

# Randoms Overlay

## Object of the Exercise:

Upgrade simulations to use as-measured noise and pileup. These can be obtained from periodic triggers, so instead of generating noise from whole cloth we would read in periodic triggers and overlay them on top of the simulated events.

The first step of the above is to employ the following strategy:

1. Generate/simulate a Monte Carlo event and run the "standard" digitization stage for each subsystem
2. Run an algorithm which will determine the appropriate bin from which to select a random event and then read that in
  - input format is standard digi's
  - input digi's will go into a "special" section of the TDS to be managed separately
3. After this, a series of subsystem specific digi "merge" algorithms run to merge the input overlay digis into the the simulated digis
  - If overlay hit is unique, simply add it into the collection
  - If overlay hit is shared then attempt to "add" it on top of the simulated hit (subsystem dependent!)
  - Mark the overlay hits so they can be identified during reconstruction/analysis, if desired.
4. Run a special step which attempts to adjust the trigger information (in particular gem bits)
5. OnboardFilter/Reconstruction/Analysis then proceed normally
  - Note that in order to move on to ghost identification and removal we will need a CAL clustering algorithm will need that can identify *multi ple* clusters. (This is not needed for the limited objective of redoing the Pass6 IRFs.)

## Subsystems

### ACD

(in progress)

### CAL

Digs are merged by a Gaudi algorithm "CalDigiMergeAlg". This algorithm simply reads in overlay digis from the special section of the TDS and then "merges" them into the existing Cal Digis from the simulation. If a CalDigi does not exist for a log, then the overlay digi is copied into the list, if a CalDigi exists in both simulation and overlay, then the adc counts are updated for the simulated digi - if the ranges are the same then they are summed (and cutoff at 4095), if the ranges are different then the adc value corresponding to the greater range is used. A status word is included which sets an "overlay" bit for the overlay digis.

### TKR

Similar philosophy to the above

### GEM

This needs to be merged to the extent that downstream analysis depends on the Gem bits. A specific example is ghost filtering. For this to work we need to merge the tkrVector bits, which give the individual TKR tower triggers. This tkrVector merge is done.

## Access to the data: digi files and TDS

Events to be overlaid are taken from periodic triggers in the data files. These are grouped into collections according to some distribution, where Mark Strickman and Eric Grove advocate the following:

Divide large sample of real periodic triggers into Mcllwain L bins (size of bins left as an exercise for the reader). Why Mcllwain L? Because background is generally more linear in Mcllwain L than in rigidity or magnetic latitude.

Implementing this requires a controlling algorithm which is able to retrieve the necessary information for determining the correct bin to use, then to handle the opening/closing/handling of the correct input file, then, finally, to "randomly" select an event from this file. The strategy is very similar to that used in the existing interleave package.

Done! (See update below...)

## Generating a Library of events

First pass done... 434K periodic trigger digis in 25 Mcllwain-L bins. (See below.)

## Issues to be dealt with

In general, merging digis will require interaction with calibration constants. Unfortunately, calibration constant updates occur at the beginning of the event, tying them to the time stamp associated with the simulated event and not with the timestamp associated with the input overlay event. In the short term, this issue can be dealt with by making sure that the simulated event time corresponds to the epoch of the overlay events, and we should strive to make sure the overlay event libraries correspond to a single calibration constant epoch (which, currently, should be the case). In the long term we need to understand how to solve this problem and, after some discussions with Joanne Bogart, this could involve some significant alterations to the calibration constant system.

An alternate approach is to make a new data object, derived from the digis, but with calibrations applied. This would give us physics-type data which would be independent of calibrations. In the process we might be able to compress the information a bit to save on disk storage, but this isn't the highest priority.

## **Latest progress**

[C&A Oct. 27, 2008](#)  
[C&A Nov. 20, 2008](#)  
[C&A Dec. 1, 2008](#)  
[C&A Dec. 4, 2008](#)  
[C&A Dec. 15, 2008](#)  
[C&A Jan. 8, 2009](#)  
[C&A Jan 26, 2009](#)