 GLAST LAT TECHNICAL DOCUMENT Document Title calibGenCAL v2r5 Description	Document # LAT-TD-	Date 21 October 2004
	Author(s) Mark Strickman	
	Subsystem/Office Calorimeter	

**Gamma Ray Large Area Space Telescope (GLAST)
Large Area Telescope (LAT)
Calorimeter Calibration Software
calibGenCAL v2r5
Description**

Change History Log

Revision	Effective Date	Description of Changes

Contents

1. Purpose	4
2. Scope	4
3. Definitions	4
3.1. Acronyms	4
3.2. Definitions	4
4. References	5
5. Measurement Conventions	5
5.1. Coordinate system	5
5.2. Alternating layers	5
5.3. Units	5
6. calibGenCAL release plans	5
7. calibGenCAL v2r5 Overview	6
8. calibGenCAL v2r5 Algorithms and Products	7
8.1. Class structure	7
8.2. Overall Flow	8
8.2.1. Inputs	8
8.2.2. Phase I	8
8.2.3. Phase II	8
8.3. Data Selection	12
8.4. Products	12
8.4.1. Pedestals	12
8.4.2. Mu Peaks	13
8.4.3. Mu Slopes	15
8.4.4. Asymmetry	15
9. calibGenCAL v2r5 Use	16

1. Purpose

This document describes the LAT calorimeter calibration process as applied during the early Integration and Test phase of the mission, using version v2r5 of the calibGenCAL package.

2. Scope

This document covers calibration procedures to be used on the flight calorimeters prior to launch. In particular, it covers calibrations performed by the calibGenCAL v2r5 package, which will be used very early in the I&T process. This will be supplanted by calibGenCAL v3 late in 2004 (and covered by another document).

This document includes discussions of both charge injection and muon calibration processes, but does not discuss on-orbit calibrations using galactic cosmic rays. It will discuss the calibration algorithms and products, and how to run the calibration software. It deals primarily with software in the Science Analysis Software (SAS) environment, although topics from on-line analysis will be touched on as required.

3. Definitions

3.1. Acronyms

CAL	Calorimeter
CDE	Crystal Detector Element
CSV	Comma Separated Values
DAC	Digital to Analog Converter
EM	Engineering Model
LAT	Large Area Telescope
GLAST	Gamma-ray Large Area Space Telescope
PDA	Pin Diode Assembly
SAS	Science Analysis Software
TKR	Tracker

3.2. Definitions

mm millimeter

Simulation To examine through model analysis or modeling techniques to verify conformance to specified requirements

4. References

LAT-TD-00035-1 “LAT Coordinate System.”

5. Measurement Conventions

5.1. Coordinate system

Dimensions and values shall use the LAT Coordinate System as their reference for describing orientations and directions (if applicable). This is detailed in LAT-TD-00035-1 “LAT Coordinate System.”

5.2. Alternating layers

Calorimeter crystals are arranged in alternating layers. In the top layer, the long axis of each crystal is parallel to the X-axis, in the next, the long axis of each crystal is parallel to the Y-axis, on so on. The former are referred to as X layers and the latter as Y layers. Dimensions referred to as “transverse” without a specification of X or Y dimension are typically the same for crystals in X and Y layers. In X layers, a transverse length would be parallel to the Y-axis, while in Y layers, it would be parallel to the X-axis. Frequently, the variable name will specify X or Y, but the dimension can apply to both.

5.3. Units

The dimensions listed here are nominal and in millimeters and do not reflect tolerances. When a number is underlined in a drawing (e.g. .360.5) it means that its value has been changed by hand for the purposes of this document.

6. calibGenCAL release plans

The calorimeter muon and charge injection calibration processes have been gathered together in the calibGenCAL package. Although this package is a part of GlastRelease and EM releases, it is independent of the Gaudi framework and GLEAM. The current version of calibGenCAL, as described in this document, is v2r5.

As part of the CAL calibration effort in support of LAT I&T, the calibGenCAL functionality is in the process of a substantial modification, involving some fundamental changes to the parameterization of calibration quantities. These changes improve performance and allow the calibration system to handle event configurations likely to appear in flight (although not during ground calibration). An example is a hit CDE, read out in single range readout mode, that uses a LEX range on one end and a HEX range on the other. This situation involves cross-calibrations that have not previously been produced. These changes will appear in calibGenCAL v3, which is described in another document.

