# LSST DMS Fault-Tolerance Use Cases

## Introduction

The subsections below present use cases for faults in the LSST DMS. These use cases answer the question: what are the possible things that could or might happen to cause a fault in the LSST DMS? Another way to look at it is that a fault-tolerance use case essentially comprises the "alternate course" of a basic use case.

Of course, it is impossible to make a complete list of everything that could possibly go wrong. However, listing the problems from past experience on prior ground-based astronomical pipeline-processing projects seems to be a good starting point for building up a set of credible fault-tolerance use cases for the LSST DMS. And delving into the details will ensure that the LSST DMS will be robust, as it is commonly held that well over half of a project's complexity is caused by dealing with alternate courses of action or, simply, faults.

These use cases are primarily fomulated from the perspective of what could or might go wrong to interfere with the transmission and/or preservation of the raw data (images and metadata), and/or production of the processed data products (nightly alerts, and science catalog data). Prevention of loss of raw image data and metadata is an important and absolutely essential job of the LSST DMS.

Another class of use cases of deep concern is the generation of nightly alerts and preservation of science catalog data. This is attuned toward meeting LSST science requirements, which is obviously important, and special attention to meeting functional requirements is also critical.

It is assumed that the LSST DMS has subsystems in the summit observatory, base facility, archive center, and data access centers. These subsystems are data-connected via a short-haul network between summit and base, and a long-haul network from base to archive center and data access centers. DMS subsystems include facilities, staff, hardware, software, database, and data. Furthermore, it is assumed that acquisition of the raw data is outside of the purview of the DMS, but, once the raw data is acquired, it falls into the DMS domain.

Any loss of raw images and their metadata are absolutely not allowed under project requirements and, hence, any such loss discussed below in the context of use cases is really only temporary loss. The data backup plan must therefore include highly reliable storage media, aggressive checksum validation, file storage cross-validation with database records, and geographically distributed redundant copies of the data. Redundant copies of the data should also be validated. Both the raw-image-data files and raw-image-metadata database tables must be backed up according to the plan.

Note that the terminology "SDQA fault" used below basically refers to mistakes made by the SDQA subsystem in 1) missing problems with the data, and 2) falsely identifying problems with the data. Since no detection algorithm is perfect, it is expected that neither SDQA-detection completeness nor reliability will be 100%. Nevertheless, the SDQA system will be tuned to achieve the best possible compromise between completeness and reliability for the LSST DMS overall.

Finally, any generic software fault could also apply to the SDQA software, and to fault-tolerance-related software, such as watchdog monitors, etc.

## Prioritization of Faults

We classify LSST DMS faults in terms of their priority and, to cover all cases, designate three levels of priority.

Priority-1 faults require the highest level of attention and resources, and are those that:

1. Delay transmission of the raw images and their metadata from the summit to the base facility
2. Prevent reliable storage of raw images and their metadata
3. Result in only temporary unavailability of raw images or their metadata, rather than complete loss.

Indeed, the data will be irrecoverably lost unless the data backup plan is comprehensive, reliable and bullet-proof.

Priority-2 faults require lower levels of attention and resources than priority-1 faults, and are those associated with

1. Inability to meet the 60-s requirement for nightly alert generation
2. Loss of science catalog data

One rationale for the content of the priority-2 level is that the nightly alerts and science catalogs, derived from the raw data, are the primary processed data products of the LSST DMS. These derived data products can be recomputed from the raw data, but at some dollar cost, as well as failing to meet the 60-second requirement. A robust plan for minimizing possible faults that hinder meeting the time constraint (for item 1) and reliably replacing lost science catalog data from redundant data backups (for item 2) is of paramount importance.

Priority-3 faults require still lower levels of attention/resources than priority-1 and priority-2 faults, and include things that can go wrong during the data release processing, which can lead to loss of processed images, their metadata, science catalog data, and other database metadata (until the associated raw data are reprocessed), especially in recent processing history, and temporary reduction in data-processing throughput.

# Use Cases for Priority-1 Faults

The following are use cases that cover loss of raw image data and metadata.

1. Faults in temporary summit storage
    - ♦ Raw image data are lost
    - ♦ Raw image metadata are lost
    - ♦ Raw image data/metadata associations are lost
2. Faults in temporary base storage
    - ♦ Raw image data are lost
    - ♦ Raw image metadata are lost
    - ♦ Raw image data/metadata associations are lost
3. Faults in primary archive storage
    - ♦ Raw image data are lost
    - ♦ Raw image metadata are lost
    - ♦ Raw image data/metadata associations are lost
4. Faults in redundant archive storage
    - ♦ Raw image data are lost
    - ♦ Raw image metadata are lost
    - ♦ Raw image data/metadata associations are lost
5. Uncorrected errors in TCP network data transfer

Metadata about the raw-image data include, but are not limited to, all copies of database records indicating where the primary and redundant copies are stored.

Data loss includes data corruption, which effectively renders the data useless.

Data corruption includes unrecoverable errors found by disk ECC and silent data corruption (not detected by disk ECC).

# Use Cases for Priority-2 Faults

The following are use cases that cover faults that prevent nightly alert generation within 60 seconds and loss of science catalog data.

1. Nightly alerts are not generated because of facility fault
2. Nightly alerts are not generated because of human fault
3. Nightly alerts are not generated because of hardware fault
4. Nightly alerts are not generated because of resource fault
5. Nightly alerts are not generated because of software fault
6. Nightly alerts are not generated because of database fault
7. Nightly alerts are not generated because of data fault
8. Nightly alerts are not generated because of SDQA fault
9. Sources and/or objects are misidentified or inaccurate because of SDQA fault
10. Sources and/or objects database records are lost from primary storage
11. Sources and/or objects database records are lost from redundant storage

Possible facility, human, hardware, resource, software, database and data faults are detailed separately below. In some cases, the specific fault leading to processing failure can be classified in multiple categories. Note that database faults are put in a separate category because of their special nature and the specialization required to address them.

