

MeritTuple status, proposals

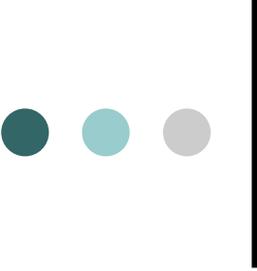
Elimination of now-redundant variables

Double to float (or int)

Arrays in the tuple

GEM status word bits set in MonteCarlo

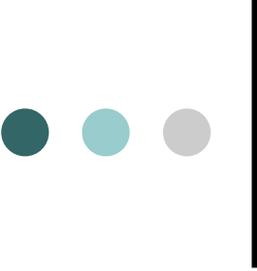
LiveTime



Redundant variables

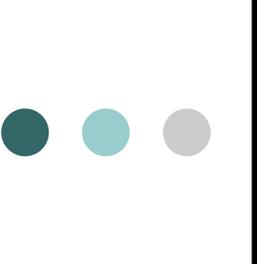
- Event-level variables set in merit are now also set in AnalysisNtuple::EvtValsTool and McValsTool. I propose, on an agreed-to time schedule, to eliminate these obsolete variables before DC2

Current (DC1)	Replacement (DC2)
Event_ID	EvtEventId
Run	EvtRun
elapsed_time	EvtElapsedTime
MC_src_Id	McSourceId



double to float (and/or int)

- Why is the MeritTuple all double now?
 - Basically, because all the AnalysisNtuple member variable set by the various ValsTools classes are double, and the visitor can only pass pointers to these
- What has changed?
 - Leon has changed the class to handle floats and ints, and the visitor now has three callbacks.
- Advantages
 - Factor of two (even compressed) in storage, download times
- Any downside?
 - The conversion program to Insightful Miner may need to be changed to use the ROOT Value() function.
- One exception: EvtElapsedTime must be double since it is copied to FT1 from the tuple.

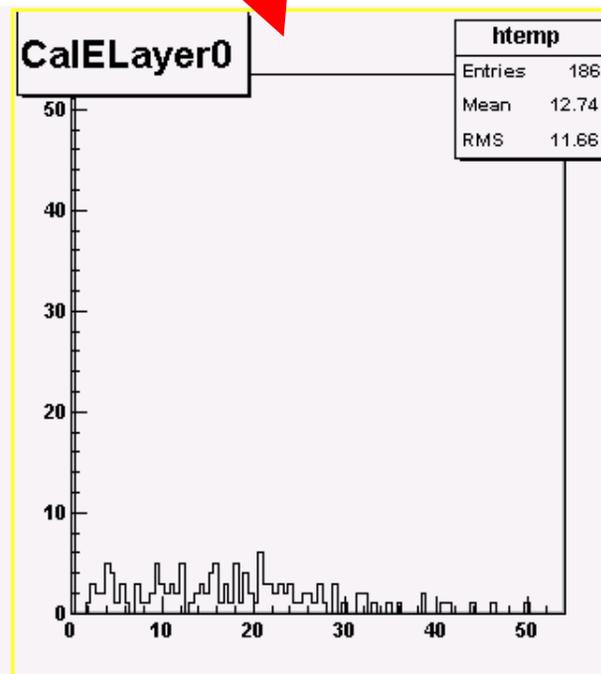
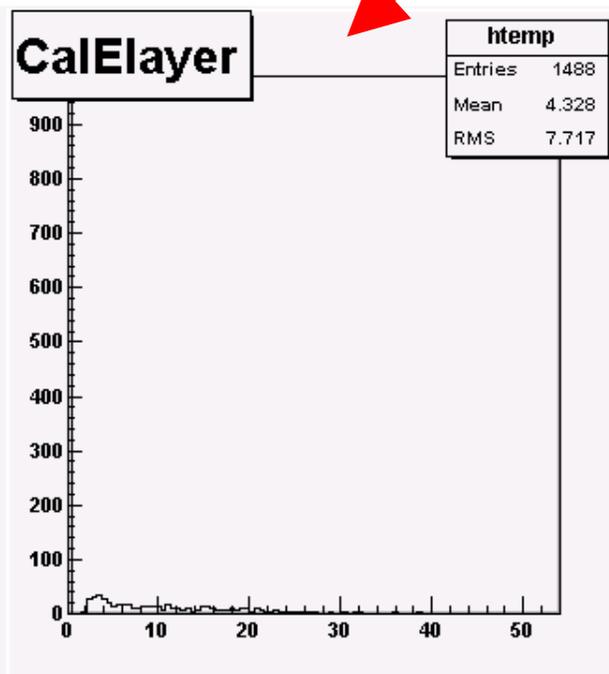


