



Vera C. Rubin Observatory
Data Management

LDM-503-13a: Science Pipelines Fall 2020 Release Test Plan and Report

Yusra AlSayyad

DMTR-302

Latest Revision: 2021-04-30

DRAFT



Abstract

This is the test plan and report for **Science Pipelines Fall 2020 Release** (LDM-503-13a), an LSST milestone pertaining to the Data Management Subsystem.

This document is based on content automatically extracted from the Jira test database on 2021-04-30 . The most recent change to the document repository was on 2023-07-10.

Draft

Change Record

Version	Date	Description	Owner name
	2021-04-08	First draft	Yusra AlSayyad
1.0	2021-04-26	Test plan completed. DM-16200	Yusra AlSayyad
2.0	2021-04-30	Test campaign completed. DM-17129	Yusra AlSayyad

Document curator: Yusra AlSayyad

Document source location: <https://github.com/lstt-dm/DMTR-302>

Version from source repository: 9aebb04

Draft

Contents

1 Introduction	1
1.1 Objectives	1
1.2 Scope	2
1.3 System Overview	2
1.4 Applicable Documents	2
1.5 Document Overview	2
1.6 References	3
2 Test Plan Details	4
2.1 Data Collection	4
2.2 Verification Environment	4
2.3 Related Documentation	4
2.4 PMCS Activity	4
3 Personnel	5
4 Test Campaign Overview	6
4.1 Summary	6
4.2 Overall Assessment	6
4.3 Recommended Improvements	6
5 Detailed Test Results	7
5.1 Test Cycle LW-C171	7
5.1.1 Software Version/Baseline	7
5.1.2 Configuration	8
5.1.3 Test Cases in LW-C171 Test Cycle	8
5.1.3.1 LVV-T362 - Installation of the LSST Science Pipelines Payloads	8
5.1.3.2 LVV-T1601 - Science Pipelines available on developer hardware	12
5.1.3.3 LVV-T363 - Science Pipelines Release Documentation	15

A Documentation	19
B Acronyms used in this document	19

Draft

LDM-503-13a: Science Pipelines Fall 2020 Release Test Plan and Report

1 Introduction

1.1 Objectives

This test plan checks for the successful release of the Fall 2020 release of the LSST Science Pipelines version 21.0.0

It will demonstrate that:

- The release has been tagged, built and made available through standard distribution channels;
- Release documentation, including release notes and a characterization report, are available on the Pipelines documentation website (<https://pipelines.lsst.io/>);
- An end-user can follow standard instructions to install the release onto some representative system;
- The release is installed into the “shared stack” on the lsst-devl shared developer systems at the Rubin Data Facility;
- The `pipelines_check` test package executes successfully in the context of the release.

This test plan does not, in itself, verify the scientific integrity or algorithmic correctness of the release, beyond checking that defined procedures for checking basic correctness and characterizing algorithmic performance have been followed.

1.2 Scope

The overall strategy for testing and verification within Rubin Data Management is described in LDM-503.

This test plan specifically verifies successful completion of milestone LDM-503-13a, which refers to the Fall 2020 release of the LSST Science Pipelines.

1.3 System Overview

The LSST Science Pipelines comprise the scientific algorithms that will be used to process data for the Rubin Observatory Legacy Survey of Space and Time (Rubin's LSST), arranged into executable pipelines by means of the LSST "task" framework. They also include execution middleware that is common across execution environments (for example, the "Data Butler" I/O abstraction is included, but schedulers or workflow management for specific clusters is not), and "camera packages", which adapt and configure the algorithms for use with specific instrumentation.

1.4 Applicable Documents

LDM-503 Data Management Test Plan
LDM-151 Data Management Science Pipelines Design
LSE-61 Data Management System Requirements

1.5 Document Overview

This document was generated from Jira, obtaining the relevant information from the LVV-P82 Jira Test Plan and related Test Cycles (LVV-C171).

