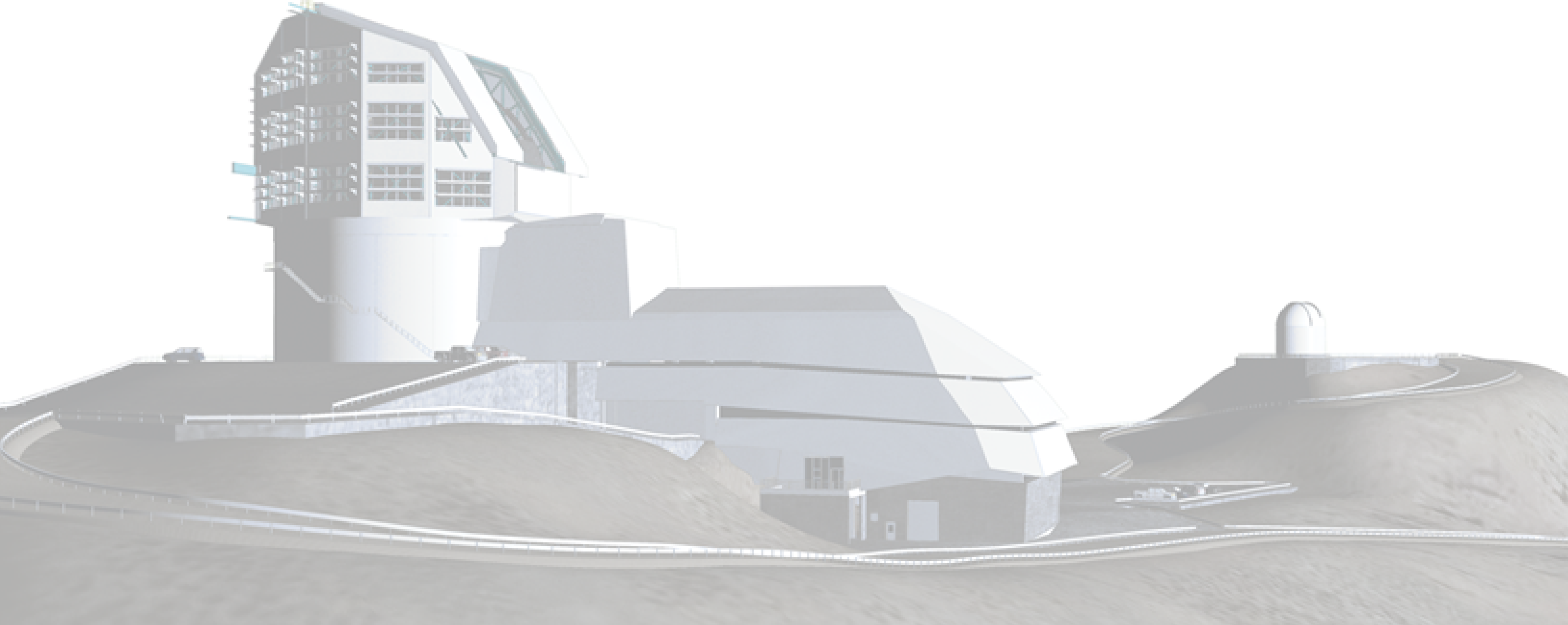
Vera C. Rubin Observatory
Systems Engineering

# Integration Milestones

Robert Lupton

SITCOMTN-006

Latest Revision: 2020-12-10



Rubin Observatory

## Abstract

A proposal for ways to work on predominantly software subsystem commissioning, building on the Rubin AuxTel (née auxTel) experience

Rubin Observatory

## Change Record

| Version | Date | Description | Owner name |
|---|---|---|---|
| 1 | YYYY-MM-DD | Unreleased. | Robert Lupton |

*Document source location:* `https://github.com/lsst-sitcom/sitcomtn-006`

Rubin Observatory

# Contents

Rubin Observatory

# Integration Milestones

## 1 Introduction

The Rubin construction project is composed of three main subsystems: DM, Camera, and T&S.[1] but in operations there is only one survey. The task of merging the projects is not easy, especially when some of the interfaces are underspecified, or turn out to need modification based on experience. This is primarily a problem for the software — the hardware interfaces, specified in mm and screw threads, are generally well defined.