V2r5 will be used during the early part of LAT I&T, extending approximately through the end of 2004. A changeover to v3 will then take place. The changeover will include a new set of calibration products to be delivered with each flight calorimeter, a new version of calibGenCAL (v3) to be made available to EM and GlastRelease, and a new descriptive document.

7. calibGenCAL v2r5 Overview

calibGenCAL v2r5 performs a variety of muon calibrations. These include:

1. Pedestals – Noise pedestal position and width are measured.
2. Mu Peaks (Gains) – Mu Peak is the position of the muon peak in ADC units, pedestal subtracted and corrected for path length. Gain, which is stored in the output xml file, is 11.2 MeV/Mu Peak (note that this is not a proper “gain”, but rather an ADC bin width).
3. Mu Slopes – Conversion from asymmetry to position, measured as the slope of the linear portion of the asymmetry curve (central half of the curve). Based on $(L_1-L_2)/(L_1+L_2)$ definition of asymmetry.
4. Light Asymmetry – Light asymmetry vs position represented by a table of 12 spline points.

While the code to analyze charge injection calibrations is present, it will not be run routinely. The current versions of CalRecon and CalDigi do not use integral nonlinearity, obviating the need for the calibration. calibGenCAL v3 and the matching new recon and digi code will make full use of CI calibrations and integral nonlinearity.

8. calibGenCAL v2r5 Algorithms and Products

8.1. Class structure

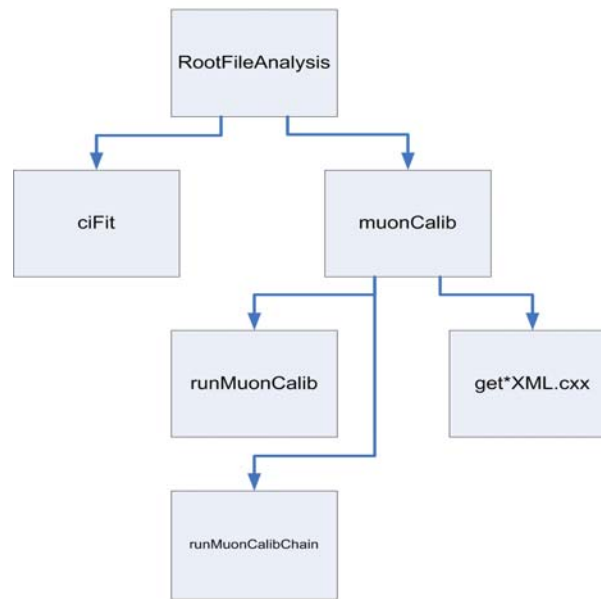


Figure 1: Class dependency tree for RootFileAnalysis

The RootFileAnalysis class dependency tree is shown in xxx.

RootFileAnalysis is a generic class containing:

- Digi/recon/mc root file input
- The event collection – allows event/playback functionality.
- Chain structure which allows single or multiple file runs

The muonCalib class contains:

- muon specific Histograms
- muonCalib data arrays
- muonCalib text/XML file output
- all muon calibration algorithms

The get*XML.cxx routines are standalone routines that read in text files and write out XML. These functions have also been added to muonCalib, hence these routines provide backward compatibility only.

8.2. Overall Flow

The overall flow of calibGenCAL is controlled by the runMuonCalib application as shown in Figure 3 and Figure 4. A pseudocode description of the process is presented in Section 0

8.2.1. Inputs

Inputs consist of one or more digiRoot file containing the data and an options file (not shown in the figures). The options file (muonCalib_option.dat) contains file names for input and output files (see Figure 2 for a sample). If multiple input files are desired, add more file names to first line, separated by spaces. The input files will be concatenated.

```
D:\GLAST_DATA\ebf031005015905_new.root d:\GLAST_DATA\ebf031005040315_new.root
../output/muped.txt
../output/mucorpped.txt
../output/muslope.txt
../output/mupeak.txt
../output/Histograms_ped_rat.root
../output/Histograms_peak.root
../xml/pedestals.xml
../xml/mucorpped.xml
../xml/muslopes.xml
../xml/light_asym.xml
../xml/gains.xml
```

Figure 2: Sample muonCalib_option.dat file. The first line is the input digiRoot file, the remainder are for various outputs, both in text and XML formats.

