LARGE SYNOPTIC SURVEY TELESCOPE

Large Synoptic Survey Telescope (LSST)
Data Management

Distributed Database Software Test Specification

Fritz Mueller

LDM-552

Latest Revision: 2019-07-25

Abstract

This document describes the detailed test specification for the Distributed Database.

Change Record

Version	Date	Description	Owner name
	2017-07-02	Initial draft.	F. Mueller
1.0	2019-07-25	Document issued. Approved in RFC-612.	G. Comoretto

Document source location: `https://github.com/lsst/LDM-552`

Contents

Distributed Database Software Test Specification

1 Introduction

This document specifies the test procedure for Distributed Database. Distributed Database is a distributed shared-nothing RDBMS which will host LSST catalogs.

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 tests that will be performed on the Distributed Database as part of the verification of the DM system.

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

1.2 Scope

This document describes the test procedures for the following components of the LSST system (as described in LDM-148):

- Parallel Distributed Database (Qserv)

1.3 Applicable Documents

LDM-135	LSST Qserv Database Design
LDM-294	LSST DM Organization & Management
LDM-502	The Measurement and Verification of DM Key Performance Metrics
LDM-503	LSST DM Test Plan
LDM-555	LSST DM Database Requirements

2 Approach

The approaches taken for the tests described here are:

- Ongoing inspection of design documents, code, and CI system logs to verify that Distributed Database design and implementation meet DM software quality standards in general, and requirements as expressed in LDM-555 in particular;
- Ongoing deployment and continuous operation of Distributed Database in a Prototype Data Access Center (PDAC) in order to assess basic reliability, fitness for purpose, and integration with adjacent subsystems;
- Annual deployment of Distributed Database to test clusters, followed by synthesis and ingestion of test datasets and scripted performance/load/stress testing. The cluster size/capabilities and the scale of the synthetic test dataset are both evolved along a path toward anticipated LSST operational scale.

2.1 Tasks and criteria

Distributed Database is a containerized, distributed, Linux application, which is deployed on machine clusters. At the scales to be tested, these clusters are comprised of one to several head ("czar") nodes and additionally on the order of tens to hundreds of shard ("worker") nodes, interconnected locally via a high-performance network. Head and shard nodes are provisioned each with on the order of 10s of gigabytes of RAM, and each with on the order of 10s of terabytes of locally attached storage.

Ongoing deployment, continuous operation, and integration tests are carried out on machines within the Prototype Data Access Center (PDAC), a dedicated machine cluster physically located at NCSA's National Peta-scale Compute Facility, maintained by NCSA staff. Catalog datasets which are maintained within the PDAC Qserv instance and which are used for this testing include, simultaneously:

- An LSST stack reprocessed version of the SDSS Stripe 82 catalog (currently from Summer 2013 [Document-15097]) (~10 TB);
- IRSA AllWISE and NEOWISE catalogs (~50 TB);
- An LSST stack reprocessed version of the HSC catalog (scheduled; ~50 TB).

Tasks required for these tests include periodic update of the software deployed on the PDAC, periodic ingest of additional test datasets, and inter-operation with adjacent subsystems. Uptime is monitored cumulatively throughout these activities to gain quantitative insight into system stability and reliability.

Scaling, load, and stress testing are carried out on an additional machine cluster located CC-IN2P3 in Lyon, maintained by CC-IN2P3 staff. Scaling tests are run annually, by issuing a representative mix of concurrent queries against a synthetic catalog while monitoring average query execution times per query type. The scaling test dataset size and query concurrency level are increased each year on a glide path toward the full scale of Data Release 1.

Tasks required for these tests include generation and ingest of each successive test dataset, and execution of scripts which issue and monitor the suites of representative test queries.

2.2 Features to be tested

This version of the Distributed Database test specification addresses only basic product verification, basic reliability, and performance/scale testing – a bare minimum required to conduct ongoing development and verify that Distributed Database remains on a realistic path towards meeting its most technically challenging requirements: those related to successful operability at the scale that will be required by LSST.

