ACD calibrations

Types of Offline Calibrations

On the ground we run these types of offline calibration

- Pedestal. Use B/13 (non zero-suppressed) run or periodic/solicited triggers from flight mode runs.
  - Caculate mean/rms of pedestal values. Check for skewness?
  - HV on or off?
  - Outlier rejection if HV on.
  - How to treat NA tiles.

- MIP peak. Use any muon runs.
  - Fit smooth curve to MIP peak.
  - Select fit range avoid pathologies
  - Select tracks near normal incidence.
  - Correct for pathlength in the tiles. (This requires pedestal subrataction first).

- Veto turn-on. Use any muon runs.
  - Estimate 50% point of veto turn on curve.
- High range calibration. Converts PHA values in high range to MIP equivalent.
  - Goes from linear (at low PHA) to saturation (at High PHA)
- CNO turn on. Use any run, but needs lots of stats
  - Estimate 50% point of CNO turn on curve.
- Range crossover. Uses any run, but needs lots of stats
  - Looks for lowest value in high-range, highest value in low range.
- Coherent noise.
  - maps out pedestal fluctuations

Pedestal, MIP peak, High range calibration and coherent noise are used in reconstruction and simulation.
Veto, CNO, Range crossover are only used in simulation.

calibGenACD Package Executables and Options

The package calibGenACD has the following executable to run calibrations

- runPedestal.exe -> make pedestal from an input digi file or files

- runMuonCalib_Roi.exe -> make mip calibrations from an input digi file or files

- runMuonCalib_Tkr.exe -> make mip calibrations from an input recon file or files (can use pathlength correction method)

- runMuonCalib_Svac.exe -> make mip calibrations from an input recon file or files (can use pathlength correction method)
- runVetoCalib.exe -> find veto set point in terms of PHA counts from an input digi file or files
- runCnoCalib.exe -> find CNO set point in terms of PHA counts from an input digi file or files
- runCoherentNoise.exe -> fit the coherent noise from an input digi file or files
- runRangeFit.exe -> find the range crossover points, for input digi file or files

- runStripChart.exe -> make a time series strip chart to look for flaring up of channels

- runMeritCalib.exe -> make an output ntuple to measure ACD performance

You can specify the input (not all jobs require all types of input files) and output files for all of these exectuable on the command line with these options:

- -d <digiFiles> : comma seperated list of digi ROOT files
- -r <reconFiles> : comma seperated list of recon ROOT files
- -S <svacFiles> : comma sepearted list of svac ROOT files

- -m <meritFiles> : comma seperated list of merit ROOT files

- -o <output> : prefix (path/filename) to add to output files

In addition to this all these executable take the following options [defaultValues] :

- -h : print help message
- -c <configFile> : name of xml file with job configuration (see below)

- -I <Instrument> : specify instument being calibrated, this is a tag in the output files [LAT] (use CU06 for beamtest)

- -n <nEvents> : run over <nEvents> (0 for all events) [0]

- -s <startEvent> : start with event <startEvent> [0]

Finally, if the calibration jobs requires other calibration files as input, these can be specified with the following options:

- -P : for runPedestal.exe only -> use only periodic triggers

- -p <pedFile> : specify the file with the pedestals
- -g <gainFile> : specify the file with the gains (AKA mip peaks)
- -b <nbins> : for runStripChart.exe only -> number of time series bins

Xml job configuration file

The xml job configuration file should have the following format

```
<?xml version="1.0" ?>
<!-- job optin file for runMuonCalib.exe -->

<!DOCTYPE ifile SYSTEM "$(CALIBGENACDROOT)/xml/ifile.dtd" >

<ifile cvs_Header="$Header: $" cvs_Revision="$Revision: $" >
  <section name="parameters"> input parameters for ACD pedestal and gain calibration
   <item name="instrument" value="LAT">instrument type to be calibrated </item>
   <item name="<parameterName>"value="<parameterValue>"> Various Job Parameter settings, one per item</item>
  </section>
</ifile>
```

The following parameter names mirror the command line arguments give above:

- instrument (same as -I) : specify instument being calibrated, this is a tag in the output files [LAT]
- digiFileList (same as -d) : blank-delimited list of input digi root files
- reconFileList (same as -r) : blank-delimited list of input recon root files
- svacFileList (same as -S) : blank-delimited list of input svac root files
- meritFileList (same as -m) : blank-delimited list of input merit root files

- outputPrefix (same as -o) : output file prefix
- pedestalFile (same as -p) : Name of text file with pedestal values
- gainFile (same as -g) : Name of text file with gain (AKA mip peak) values
- requirePeriodic (same as -P) : use only periodic triggers
- pathLengthCorrection (same as -L) : correct for pathlength in tile"

Calibration Output Files

The various calibrations produce output files in both xml and text format. Both files consist of headers that give information about the calibration version & data set
and bodies that give the calibration values.

An example calibration header might look like this (in text format)

```
#SYSTEM = acdCalib
#instrument= LAT
#timestamp = Wed Jan 18 16:35:58 2006
#calibType = ACD_ElecGain
#fmtVersion = v1r0p0
#startTime = 135005345:1
#stopTime = 135005389:466011
#triggers = 9796697/10799542
#source = 0
#mode = 0
```

startTime and stopTime show the first and last triggers seen in the data used for this calibration.

fmtVersion is the xml (and text) format version. The should match the dtd in the calibUtil/xml/acdCalib_<fmtVersion>.dtd file.

triggers shows the number of triggers taken and used for the calibration.

source and mode don't mean anything yet.