8.2.2. Phase I

The calibration process is iterative, as shown by the presence of two phases of operation. After performing data selection on the basis of CAL-determined muon track, Phase I (Figure 3) computes pedestals and performs a linear fit to the asymmetry measured from the central half of each crystal. The slope of the resulting best fit line (muSlope) is a conversion factor from measured asymmetry to position. The normalization of the line yields a first estimate of gain (muPeak), under the assumption that the gain is the same for both ends of any given crystal.

8.2.3. Phase II

Phase II (Figure 4) uses the results (pedestals, muSlopes, muPeaks) from Phase I, then the Phase I data set to produce an asymmetry table (asymmetry vs position with crystal width granularity) which is written out as text and XML files. It also collects a gain, position, and path-length corrected pulse height histogram and fits a Landau model to the muon peaks measured at each end. The results of these fits yield more realistic muPeaks. Another iteration is performed so that small changes in data

selection due to the refined muPeaks can be taken into account. Finally, muPeaks are written out to text and XML files.

8.2.4. Note on readXXX() functions

The application is interspersed with several read() functions which read calibration data from text files that were created earlier in the application. This allows the developers to comment out earlier portions of the code and save processing time while they are testing later passes.

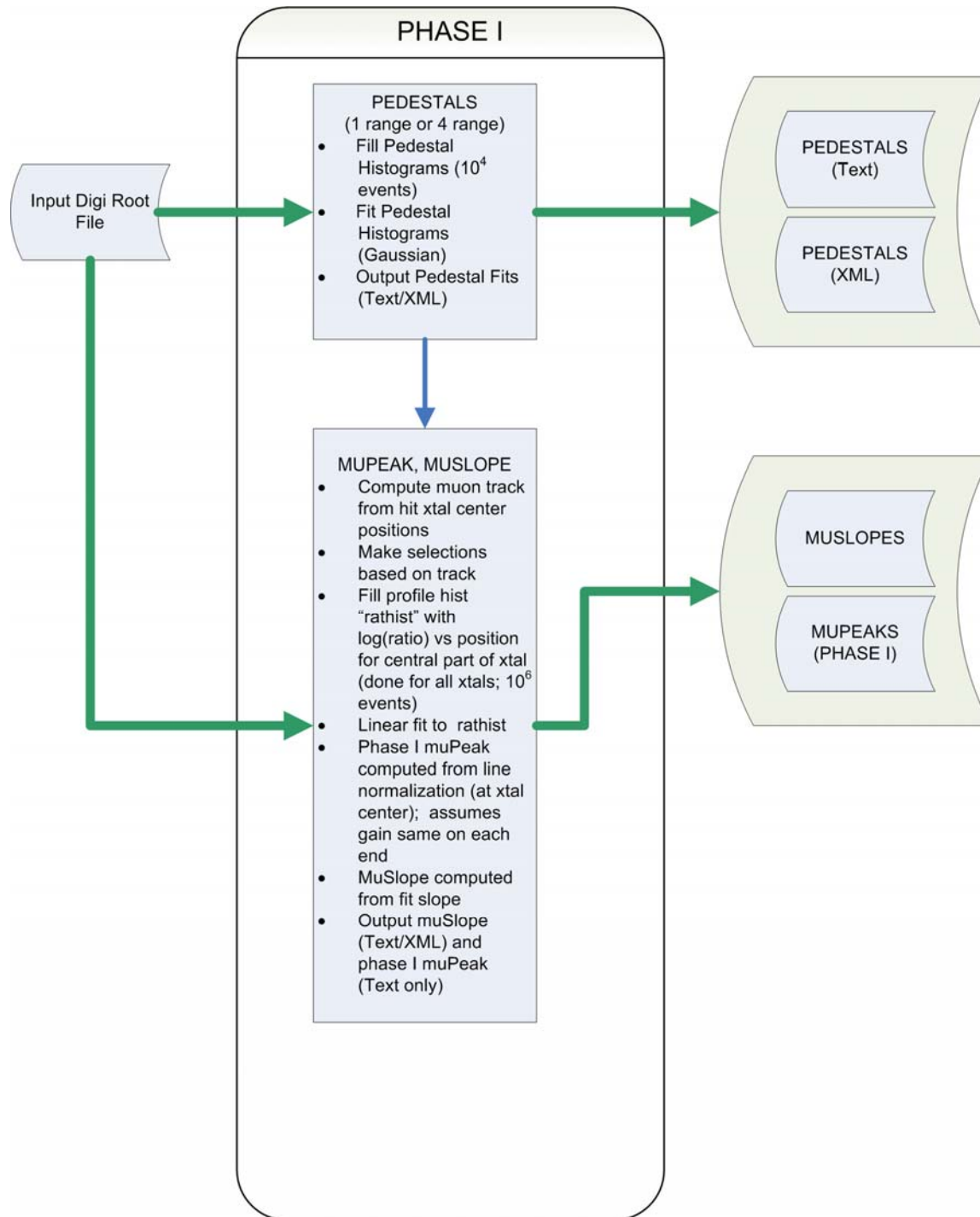


Figure 3: Phase I of muonCalib

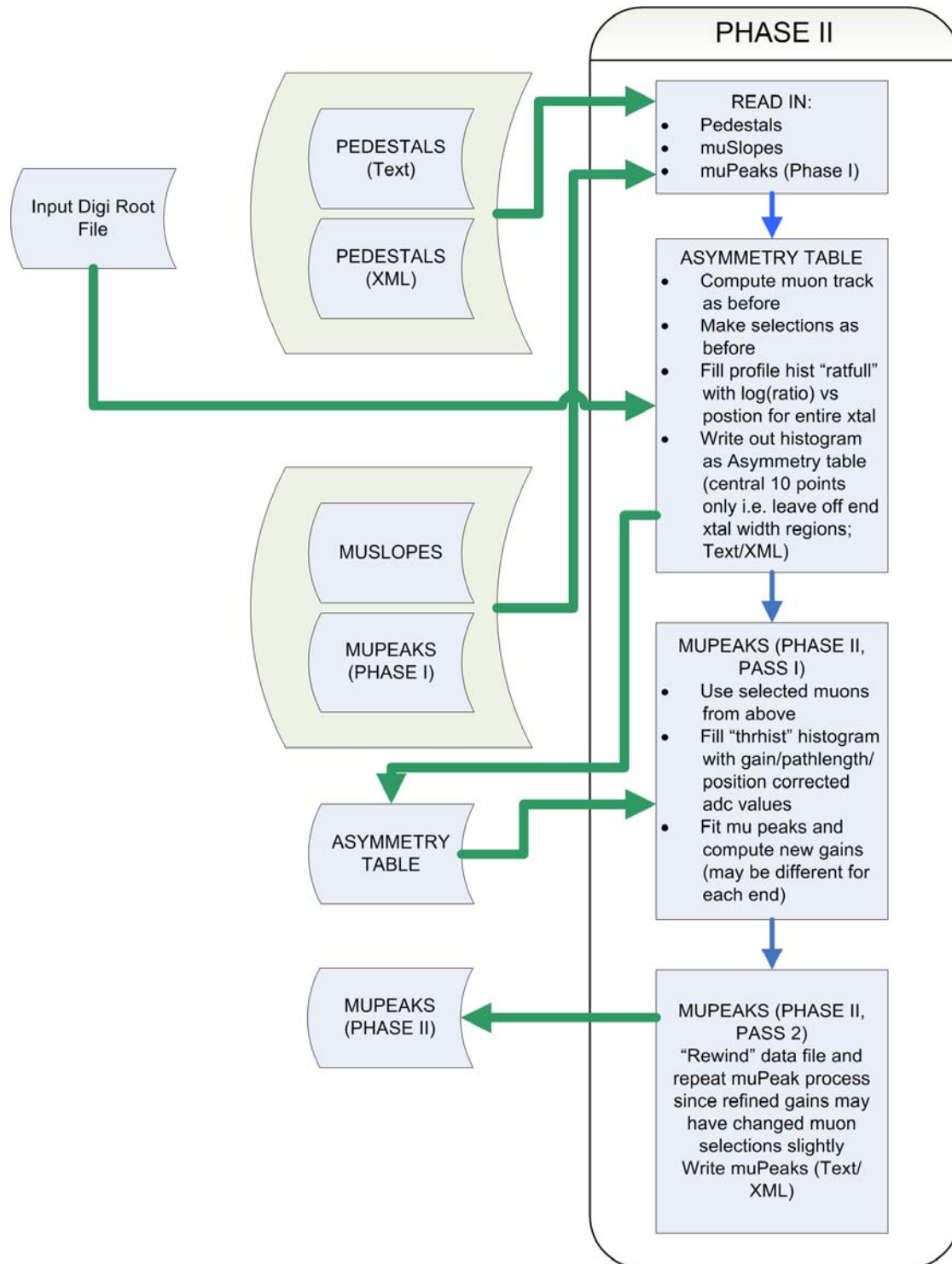


Figure 4: Phase II of muonCalib

8.3. Data Selection

The following track related data selections are applied to all muons:

