 in Jira

5.10.0.1 Test Items

Execute the Daily Calibration Products Update payload to create a subset of Master Calibration images and Calibration Database entries.

5.10.0.2 Test Procedure

Step 1	Description

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

Expected Result

Step 2	Description

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

Expected Result

5.11 LVV-T1060 - Run Periodic Calibration Products Production Payload

Version	Status	Priority	Verification Type	Owner

1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1060 in Jira

5.11.0.1 Test Items

Execute the Calibration Products Production payload to create a subset of Master Calibration images and Calibration Database entries.

5.11.0.2 Test Procedure

Step 1	Description

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

	Expected Result

Step 2	Description

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

	Expected Result

5.12 LVV-T1064 - Run Data Release Production Payload

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1064 in Jira

5.12.0.1 Test Items

Execute the Data Release Production payload, starting from raw images and producing science data products.

5.12.0.2 Test Procedure

Step 1	Description

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

Expected Result

5.13 LVV-T1207 - Execute a simple ADQL query using the TAP service in the notebook aspect

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Jeffrey Carlin

Open LVV-T1207 in Jira

5.13.0.1 Test Items

Extract a small amount of data from a catalog via the LSST TAP service.

5.13.0.2 Input Specification

One must have access to the LSST Notebook Aspect, and have logged in and opened an empty notebook.

5.13.0.3 Test Procedure

Step 1	Description

Execute a query in a notebook to select a small number of stars. In the example code below, we query the Data Preview 0.2 (DP0.2) catalog, then extract the results to an Astropy table.

Example Code

```
CELL 1:

from IPython.display import Markdown as md
from lsst.rsp import get_tap_service, retrieve_query

service = get_tap_service()
md(f'The service endpoint for TAP in this environment is:\n\n &#10145;     {service.baseurl}')

CELL 2:
```

```
results = service.search("SELECT coord_ra, coord_dec, g_cModelFlux, r_cModelFlux \
                          FROM dp02_dc2_catalogs.Object \
                          WHERE CONTAINS(POINT('ICRS', coord_ra, coord_dec), \
                          CIRCLE('ICRS', 60.0, -30.0, 0.05)) = 1")
```

Expected Result

Screen output from CELL 1:

The service endpoint for TAP in this environment is:
âž¡ `https://data.lsst.cloud/api/tap`

Example screen output from CELL 2 (may not contain the same 10 entries):

Table length=5533

coord_ra

coord_dec

g_cModelFlux

r_cModelFlux

deg

deg

nJy

nJy

float64

float64

float64

float64

59.9987401

-29.9728812

62.7060123

49.3496319

59.9995813

-29.9743232

166.0433743

394.8261645

59.9989853

-29.9750457

78.9557388

85.2691232

59.9993731

-29.9732406

111.0082072

165.6229656

60.0477786

-29.9736805

68.4818592

49.4783714

60.0400024

-29.9731507

52.0567337

114.2562171

60.0054666

-29.9728639

146.053072

134.1795803

60.00489

-29.9732239

1436.7150639

3606.8163133

60.0469583

-29.9735655

64.8838762

56.5677789

...

...

...

...

60.0053313

-30.0240394

125.6977786

379.8120713

59.9574061

-30.0163726

181.050889

200.8032979

60.0294415

-30.0241709

133.662163

230.8673464

59.9563419

-30.0239843

1551.2308712

4611.0406542

59.9879157

-30.0181116

76.3796313

46.5682713

60.0204061

-30.0228981

174.7738892

304.9991558

60.001638

-30.0183336

43.9593753

46.9695823

59.9861714

-30.0173405

164.6261404

288.8650875

59.9537443

-30.0160515

2228.7204658

5091.2041475

59.9683498

-30.0239539

835.415374

1101.0548649

—

—

5.14 LVV-T1208 - Log out of the notebook aspect of the RSP

Version	Status	Priority	Verification Type	Owner
1	Draft	Normal	Test	Simon Krughoff

Open LVV-T1208 in Jira

5.14.0.1 Test Items

Leave the notebook aspect of the Rubin Science Platform in a clean state

5.14.0.2 Test Procedure

Step 1	Description

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

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

Expected Result

You will be returned to the RSP landing page: either `https://data.lsst.cloud/nb` (for the interim data facility, or IDF) or `https://usdf-rsp.slac.stanford.edu/nb` (for the US data facility, or USDF). It is now safe to close the browser window.