Section 1 provides an overview of the test campaign, the system under test (Science Pipelines SW), the applicable documentation, and explains how this document is organized. Section 2 provides additional information about the test plan, like for example the configuration used

for this test or related documentation. Section 3 describes the necessary roles and lists the individuals assigned to them.

Section 4 provides a summary of the test results, including an overview in Table 2, an overall assessment statement and suggestions for possible improvements. Section 5 provides detailed results for each step in each test case.

The current status of test plan LVV-P82 in Jira is **Approved**.

1.6 References

- [1] **[DMTN-140]**, Comoretto, G., 2021, *Documentation Automation for the Verification and Validation of Rubin Observatory Software*, DMTN-140, URL <https://dmtn-140.lsst.io/>, Vera C. Rubin Observatory Data Management Technical Note
- [2] **[DMTN-178]**, Comoretto, G., 2021, *Docsteady Usecases for Rubin Observatory Constructions*, DMTN-178, URL <https://dmtn-178.lsst.io/>, Vera C. Rubin Observatory Data Management Technical Note
- [3] **[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
- [4] **[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
- [5] **[LSE-160]**, Selvy, B., 2013, *Verification and Validation Process*, LSE-160, URL <https://ls.st/LSE-160>
- [6] **[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

2 Test Plan Details

2.1 Data Collection

Observing is not required for this test campaign.

2.2 Verification Environment

Several of the tests described in this plan are agnostic of environment: they involve checking that certain content has been properly published. This can be performed from any internet-connected system with a web browser, and will, in this case, likely be executed from the tester's laptop.

Where tests require installation or execution of specific Science Pipelines components, this will be carried out on the "lsst-devl" shared developer infrastructure at the Rubin Data Facility. This infrastructure provides a number of powerful (24core, 256GB RAM) systems accessible to Rubin developers. At time of writing, they are running CentOS 7.9.2009; in practice, any version of CentOS (or a similar operating system) is appropriate for this test plan, as long as it complies with the published installation prerequisites of the LSST Science Pipelines.

2.3 Related Documentation

No additional documentation provided.

2.4 PMCS Activity

Primavera milestones related to the test campaign:

- LDM-503-13a

3 Personnel

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

T. Plan LVW-P82 owner:		Yusra AlSayyad	
T. Cycle LVW-C171 owner:		Yusra AlSayyad	
Test Cases	Assigned to	Executed by	Additional Test Personnel
LVW-T362	Yusra AlSayyad	Yusra AlSayyad	
LVW-T1601	Yusra AlSayyad	Yusra AlSayyad	
LVW-T363	Yusra AlSayyad	Yusra AlSayyad	

Draft

4 Test Campaign Overview

4.1 Summary

T. Plan LVV-P82:		LDM-503-13a: Science Pipelines Fall 2020 Release		Approved
T. Cycle LVV-C171:		LDM-503-13a: Science Pipelines Fall 2020 Release		Done
Test Cases	Ver.	Status	Comment	Issues
LVV-T362	1	Pass w/ Deviation	Test completed with deviation; see linked issue	DM-29904
LVV-T1601	1	Pass w/ Deviation	Test completed with deviation	DM-28988 DM-29963
LVV-T363	1	Pass	Test is completed	

Table 2: Test Campaign Summary

4.2 Overall Assessment

All the tests described in this plan were completed successfully.

Two minor deviations were encountered.

(1) This release marks the full replacement of the `lsst_dm_stack_demo` package with the `pipelines_check` package. The instructions on the `pipelines_check` README were missing a step if followed directly from the stack installation procedure in the previous test case step. This was resolved on DM-29904.

(2) The shared stack maintained for developers at the Data Facility included a tagged version for the v21 release candidate, it did not contain a tagged version for the final v21 release. This distinction is minor because the two versions are equivalent, but this was resolved on DM-29963.

4.3 Recommended Improvements

No recommendations beyond the tickets filed

5 Detailed Test Results

5.1 Test Cycle LVV-C171

Open test cycle *LDM-503-13a: Science Pipelines Fall 2020 Release* in Jira.

