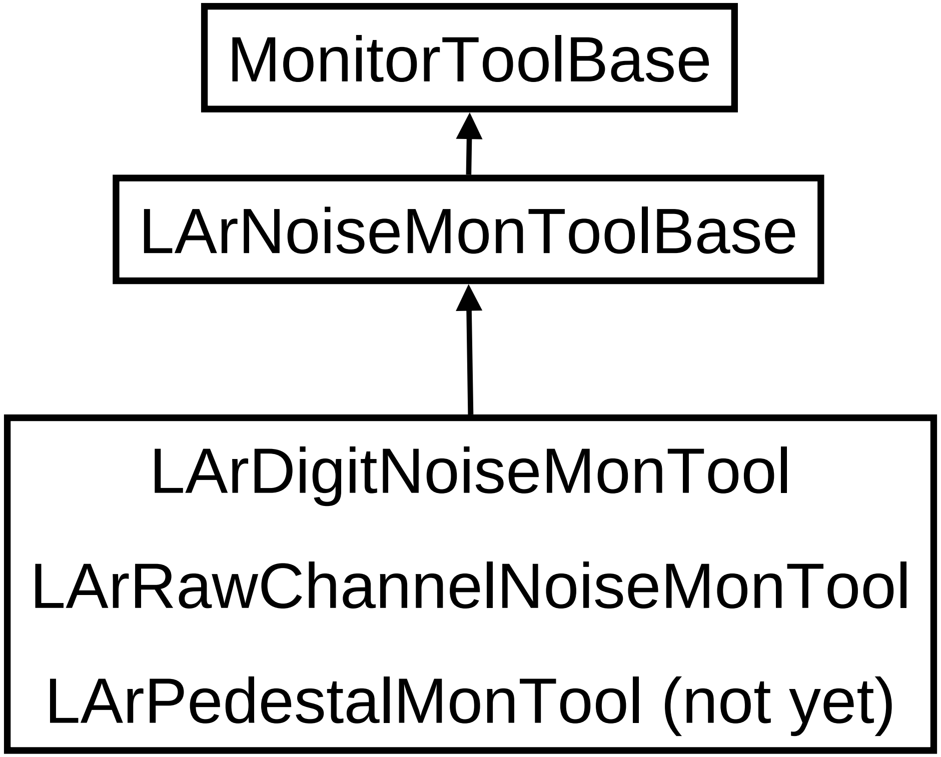
Athena LAr Noise Monitoring Software

Brief Status Report

Michel Lefebvre
University of Victoria 02 Nov 2005

- Following discussions with Remi Lafaye, the design is evolving
- Try to avoid computing the pedestal and rms in different places in the code
  - Pedestal and their rms are to be computed in dedicated algorithms and, optionally, loaded in the database
  - LArDigit: LArPedestalMaker produces LArPedestal
    - needs improvements (Kai)
  - LArRawChannel: a similar algorithm perhaps to be written
  - One way to monitor the pedestals and their rms is to take info from these objects
    - LArDigit: from LArPedestal
  - LArNoiseMonToolBase design (Rolf, Michel) still good
    - Continue implementation!