- no more than 2 hits on any layer
- For pedestal, muPeak calibration of x layer crystals:
 - 4 x-layers containing exactly 1 hit each
 - 2 y-layers containing 1-2 hits each
 - Reverse for y-layer crystals
- For each projection (x and y), fit line using hit crystal center coordinates
- Line defines angles from vertical, θ_x and θ_y . For x-crystals, require:
 - $|\tan \theta_y| < 0.01$
 - $|\tan \theta_x| < 0.5$
 - Reverse for y-layer crystals

These selections result in muons that are close to vertical when viewed in the plane perpendicular to the crystals to be calibrated and within 45 degrees of vertical when viewed in the plane parallel to the crystals to be calibrated. This scheme results in better statistics than requiring muons to be vertical in both planes, and the varying path lengths along the crystal longitudinal direction are known and can be corrected using the angles from vertical defined above.

For asymmetry, we want just the opposite: We want the direction of the muons to be tightly constrained in the longitudinal direction, so we define a narrow range of longitudinal positions, but we don't care too much about constraining the muon track in the transverse direction. So we use the selection of muon described above but reversed i.e. the x-layer selection for muPeaks is used for the y-layer crystals for asymmetry and vice versa.

8.4. Products

8.4.1. Pedestals

In the first phase, pedestals are computed by accumulating pulse height histograms from crystals not hit by muons (10^4 events). The resulting pedestal peaks are fit with Gaussians. (The program is also capable of using the correlated Gaussian model for pedestal peak fitting, but this is not currently part of routine processing.) Pedestal results, including peak positions and widths, are stored in both a text file and an XML file.