2.2.1 Performances

In order to ensure that QSERV is able to meet the performance, specific test cases have been designed. These test cases will be executed each year, in order to denmonstrate that query performances are as described in the following table.

Query Class		2015	2016	2017	2018	2019	2020
Dataset size, relative to DR-1		10%	20%	30%	50%	75%	100%
LV	**# queries**	50	60	70	80	90	100
	time (sec)	10	10	10	10	10	10
FTSObj	**# queries**	3	4	8	12	16	20
	time (hours)	12	1	1	1	1	1
FTSSrc	**# queries**	1	1	2	3	4	5
	time (hours)	12	12	12	12	12	12
FTSFSrc	**# queries**		1	2	3	4	5
	time (hours)		12	12	12	12	12
joinObjSrc	**# queries**	1	2	4	6	8	10
	time (hours)	12	12	12	12	12	12
joinObjFSrc	**# queries**		1	2	3	4	5
	time (hours)		12	12	12	12	12
nearN	**# queries**		1	2	3	4	5
	time (hours)		1	1	1	1	1

2.3 Features not to be tested

Testing of the following are NOT YET COVERED in this specification:

- Fault-tolerance and disaster recovery;
- Schema evolution;
- Data ingest performance;
- Query reproducibility;
- Cross-match with external datasets.

It is anticipated that test specifications and cases for all of the above will be developed and added to future revisions of this document.

2.4 Pass/fail criteria

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

2.5 Suspension criteria and resumption requirements

Refer to individual test cases where applicable.

2.6 Naming convention

With the introduction of the Jira Test Management, the following definitions have to be considered:

LVV : Is the label for the "LSST Verification and Validation" project in Jira where all information regarding tests are managed.

LVV-XXX : Are Verification Elements, where XXX is the Verification Element identifier. Each Verification Element is derived from a requirement and has at least one Test Case associated. There can be multiple Verification Elements associated with a requirement.

LVV-TYYY : Are Test Cases. Each Test Case is associated with a Verification Element, where YYY is the Test Case identifier. There can be multiple test cases associated with a Verification Element.

3 Test Cases Summary

Test Id	Test Name
LVV-T1017	Qserv Preparation
LVV-T1085	Short Queries Functional Test
LVV-T1086	Full Table Scans Functional Test
LVV-T1087	Full Table Joins Functional Test
LVV-T1088	Concurrent Scans Scaling Test
LVV-T1089	Load Test
LVV-T1090	Heavy Load Test

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 LVV-T1017 - Qserv Preparation

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

4.1.1 Verification Elements

None.

4.1.2 Test Items

Before running any of the performance test cases, Qserv must be installed on an appropriate test cluster (e.g. the test machine cluster at CC-IN2P3). To upgrade Qserv software on the cluster in preparation for testing, follow directions at http://www.slac.stanford.edu/exp/lsst/qserv/2015_10/HOWTO/cluster-deployment.html.

The performance tests will also require an appropriately sized test dataset to be synthesized and ingested, per the yearly dataset sizing schedule described in LDM-552, section 2.2.1. Tools for synthesis of ingest of test datasets may be found in the LSST GitHub repot at https://github.com/lsst-dm/db_tests_kpm*. Detailed use and context information for the tools is described in https://jira.lsstcorp.org/browse/DM-8405.

It has also been found that the Qserv shard servers must have engine-independent statistics loaded for the larger tables in the test dataset, and be properly configured so that the MariaDB query planner can make use of those statistics. More information on this issue is available at https://confluence.lsstcorp.org/pages/viewpage.action?pageId=58950786.