5.15 LVV-T1744 - Run faro on a repository of data

Version	Status	Priority	Verification Type	Owner
1	Defined	Normal	Analysis	Jeffrey Carlin

Open LVV-T1744 in Jira

5.15.0.1 Test Items

Run the faro code on a dataset containing the appropriate data products, to evaluate the metrics that have been implemented in faro.

5.15.0.2 Test Procedure

Step 1	Description

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

Example Code

```
pipetask –long-log run -j 2 -b DATA/path/butler.yaml –register-dataset-types -p $FARO_DIR/pipelines/metrics_pipeline.yaml -
```

d “band in (’g’, ‘r’, ‘i’) AND tract=9813 AND skymap=’hsc_rings_v1’ AND instrument=’HSC”’ –output u/username/faro_metrics -i HSC/runs/RC2/w_2021_06 2>&1 | tee w06_2021_tract9813_faro.txt

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

6 Deprecated Test Cases

This section includes all test cases that have been marked as deprecated. These test cases will never be executed again, but have been in the past. For this reason it is important to keep them in the baseline as a reference.

6.1 LVV-T10 - DRP-00-00: Installation of the Data Release Production v14.0 science payload

Version	Status	Priority	Verification Type	Owner
1	Deprecated	Normal	Test	Jim Bosch

Open LVV-T10 in Jira

6.1.0.1 Verification Elements

None.

6.1.0.2 Test Items

This test will check:

- That the Data Release Production science payload is available for distribution from documented channels;
- That the Data Release Production science payload can be installed on LSST Data Facility-managed systems.

6.2 LVV-T11 - DRP-00-05: Execution of the DRP Science Payload by the Batch Production Service

Version	Status	Priority	Verification Type	Owner
1	Deprecated	Normal	Test	Jim Bosch

Open LVV-T11 in Jira

6.2.0.1 Verification Elements

None.

6.2.0.2 Test Items

This test will check that the DRP Science Payload can be executed using a specific version of the Batch Production Service provided by the LSST Data Facility. Since the outputs are stored in the Data Backbone, it too is a component of this test.

6.3 LVV-T12 - DRP-00-10: Data Release Includes Required Data Products

Version	Status	Priority	Verification Type	Owner
1	Deprecated	Normal	Test	Jim Bosch

Open LVV-T12 in Jira

6.3.0.1 Verification Elements

None.

6.3.0.2 Test Items

This test will check that the basic data products which should be in an data release are generated by execution of the science payload.
These products will include:

- Source catalogs, derived from PVIs and coadded images (DMS-REQ-0267 & DMS-REQ-0277);
- Forced source catalogs (DMS-REQ-0268);
- Object catalogs (DMS-REQ-0275);
- Processed visit images (PVIs; DMS-REQ-0069);
- Coadded images (DMS-REQ-0279);

6.4 LVV-T13 - DRP-00-15: Scientific Verification of Source Catalog

Version	Status	Priority	Verification Type	Owner
1	Deprecated	Normal	Test	Jim Bosch

Open LVV-T13 in Jira

6.4.0.1 Verification Elements

None.

6.4.0.2 Test Items

This test will check that the source catalogs delivered by the DRP science payload meet the requirements laid down by LSE-61.
Specifically, this will demonstrate that:

- Measurements in the catalog are presented in flux units (DMS-REQ-0347);
- Derived quantities are provided in pre-computed columns (DMS-REQ-0331);
- Aperture corrections for different photometry algorithms are consistent.
- Photometry measurements are consistent with reference catalog photometry (including sources not used in photometric calibration).
- Astrometry measurements are consistent with reference catalog positions (including sources not used in astrometric calibration).

This test does not include quantitative targets for the science quality criteria; we instead require for each test that we be able to quickly construct a plot in which such a target can be visualized.

6.5 LVV-T14 - DRP-00-25: Scientific Verification of Object Catalog

Version	Status	Priority	Verification Type	Owner
1	Deprecated	Normal	Test	Jim Bosch

Open LVV-T14 in Jira

6.5.0.1 Verification Elements

None.