A sample pedestal XML file for the first two crystals is shown in Figure 5 (pedestals.xml). The pedestal distribution peak and width (sigma) are in adc units. They are supplied for each range of each crystal end.

```

<?xml version="1.0" ?>
<!DOCTYPE calCalib SYSTEM "$(CALIBUTILROOT)/xml/calCalib_v2r1.dtd" [] >
<calCalib>
<generic instrument="EM" timestamp="2003-10-1-12:56" calibType="CAL_Ped"
fmtVersion="v3r3p2">
</generic>
<dimension nRow="1" nCol="1" nLayer="8" nXtal="12" nFace="2" />
<tower iRow="0" iCol="0">
  <layer iLayer="0">
    <xtal iXtal="0">
      <face end="NEG">
        <calPed avg="419.112" sig="5.71859" range="LEX8" />
        <calPed avg="225.457" sig="0.73131" range="LEX1" />
        <calPed avg="514.758" sig="4.32912" range="HEX8" />
        <calPed avg="257.9" sig="0.597832" range="HEX1" />
      </face>
      <face end="POS">
        <calPed avg="634.602" sig="5.78778" range="LEX8" />
        <calPed avg="203.673" sig="0.733355" range="LEX1" />
        <calPed avg="618.355" sig="4.35142" range="HEX8" />
        <calPed avg="238.426" sig="0.589113" range="HEX1" />
      </face>
    </xtal>
    <xtal iXtal="1">
      <face end="NEG">
        <calPed avg="576.902" sig="5.67201" range="LEX8" />
        <calPed avg="223.758" sig="0.72788" range="LEX1" />
        <calPed avg="645.682" sig="4.34322" range="HEX8" />
        <calPed avg="224.903" sig="0.592412" range="HEX1" />
      </face>
      <face end="POS">
        <calPed avg="551.953" sig="5.784" range="LEX8" />
        <calPed avg="219.657" sig="0.733635" range="LEX1" />
        <calPed avg="610.173" sig="4.38797" range="HEX8" />
        <calPed avg="217.313" sig="0.604085" range="HEX1" />
      </face>
    </xtal>
  </layer>
</tower>