4.1.3 Predecessors

4.1.4 Environment Needs

4.1.4.1 Software

4.1.4.2 Hardware

4.1.5 Input Specification

4.1.6 Output Specification

4.1.7 Test Procedure

Step	Description, Input Data and Expected Result	
1	Description	Install/upgrade Qserv on a test cluster, following directions at http://www.slac.stanford.edu/exp/lsst/qserv/2015_10/HOW-TO/cluster-deployment.html
	Test Data	No data.
	Expected Result	Qserv installed
2	Description	Synthesize and load and appropriately sized test dataset per the yearly dataset sizing schedule described in LDM-552, section 2.2.1. Tools for synthesis of ingest of test datasets may be found in the LSST GitHub repot at https://github.com/lsst dm/db_tests_kpm*. Detailed use and context information for the tools is described in https://jira.lsstcorp.org/browse/DM-8405.
	Test Data	No data.
	Expected Result	Test dataset loaded

4.2 LVV-T1085 - Short Queries Functional Test

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

4.2.1 Verification Elements

- LVV-33 - DMS-REQ-0075-V-01: Catalog Queries
- LVV-9787 - DMS-REQ-0356-V-04: Max time to retrieve low-volume query results

4.2.2 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.

4.2.3 Predecessors

4.2.4 Environment Needs

4.2.4.1 Software

4.2.4.2 Hardware

4.2.5 Input Specification

QSERV has been set-up following procedure at LVV-T1017.

4.2.6 Output Specification

4.2.7 Test Procedure

Step	Description, Input Data and Expected Result	
1	Description	Execute single object selection: **SELECT** * **FROM** Object **WHERE** deepSourceId = 9292041530376264 and record execution time.
	Test Data	No data.
	Expected Result	Query runs in less than 10 seconds.

Step	Description, Input Data and Expected Result	
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.
	Test Data	No data.
	Expected Result	Query runs in less than 10 seconds.

4.3 LVV-T1086 - Full Table Scans Functional Test

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

4.3.1 Verification Elements

- LVV-33 - DMS-REQ-0075-V-01: Catalog Queries
- LVV-188 - DMS-REQ-0357-V-01: Result latency for high-volume full-sky queries on the Object table
- LVV-185 - DMS-REQ-0354-V-01: Result latency for high-volume complex queries

4.3.2 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.

4.3.3 Predecessors

4.3.4 Environment Needs

4.3.4.1 Software

4.3.4.2 Hardware

4.3.5 Input Specification

QSERV has been set-up following procedure at LVV-T1017.

4.3.6 Output Specification

4.3.7 Test Procedure

Step	Description, Input Data and Expected Result	
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.
	Test Data	No data.
	Expected Result	Query expected to run in less than 1 hour.
2	Description	Execute query: **SELECT** COUNT(*) **FROM** Source **WHERE** flux_sinc **BETWEEN** 1 **AND** 1.1 and record the execution time
	Test Data	No data.
	Expected Result	Query expected to run in less than 12 hours.
3	Description	Execute query: **SELECT** COUNT(*) **FROM** ForcedSource **WHERE** psfFlux **BETWEEN** 0.1 **AND** 0.2 and record the execution time

Step	Description, Input Data and Expected Result	
	Test Data	No data.
	Expected Result	Query expected to run in less than 12 hours.

4.4 LVV-T1087 - Full Table Joins Functional Test

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

4.4.1 Verification Elements

- LVV-33 - DMS-REQ-0075-V-01: Catalog Queries
- LVV-185 - DMS-REQ-0354-V-01: Result latency for high-volume complex queries

4.4.2 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.

4.4.3 Predecessors

4.4.4 Environment Needs

4.4.4.1 Software

4.4.4.2 Hardware

4.4.5 Input Specification

QSERV has been set-up following procedure at LVV-T1017.

4.4.6 Output Specification

4.4.7 Test Procedure

Step	Description, Input Data and Expected Result	
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.
	Test Data	No data.
	Expected Result	Query expected to run in less than 12 hours.
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.
	Test Data	No data.
	Expected Result	Query expected to run in less than 12 hours.

