

CCB Action 20060515

Code Versions

Engineering Model (sim/recon): [v6r070329p16](#)

This release uses calibrated ACD quantities. In addition, the ACD VETO threshold is changed back to 0.3 MIP (Real data from April and May use both 0.1 and 0.3), and the CNO threshold is set to 20 MIPs (data threshold). Run and event ID in AnalysisNtuple are doubles to avoid truncation for real data. We also now have better error decoding (can distinguish TKR FIFO full errors from the generic TEM error for example). The first step on a long road

[Differences wrt the previous pipeline EM release \(EM v6r070329p5\)](#)

New tags:

Event v11r8em4: I

- EAC Fixes AcdHit tile ID not set bug.

xmlGeoDbs v1r29gr2: R

- hmk Merge VDG updates from Anders into this branch for EM
- jrb Update requirements and this file correctly
- jrb Update requirements and this file correctly (I hate having to do this by hand)
- jrb Merge in 2 little fixes from head
- jrb Get rid of ACDTop composition; put contents in new fragment ACDTop.xml

merit v6r28p5: E

- THB fix FT1Energy; new define required by OnboardFilter on Win32
- THB FT1 mod to set pointing stuff for background
- THB fix FT1Livetime

ldfReader v3r19p2: A

- hmk Forgot to add ErrParser.cxx
- hmk print eventId and apid when reading LSF with EbfDebug set
- hmk Store TKR fifo errors in all their glory and update
- hmk Eventflags to include common error bits in error contribution
- hmk Modify ACD parsing to create AcdDigi objects during the header method rather than just during PHA. This takes care of the unlikely occurrence of veto latching occurring after GEM.

enums v0r6: L

- HMK Check in new include DetectorConstants.h
- HMK New eventflags for error contribution

idents v2r17: L

- jrb Add a constructor and some get fctns to AcdId
- eac add AcdGapId

commonRootData v2r8: Y

- EAC Added AcdMap to convert back to electronics space

calibUtil v1r6: H

- jrb Add new dtd for Acd calibs & a couple sample files

calibGenCAL v3r9p6: A

- fewtrell: fixed stupid bug in MuonCalib::readAsymTXT(), probably introduced when I added tower info to output txt files in v3r9
- dwood/fewtrell XML parser - switched base XML parser from python standard (minidom) to PyXML package (4DOM)
- dwood/fewtrell allows for DTD validation:
 - DTD validation is now enabled for XML files required to have DTD information prefixed
 - DTD validation still disabled for XML files which have no DTD yet
 - better for quality control, but quite a bit slower
- dwood/fewtrell adcsmooth - numerous changes to work correctly with sparsely sampled characterization data
- dwood/fewtrell extrapolate data near the pedestal noise region; promotes sharp cutoff at low end of curves
- dwood/fewtrell LAC pedestal noise is zeroed out:
 - smoothing filter does not run at endpoints of curves; promotes sharp cutoff
 - better extrapolation of FLE coarse range values past measurement range
 - special processing for ULD data to reconstruct saturation plateau with sparse data
- dwood/fewtrell charplot - new utility to plot raw and smoothed characterization data. Mainly a tool to diagnose the output of adcsmooth
- dwood/fewtrell charVal - new validation application for characterization data XML files. Will catch gross errors, but probably still needs work to improve validation criteria
- dwood/fewtrell dacVal - new validation application for DAC settings XML files:
 - works backwards from DAC settings to calculate threshold in energy units
 - requires a complete set of files to run
 - nothing yet for ULD
- dwood/fewtrell adc2nrgVal - fixed bug with ROOT output option
- dwood/fewtrell xxxXML2TXT.py - new scripts generate columnar text files from offline Cal calibration files
- dwood/fewtrell dacDiff.py - generates per channel differences between any 2 cal dac settings snapshot files.
- dwood/fewtrell roothist2csv.py - generates columnar text files from all histograms in ROOT file.
- dwood/fewtrell dumpROOTPlots.py - generates images files for all histograms & Canvases in ROOT file.
- dwood/fewtrell unit_test - now 2 tower instead of 8 to save processing time
- dwood/fewtrell runSuiteParallel.py - now supports commandline override of twr_id as well as each individual processing phase. Allows work to be easily broken up into several threads w/ same cfg file.
- dwood/fewtrell C++ - TXT format output now supports new format used by python scripts. improved format includes tower information & contains all 4 asym types & both MeVPerDAC types into single file each.
- fewtrell C++ apps now use TH1S instead of TH1F across board. Results are the same & it saves memory
- fewtrell future updates w/ 16 tower support will use a lot of memory so this is important step. C++ apps now use float instead of double in most places. the precision is not needed & it saves some memory, particularly in larger arrays.
- fewtrell All python scripts now have .bat & .sh shell launcher scripts. All .sh scripts have executable bit set.
- fewtrell runSuiteParallel.py, gensettings.py, build_adcsmooth.py all generate appropriate validation outputs along w/ their primary output.
- dwood/fewtrell For DAC settings XML file produced from genXXXsettings tools, modified the values of the element 'hierarchy' and 'shape' attributes. Hierarchy is set to the value ['GCCC','GCRC','GCFE','J'] and shape is set to the value '(8,2,12)'.
- dwood/fewtrell Added support to tools for handling LEX1 energy range FLE DAC characterization files. Currently, a simple scale factor is used to convert to LEX8 ADC units and process as before.
- dwood/fewtrell Better file type checking for files which use the ADC/DAC XML tables as input.
- dwood/fewtrell Muon asymmetry now smarter for cross diode asymmetry. Uses average ratio of Large2Small diode signal instead of the slope. This will make energy & position response @ 1 GeV +- 100 MeV better.
- dwood/fewtrell runCIFit can now process broadcast mode calibGen as well as the older, 1 column at a time method.
- dwood/fewtrell uldVal: new application to validate ULD DAC settings XML files (from genULDsettings)
- dwood/fewtrell dacVal:
 - fixed report of FAILED tests
 - relaxed FHE error and warning limits
- dwood/fewtrell charVal:
 - added support for ULD characterization files
 - added check for incorrect pedestal subtraction
 - relaxed FHE and LAC error and warning limits
- dwood/fewtrell build_adcsmooth: added call to charVal to do validation on ULD characterization files
- dwood/fewtrell gensettings:
 - added call to uldVal to do validation on ULD DAC settings files
 - fixed bug which put 'cp' instead of 'copy' in Windows BAT output files
 - made processing of muon gain FHE files optional; command line -m or --muon
- dwood/fewtrell build_tholdci_cfg: make sure FHE threshold is properly converted from GeV to MeV
- dwood/fewtrell runSuiteParallel: fixed undefined environment variable error
- dwood Added titles and axes labels to ROOT output graphs generated by plot and validation scripts
- dwood genULDsettings - removed one DAC step safety margin; output margins are closer now to user requested value
- dwood uldVal:
 - look at all energy ranges when comparing threshold to saturation level
 - add check to make sure (saturation - threshold) is not less than margin paramter
 - tholdciVal - removed check of FLE threshold verses LEX8 ULD value; not needed now that FLE data is LEX1
- fewtrell fixed bug introduced w/ broadcast mode in v3r9p