6.5.0.2 Test Items

This test will check that the object catalogs delivered by the DRP science payload meet the requirements laid down by LSE-61.
Specifically, this will demonstrate that:

- Measurements in the catalog are presented in flux units (DMS-REQ-0347);
- Derived quantities are provided in pre-computed columns (DMS-REQ-0331);
- Aperture corrections for different photometry algorithms are consistent.
- PSF models correctly predict the ellipticities of stars over each tract.
- Photometry measurements are consistent with reference catalog photometry (including sources not used in photometric calibration).
- Astrometry measurements are consistent with reference catalog positions (including sources not used in astrometric calibration).
- Forced and unforced photometry measurements are consistent.
- The slope of the stellar locus in color-color space is not a function of position on the sky.

This test does not include quantitative targets for the science quality criteria; we instead require for each test that we be able to quickly construct a plot in which such a target can be visualized.
All science quality tests in this section shall distinguish between blended and isolated objects.

6.6 LVV-T15 - DRP-00-30: Scientific Verification of Processed Visit Images

Version	Status	Priority	Verification Type	Owner
1	Deprecated	Normal	Test	Jim Bosch

Open LVV-T15 in Jira

6.6.0.1 Verification Elements

None.

6.6.0.2 Test Items

This test will check that the Processed Visit Images (PVIs) delivered by the DRP science payload meet the requirements laid down by LSE-61.
Specifically, this will demonstrate that:

- Processed visit images have been generated and persisted during payload execution;
- Each PVI includes a background model (DMS-REQ-0327), photometric zero-point (DMS-REQ-0029), spatially-varying PSF (DMS-REQ-0070) and WCS (DMS-REQ-0030).
- Saturated pixels are correctly masked.
- Pixels affected by cosmic rays are correctly masked.
- The background is not oversubtracted around bright objects.

This test does not include quantitative targets for the science quality criteria; we instead require for each test that we be able to quickly construct a plot or display summary images that allow such a target can be visualized.

6.7 LVV-T16 - DRP-00-35: Scientific Verification of Coadd Images

Version	Status	Priority	Verification Type	Owner
1	Deprecated	Normal	Test	Jim Bosch

Open LVV-T16 in Jira

6.7.0.1 Verification Elements

None.

6.7.0.2 Test Items

This test will check that the coadded images delivered by the DRP science payload meet the requirements laid down by LSE-61.
Specifically, this will demonstrate that:

- Coadds have been generated and persisted during payload execution;

- Each coadd provides a spatially varying PSF model (DMS-REQ-0047).
- Saturated pixels are correctly masked.
- Pixels affected by satellite trails and ghosts are rejected from the coadd.
- The background is not oversubtracted around bright objects.

This test does not include quantitative targets for the science quality criteria; we instead require for each test that we be able to quickly construct a plot or display summary images that allow such a target can be visualized.

6.8 LVV-T17 - AG-00-00: Installation of the Alert Generation v16.0 science payload.

Version	Status	Priority	Verification Type	Owner
1	Deprecated	Normal	Test	Eric Bellm

Open LVV-T17 in Jira

6.8.0.1 Verification Elements

None.

6.8.0.2 Test Items

This test will check:

- That the Alert Generation science payload is available for distribution from documented channels;
- That the Alert Generation science payload can be installed on LSST Data Facility-managed systems.

6.9 LVV-T18 - AG-00-05: Alert Generation Produces Required Data Products

Version	Status	Priority	Verification Type	Owner
1	Deprecated	Normal	Test	Eric Bellm

Open LVV-T18 in Jira

6.9.0.1 Verification Elements

None.

6.9.0.2 Test Items

This test will check that the basic data products produced by Alert Generation are generated by execution of the science payload.
These products will include:

- Processed visit images (PVIs; DMS-REQ-0069);
- Difference Exposures (DMS-REQ-0010);
- DIASource catalogs (DMS-REQ-0269);
- DIAObject catalogs (DMS-REQ-0271);

6.10 LVV-T19 - AG-00-10: Scientific Verification of Processed Visit Images

Version	Status	Priority	Verification Type	Owner
1	Deprecated	Normal	Test	Eric Bellm

Open LVV-T19 in Jira

6.10.0.1 Verification Elements

None.

6.10.0.2 Test Items

This test will check that the Processed Visit Images (PVIs) delivered by the alert generation science payload meet the requirements laid down by LSE-61.