4.5 LVV-T1088 - Concurrent Scans Scaling Test

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

4.5.1 Verification Elements

- LVV-185 - DMS-REQ-0354-V-01: Result latency for high-volume complex queries

- LVV-188 - DMS-REQ-0357-V-01: Result latency for high-volume full-sky queries on the Object table
- LVV-3403 - DMS-REQ-0361-V-01: Simultaneous users for high-volume queries

4.5.2 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.

4.5.3 Predecessors

4.5.4 Environment Needs

4.5.4.1 Software

4.5.4.2 Hardware

4.5.5 Input Specification

1. A test catalog of appropriate size (see schedule detail in LDM-552, section 2.2.1), prepared and ingested into the Qserv instance under test as detailed in LVV-T1017.
2. The concurrency load execution script, runQueries.py, maintained in the LSST Qserv github repository here: https://github.com/lsst/qserv/blob/master/admin/tools/docker/deployment/ir

4.5.6 Output Specification

4.5.7 Test Procedure

Step	Description, Input Data and Expected Result	
1	Description	Repeat steps 2 through 5 below, where “pool of interest” is taken first to be “FTSObj” and subsequently “FTSSrc”:
	Test Data	No data.
	Expected Result	At end of each pass, a graph indicating scan scaling rate and machine resource exhaustion cutoff.

Step	Description, Input Data and Expected Result	
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.
	Test Data	No data.
	Expected Result	rueQueries.py script updated with appropriate values for test iteration
3	Description	Execute the runQueries.py script and let it run for at least one, but preferably several, query cycles.
	Test Data	No data.
	Expected Result	Test script executes producing log file.
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.
	Test Data	No data.
	Expected Result	Logs indicate either successful test run, providing another growth point for curve, or errors indicating machine resource exhaustion cutoff has been reached.
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.
	Test Data	No data.
	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.

4.6 LVV-T1089 - Load Test

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

4.6.1 Verification Elements

- LVV-9786 - DMS-REQ-0356-V-03: Min number of simultaneous low-volume query users
- LVV-9787 - DMS-REQ-0356-V-04: Max time to retrieve low-volume query results

- LVV-188 - DMS-REQ-0357-V-01: Result latency for high-volume full-sky queries on the Object table
- LVV-185 - DMS-REQ-0354-V-01: Result latency for high-volume complex queries
- LVV-3403 - DMS-REQ-0361-V-01: Simultaneous users for high-volume queries

4.6.2 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.

4.6.3 Predecessors

4.6.4 Environment Needs

4.6.4.1 Software

4.6.4.2 Hardware

4.6.5 Input Specification

QSERV has been set-up following procedure at LVV-T1017

4.6.6 Output Specification

4.6.7 Test Procedure

Step	Description, Input Data and Expected Result	
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.
	Test Data	No data.
	Expected Result	Script updated with appropriate values.
2	Description	Execute the runQueries.py script and let it run for 24 hours.

Step	Description, Input Data and Expected Result	
	Test Data	No data.
	Expected Result	Script runs without error and produces output log.
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.
	Test Data	No data.
	Expected Result	Average query times per query type equal or less than corresponding yearly target values in LDM-552, section 2.2.1.

4.7 LVV-T1090 - Heavy Load Test

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

4.7.1 Verification Elements

- LVV-9786 - DMS-REQ-0356-V-03: Min number of simultaneous low-volume query users
- LVV-9787 - DMS-REQ-0356-V-04: Max time to retrieve low-volume query results
- LVV-188 - DMS-REQ-0357-V-01: Result latency for high-volume full-sky queries on the Object table
- LVV-185 - DMS-REQ-0354-V-01: Result latency for high-volume complex queries
- LVV-3403 - DMS-REQ-0361-V-01: Simultaneous users for high-volume queries

4.7.2 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.