calibGenACD v1r4p1: T

- EAC Fix a spurious include in cmt/requirements
- EAC Added runMuonCalib_Svac.cxx to run mip peak calibrations on SVAC ntuple files.
- EAC Make changes so that xml format agrees with the dtd file that joanne has
- EAC Remove a dead src/runMuonCalib_Roi.cxx file which was moved to apps subdir
- EAC Initilize pedestal file to zero to prevent crash if it don't exist
- EAC Fairly big update, added AcdMeritCalib, changed the structure a bit.
- EAC Fairly big update, added AcdVetoCalib, changed the structure a bit.
- EAC Minor update. Add information about the datasets used to generate calibrations to the txt and xml output files. Also, added apps/ subdir for application main() routines.
- EAC Major re-write. Brings in some code from the ACD. Uses polynomials to smooth pha distributions and find the peak values. Uses a simple algorithm to decide what range to fit over. Supports both ped-subtracted and raw values.

digiRootData v9r5: E

- HMK Add Tkr FIFO Full bits to Error class

Rootlo v17r9gr7: C

- HMK TKR FIFO errors from LDF

LdfEvent v3r7: O

- HMK store TKR fifo errors from err contribution

LdfConverter v2r7: N

- HMK store error contribution in eventFlags and really store TKR fifo errors in their full glory

TkrRecon v10r8p1em0: F

- LSR Fix the Vtx status bit thing for latest EM

LatIntegration v2r58: L

- hmk Patch calibration.txt JO file for AcCalibSvc
- awb Added JO so we get CTB variables in data and MC.
- awb Added LICOS/FSW LPA run for system test.
- awb More ACD calib JO updates. Removed 'LivetimeSvc.Deadtime'.
- awb Added 'ideal' as valid calibration flavor for MC (for the ACD simulations - for now).
- awb Turn off LivetimeSvc in ldf2digi.
- awb Updated single tower VDG source location.

CalibSvc v0r21p8em0: U

- jrb Merge in ACD calibration support

CalibData v0r12p3em0: E

- jrb Merge in support for Ac calibrations

CHS/eventFile v1r0p3: N

- RBL added dumpEvent single-dump utility

AnalysisNtuple v2r9p3gr7: C

- LSR EvtRun and EvtEventId to double
- (awb) This is just an emergency fix for GRINF-29. The correct solution is to make the ULong64_t.