Test Cycle name: LDM-503-13a: Science Pipelines Fall 2020 Release

Status: Done

This test cycle describes tests performed on the Science Pipelines Fall 2020 (v21.0.0) release, ensuring that the release is properly identified, documented, distributed, installable and tested.

5.1.1 Software Version/Baseline

A web browser is required for inspecting release artifacts published to the web.

Testing the software installation procedures, and demonstrating that the release's integration tests can be executed successfully, require a supported operating system with the documented prerequisites of the release installed. This will be carried out on the "lsst-dev" shared developer systems at the LSST Data Facility.

At time of writing, Data Facility systems run CentOS Linux release 7.9.2009, and it is anticipated that this will be a supported platform for the Science Pipelines release.

Science Pipelines prerequisites are documented at pipelines.lsst.io; all of these must be installed.

It is possible that the software release will involve a reorganization of documentation describing prerequisites; in this case, the documentation corresponding to the new release must be consulted.

5.1.2 Configuration

No specific configuration is required beyond the availability of test systems with the prerequisite software, described above, installed.

5.1.3 Test Cases in LVV-C171 Test Cycle

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

Version 1. Open *LVV-T362* test case in Jira.

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 a 2018-era packaging of the Science Pipelines software, in which all the above payloads are represented by a single “meta-package”, `lsst_distrib`.

Preconditions:

Execution status: **Pass w/ Deviation**

Final comment:

Test completed with deviation; see linked issue

Issues found during the execution of LVV-T362 test case:

- DM-29904 Instruction on pipelines_check README are incomplete

Detailed steps results:

Step 1	Step Execution Status: Pass
<hr/>	
Description	
The LSST Science Pipelines, described by the <code>lsst_distrib</code> 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.	
<hr/>	
Expected Result	
Detailed output will depend on the installation method chosen, but will confirm the successful installation of the Science Pipelines.	
<hr/>	
Actual Result	
The Science Pipelines were installed on the shared server <code>lsst-devl01.ncsa.illinois.edu</code> which is running CentOS 7.9.2009, using the procedure described in https://pipelines.lsst.io/install/newinstall.html	

Following the instructions in step 2 and 3, I made an installation directory and ran `newinstall.sh`:

```
mkdir -p lsst_stack
cd lsst_stack
curl -OL https://raw.githubusercontent.com/lsst/lsst/21.0.0/scripts/newinstall.sh
bash newinstall.sh -ct
```

answering “yes” when prompted to install miniconda:

```
> Would you like us to install Miniconda distribution (if unsure, say yes)?yes
```

Then following step 4, I installed the Science Pipelines:

```
source loadLSST.bash # for bash
eups distrib install -t v21_0_0 lsst_distrib
```

```
curl -sSL https://raw.githubusercontent.com/lsst/shebangtron/master/shebangtron | python
setup lsst_distrib
```

Step 2 Step Execution Status: **Pass**

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.

Actual Result

```
(lsst-scipipe-cb4e2dc) [yusra@lsst-devl01 lsst_stack]$ source /project/yusra/LVV-C171/TVW-T362/lsst_stack/loadLSST.bash
(lsst-scipipe-cb4e2dc) [yusra@lsst-devl01 lsst_stack]$ setup lsst_distrib
(lsst-scipipe-cb4e2dc) [yusra@lsst-devl01 lsst_stack]$ eups list -s lsst_distrib
21.0.0+973e4c9e85 current v21_0_0 setup
```

Step 3 Step Execution Status: **Pass**

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 the version of the Science Pipelines under test should be chosen.

Expected Result

Depends on the tool selected by the user for downloading.

Actual Result

The 21.0.0 pipelines_check was downloaded from https://github.com/lsst/pipelines_check/releases/tag/21.0.0 by executing the command:

```
(lsst-scipipe-cb4e2dc) [yusra@lsst-devl01]$ curl -OL https://github.com/lsst/pipelines_check/archive/refs/tags/21.0.0.tar.gz
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 138 100 138 0 0 862 0 --:--:-- --:--:-- --:--:-- 862
100 25.8M 0 25.8M 0 0 13.6M 0 --:--:-- 0:00:01 --:--:-- 16.8M
```

Step 4 Step Execution Status: **Pass**

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.