```

Figure 5: Sample from Pedestals.xml file for first two crystals

8.4.2. MuPeaks

muPeaks are simply the most probable value of the Landau model used to fit muon peaks accumulated from selected data. They are stored together with a fractional width of the peak (i.e. sigma/mean). They are fit in Phase II from pulse height spectra collected from the central region of each crystal. The pulse heights are corrected for possible differences in gain between the crystal ends, for longitudinal position of the event and for path length of the muon track through the crystal. Fitting is done twice during Phase II. The first fit gives the relative gains between the two ends (remember that the “muPeak” stored during Phase I assume the ends are the same). The second fit

then iterates to account for changes in event selection and pulse height due to the refined gains from the first iteration. Results are written out in both text and XML files.

The muPeak XML file is slightly different than the text file in that it is referred to as “gains.xml” and contains 11.2 MeV/muPeak for each range for each crystal end. This quantity is not really a gain, but rather an adc bin width, actually the reciprocal of a gain. The fit is done strictly for LEX8, with the other ranges computed from known gain ratios.

Figure 6 shows a sample from a gains.xml file. The gain ratios in this sample are not necessarily flight values. The “gains” are in MeV/adc bin and the sigmas are fractional Landau peak widths.

```
<?xml version="1.0" ?>
<!DOCTYPE calCalib SYSTEM "$(CALIBUTILROOT)/xml/calCalib_v2r1.dtd" [] >
<calCalib>
<generic instrument="EM" timestamp="2003-10-1-12:56"
calibType="CAL_ElecGain" fmtVersion="v3r3p2">
</generic>
<dimension nRow="1" nCol="1" nLayer="8" nXtal="12" nFace="2" />
<tower iRow="0" iCol="0">
  <layer iLayer="0">
    <xtal iXtal="0">
      <face end="NEG">
        <calGain avg="0.0356871" sig="0.0685552" range="LEX8" />
        <calGain avg="0.321184" sig="0.0685552" range="LEX1" />
        <calGain avg="0.192711" sig="0.0685552" range="HEX8" />
        <calGain avg="1.73439" sig="0.0685552" range="HEX1" />
      </face>
      <face end="POS">
        <calGain avg="0.0349193" sig="0.066587" range="LEX8" />
        <calGain avg="0.314274" sig="0.066587" range="LEX1" />
        <calGain avg="0.188564" sig="0.066587" range="HEX8" />
        <calGain avg="1.69708" sig="0.066587" range="HEX1" />
      </face>
    </xtal>
    <xtal iXtal="1">
      <face end="NEG">
        <calGain avg="0.0341574" sig="0.0671152" range="LEX8" />
        <calGain avg="0.307417" sig="0.0671152" range="LEX1" />
        <calGain avg="0.18445" sig="0.0671152" range="HEX8" />
        <calGain avg="1.66005" sig="0.0671152" range="HEX1" />
      </face>
      <face end="POS">
        <calGain avg="0.0344676" sig="0.0672717" range="LEX8" />
        <calGain avg="0.310209" sig="0.0672717" range="LEX1" />
        <calGain avg="0.186125" sig="0.0672717" range="HEX8" />
        <calGain avg="1.67513" sig="0.0672717" range="HEX1" />
      </face>
    </xtal>
  </layer>
</tower>
</calCalib>
```

Figure 6: Sample from gains.xml file for first two crystals

8.4.3. *MuSlopes*

The muSlopes, as written to the XML file, are proportional to the inverse of the slope of the best fit line to the asymmetry vs position profile histogram. Since they are the inverse of the slope, they are a conversion factor from asymmetry to position, under the assumption that asymmetry for the central part of the crystal can be represented by a linear function. By convention, the muSlopes values are normalized to the (P-M)/(P+M) style of asymmetry. Since this has been supplanted by $\log(P/M)$ in the calculation, a factor of 2 has been introduced when the values are written out to XML (since old style asymmetry is normalized to the sum of the ends rather than the average).