AcUtil v1r1p3em4: E

- EAC "ideal" now uses built-in code instead of file in calibDB
- EAC Added apply_pattern use_CalibData_symbols for windows
- EAC Work some obscure magic so that windows understands what to do
- EAC Pull in updates from branch EM-v6r070329p0, tag v0r2p10em0
- EAC Added AcCalibSvc: So far this handles pedestals and gains (aka mip peaks). Adds a dependency of CalibData
- HMK updates for CLHEP 1.9.2.2
- Along Branch EM-v6r070329p0:
 - EAC No really, actually tag the stuff this time
 - EAC And tag correct versions of cmt requirements
 - EAC And tag correct versions of the dll stuff
 - EAC pull in requirements file from main branch for AcCalibSvc
 - EAC pull in calibration stuff from main branch
 - EAC Cache transform to local coords in AcTileDim and AcRibbonDim

AcRecon :v3r0em8 !

- EAC Arggh fix some missing bits from v3r0em7
- EAC Pull back in calibration stuff, which seems to have gotten lost in v3r0em6
- EAC Pull in a bug fix to the AcGap stuff
- EAC Pull in use of AcCalibSvc to calibration hits from main branch

AcDigi v1r14p2em3: !

- HMK Set veto threshold hold back to 0.3 MIP and update CNO threshold to 20 MIPs as it probably always should have been

System Tests for this [version](#)

System Tests results:

Differences in vdg_oneTower test due to changes in VDG source location. Changes in VerticalGamma100MeV due to ACD veto change from 0.1 MIP to 0.3 MIP: All trigger bits with the ROI are lower thus changing the total number of events in the other histograms. The current system tests do not include data with the full LAT (including the calibrated ACD) yet. I've done additional checks running recon myself and looked at the ACD variables in the SVAC ntuple.

The test all_gamma_sixTower has a frontend problem - only the first nine histograms are visible and the statistics page is blank. I've opened a JIRA. The plots for SixTower_Data and TowerA_Data are missing for EM p16. I looked at the ones for EM p14 (wrt EM p5). No differences.

Fred version:

[v0r99](#)

Pipeline version:

v1.4.3

GRITS tag (web browsing and task configuration)

glast-ground v0r3p7
grits-gino-web version 0.55 (v0r5p5)
grits-gino version 0.95 (v0r9p5)
grits-gino-xml version 1.42 (v1r4p2)
grits-common version 0.32 (v0r3p2)

online/svac (task defs, scripts):

pipeline tasks:

online: v2r4p1

svac pipeline code and tasks:

code/tasks v3r6p0:

Make LicosLaunch die if run directory isn't in rawData.

More & better timestamps in recon chunk script.

Put output from each chunk in a separate subdir of staging dir.

Merge chunks into staging dir, then move merged files to destination.

Merge recon chunks before merit & cal.

Use new AFS staging dir.

Switch rootData to u34.

Split recon/recon into 6 processes to get better rollback granularity.

When recon of some chunks fail, save the output from the successful ones and don't re-run them.

Don't do merge on single-chunk recon runs. # Disabled! - do do merge.

All file copies & moves performed by recon are attempted up to 5 times, then fail (used to try until the job timed out on copies to local disk, and only once for moves away from local disk or staging).

pipelineDatasets v0r3

ISOC code and tasks:

v0r5p0

Apps that run in pipeline:

eLog: v2r3p1

ConfigTables: v3r2p1

TestReport: v3r6p4 (digi & recon reports): EAC and awb

- Added ACD electronics space and ACD trigger quantities. Also improved descriptions.
- Protect code against time hacks being the same in two events.
- Get GEM discarded counter from context information instead of directly from the GEM.
- Added number of prescaled and dead zone events. Added histogram with delta deadzone events.
- Added check on the consistency between the GEM sequence counter and the number of events in the digi file.
- Added trigger/deadzone/discarded rates for 30 time intervals. In case of problems it may tell us where in a run it happened and speed up troubleshooting.
- Added consistency check on all extended counters so we'll know when the go backwards.
- Added much better error decoding. Will now know what the TEM errors are (TKR FIFO full or not).
- Added trigger arrival time histograms.
- Added ACD MIP peak histograms.

EngineeringModelRoot: v2r2 (SVAC tuple)

- Added a few useful AcdMips variables (sum and max and error flag).
- Added more error flags.
- Updated documentation.

Approved: unanimous (Bill not available) - 5/19/06

Added 06.01.2006 for book keeping purposes (for the moment): Bryson's CHS task

- Bryson IExternal/fsw v0r069 -- incorporates DFI, etc. from FSW B0-6-9 with the 24/25-bit fix for the livetime counter.
- Bryson CHS/eventFile v1r0p9 -- bunch of tweaks mostly to rqts file for building on Windows.
- Bryson CHS v2r2 -- updated checkout package to pull in these two new tags.