



LARGE SYNOPTIC SURVEY TELESCOPE

Large Synoptic Survey Telescope (LSST)
Data Management

LSST Software Release Management

G. Comoretto, L. P. Guy, W. O'Mullane, K.-T. Lim, M. Butler, M.
Gelman, J.D. Swinbank

LDM-672

Latest Revision: 2019-07-25

Draft Revision NOT YET Approved - This LSST document has been approved as a Content-Controlled Document by the LSST DM Change Control Board. If this document is changed or superseded, the new document will retain the Handle designation shown above. The control is on the most recent digital document with this Handle in the LSST digital archive and not printed versions. Additional information may be found in the corresponding DM RFC. - **Draft Revision NOT YET Approved**

Abstract

This document outlines the policies and high level management approach for Large Synoptic Survey Telescope (LSST) Data Management (DM) software product releases.

Change Record

Version	Date	Description	Owner name
	2018-12-03	First draft	L. Guy, G. Comoretto
1.0	2019-07-25	First issue, approved on RFC-619	G. Comoretto

Document curator: Gabriele Comoretto

Document source location: <https://github.com/lst/lst/lst-672>

Version from source repository: 9072d9d

Draft



Contents

- 1 Introduction 1**
 - 1.1 Releases Status 1
 - 1.2 Definitions 2
- 2 Stakeholders Requirements 3**
 - 2.1 Release Requirements to meet LSST’s Operational Goals 3
 - 2.2 Release Requirements for Interacting with the Scientific Community 4
 - 2.3 Release Requirements for Operational Data Processing Centers 4
 - 2.4 Release Requirements for Observatory Operations 5
 - 2.5 Release Requirements for User-Facing Infrastructure 5
 - 2.6 Release Requirements for LSST Construction Activities 6
- 3 Consolidated Requirements 7**
 - 3.1 General Requirements 7
 - 3.1.1 Software Products Identification Requirement 7
 - 3.1.2 Software Release Documentation Requirement 7
 - 3.1.3 Software Release Test Requirement 7
 - 3.2 Stakeholders Requirements 7
 - 3.2.1 Release Schedule Requirement 8
 - 3.2.2 Patch Release Requirement 8
 - 3.2.3 Third-Party Software Inclusion Requirement 8
 - 3.2.4 Stable public APIs and Schemas Requirement 8
 - 3.2.5 License Requirement 8
 - 3.3 Requirements Summary Overview 8
- 4 Release Policy 10**
 - 4.1 Versioning Policy 10
 - 4.2 Release Schedule Policy 10
 - 4.3 Patch Releases and Backporting Policy 10
 - 4.4 Release Testing Policy 11



4.5 Release Note Policy	11
4.6 Third-Party Software Policy	11
4.7 Software License Policy	12
5 Applicability of policies	13
5.1 Non-Compliance	13
6 Release High Level Process	14
A References	15
B Glossary	16

Draft

LSST Software Release Management

1 Introduction

This document presents the release management approach for LSST DM software products.

First of all, in section 2, the release requirements for the various stakeholders are identified. The release requirements are then consolidated in section 3. These requirements are not formal project requirements, as given for example in the Data Management Subsystem (DMS) requirements specification LSE-61, but are nevertheless important to ensure the project's goals.

Based on the consolidated requirements, a set of policies are derived in section 4 and guidelines on their applicability is provided in section 5.1.

Finally section 6 gives a high level overview of the release process.

1.1 Releases Status

Currently, only the Science Pipelines product is released. Builds and releases are made on the following time-based cadence:

- Nightly builds
- Weekly builds
- Official releases every 6 months

The time needed to consolidate an official release from a weekly build is considerable. Usually 2 or 3 weeks are sufficient but in some cases it may take more than a month. Consequently, by the time a release becomes available to the users, it is already old. For this reason, users generally prefer to work with weekly builds that are sufficiently stable and include all new functionalities completed in the last week.

The Science Pipelines release checklist is documented in SQR-016. The technical note DMTN-106 generalizes the process and summarizes the technical problems that need to be solved to make the procedure applicable to other software products.

1.2 Definitions

The relevant definition to be considered when working on release policy and process are given in DMTN-106, section 2.

Draft

2 Stakeholders Requirements

This section address the questions:

Why are DM software releases needed? Who is requesting them?

The classic answer to the second questions is that stakeholders request releases for various reasons.

The following subsections summarize the release requirements on DM software products from the different stakeholders' points of view.

It is important to identify these requirements and the corresponding policies during the construction phase, in order to have them consolidated when operations start.