Once we accept that the only way to complete the integration of the subsystems is to actually integrate them, while expecting that the interfaces and requirements will evolve, we recognise that our situation is analogous to the way that software development has moved from waterfall to agile development models.

This technote proposes that we introduce the concept of an IM which defines a specified set of functionalities that are available to the Rubin project. This is slightly different from DM's current OR concept which tests the state of DM systems at a certain time rather than guaranteeing continued cross-subsystem functionality. I envisage that an IM would be followed by an OR to test the deliverable, and to carry out an internal ORR for the functionality delivered by the IM.

[1] And also EPO, but this is probably not relevant here.

# 2 $IM_1$: ComCam Image Capture and Archive (2021-01-08)

Executive Summary:

Run ComCam from notebooks, with generation and certification of ComCam calibrations

## 2.1 Goals of IM

- Taking ComCam images in Chile using nublado
- Taking calibration and other images using the scriptQueue
- Automatic ingestion into a gen3 butler in Chile
- Transfer over DBB Buffer manager and the LHN to a gen3 repo on the RSP
- Human generation and availability of master calibrations in Chile

## 2.2 Prerequisites

- ComCam on summit
  - cold and functional
  - incoherent light source available to take flats
- gen3 butler ingestion for ComCam
- Nublado running in Chile

## 2.3 Procedure

The following procedure is to be executed by a general commissioning team member. The script creation and scriptQueue requires a minor amount of training and may require assistance.

1. Following a procedure, instantiate the OCS bridge
2. Using Nublado, bring to enabled state using Notebook

3. Using Nublado and the ComCam class:
   (a) Take a single OBJECT, BIAS, FLAT, DARK image
   (b) For each image, monitor event for CCCamera completion, monitor OODS event saying image is ready, use butler to grab image, display image locally using Firefly/DS9 or camera display tool
4. In a notebook, create cells to take bias, dark, flat, and PTC calibration data
5. Convert Nublado cells to scriptQueue, creating a "standard Calibration" script and execute them
6. From the Commissioning Cluster at the base:
   (a) Display one of each image type locally using Firefly/DS9 or camera display tool
   (b) Run gen3 cp_pipe by hand from a Nublado terminal
      i. Create master biases, flats, darks using "auto-certify" mode which assumes that the derived products are good [2]
      ii. Copy images to summit and include in summit Butler repo.
7. Take further exposures with structured illumination, preferably different from what was used to generate the flat.
8. From the summit, run ISR processing, display images and confirm new calibs are being applied
9. Once data is synced to NCSA via the LHN, Repeat the generation of calibration images using the RSP
10. Re-verify that RSP generated calibration images can be used on the summit.

[2] This results in the calibrations being available for use

# 3 $IM_B$: ComCam Calibration Acquisition, Transfer, and Processing (2021-XX-YY)

Executive Summary:

Run ComCam from notebooks and the scriptQueue, with automated generation and certification of ComCam calibrations

## 3.1 Goals of IM

- Triggering OCPS processing from nublado/scriptQueue
- Automated generation and availability of ComCam master calibrations in Chile
- Use of RLS and LOVE
  - including monitoring of data transfer to NCSA
- Demonstration that the operator can cycle through states, bring the system up from `STANDBY`, and take data without any intervention from developers
- Monitor the health of the CCS from the observatory environment (e.g. Chronograph/LOVE)
- Explore relationship between EFD and Camera trending databases
- Query capability (TAP/ADQL) access to EFD from RSP

## 3.2 Prerequisites

- $IM_1$ completed with functionality maintained
- ComCam cold and functional with light source available for flat

## 3.3 Procedure

The following procedure is to be executed by an observing specialist. Assistance may be provided in the script editing and database querying by other commissioning team members.