Actual Result

The demo package was uncompressed by running:

```
$ tar -xzf 21.0.0.tar.gz
```

There is now an uncompressed directory called `pipelines_check-21.0.0`.

Step 5 Step Execution Status: **Pass w/ Deviation**

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.

Actual Result

The pipeline_check completed successfully with a minor deviation of the instructions on the package README.

The instructions for execution were to

```
$ source $LSST_HOME/loadLSST.sh$ setup -r . $ ./bin/run_demo.sh
```

loadLSST is spelled loadLSST.bash for bash, and the ‘setup’ command requires either the ‘-j’ argument or an extra stack installation step of cloning the v21_0_0 tag as current first:

```
(lsst-scipipe-cb4e2dc) [yusra@lsst-dev101 pipelines_check-21.0.0]$ setup -r .
setup: in file /project/yusra/LVV-C171/TVV-T362/pipelines_check-21.0.0/ups/pipelines_check.table: Product pipe_tasks not found
(lsst-scipipe-cb4e2dc) [yusra@lsst-dev101 pipelines_check-21.0.0]$ eups list -s pipe_tasks
21.0.0+44ca056b81 v21_0_0 setup
(lsst-scipipe-cb4e2dc) [yusra@lsst-dev101 pipelines_check-21.0.0]$ eups tags --clone v21_0_0 current
(lsst-scipipe-cb4e2dc) [yusra@lsst-dev101 pipelines_check-21.0.0]$ setup -r .
(lsst-scipipe-cb4e2dc) [yusra@lsst-dev101 pipelines_check-21.0.0]$ ./bin/run_demo.sh
+ '[' !' -f DATA_REPO/butler.yaml ']'
+ butler create DATA_REPO
+ butler register-instrument DATA_REPO lsst.obs.subaru.HyperSuprimeCam
+ '[' !' -d DATA_REPO/HSC/calib ']'
+ butler import DATA_REPO /project/yusra/LVV-C171/TVV-T362/pipelines_check-21.0.0/input_data --export-file /project/yusra/LVV-C171/TVV-T362/pipelines_check-21.0.0/output_data
<log snipped>
INFO 2021-04-26T23:02:25.852-0500 calibrate (calibrate:{band: r, instrument: HSC, detector: 10, physical_filter: HSC-R, visit_id: 1})
+ pipetask qgraph -b DATA_REPO/butler.yaml --input HSC/calib,HSC/raw/all,refcats -p /project/yusra/LVV-C171/TVV-T362/lsst_s
ctrl.mpexec.cmdLineFwk INFO: QuantumGraph contains 3 quanta for 3 tasks
(lsst-scipipe-cb4e2dc) [yusra@lsst-dev101 pipelines_check-21.0.0]$
```

5.1.3.2 LVV-T1601 - Science Pipelines available on developer hardware

Version 1. Open *LVV-T1601* test case in Jira.

This test will check that a given release of the LSST Science Pipelines is available for use in a “shared stack” on developer infrastructure.

Preconditions:

Execution status: **Pass w/ Deviation**

Final comment:

Test completed with deviation

Issues found during the execution of LVV-T1601 test case:

- DM-28988 Update <https://developer.lsst.io/services/software.html#software-stack>
- DM-29963 Build version 21.0.0 in the shared stack

Detailed steps results:

Step 1	Step Execution Status: Pass
<hr/> Description	
Consult the LSST Developer Guide (http://developer.lsst.io/) to establish:	
<ul style="list-style-type: none">• An appropriate hostname and login instructions for the shared developer infrastructure at the LSST Data Facility;• Instructions for initializing the shared stack on the developer infrastructure.	
<hr/> Expected Result	
The Developer Guide clearly presents information about connecting to and using shared infrastructure.	
<hr/> Actual Result	
<ul style="list-style-type: none">• Instructions for logging into the head nodes <code>lsst-devl0*</code> at the NCSA development cluster are available at https://developer.lsst.io/services/lsst-login.html	

- Instructions for accessing the shared stack are available at <https://developer.lsst.io/services/software.html#software-stack>

Step 2 Step Execution Status: **Pass**

Description

Connect to the shared infrastructure following the Developer Guide instructions.