4.7.3 Predecessors

4.7.4 Environment Needs

4.7.4.1 Software

4.7.4.2 Hardware

4.7.5 Input Specification

QSERV has been set-up following procedure at LVV-T1017

4.7.6 Output Specification

4.7.7 Test Procedure

Step	Description, Input Data and Expected Result	
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.
	Test Data	No data.
	Expected Result	Script updated with appropriate values.
2	Description	Execute the runQueries.py script and let it run for 24 hrs.
	Test Data	No data.
	Expected Result	Script runs without error and produces output log.
3	Description	Examine log file output and compile average query execution times per query type.
	Test Data	No data.
	Expected Result	Average query times per query type equal or less than corresponding yearly target values in LDM-552, section 2.2.1.

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.

No library test cases found.

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.

No deprecated test cases found.

A Traceability

Verification Elements	Test Cases
LVV-33 - DMS-REQ-0075-V-01: Catalog Queries	LVV-T1085 LVV-T1086 LVV-T1087
LVV-9787 - DMS-REQ-0356-V-04: Max time to retrieve low-volume query results	LVV-T1085 LVV-T1089 LVV-T1090
LVV-188 - DMS-REQ-0357-V-01: Result latency for high-volume full-sky queries on the Object table	LVV-T1086 LVV-T1088 LVV-T1089 LVV-T1090
LVV-185 - DMS-REQ-0354-V-01: Result latency for high-volume complex queries	LVV-T1086 LVV-T1087 LVV-T1088 LVV-T1089 LVV-T1090
LVV-3403 - DMS-REQ-0361-V-01: Simultaneous users for high-volume queries	LVV-T1088 LVV-T1089 LVV-T1090
LVV-9786 - DMS-REQ-0356-V-03: Min number of simultaneous low-volume query users	LVV-T1089 LVV-T1090

B References

[1] **[LDM-555]**, Becla, J., 2017, *Data Management Database Requirements*, LDM-555, URL https://ls.st/LDM-555

[2] **[LDM-135]**, Becla, J., Wang, D., Monkewitz, S., et al., 2017, *Data Management Database Design*, LDM-135, URL https://ls.st/LDM-135

[3] **[Document-15097]**, Lim, K.T., 2013, *LSST Data Challenge Report: Summer 2013*, Document-15097, URL https://ls.st/Document-15097

[4] **[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

[5] **[LDM-552]**, Mueller, F., 2017, *Qserv Software Test Specification*, LDM-552, URL https://ls.st/LDM-552

[6] **[LDM-502]**, Nidever, D., Economou, F., 2016, *The Measurement and Verification of DM Key Performance Metrics*, LDM-502, URL https://ls.st/LDM-502

[7] **[LDM-294]**, O'Mullane, W., Swinbank, J., Jurić, M., DMLT, 2018, *Data Management Organization and Management*, LDM-294, URL https://ls.st/LDM-294