- Following a procedure, bring up the LOVE interface for ComCam
- Using Nublado, without assistance from developers, the operator should bring ComCam to ENABLED from STANDBY
- Take an OBJECT image using the ComCam class
  - Verify functionality of the LOVE interface, including the state and health
  - Display the raw image locally
  - Using the OCPS, perform basic ISR on the image and wait for the result (e.g. "await ocps.process(visitID, task_name)")
  - Display the ISR-corrected image
  - Monitor the transfer status and elapsed time between when the image is written on the summit to when it can be accessed at NCSA.
- Modify the script from $IM_1$ used to take calibrations to also command the OCPS to reduce the data
  - Take *only* a stack of ~10 bias frames
  - Using the OCPS, build a master bias from inside the script. Do not wait for the processing to finish
    - One could use "asyncio.future(ocps.process(visitID, task_name))"
  - Ensure the bias builds and is certified
- From the RSP, query the local copy of the EFD and plot exposure time versus number since the start of this exercise
- From the RSP, make the same plot using data from the Camera Trending Database

Rubin Observatory

# 4 $IM_C$: Rubin AuxTel Standard Star Campaign (2021-XX-YY)

Executive Summary:

Demonstrates ability to take and reduce AuxTel data using the scriptQueue.

## 4.1 Goals of IM

- Automated generation and availability of AuxTel master calibrations in Chile (cf. $IM_B$)
- Annotate exposures using RLS
- Take data in a standard star field, including interactive re-centering
- Run AuxTel data analysis package in Chile and NCSA, triggered by OCPS
  - Display results to operators

## 4.2 Prerequisites

- $IM_B$ Completed
- AuxTel and LATISS functional and ready to track a target. Rough focus should be achieved.
- Available staff for night time operations
- User commenting in RLS available

## 4.3 Procedure

The following procedure is to be executed by an observing specialist.

1. Using Nublado, slew to $\sim$ 60 degree elevation target and start tracking
2. Using Nublado, measure the focus offset using the CWFS focus script, but launched from the notebook.
3. Annotate results using the RLS

4. Apply the focus offset using the ATAOS, record the applied offset in the RLS
5. Manually take a single OBJECT image:
   (a) Verify functionality of the LOVE interface, including instrument setup and applied focus offset(s).
   (b) Using the OCPS, perform basic ISR on the image and wait for the result (e.g. ”await ocps.process(visitID, task_name)”)
   (c) Display the image locally after running ISR
6. Using the standard visit script launched from a notebook, slew to new target and perform automated acquisition only, script must have user verify offset prior to motion.
7. Perform the standard data taking sequence using the standard visit script
8. Perform a single image with the instrument in it’s standard spectral mode, then send for reduction with the OCPS. Wait for reduction to complete.
9. View the results locally in the notebook.
10. Perform a custom sequence using at least two items in the grating wheel and two in the filter wheel, all with different exposure times (OCPS should only perform ISR)
    (a) Verify functionality of the LOVE interface, including filter/grating changes/shutter movement, focus/pointing offsets, OCPS status
    (b) Display the ISR processed images locally as they are available
    (c) Monitor the transfer status and elapsed time between when the image is written on the summit to when it can be accessed from the RSP.
11. Using the scriptQueue:
    (a) Perform standard observations of 3 targets spanning the elevation range sequentially, script must send data to OCPS for reduction (but not await results)
    (b) Display extracted atmospheric parameters for each target
    (c) Comment in RLS about the weather or image quality
    (d) Perform same observations, but with a custom modified configuration (exposure time or filter adjustment)
12. From the Base and the RSP, redo reduction of scriptQueue visit data
13. From the Base and the RSP, view the log and correlate comments against images.

Rubin Observatory

# 5 $IM_D$: MT Full Slew (2021-XX-YY)

Executive Summary:

Demonstrates ability to slew the MT with Group 2 systems running

## 5.1 Goals of IM

- Slew MT to a field using a nublado notebook
- Status displays using LOVE
- Create observing log using RLS

## 5.2 Prerequisites

- $IM_B$
  - RLS, LOVE
- Group 2 hardware and CSCs all online and enabled
  - RHL which of these will be real hardware on 2021-01-15?
  - Not required for fully simulated exercise at NCSA, but CSCs must be simulated, however, no low-level M2-type simulators required
- User commenting in RLS available

# 6 $IM_E$: Rubin AuxTel Standard Star Campaign (2021-XX-YY)

Executive Summary:

Demonstrates ability to carry out sustained AuxTel observing using the scriptQueue

## 6.1 Goals of IM

- Display images using Camera Image Visualization Tool
- Take pointing model using the script queue + OCPS, possibly including interactive re-centering
- Take standard set of standard star fields using the script queue
- Visualise the reduced data

## 6.2 Prerequisites

- $IM_C$
- Camera Image Visualisation Tool or other equivalent tooling

# 7 $\mathrm{IM}_\mathrm{F}$: Corner raft AOS (2021-XX-YY)

Executive Summary:

Use a notebook to process corner raft `SW0/1` simulated data, publishing wavefront error events

## 7.1 Goals of IM

Demonstrate ability to:

- Slew to a field (only needed to provide ra, dec, rotator telemetry)
  - Question: does AOS need anything from the scheduler about the next field?
- Command exposure (or pair, if that's required by AOS) from ComCam
- have MTAOS process images, triggered by appropriate events, and publish `{{camera,m2}Hexapod,{m1m3,` events

## 7.2 Prerequisites

- Recover PhoSim images with donuts from ComCam
  - we may need to fake something here, if ComCam cannot play back corner raft data
- Pointing component generating telemetry

# 8 $IM_{pre\text{-}G}$: Standalone ComCam AOS (2021-XX-YY)

Executive Summary: Process phosim Intra/Extra simulated ComCam images with the MTAOS. Assuming that the images where acquired and ingested previously, drive the MTAOS through the data processing steps and produce corrections.

## 8.1 Goals of IM

Demonstrate ability to:

- Have MTAOS process images, triggered by appropriate commands, and publish `{{camera,m2}Hexapod,{m1m3,m2}}Correction` events

## 8.2 Prerequisites

- PhoSim images with donuts from ComCam ingested into a butler repo.
- Working version of MTAOS with access to a butler repo containing simulated ingested data.
- Working version of wavefront estimation pipeline (wep).

## 8.3 Procedure

- Generate and ingest simulated images with desired pistons (e.g. $[0, -10, 0, 10, 0]$) and appropriate metadata
- Manually issue commands to AOS system to process the data; in $IM_G$ we will generate these commands from the scriptQueue, triggered by OODS events. *N.b.* we may need an extra config file for this IM providing information which will come from SAL in $IM_G$
- Check that desired events are in the EFD

# 9 $IM_G$: ComCam AOS (2021-XX-YY)

Executive Summary:

Perform wave-front sensing using ComCam and a notebook

## 9.1 Goals of IM

Demonstrate:

- Slew to field, generating telemetry (*cf.* $IM_F$)
- piston camera (e.g. in, -out, in, +out, in)
- take data using ComCam in playback mode
- uses OCPS to reduce data on commissioning cluster
- load Zernikes into EFD

## 9.2 Prerequisites

- ability to control camera hexapod
- $IM_{pre\text{-}G}$
- $IM_F$
  - No need for replaying corner raft data
- notebook/script to perform pistoning of camera hexapod with image acquisition
- OCPS

## 9.3 Procedure

- Generate simulated images with desired pistons (e.g. $[0, -10, 0, 10, 0]$) and appropriate metadata and load into comCam

- Use scriptQueue to issue slew and camera commands, sending commands to OCPS to process the data. *N.b.* will require the AOS to synchronise the piston values with the OCPS exposure numbers (`seqNum`).
- Check that desired events are in the EFD

# 10 $IM_H$: Active Optics System (2021-XX-YY)

Executive Summary:

Simulate AOS in regular operations

Outstanding questions:

- Are we verifying anything here?
- Should this be split into two IMs?

## 10.1 Goals of IM

Demonstrate running full AOS functionality script at 3 elevations

- scriptQueue runs SAL script
  - slew to field
  - take an exposure (or snap pair)
  - MTAOS processes data and broadcasts events
    - before starting next field

## 10.2 Prerequisites

- $IM_D$
- $IM_F$

Rubin Observatory

# 11 $IM_I$: Commission Active Optics System (2021-XX-YY)

Executive Summary:

Simulate AOS commissioning, including building LUTs

Outstanding questions:

- Are we verifying anything here?
- Should this be split into two IMs?