simple design idea



MonitorToolBase

LArNoiseMonToolBase

LArDigitNoiseMonTool

LArRawChannelNoiseMonTool

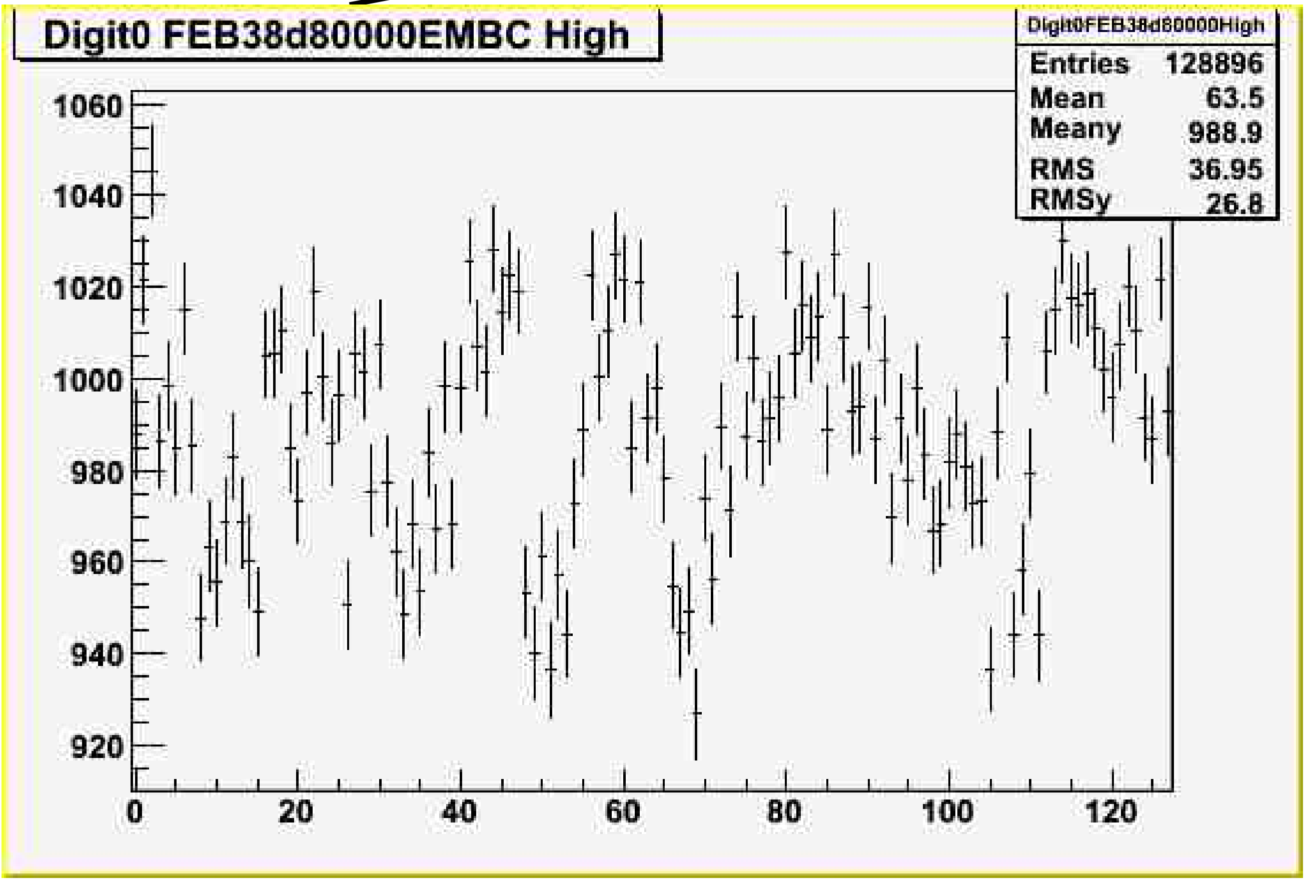
LArPedestalMonTool (not yet)

- implements common code
  - implemented
- handles specifics
  - handles loop over relevant objects

- First implementation working well
- Expect the design to be stable
  - implementation will evolve with profiling info

a few histograms (Phase1 run 18720)