During operations, some of the construction-era subsystems will no longer exist. For example, the Data Management Subsystem will have been disbanded, but the DM products will continue to be used and developed under the operational project structure. The requirements and policies defined here will still be applicable since many of the stakeholders will still expect software releases to be managed following the process consolidated during construction.

2.1 Release Requirements to meet LSST's Operational Goals

A primary reason to make software releases is provide a mechanism by which the versions of the software used to generate LSST's data products, and hence the provenance of those data products, may be managed and controlled.

During the operational era, the Data Management subsystem will have disbanded. However, it is important that best practices for release management are established now and can be supplied to the LSST Operations project. In particular, the process should deliver software releases which:

- Enable data processing to be carried out according to the project requirements. This implies that the process should:
 - fulfill the LSST construction requirements as defined in §2.6
 - fulfill the operational requirements as defined in §2.3 and §2.4

- deliver documented functionality as described by construction-era milestones and emergent operational requirements.
- Enable the scientific community to contribute to the project. This implies that the process should:
 - fulfill the requirements from the scientific community as defined in §2.2
 - include third-party software provided by the scientific community when this is beneficial to the project outcome
 - carry an appropriate source code license.

2.2 Release Requirements for Interacting with the Scientific Community

In preparation for working with the LSST data products and software during operations, several LSST science collaborations have begun using the DM software to run data challenges using precursor data or simulations, and to do performance studies. These activities effectively increase the number of beta-testers of DM software products, providing valuable feedback to DM on the state of the system.

In order to work effectively with the DM software while it is still under development, the scientific community requires:

- timely access to new functionality,
- stable public APIs and schemas in order to build software for user-generated analysis workflows,
- select bug fixes and other back-ported to the current stable version of the software,
- select software provided by external contributors be included in the software release or distribution (due to the collaborative nature of the project).

2.3 Release Requirements for Operational Data Processing Centers

The LSST Data Facility (LDF) will be responsible for generating the data products as specified in the [DPDD](#) during commissioning and operations. LDF policies require officially released software to be used in production for the various operational activities. Software releases

will be run in production at the National Center for Supercomputing Applications (NCSA), CC-IN2P3, in Chile, and possibly at independent Data Access Centers (iDACs).

Software releases are required to be fully tested and well documented.

Release frequencies will depend on the processing type:

- Rapidly responding to issues involving Prompt Processing will require releases to be made available rapidly (i.e., on a timescale of no more than days) and frequently (perhaps daily). It may also be necessary to provide patch releases addressing specific problems during the observing night (this would require approval from the Associate Director (AD) for Science Operations).
- Data Release Production (DRP) processing must be stable for long periods (currently, processing is expected to take 9 months). Before such a long processing run the release must be very well tested; during the run, any updates must be extremely tightly controlled.
- DM software involved in image acquisition, either at the LDF or on the mountain, and including the Header Service, also require strict change control. Releases for this could be on monthly or even longer timescales; however, if there is a problem a patch will be needed immediately.

In short, patch releases will need to be provided with a frequency that depends on the type of processing and on the urgency of the problems to be fixed.

2.4 Release Requirements for Observatory Operations

Other parts of the LSST Observatory are also expected to be consumers of DM software products. For example, the telescope control software makes extensive use of code provided by DM. These must be addressed on a case-by-case basis with the consumers of the DM-provided software.

2.5 Release Requirements for User-Facing Infrastructure

A significant subset of DM software products are used to provide services to LSST science users and staff that are not directly used to generate LSST science data products. An example of this is the software that implements the LSST Science Platform (LSP) [LSE-319].

Releases of this type of software are typically on their own cadence and need to be adequately tested before deployment to ensure a stable infrastructure. The releases may be tied to processing milestones if the services or features thereof are required for the processing (e.g. functionality of the workflow service may be required for Data Release processing and features in the LSP may be needed for Quality Assurance (QA) of data products).

Patch releases may need to be provided depending on the urgency and severity of the problems to be fixed.

2.6 Release Requirements for LSST Construction Activities

This includes activities done in preparation for operations, such as commissioning, large scale integration/validation test campaigns, etc. These activities should use use, as far as is possible, officially released software.

In some cases however, it is necessary to use unreleased software, such as release candidates or stable builds. In all cases, the software used must be clearly identified, e.g. by Git SHA1, and distribution and deployment must be strictly controlled.

During construction software releases are related to construction milestones. In most of cases, this implies releases should be made on a regular cadence — for example, every 6 months.

3 Consolidated Requirements

The main purpose of this section is to identify all possible release requirements. In the first subsection a few general requirements are given. The second subsection summarizes the requirements given by the stakeholders.

A summary overview of the requirements per stakeholder is given at the end of the section.