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

C Acronyms

Acronym	Description
ATM	Adaptavist Test Management
AURA	Association of Universities for Research in Astronomy
Archive	The repository for documents required by the NSF to be kept. These include documents related to design and development, construction, integration, test, and operations of the LSST observatory system. The archive is maintained using the enterprise content management system DocuShare, which is accessible through a link on the project website www.project.lsst.org.
CC	Change Control
CI	Cyber Infrastructure
Center	An entity managed by AURA that is responsible for execution of a federally funded project
Change Control	The systematic approach to managing all changes to the LSST system, including technical data and policy documentation. The purpose is to ensure that no unnecessary changes are made, all changes are documented, and resources are used efficiently and appropriately.
DAC	Data Access Center
DM	Data Management
DMS	Data Management Subsystem
DR	Data Release
DRP	Data Release Production
Data Access Center	Part of the LSST Data Management System, the US and Chilean DACs will provide authorized access to the released LSST data products, software such as the Science Platform, and computational resources for data analysis. The US DAC also includes a service for distributing bulk data on daily and annual (Data Release) timescales to partner institutions, collaborations, and LSST Education and Public Outreach (EPO).
Data Management	The LSST Subsystem responsible for the Data Management System (DMS), which will capture, store, catalog, and serve the LSST dataset to the scientific community and public. The DM team is responsible for the DMS architecture, applications, middleware, infrastructure, algorithms, and Observatory Network Design. DM is a distributed team working at LSST and partner institutions, with the DM Subsystem Manager located at LSST headquarters in Tucson.
Data Management Subsystem	The subsystems within Data Management may contain a defined combination of hardware, a software stack, a set of running processes, and the people who manage them: they are a major component of the DM System operations. Examples include the 'Archive Operations Subsystem' and the 'Data Processing Subsystem'".'
Data Management System	The computing infrastructure, middleware, and applications that process, store, and enable information extraction from the LSST dataset; the DMS will process peta-scale data volume, convert raw images into a faithful representation of the universe, and archive the results in a useful form. The infrastructure layer consists of the computing, storage, networking hardware, and system software. The middleware layer handles distributed processing, data access, user interface, and system operations services. The applications layer includes the data pipelines and the science data archives' products and services.
Data Release	The approximately annual reprocessing of all LSST data, and the installation of the resulting data products in the LSST Data Access Centers, which marks the start of the two-year proprietary period.
Data Release Production	An episode of (re)processing all of the accumulated LSST images, during which all output DR data products are generated. These episodes are planned to occur annually during the LSST survey, and the processing will be executed at the Archive Center. This includes Difference Imaging Analysis, generating deep Coadd Images, Source detection and association, creating Object and Solar System Object catalogs, and related metadata.
DocuShare	The trade name for the enterprise management software used by LSST to archive and manage documents

Document	Any object (in any application supported by DocuShare or design archives such as PDMWorks or GIT) that supports project management or records milestones and deliverables of the LSST Project
EPO	Education and Public Outreach
FITS	Flexible Image Transport System
ForcedSource	DRP table resulting from forced photometry.
HSC	Hyper Suprime-Cam
Handle	The unique identifier assigned to a document uploaded to DocuShare
IRSA	Infrared Science Archive
LDM	LSST Data Management (Document Handle)
LSST	Large Synoptic Survey Telescope
NCSA	National Center for Supercomputing Applications
NSF	National Science Foundation
Object	In LSST nomenclature this refers to an astronomical object, such as a star, galaxy, or other physical entity. E.g., comets, asteroids are also Objects but typically called a Moving Object or a Solar System Object (SSObject). One of the DRP data products is a table of Objects detected by LSST which can be static, or change brightness or position with time.
Operations	The 10-year period following construction and commissioning during which the LSST Observatory conducts its survey
PDAC	Prototype Data Access Center
Project Manager	The person responsible for exercising leadership and oversight over the entire LSST project; he or she controls schedule, budget, and all contingency funds
QA	Quality Assurance
Qserv	LSST's distributed parallel database. This database server is used for collecting, storing, and serving LSST Data Release Catalogs and Project metadata, and is part of the Software Stack.
Quality Assurance	All activities, deliverables, services, documents, procedures or artifacts which are designed to ensure the quality of DM deliverables. This may include QC systems, in so far as they are covered in the charge described in LDM-622. Note that contrasts with the LDM-522 definition of "QA" as "Quality Analysis", a manual process which occurs only during commissioning and operations. See also: Quality Control.
RAM	Random Access Memory
RFC	Request For Comment
SDSS	Sloan Digital Sky Survey
Science Pipelines	The library of software components and the algorithms and processing pipelines assembled from them that are being developed by DM to generate science-ready data products from LSST images. The Pipelines may be executed at scale as part of LSST Prompt or Data Release processing, or pieces of them may be used in a standalone mode or executed through the LSST Science Platform. The Science Pipelines are one component of the LSST Software Stack.
Science Platform	A set of integrated web applications and services deployed at the LSST Data Access Centers (DACs) through which the scientific community will access, visualize, and perform next-to-the-data analysis of the LSST data products.
Scope	The work needed to be accomplished in order to deliver the product, service, or result with the specified features and functions
Sloan Digital Sky Survey	is a digital survey of roughly 10,000 square degrees of sky around the north Galactic pole, plus a 300 square degree stripe along the celestial equator.
Software Stack	Often referred to as the LSST Stack, or just The Stack, it is the collection of software written by the LSST Data Management Team to process, generate, and serve LSST images, transient alerts, and catalogs. The Stack includes the LSST Science Pipelines, as well as packages upon which the DM software depends. It is open source and publicly available.