## 11.1 Goals of IM

Demonstrate running full AOS functionality script at 3 elevations

- scriptQueue runs SAL script
  - pistons camera (in, -out, in, +out, in)
  - uses OCPS to reduce data on commissioning cluster
    - Returns Zernikes
  - Script awaits on OCPS reduction finishing
  - AOS uses returned Zernikes to calculate mirror offsets as part of loop
  - Applies offsets and repeat
  - in-focus images get sent off via OCPS for reduction
- Build LUT for each mirror and hexapod
- Load the LUTs into components
  - Ideally would use new configuration handling tstn-017
- Repeat elevation testing with new LUTs
  - results should show zero WFE
- Analysis should be repeated at NCSA

Rubin Observatory

## 11.2 Prerequisites

- $IM_f$

Rubin Observatory

# 12 $IM_J$: Start/End of Night (with AuxTel) (2021-XX-YY)

Executive Summary:

Exercise handoffs between day crew and night crew. Perform afternoon calibrations. Perform observatory opening/closing. Requires definition of opening/closing procedures and appropriate observing conditions. Also requires mechanism to get knowledge from night crew into Jira tickets which are then subsequently. How do we use LOVE/EUI/notebooks to troubleshoot/resolve faults?

## 12.1 Goals of IM

- Have observing specialist run the AuxTel, including
  - Perform nighttime hand off sequence
- Run afternoon calibrations, reduce them using OCPS, run QA, and possibly certify for use
- Test Camera Image Visualization Tool
  - This is a camera deliverable and I'm completely unaware of it's status.
- Verify current environmental conditions against acceptable conditions (Where are these hosted?)
  - Winds, seeing, humidity, windshake conditions (shutter/vent gate configurations), temperature
- Run opening script
  - Functionality requires review to ensure component safety etc.
  - When do we start observing?
- Perform several standard (spectroscopic) AuxTel visits
  - Simulate a fault, have pre-derived written instructions to "reset"
    - Instructions must not include syntax!
      - e.g. unwrap rotator, manually close vent gates 1-3, put down and push up mirror, home hexapod and re-position etc,

    - Use logger to record important information
    - Create Jira ticket(s)
    - Run pre-derived commands

- Recover, Perform more visits
    - Simulate weather event requiring closure of observatories
- Recover, perform more visits
- Sun rising, Close observatory, publish night report
- Perform daytime handoff, including the procedure to getting the JIRA tickets filtered and addressed accordingly

## 12.2 Prerequisites

- AuxTel system fully functional
- LOVE screens should be operational for mount/dome/mirrors(?)
- RLS system with fault reporting
- EAS

# 13 $IM_K$: LSSTCam Image Capture and Archive (2021-XX-YY)

Executive Summary:

Run LSSTCam from the scriptQueue, with automated generation and certification of LSSTCam calibrations

## 13.1 Goals of IM

- Taking complete set of LSSTCam calibration data using the scriptQueue
- Reduce the data using OCPS:
  - generating a set of LSSTCam master calibrations
  - characterizing the detectors (noise, gain, tearing)

## 13.2 Prerequisites

- $IM_b$
- Remote OCPS, if necessary

# 14 Milestone Activity backlog

Activities that need to be included in an IM

## 14.1 Have operators bring system up from power-off

## 14.2 Test software deployment (daytime) and rollback (nighttime) scenarios.

- Also create a test of setting up individual environments (custom packages/modifications),
- how to run a script that uses a modification to a function.
- How can someone else run someone's code from a previous night if they used a modified environment?

## 14.3 Alert Processing using AuxTel Imaging Data

## 14.4 Restart from power-loss on the Mountain

## 14.5 Promote any remaining CCS functionality needed by observers to OCS

## 14.6 Write data from the CCS, DMTN-143

- Including ComCam, AuxTel

## 14.7 Use PPI to transfer data from Chile

# A References

# B Glossary

**AOS** Active Optics System.

**CCS** Camera Control System.

**cp_pipe** The DM package used to generate calibrations products (*e.g.* master biases).

**CSC** Commandable SAL Component.

**DBB Buffer manager** OODS-aware transfer of data from Chile to the USDF.

**DM** Data Management.

**EFD** Engineering and Facility Database.

**IM** Integration Milestone.

**ISR** Instrument Signal Removal.

**LHN** Long Haul Network.

**LOVE** LSST Operations Visualization Environment.

**MT** Main Telescope.

**MTAOS** Main Telescope Active Optics System.

**OCPS** OCS Controlled Processing System.

**OODS** Observatory Operations Data Service.

**OR** Operation Rehearsal.

**ORR** Operations Readiness Review.

**RLS** Rubin Observatory Logging (may be equivalent to OWL).

**RSP** Rubin Science Platform.

**SAL** Service Abstraction Layer.

**scriptQueue** The observatory facility for running sequences of commands and interact with OCPS.