Expected Result

A shell prompt on a shared machine.

Actual Result

login successful and a shell prompt is available.

Step 3 Step Execution Status: **Pass**

Description

Initialize the LSST environment following the Developer Guide instructions.

Example Code

```
source /software/lsstsw/stack/loadLSST.bash
```

Expected Result

No errors are shown.

Actual Result

The shared stack for the v21 release is available at `/software/lsstsw/stack_20200922`. The command executed was:

```
source /software/lsstsw/stack_20200922/loadLSST.bash
```

Step 4 Step Execution Status: **Pass w/ Deviation**

Description

List available software products using EUPS, and check that the release under test is available.

Example Code

```
eups list lsst_distrib | grep vXX_XX_XX # replace XX_XX_XX with the version number of the release under test
```

Expected Result

The provided version number should be available in the shared stack.

Actual Result

There is no v21_0_0 tag in the shared stack, however developers have been using the tag for release candidate v21_0_0_rc1 instead when a v21 stack is needed. v21_0_0_rc1 became v21_0_0, and they are equivalent:

```
(lsst-scipipe) [yusra@lsst-dev101]$ eups list lsst_distrib | grep v21_0_0
20.0.0-3-gdc8c9d3+973e4c9e85 v21_0_0_rc1
```

5.1.3.3 LVV-T363 - Science Pipelines Release Documentation

Version 1. Open *LVV-T363* test case in Jira.

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.

Preconditions:

Execution status: **Pass**

Final comment:

Test is completed

Detailed steps results:

Step 1 Step Execution Status: **Pass**

Description

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

Expected Result

The website is displayed.

Actual Result

The website <https://pipelines.lsst.io> is displayed

Step 2 Step Execution Status: **Pass**

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.

Actual Result

<https://pipelines.lsst.io> is for the release under test and clearly states: "This documentation covers version **v21_0_0**" It was also simple to navigate to "Other Versions", and click on v21_0_0 and land on the permanent page address for this release: https://pipelines.lsst.io/v/v21_0_0/index.html

Step 3 Step Execution Status: **Pass**

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.

Actual Result

The documentation for the following are all available on the front page or linked from the front page:

- Release notes: https://pipelines.lsst.io/v/v21_0_0/releases/v21_0_0.html#release-latest
- Installation instructions: https://pipelines.lsst.io/v/v21_0_0/index.html#installation
- Getting started information: https://pipelines.lsst.io/v/v21_0_0/index.html#getting-started

Step 4 Step Execution Status: **Pass**

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.

Actual Result

The Characterization Report for this release Characterization Metric Report (DMTR-281) is clearly linked from the Release Notes: https://pipelines.lsst.io/v/v21_0_0/releases/v21_0_0.html#release-latest

Step 5 Step Execution Status: **Pass**

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.

Actual Result

The Characterization Report includes the expected metrics.

Draft

A Documentation

The verification process is defined in LSE-160. The use of Docsteady to format Jira information in various test and planing documents is described in DMTN-140 and practical commands are given in DMTN-178.

B Acronyms used in this document

Acronym	Description
DM	Data Management
DMTN	DM Technical Note
DMTR	DM Test Report
DRP	Data Release Production
EUPS	Extended Unix Product System
HSC	Hyper Suprime-Cam
LDM	LSST Data Management (Document Handle)
LPM	LSST Project Management (Document Handle)
LSE	LSST Systems Engineering (Document Handle)
LSST	Legacy Survey of Space and Time (formerly Large Synoptic Survey Telescope)
LVV	LSST Verification and Validation
NCSA	National Center for Supercomputing Applications
PMCS	Project Management Controls System
RAM	Random Access Memory