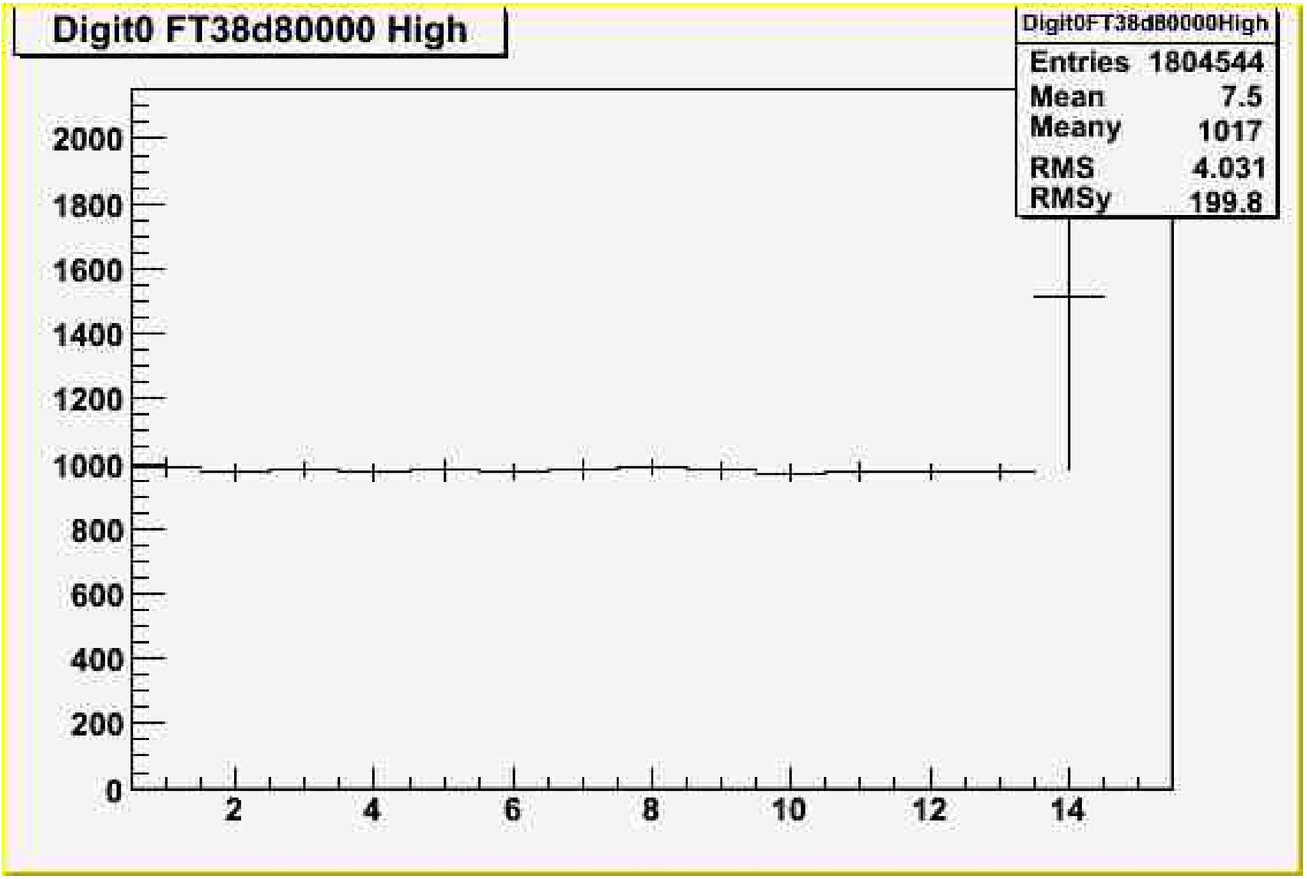
FebID, LAr partition and gain



- sample 0 (ADC) per channel for a given FEB

a few histograms (Phase1 run 18720)