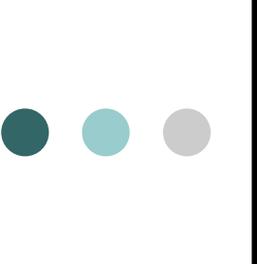
Arrays (constant length)

- Supported by ROOT and RootTupleSvc:
 - Simply add “[*n*]” to the variable name, where *n* is a constant integer, and that is it!
 - Pointer is then interpreted as being to an array of the pointer type.
- Advantages:
 - Economy of variable names
 - Possibility of implied loop over the index
- Example: CalElayer[8] (see next slide)
- Proposal
 - Add the array definition(s) as aliases initially, consider need for specific entries (like CalELayer5).

Array test example

- CalTrackAngle
- CalLayer2
- CalLayer7
- CalTwrGap
- CalELayer3
- CalLyr0Ratio
- CalELayer[8]
- CalELayer4
- CalLyr7Ratio
- CalELayer0
- CalELayer5
- CalBkHalfRatio
- CalELayer1
- CalELayer6
- CalXtalsTrunc



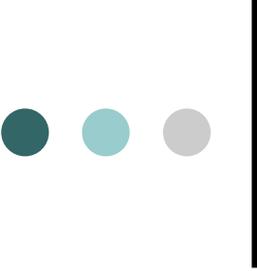


Gem condition summary word

The summary word bit definition is

bit 0	ROI	Meaning depends on whether ACD is being used as veto or trigger
bit 1	TKR	OR of 3-in-a-row for each tower
bit 2	CAL (LE)	OR of CAL low energy for each tower
bit 3	CAL (HE)	OR of CAL high energy for each tower
bit 4	CNO	OR of 12 ACD CNO inputs
bit 5	Periodic	set for periodic trigger
bit 6	Solicited	set for solicited trigger.

For MC events, we are now setting the bits in the tuple word GemConditionSummary corresponding to TKR, CAL(LE), CAL(HE) and CNO.

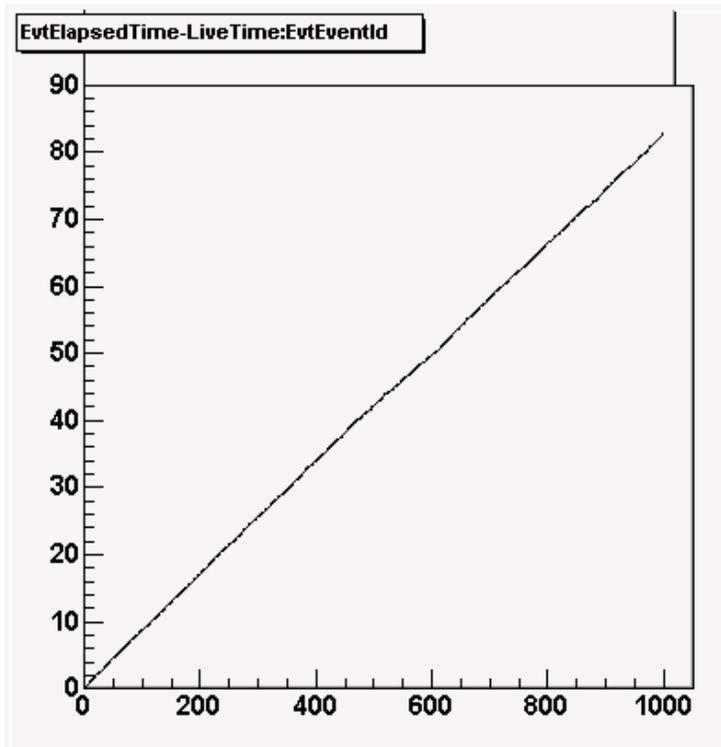


Live Time

- New temporary variable: *LiveTime*
 - Will be named *EvtLiveTime*, if/when it is eventually set in `AnalysisNtuple::EvtValsTool`.
 - Definition: cumulative live time up to current event
- MC implementation, in `TriggerAlg`
 - Add elapsed time from previous trigger, less the value of *TriggerAlg.deadtime*.
 - If an event occurs sooner than this, do not set new trigger bit. (is there ever a reason to keep such an event?)
- LDF implementation
 - Not set yet.

Plot example

- Set TriggerAlg.deadtime to 0.1 s (like EGRET), require pass dead time trigger condition:



Shows the accumulated deadtime increasing with event id. Total is 83 s corresponding to 829 triggers.