FSD High Level Apps

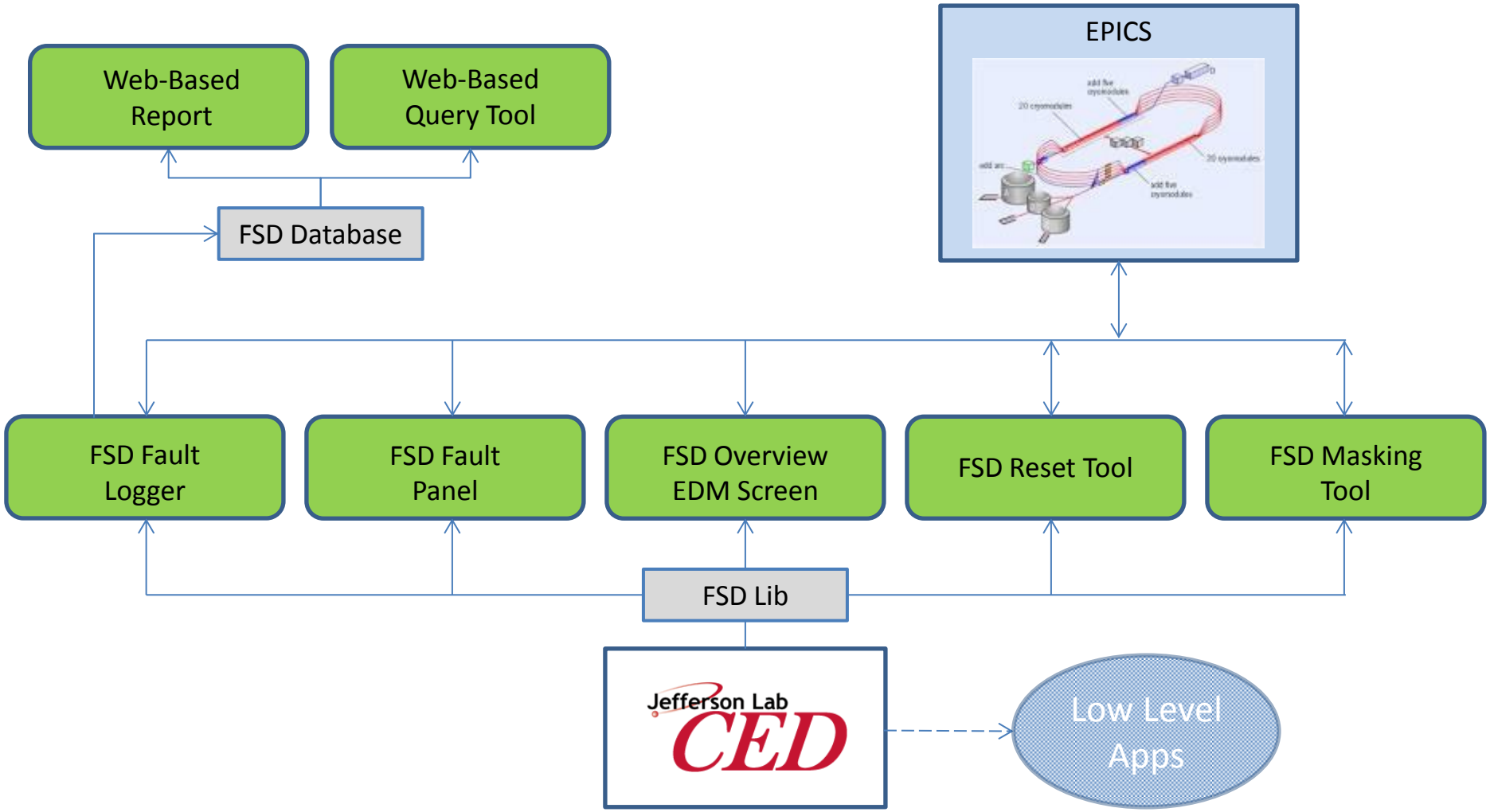
Ryan Slominski

HLA Group is Michele Joyce, Marie Keese, Theo Larrieu, Chris Slominski, Ryan Slominski

Outline

- Overview and Problem Statement
- Catching and Recording
- Alerting, Resetting, and Masking
- Reporting and Analyzing
- Known Issues
- Conclusion

Overview


Web-Based Report
Web-Based Query Tool
FSD Database
EPICS
FSD Fault Logger
FSD Fault Panel
FSD Overview EDM Screen
FSD Reset Tool
FSD Masking Tool
FSD Lib
Jefferson Lab CED
Low Level Apps

What problems are we solving?