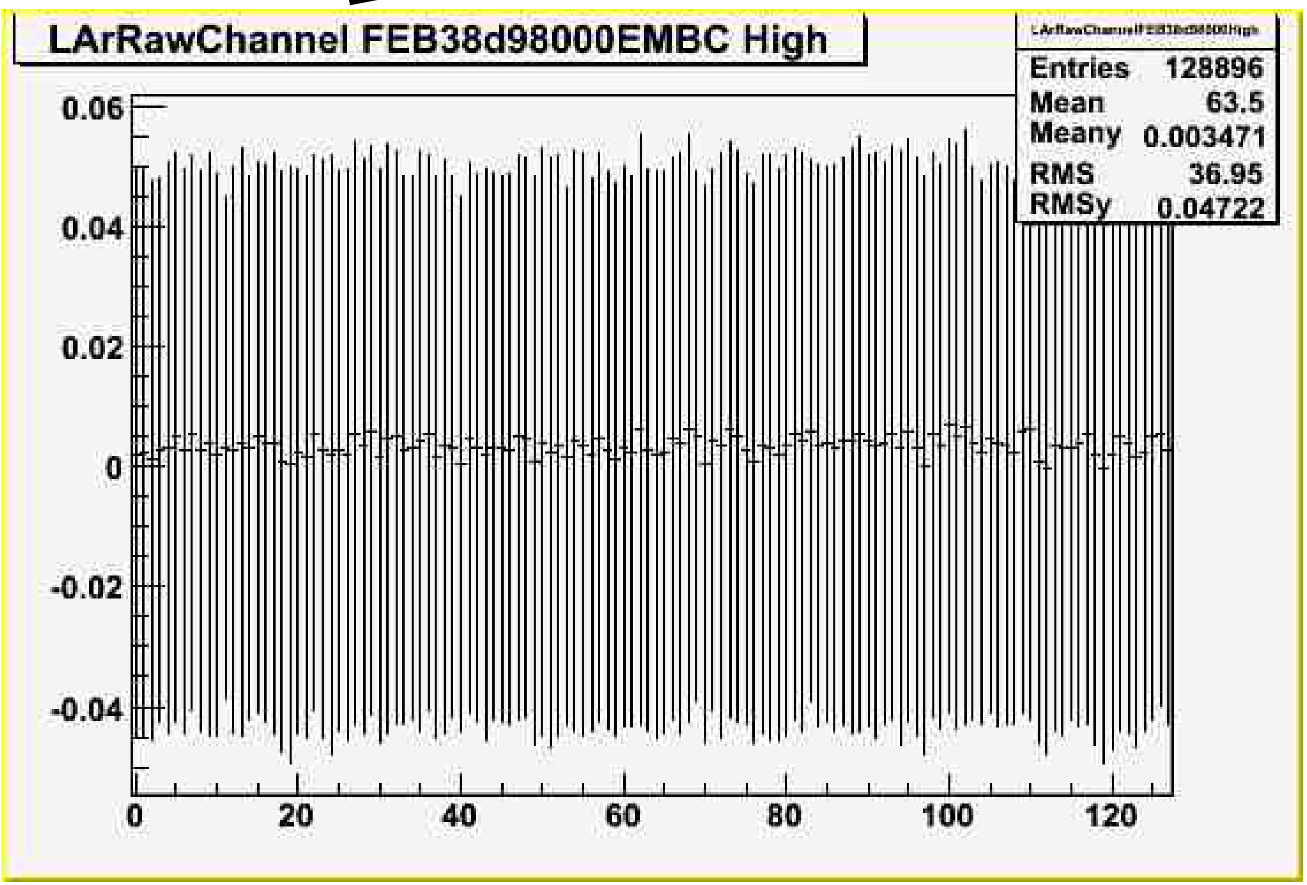
Feedthrough ID and gain



- sample 0 (ADC) per FEB for a given feedthrough

a few histograms (Phase1 run 18720)