3.1 General Requirements

The following general requirements are needed in order to properly implement the release process.

3.1.1 Software Products Identification Requirement

All software products shall be clearly and unequivocally identifiable in the source repository (currently, GitHub) and documented¹. This requirement needs to be fulfilled in order to ensure the applicability of the release policy and process. If the software products are not properly identified, it will not be possible to do releases.

3.1.2 Software Release Documentation Requirement

All software releases shall be properly documented with a software release note.

3.1.3 Software Release Test Requirement

A software release should be fully tested before making it available for use. The test should be documented in a test report.

3.2 Stakeholders Requirements

The following list of requirements is derived from the above section 2.

¹ DMTN-106 §2.2 (see 1.2) provides a software product definition that can be used as a starting point to identify the DM software products. The DM Product Tree is provided with LDM-294 and may be accessed directly at <https://ldm-294.lsst.io/ProductTreeLand.pdf>.

3.2.1 Release Schedule Requirement

Releases on a software product shall be scheduled in advance.

Two types of release schedule can be identified:

- Functional Based Release Schedule: a release shall provide specific functionality.
- Time-Based Release Schedule: a release shall be provided on a specific date or cadence.

In both cases, releases may be associated with project milestones. Additional releases may be made available upon request to the DM-CCB using the “RFC” mechanism as described in §6.

3.2.2 Patch Release Requirement

It shall be possible to back-port a fix on a stable release and provide a patch release including only the backported fix.

3.2.3 Third-Party Software Inclusion Requirement

It shall be possible to include a software package provided by a third-party contributor in a software product release or distribution.

3.2.4 Stable public APIs and Schemas Requirement

Public APIs and schemas shall follow a well-defined deprecation mechanism in order to give time to the stakeholder to adapt to the new API.

3.2.5 License Requirement

DM software shall be released with an appropriate license that permits distribution to, use by, and contributions from, external collaborators.

3.3 Requirements Summary Overview

The following table gives an overview of the release requirements applicable for each stakeholder.



	<i>SW Ident.</i>	<i>Release Doc.</i>	<i>Release Test</i>	<i>Funct. Based</i>	<i>Time Based</i>	<i>Patch</i>	<i>3rd Party SW</i>	<i>Stable API</i>	<i>License</i>
LSST Project (§2.1)	✓	✓	✓	✓	✓		✓		✓
Scientific Community (§2.2)	✓	✓			✓	✓		✓	
Proc. Centers (§2.3)	✓	✓	✓	✓		✓			
Other Ops. (§2.4)	✓	✓	✓	✓		✓			
Infrastructure (§2.5)	✓	✓	✓		✓	✓		✓	
Construction (§2.6)	✓	✓	✓	✓	✓	✓		✓	

4 Release Policy

The following policies are derived from the consolidated requirements described in the previous section 3.

4.1 Versioning Policy

The DM release versioning shall follow Semantic Versioning² as described in DMTN-106, section 3.3.

This policy partially addresses the requirement for stable APIs (§3.2.4).

4.2 Release Schedule Policy

Major and minor releases should be scheduled accordingly to the requirements of each stakeholder.

Each release shall be made following a release plan, which provides the following information:

- when the release is expected;
- the corresponding milestone or RFC;
- if it is a time-based release: specify the release cadence (for example, every 6 months);
- if it is a functionality based release: specify which features shall be included in the release.

This policy addresses requirement §3.2.1.

4.3 Patch Releases and Backporting Policy

Patch releases shall be scheduled, tested and approved by the DMCCB wherever possible. However, the DM Project Manager can authorize urgent releases if required.

The mechanisms by which fixes are back-ported are a matter of development practice, rather than policy; as such, they are addressed in the Developer Guide [6].

²<https://semver.org/>

This policy addresses requirement §3.2.2 .

4.4 Release Testing Policy

All releases must be fully tested and the results recorded according to standard DM practice³.

The scope of the test is to ensure that:

- All planned functionality is provided and working;
- There is no regression in functionality or performance relative to previous releases.

This policy addresses requirement §3.1.3 .

Note that, in some cases, and at the discretion of the DM-CCB, it may be appropriate to relax this policy and release based solely on successful execution of unit and/or integration tests.

4.5 Release Note Policy

Each release shall be documented by an accompanying set of release notes.

DMTN-106 §3.2 provides a definition of software release note that can be used as is or tailed depending on the needs (see also §1.2 of this document).

This policy addresses requirement §3.1.2.