Specifically, this will demonstrate that:

- Processed visit images have been generated and persisted during payload execution;
- Each PVI includes a science pixel array, a mask array, and a variance array. (DMS-REQ-0072).
- Each PVI includes a background model (DMS-REQ-0327), photometric zero-point (DMS-REQ-0029), spatially-varying PSF (DMS-REQ-0070) and WCS (DMS-REQ-0030).
- Saturated pixels are correctly masked.
- Pixels affected by cosmic rays are correctly masked.
- The background is not oversubtracted around bright objects.

This test does not include quantitative targets for the science quality criteria.

6.11 LVV-T20 - AG-00-15: Scientific Verification of Difference Images

Version	Status	Priority	Verification Type	Owner
1	Deprecated	Normal	Test	Eric Bellm

Open LVV-T20 in Jira

6.11.0.1 Verification Elements

None.

6.11.0.2 Test Items

This test will check that the difference images delivered by the Alert Generation science payload meet the requirements laid down by LSE-61.
Specifically, this will demonstrate that:

- Difference images have been generated and persisted during payload execution;
- Each difference image includes information about the identity of the input exposures, and metadata such as a representation of the PSF matching kernel (DMS-REQ-0074);
- Masks are correctly propagated from the input images.

This test does not include quantitative targets for the science quality criteria.

6.12 LVV-T21 - AG-00-20: Scientific Verification of DIASource Catalog

Version	Status	Priority	Verification Type	Owner
1	Deprecated	Normal	Test	Eric Bellm

Open LVV-T21 in Jira

6.12.0.1 Verification Elements

None.

6.12.0.2 Test Items

This test will check that the difference image source catalogs delivered by the Alert Generation science payload meet the requirements laid down by LSE-61.

- Specifically, this will demonstrate that:
- Measurements in the catalog are presented in flux units (DMS-REQ-0347);
- Each DIASource record contains an appropriate subset of the attributes required by DMS-REQ-0269. In particular, the LDM-503-3-era pipeline is expected to provide DIASource positions (sky and focal plane), fluxes, and flags indicative of issues encountered during processing.
- Faint DIASources satisfying additional criteria are stored (DMS-REQ-0270).
- Derived quantities are provided in pre-computed columns (DMS-REQ-0331);

This test does not include quantitative targets for the science quality criteria.

6.13 LVV-T22 - AG-00-25: Scientific Verification of DIAObject Catalog

Version	Status	Priority	Verification Type	Owner
1	Deprecated	Normal	Test	Eric Bellm

Open LVV-T22 in Jira

6.13.0.1 Verification Elements

None.

6.13.0.2 Test Items

This test will check that the DIAObject catalogs delivered by the Alert Generation science payload meet the requirements laid down by LSE-61.
Specifically, this will demonstrate that:

- DIAObjects are recorded with unique identifiers (DMS-REQ-0271);
- Measurements in the catalog are presented in flux units (DMS-REQ-0347);
- EachDIAObjectrecordcontainscontainsanappropriatesetofsummaryattributes(DMS- REQ-0271 and DMS-REQ-0272). Note:
  - This test is executed independently of the Data Release Production system. Hence, DIAObjects are not associated to Objects, and the association metadata specified by DMS-REQ-0271 is not expected to be available.
  - TheLDM-503-3erapipelineisnotexpectedtocalculateorpersistallattributesspec- ified by DMS-REQ-0272 requirement.
- Relevant derived quantities are provided in pre-computed columns (DMS-REQ-0331);

This test does not include quantitative targets for the science quality criteria.

6.14 LVV-T31 - Verify implementation of Crosstalk Corrected Science Image Data Acquisition

Version	Status	Priority	Verification Type	Owner
1	Deprecated	Normal	Test	Kian-Tat Lim

Open LVV-T31 in Jira

6.14.0.1 Verification Elements

None.

6.14.0.2 Test Items

Verify successful ingestion of crosstalk corrected data from L1 Test Stand DAQ while simulating all modes.

6.15 LVV-T378 - Verify Calculation of Astrometric Performance Metrics

Version	Status	Priority	Verification Type	Owner
1	Deprecated	Normal	Test	Leanne Guy

Open LVV-T378 in Jira

6.15.0.1 Verification Elements

None.

6.15.0.2 Test Items

Verify that the DMS system provides software to calculate astrometric performance metrics, and that the algorithms are properly calculating the desired quantities. Note that because the DMS requirement is that the software shall be provided (and not on the actual measured values of the metrics), we verify all of the requirements via a single test case.

A Traceability

Verification Elements	High Level Requirements	Test Cases