EPICS General Messaging and Error Handling for NLC

• What we have now
• Work in progress
• What is needed for NLC
What We Have Now

• Status codes - passed back to caller or logged
  – Format is: 8bit Subsystem + 8bit Status

• Can store device status in database.

• SEVCHECK. CA allows private exception handler.

• Currently we can trap error messages and direct to system wide log file (see IOC application developer’s guide).
  – errlogAddListener() is available in EPICS 3.13.1
  – iocLogClient and iocLogServer

• Likewise, can trap to CMLOG

• CMLOG is available for logging general (event) messages or trapping error messages.
Enhanced status codes have been batted around since ‘95
William Lupton did a good write-up and summary of various points of view.
Proposed Status Code Format:
  - 8 bits - severity, group, subsystem
  - 8 bits - status value
Have we arrived at a consensus?
Standardization of what gets sent where is needed
  - Sending codes back to client vs. Sending msgs to central log system
Possible message string server.
• Ability to trap messages and send to central logging.
  – Server needed
  – Monitor/retrieval system is needed.

• CMLOG
  – A separate product from EPICS.
  – Complete talk tomorrow.
  – Can log lots of CDEV tags like pid, severity, host, etc.
  – Not just for error messages.
  – `errlogAddListener` used to trap to CMLOG client.
  – Uses `cdevData` C++ objects
CMLOG (continued)
- Messages logged in B+ tree for fast lookup by time.
- Multi-threaded server.

CMLOG could be part of comprehensive solution.
- Could be distributed with EPICS?
- Could provide flag to trap messages to client Daemon
  - (under logMsg, errlogPrintf, errPrintf, etc) CMLOG

SLC is integrating CMLOG with the control system

BaBar using CMLOG also.
NLC Requirements

• Message volume/rate
  - 1300 IOC’s
    x 200 bytes per message average
    x 100 msgs/hour/ioc average

  = 26 MB / hour
  x 24 hours
  x 200 days of online storage

  = 124 GB of online storage
NLC Requirements Continued

• Burstiness is not determinate so we need:
  – Error message metering on IOC (errVerbose flag now).
  – Metering summary messages.
  – Filter by any tag on the ioc (like severity)
  – Storm prevention.

• Implementation of status code processing above is needed.
NLC - The Next Linear Collider Project

SLC Errorlogging As A Requirements Baseline For NLC

• Robust and Very Useful
• Status codes mapped to strings across all platforms
• Pass status codes to save network bandwidth
  – Originator only passes status code and parameters.
  – Text string provided when displayed for user (browser)
• Monitor in real time or look at historical messages
  – Filtering provided by any tag type.
• Can correlate messages with events (agrees w/archiver)
• Tags include: error code, source micro, severity, time.
NLC Requirements

• Enhanced central message logging system is needed.
  – Just pass status codes, not text (like VMS) when possible.
    • Saves bandwidth
    • What are the disadvantages to this?
  – Separate streams for central logger.
    • Partition Accelerator into N sections

• CMLOG could be that logging system with modifications
  – Gateway for multiple streams to one huge logging system
  and/or-
  – One logging server per section.
NLC Requirements

• CMLOG would provide an arbitrary number of CDEV tags to be logged.
  – This is useful for filtering and metering.
  – Tags would include: error code, source micro, severity, time, subsystem, source line, line number, etc.

• Central Logger failover.

• Useful monitor and retrieval tools (like CMLOG browser).

• Compression and backup/merge of historical data.

• Ability to Route messages by facility or other tag to separate server.
Conclusion & Questions

• Good work is underway now which will be needed for NLC (like status code standardization)
• Overall architecture for error logging needs to be developed
• Centralized message logging system will also be developed. This could be CMLOG
• Do we need web-based browsers?
• C++ exception handling needs some thought.
• What work are others doing in error codes/logging?