- Maintainable, consistent, correct: CED / OTF
- Transparent, accountable: web-accessible archived data
- Easy to use: mask by destination for example
- Improve machine performance: understand / minimize trips

FSD High Level Apps

CATCHING AND RECORDING

FSD Lib

- Common library of FSD functions
- Used by all HLA FSD applications
- Monitors FSD System status
- CED driven
- Logic to interrogate devices

Who Faulted?

FSD Database

- Stores Trips
- Each Trip is due to a fault in the master node and zero or more child node faults
- Each faulted Node has zero or more faulted channels (zero = Phantom)
- Each faulted channel references either a child node or one or more devices
  - Referenced entity may not be faulted (Phantom)

FSD Fault Logger

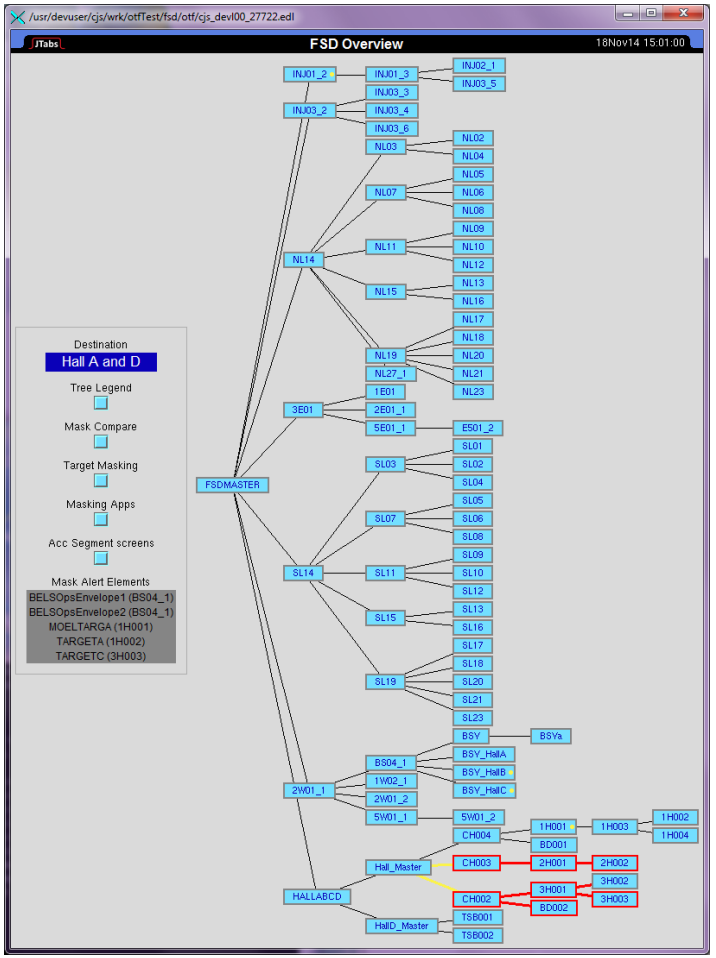
- Continuously running daemon process
- Logs information into the FSD Database