In addition to their use for computing event position, and given the assumption that the light taper (variation with light with distance from the diode) is equal for each end (not strictly true), the muSlopes can be used to compute light taper and hence normalize events to the center of the crystal.

MuSlopes are stored for each crystal and range. Their units are mm/unit asymmetry. Figure 7 shows a sample from a muslopes.xml file.

```
<?xml version="1.0" ?>
<!DOCTYPE calCalib SYSTEM "$(CALIBUTILROOT)/xml/calCalib_v2r1.dtd" [] >
<calCalib>
<generic instrument="EM" timestamp="2003-10-1-12:56"
calibType="CAL_MuSlope" fmtVersion="v3r3p2">
</generic>
<dimension nRow="1" nCol="1" nLayer="8" nXtal="12" nFace="1" />
<tower iRow="0" iCol="0">
  <layer iLayer="0">
    <xtal iXtal="0">
      <face end="NA">
        <muSlope slope="749.548" range="LEX8" />
        <muSlope slope="749.548" range="LEX1" />
        <muSlope slope="749.548" range="HEX8" />
        <muSlope slope="749.548" range="HEX1" />
      </face>
    </xtal>
    <xtal iXtal="1">
      <face end="NA">
        <muSlope slope="618.747" range="LEX8" />
        <muSlope slope="618.747" range="LEX1" />
        <muSlope slope="618.747" range="HEX8" />
        <muSlope slope="618.747" range="HEX1" />
      </face>
    </xtal>
  </layer>
</tower>
</calCalib>
```

Figure 7: Sample from file muslopes.xml for first two crystals

8.4.4. *Asymmetry*

The asymmetry product, stored in light_asym.xml, is just the histogram of asymmetry ($\log(P/M)$) vs position for LE and HE diodes for each crystal. The current product has identical values for the two

diodes, but this will change in v3. Each table contains ten numbers representing ten positions at the centers of the orthogonal crystals, excluding the end crystals. Currently, the read routine for the table linearly extrapolates from the last two positions to get values at the ends. Figure 8 shows a sample of a `light_asym.xml` file for the first two crystals.

```
<?xml version="1.0" ?>
<!DOCTYPE calCalib SYSTEM "$ (CALIBUTILROOT)/xml/calCalib_v2r1.dtd" [] >
<calCalib>
<generic instrument="EM" timestamp="2003-10-1-12:56"
calibType="CAL_LightAsym" fmtVersion="v3r3p2">
</generic>
<dimension nRow="1" nCol="1" nLayer="8" nXtal="12" nFace="1" nRange="2"
/>
<tower iRow="0" iCol="0">
  <layer iLayer="0">
    <xtal iXtal="0">
      <face end="NA">
        <lightAsym diode="LE" error="0.03"
          values=" -0.351856 -0.268391 -0.186048 -0.109585 -0.0366382
0.0349797 0.110767 0.187158 0.261037 0.338278" />
        <lightAsym diode="HE" error="0.03"
          values=" -0.351856 -0.268391 -0.186048 -0.109585 -0.0366382
0.0349797 0.110767 0.187158 0.261037 0.338278" />
      </face>
    </xtal>
    <xtal iXtal="1">
      <face end="NA">
        <lightAsym diode="LE" error="0.03"
          values=" -0.320841 -0.273547 -0.213502 -0.144366 -0.0601645
0.0450107 0.143692 0.226079 0.292944 0.349141" />
        <lightAsym diode="HE" error="0.03"
          values=" -0.320841 -0.273547 -0.213502 -0.144366 -0.0601645
0.0450107 0.143692 0.226079 0.292944 0.349141" />
      </face>
    </xtal>
  </layer>
</tower>
</calCalib>
```

Figure 8: Sample from a `light_asym.xml` file showing values for the first two crystals

9. calibGenCAL v2r5 Use

CalibGenCAL v2r5 is extremely simple to use. After build within the appropriate EM environment, an options file containing input and output file specifications must be created and named `muonCalib_option.dat` (see Figure 2 for an example). Then the `runMuonCalib` application is executed to perform the calibration analyses.