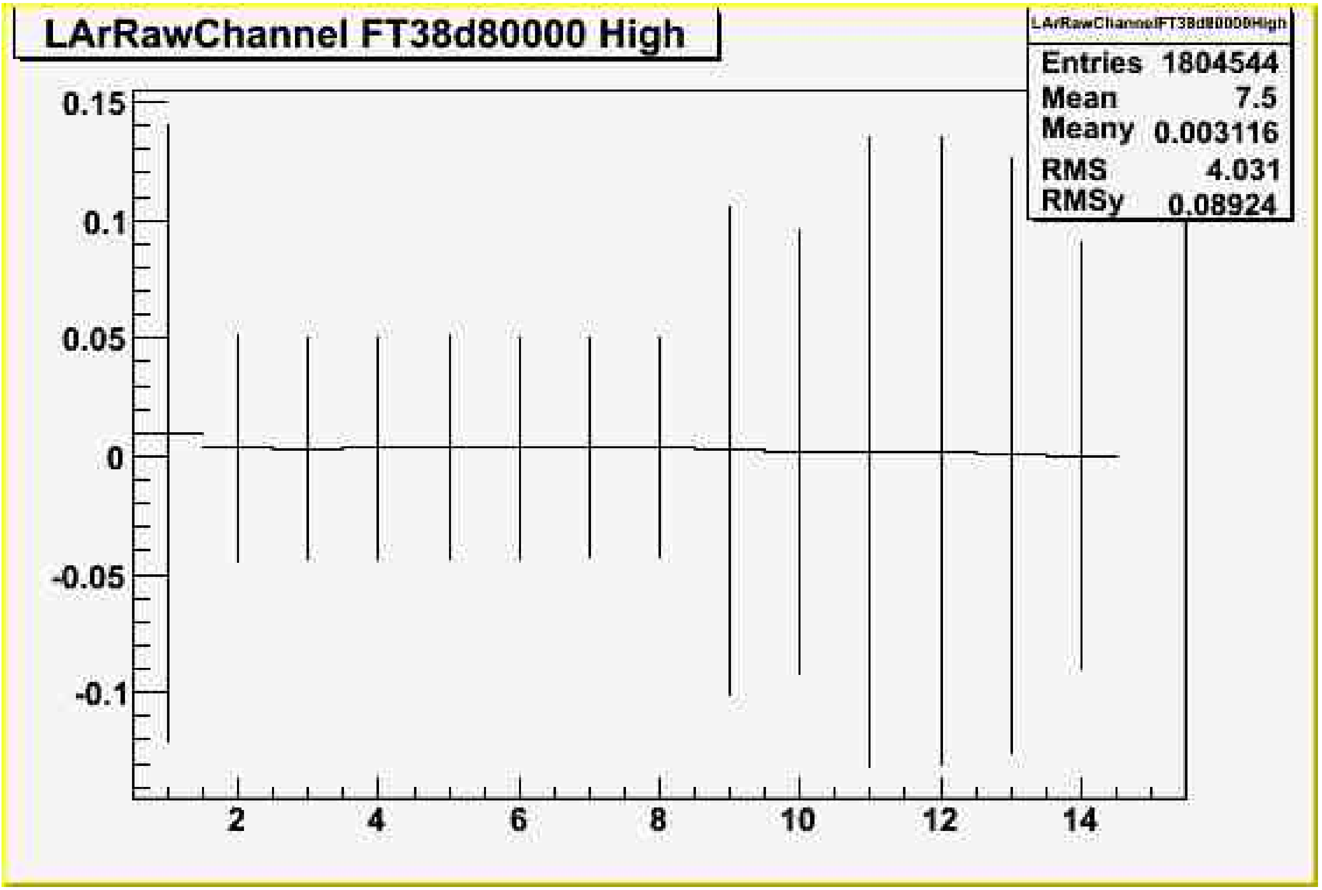
FebID, LAr partition and gain



- LArRawChannel signal (GeV) per channel for a given FEB

a few histograms (Phase1 run 18720)

FebID, LAr partition and gain



- LArRawChannel signal (GeV) per FEB for a given feedthrough

coherent/incoherent noise

- Consider implementing (relative) coherent noise monitoring
  - the issue was discussed with Petr Gorbanov and Remi Lafaye and a good solution exist
  - difficulty is free gain
  - speed is an issue

- Histogram checking to be tried very soon

- root macros implementation (Tayfun, Michel) stating NOW