FSD High Level Apps

ALERTING, RESETTING, MASKING

FSD Overview Screen

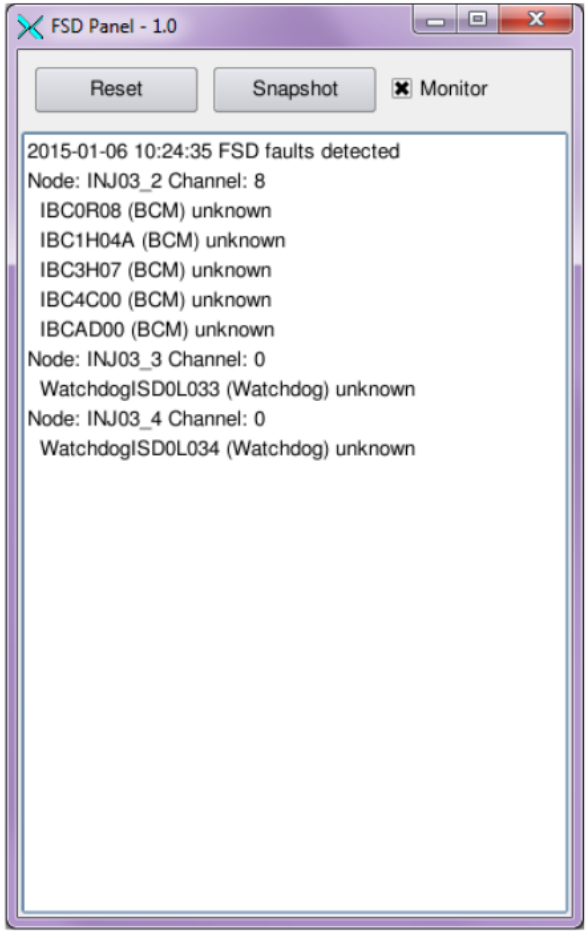
- Graphical view of FSD Tree and its current masking and fault state
- On-the-fly (OTF)



JTabs > Operations > FSD > Overview

FSD Fault Panel

- Displays textual description of faulted devices
- Reset option
- Current snapshot on-demand
- Continuously monitor root node state changes (faulted/reset) and display tree snapshot



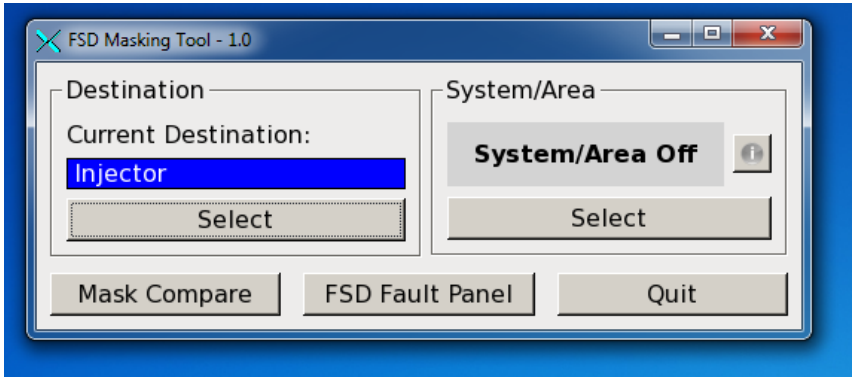
JTabs > Operations > FSD > Reset

FSD Reset Tool

- Command line application
- Used to reset the FSD Tree
- Can be invoked from Overview, Panel, or Masking GUIs via button

FSD Masking Tool

- New (reworked); still in acceptance testing
- Use to setup destination and system based masking of devices that should not propagate faults