Solar System Object	A solar system object is an astrophysical object that is identified as part of the Solar System: planets and their satellites, asteroids, comets, etc. This class of object had historically been referred to within the LSST Project as Moving Objects.
Source	A single detection of an astrophysical object in an image, the characteristics for which are stored in the Source Catalog of the DRP database. The association of Sources that are non-moving lead to Objects; the association of moving Sources leads to Solar System Objects. (Note that in non-LSST usage "source" is often used for what LSST calls an Object.)
Specification	One or more performance parameter(s) being established by a requirement that the delivered system or subsystem must meet
Stripe 82	A 2.5° wide equatorial band of sky covering roughly 300 square degrees that was observed repeatedly in 5 passbands during the course of the SDSS, In part for calibration purposes.
Subsystem	A set of elements comprising a system within the larger LSST system that is responsible for a key technical deliverable of the project.
Subsystem Manager	responsible manager for an LSST subsystem; he or she exercises authority, within prescribed limits and under scrutiny of the Project Manager, over the relevant subsystem's cost, schedule, and work plans
TB	TeraByte
US	United States
Validation	A process of confirming that the delivered system will provide its desired functionality; overall, a validation process includes the evaluation, integration, and test activities carried out at the system level to ensure that the final developed system satisfies the intent and performance of that system in operations
Verification	The process of evaluating the design, including hardware and software - to ensure the requirements have been met; verification (of requirements) is performed by test, analysis, inspection, and/or demonstration
astronomical object	A star, galaxy, asteroid, or other physical object of astronomical interest. Beware: in non-LSST usage, these are often known as sources.
calibration	The process of translating signals produced by a measuring instrument such as a telescope and camera into physical units such as flux, which are used for scientific analysis. Calibration removes most of the contributions to the signal from environmental and instrumental factors, such that only the astronomical component remains.
camera	An imaging device mounted at a telescope focal plane, composed of optics, a shutter, a set of filters, and one or more sensors arranged in a focal plane array.
flux	Shorthand for radiative flux, it is a measure of the transport of radiant energy per unit area per unit time. In astronomy this is usually expressed in cgs units: erg/cm2/s.
forced photometry	A measurement of the photometric properties of a source, or expected source, with one or more parameters held fixed. Most often this means fixing the location of the center of the brightness profile (which may be known or predicted in advance), and measuring other properties such as total brightness, shape, and orientation. Forced photometry will be done for all Objects in the Data Release Production.
metadata	General term for data about data, e.g., attributes of astronomical objects (e.g. images, sources, astroObjects, etc.) that are characteristics of the objects themselves, and facilitate the organization, preservation, and query of data sets. (E.g., a FITS header contains metadata).
monitoring	In DM QA, this refers to the process of collecting, storing, aggregating and visualizing metrics.
shape	In reference to a Source or Object, the shape is a functional characterization of its spatial intensity distribution, and the integral of the shape is the flux. Shape characterizations are a data product in the DIASource, DIAObject, Source, and Object catalogs.
stack	a grouping, usually in layers (hence stack), of software packages and services to achieve a common goal. Often providing a higher level set of end user oriented services and tools

transient	A transient source is one that has been detected on a difference image, but has not been associated with either an astronomical object or a solar system body.