4.6 Third-Party Software Policy

It shall be possible to include in a software product or distribution a software package provided by an external collaborator. The use of all third-party packages shall be approved by the DMCCB using the RFC mechanism. External software may be included in the following ways:

- The package may be included in the standard environment used to execute DM code⁴

³At time of writing, this means that test activities are managed using the Adaptavist Test Management framework for Jira.

⁴At time of writing, this environment is based on Conda.

- this implies that the third party software is already packaged for that environment, or, alternatively, that DM must itself provide such a package.
- the software will be provided *as is* and DM is not responsible for it
- include the software as part of a DM software distribution, as an additional package⁵
 - the software is provided *as is* and DM is not responsible for it
- Create a new DM-provided software product to contain the third-party software
 - This is appropriate if DM-specific patches or customization is required.
 - DM takes responsibility for the customized package.

In all cases, a member of the DM team shall be identified as point of contact and reference for the external package.

This policy addresses requirement §3.2.3.

4.7 Software License Policy

All DM-produced software must carry an Open Source Initiative (OSI)-approved license⁶. Specific requirements for particular aspects of the DM codebase are addressed in the Developer Guide [6].

⁵At time of writing, this would mean packaging for EUPS.

⁶<https://opensource.org/licenses>

5 Applicability of policies

The policies described in §4 of this document are mandatory for all DM software products, as defined in LDM-294, except where otherwise specified.

5.1 Non-Compliance

Individual T/CAMs or Product Owners may request that the DM-CCB issue exemptions from specific aspects of policy for particular software products. The DM-CCB is responsible for ensuring that the resulting system still meets the needs of applicable stakeholders, and for ensuring that the overall integrity of the DM system is maintained.

Draft

6 Release High Level Process

The DM-CCB maintains the release plan, LDM-564, synchronized with the project milestones.

⁷

Any unscheduled release — major, minor or patch, and including a request to backport fixes to an earlier release — must be requested of the DM-CCB using a Jira issue of type “RFC”. The issue shall describe:

- The justification for the release.
- The date by which the release should be available.
- A list of specific functionality and/or bug fixes which should be contained in the release, specified in terms of Jira tickets.

The DM-CCB will assess the release request within one week. If the release is urgent, DM-CCB may assess it within 24 hours. The DM-CCB will approve or reject the proposed release and add a comment to the RFC with the reason of rejection or, in case of approval, with the following information:

- The release identifier (version number M.N.p).
- The estimated release date.
- The list of Jira issues that will be included.

In the case an immediate fix is required for a critical operations activity, a quick decision can be taken by the DM Project Manager and a requested fix can be implemented and released as rapidly as possible. However, an Request For Comment (RFC) has to be filed a posteriori, and the DM-CCB is required to review it.

⁷As of June 2019 the release plan needs to be reviewed and the release milestones listed therein need to be made consistent with the scope of the document. Issue DM-17001 is tracking this activity.

A References

References

- [1] **[DMTN-106]**, Comoretto, G., 2019, *DM Release Process*, DMTN-106, URL <http://DMTN-106.lsst.io>
- [2] **[LSE-61]**, Dubois-Felsmann, G., Jenness, T., 2018, *LSST Data Management Subsystem Requirements*, LSE-61, URL <https://ls.st/LSE-61>
- [3] **[SQR-016]**, Economou, F., 2018, *Stack release playbook*, SQR-016, URL <https://sqr-016.lsst.io>
- [4] **[LSE-319]**, Jurić, M., Ciardi, D., Dubois-Felsmann, G., 2017, *LSST Science Platform Vision Document*, LSE-319, URL <https://ls.st/LSE-319>
- [5] **[LSE-163]**, Jurić, M., et al., 2017, *LSST Data Products Definition Document*, LSE-163, URL <https://ls.st/LSE-163>
- [6] LSST Data Management, LSST DM Developer Guide, URL <https://developer.lsst.io/>
- [7] **[LDM-564]**, O'Mullane, W., Economou, F., Jenness, T., Loftus, A., 2018, *Data Management Software Releases for Verification/Integration*, LDM-564, URL <https://ls.st/LDM-564>
- [8] **[LDM-294]**, O'Mullane, W., Swinbank, J., Jurić, M., DMLT, 2018, *Data Management Organization and Management*, LDM-294, URL <https://ls.st/LDM-294>

B Glossary

AD Associate Director.

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 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..

DM Data Management.

DMS Data Management Subsystem.

DRP Data Release Production.

LDF LSST Data Facility.

LSP LSST Science Platform.

LSST Large Synoptic Survey Telescope.

NCSA National Center for Supercomputing Applications.

Operations The 10-year period following construction and commissioning during which the LSST Observatory conducts its survey.

QA Quality Assurance.

RFC Request For Comment.

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..