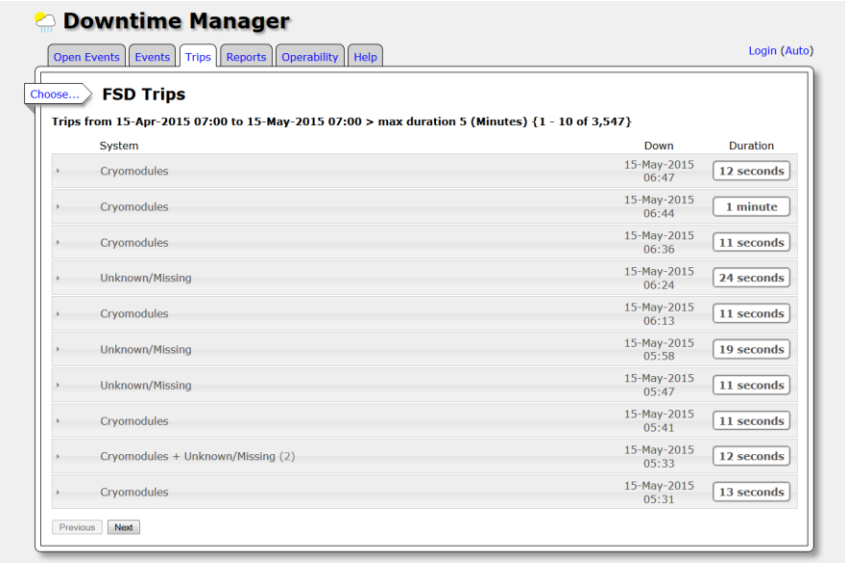
JTabs > Operations > FSD > Masking

FSD High Level Apps

REPORTING AND ANALYSIS

Trip Database Query Tool

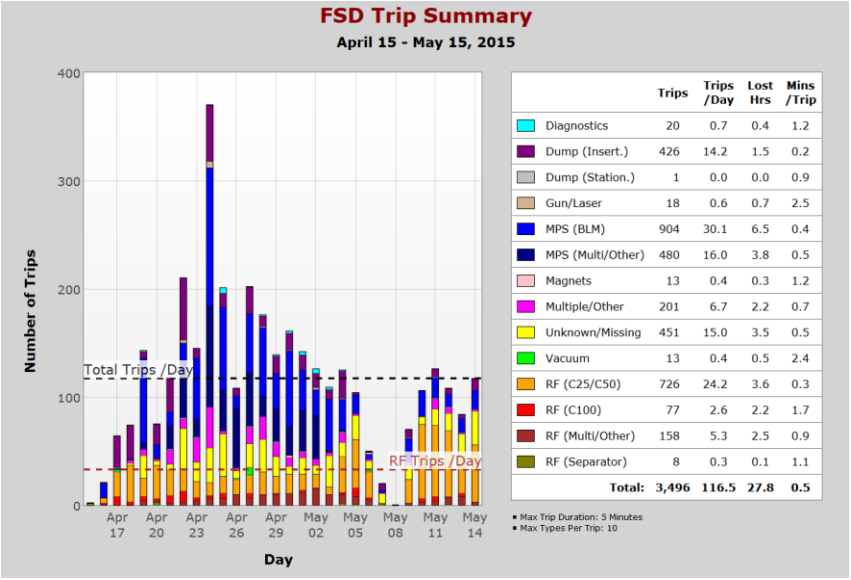
- Query Trip History
- Filter results
  - Machine beam state
  - Trip duration
  - Date range
  - CED Type
  - CED Component
  - HCO System
  - And more…



https://accweb.acc.jlab.org/dtm/trips

Trip Summary Report

- MCC 8:00 AM Summary
- Configurable Histogram
  - Date range + bin size
  - Legend Data
  - And More…



https://accweb.acc.jlab.org/dtm/reports/fsd-summary

FSD High Level Apps

KNOWN ISSUES

Device Interrogation

- We don’t always know how to query various devices on a faulted channel to find culprit(s)
  - We must record all devices on channel as faulted
  - If only one device on channel then no issue

First Fault Tracing

- Faults cascade; but difficult to know which came first; some may truly be concurrent
- FSD Lib just reports all faulted nodes
  - Web Histogram indicates “Multiple/Other” when more than one of differing types
- Scan rate and clock skew = race condition
  - root node may indicate fault before leaf node that generated it does! (shown in archiver)