# Use Cases for Priority-3 Faults

The following are use cases that cover loss of processed image data, especially in recent processing history.

1. Data release processing fails because of facility fault
2. Data release processing fails because of human fault
3. Data release processing fails because of hardware fault
4. Data release processing fails because of a resource fault
5. Data release processing fails because of software fault
6. Data release processing fails because of database fault
7. Data release processing fails because of data fault
8. Data release processing fails because of SDQA fault
9. Sources and/or objects are misidentified or inaccurate because of SDQA fault
10. Sources and/or objects database records are lost from primary storage
11. Sources and/or objects database records are lost from redundant storage

Possible facility, human, hardware, resource, software, database and data faults are detailed separately below. In some cases, the specific fault leading to processing failure can be classified in multiple categories. Note that database faults are put in a separate category because of their special nature and the specialization required to address them.

# Underlying Causes of Faults

## Facility Faults

- Natural disaster (fire, earthquake, flood, tornado, etc.)
- Man-made catastrophe (radiactive contamination, airline crash, poisonous gas, etc.)
- Act of war (attack, seige, sabotage, etc.)
- Security
    - ♦ Computer firewall breach (hacker, virus, etc.)
    - ♦ Unauthorized computer-room access
- System resets (e.g., checksum mismatches correlate with this)
- Air-conditioning malfunction
- Electrical fuse blown

## Human Faults

- Staff problems (malicious intent, negligence, retention/turnaround, labor strike, slow down or sick out, etc.)
- Pipeline-operator procedural error
- Specialist unavailability (e.g., DBA or mySQL expert during crisis)
- Slow turnaround in fixing/delivering software bugs

## Hardware Faults

- Summit-base fiber link/interfaces failure ("short haul")
- Global data-transer link/interfaces failure ("long haul")
- CPU failure
- RAM failure
- Local disk failure
- Power supply failure
- Network switch failure
- Network disk problems
    - ♦ Catastrophic failure (disk media, disk controller, etc.)
    - ♦ Corrupted data
        - ◊ Latent sector errors caught by disk ECC
        - ◊ Silent corruption (checksum mismatches)
- Hardware upgrade not compatible with software (portability issues, backward uncompatibility, etc.)
- Unsuccessful machine reboot

## Resource Faults

- Power failure (black out, brown out, etc.)
- Insufficient disk space
- Disk performance degradation (can occur for disks > 90% full, fragmentation, etc.)
- Disk threshing caused by insufficient memory
- Disk/network speed mismatch (bandwidth, maximum number of reads/writes per second, etc.)
- Database resource faults
    - ♦ Bandwidth limitations caused by resource over-allocation
    - ♦ Too many database connections

        - Performance degradation caused by
            - Large tables filling up
            - Too many queries running
            - Large queries running
            - Usage statistics not updated
            - Insufficient table-space allocation
            - Progressive index computation slowdown
            - Transaction logging disk space filling up
            - Transaction rollback taking too long
            - Miscellaneous mistunings
- Insufficient disk-space allocation
- Network bandwidth limitation (sustained or peak specifications exceeded)
- Memory segment fault (stack size exceeded, insufficient heap allocation, misassignment of large-memory process to small-memory machine, etc.)
- OS limits exceeded (queue length for file locking, number of open files per process, etc.)
- Bottleneck migration (e.g., increase in processor throughput hammers database harder)

## Software Faults

- Software inadequacies and bugs flushed out by data-dependent processing
- Incorrect software version installed
- Incompatibility with operation system software
- OS, library, database software, or third-party-software upgrade problem
- Cron job, client, or deamon inadvertently stopped
- Environment misconfiguration or loss (binary executable or third-party software not in path, dynamic library not found, etc.)
- Processing failures due to algorithmic faults
    - Division by zero
    - No convergence of iterative algorithm
    - Insufficient input data
- Processing failures related to files
    - Can't open file
    - File not found
- Processing failures related to sockets
    - Port number not available
    - Socket connection broken
- Processing failures related to database (also see section on database faults below)
    - Can't connect to database
    - Missing stored function
- Faults associated with user-contributed software
- Problems with user retrieving data from archive
- Problems reverting to previous build (incomplete provenance of software and builds)

## Database Faults

- Database server goes down
- Database client software incompatible with database server software
- Bugs in upgraded versions of database server software
- Can't connect to database
- Can't set database role

- Can't execute query
- Can't execute stored function
- Missing stored function
- Queries take too long
- Table locking
- Transaction rollback error
- Transaction logging out of disk space
- Record(s) missing
- More than one record unexpectedly returned
- Inserting record with primary key violation or missing foreign key

## Data Faults

- Uncorrected errors in TCP communications
- Missing or bad input data
    - ♦ Bad images (missing, noisy data, or instrument-artifact-contaminated pixels; not enough good sources for sufficient asterometric and/or photometric calibration; etc.)
    - ♦ Missing/unavailable database data (e.g., PM and operations activities not syncronized)
    - ♦ Bad or wrong calibration data used in processing
    - ♦ Unavailability of calibration images (missing observations, calibration-pipeline error, etc.)
        - ◊ Use lower quality fallback calibration data (affects SDQA)
        - ◊ Missing fallback calibration data
    - ♦ Unavailability of configuration or policy data files
- Failure to flag dead, dying, or hot pixel-detectors in data mask
- Publicly release data is found to have problems after it is already released

## SDQA Faults

- Incorrect or mistuned QA-metric threshold setting(s) for automatic SDQA
- Failure to do sufficient manual SDQA on a particular data set