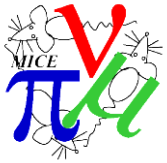

Configuration Database

Janusz Martyniak, Imperial College London

MICE CM43 Analysis, Software and Reconstruction

Configuration Database

project status

- We maintain a primary CDB server in the Control Room, including a supermouse WebService front end.
- A hot standby in the PPD which mirrors the primary DB, read only – have a new machine as the second standby now
- A preprod DB with write access

Configuration Database (contd.)

An extension to the server API:

- Batch Iteration number which allows storing multiple MAUS data cards for the same MAUS version – used in production now (however overused)
- We should make sure all information needed for MAUS to run is in the CDB, so we can run with a default B.I.N (next slide)
- MC serial number, not used yet
- Server-site code extended
- Python API for the client written

Configuration DB (contd.)

Beamline C API

- The API has been written in C (gSOAP) to facilitate integration with run control code
- Successfully implemented for:

1. Retrieving beamline settings (tags)form the database
2. Storing beamline magnet data to the hardware
3. Storing the settings to the CDB for a given run
4. Storing new tags (in progress)

- Considered stable, although some minor changes are still expected
- Project on Launchpad:

bzr+ssh://bazaar.launchpad.net/~janusz-martyniak/mcdb/mice.cdb.client.api-C/

Configuration DB

(coolinchannel data for previous runs)

- Cooling channel magnet setting are not yet written to the CDB during data taking (work in progress).
- The settings are stored in from of tags in the CDB, and we know the run ranges the tags apply to.
- A small program was written to retrofit missing cooling channel data based on known run ranges and the CDB tags.
- The cc data from day 1 are now on preprod.
- Ready to be stored in the CDB after checks for consistency are done (a matter of days)
- This will allow MAUS to operate with the default settings.

Cooling Channel CDB API

- First version created in mid 2014
- No concept changes, some refactoring done to match the Beamline API as close as possible
- Implementation in the Control Room in progress
- Workflow similar as for the Beamline API
- Should be ready when data taking starts (otherwise we have to repeat a ‘retrofit’ procedure for previous runs, as described earlier)
- Unit tests still to be refactored

CDB - Summary

- CDB is fully operational, bugs discovered have been fixed (latest fix #1778, for retrieving geometry by run number)
- C-API for Beamline implemented in the Control room
- Cooling Channel API being implemented, should be ready in a week
- State machine C-API written (user interface only, no supermouse)
- Extensive usage examples exist for all APIs