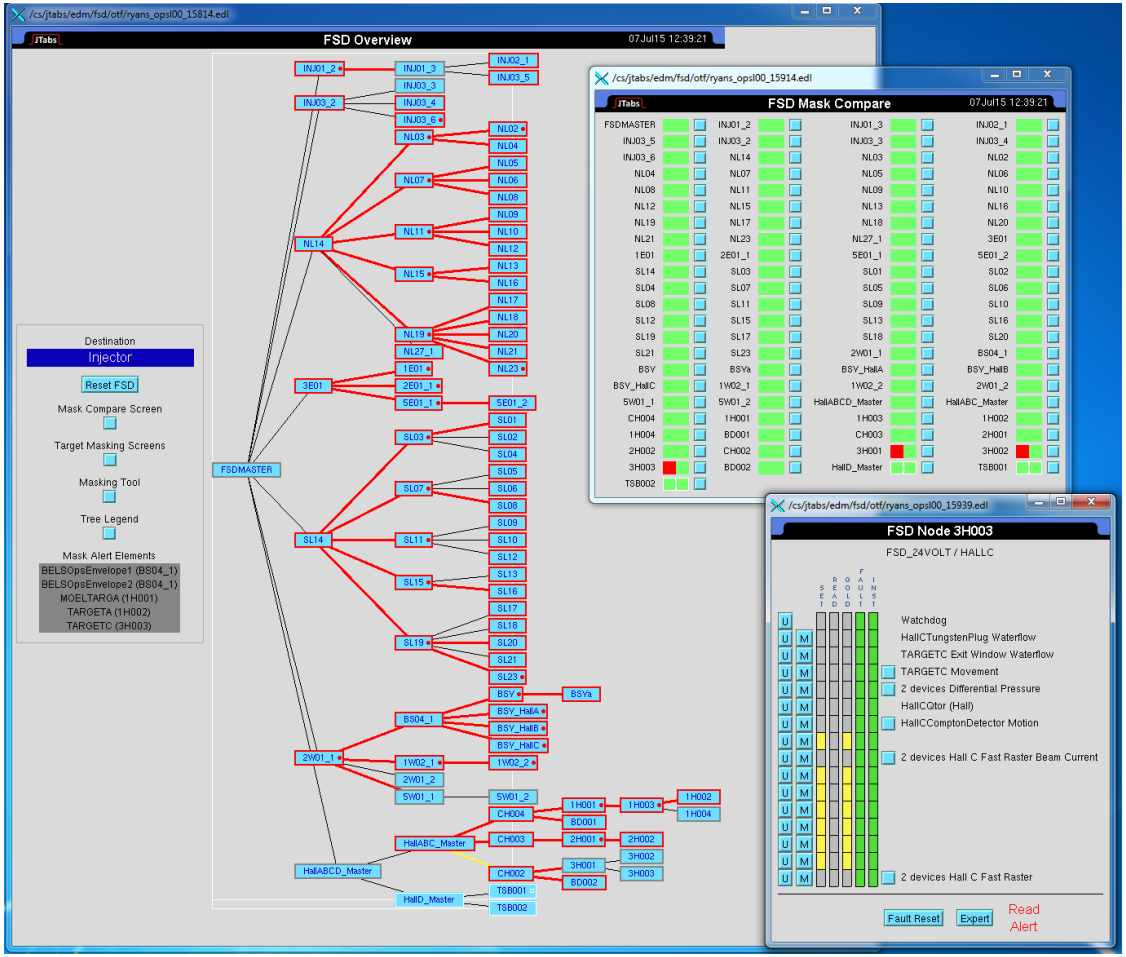
Phantom Faults

- Master node signaled, but either:
  - No leaf node admits fault
  - A leaf node admits fault, but no channel/device does
- Costs downtime / confusion
  - 685 Phantoms in Spring
- Many possible causes
  - Hardware / IOC software sync
  - Incomplete / Incorrect device interrogation rules (dtm1442)
  - Scan-rate timing issues
  - And more…

Conclusion

- CED and FSD Lib ensure all apps have consistent view
- Trip reporting available on web
- To Improve FSD Apps & Operator experience we need to:
  - Minimize Phantom Faults
  - Explain device interrogation details
  - ~~Synchronize FSD System?~~

Bonus: What is wrong here?



Interesting Read

- J. Perry and E. Woodworth. The CEBAF Fast Shutdown System. CEBAF PR-90-15. September 1990
  - In 1990 we needed 24 µs to shutdown, and at that time burn through was in 30 µs.
  - We improved FSD speed for 12GeV, right?