

Start Counter

A.S. & Giacomo Traini

Relevant files

→ Geometry constructor

- TASTbase/TASTparGeo. Handles what is needed both by Event display and FLUKA.

→ Definition

- Reconstruction/level0/geomaps/TASTdetector.map
- quite easy / trivial. No 'custom' detail on the cage is added. The mylar windows are added by hand

→ Global positioning is done in:

- Reconstruction/level0/geomaps/FOOT_geo.map
 - StartBaseName: "ST"
 - StartPosX: 0. StartPosY: 0. StartPosZ: -29.
 - StartAngX: 0. StartAngY: 0. StartAngZ: 0.

Getting ready for data

→ Handling the mapping:

- TASTparMap.cxx : to be used to define board/channel etc etc in the WaveDAQ interface
- TASTactDatRaw.cxx:
 - takes as input the data fragment : `AddDataIn(p_datdaq, "TAGdaqEvent");`
 - use the TASTparMap to decode it
 - fills the TASTdatRaw object that holds the HW infos: charge, time, channel, trig time
- TASTactNtuRaw.cxx
 - to be coded... Not sure if really needed... Can be used to take as input the dat raw and provide as output some more elaborate quantities.

Decoding data

- ➔ To be coded.
- ➔ For now:
 - `const WDEvent* evt = static_cast<const WDEvent*> (p_datdaq->GetFragment(i));`
 - `DecodeHits(evt);`
- ➔ This solution is not optimal: we read the WDEvent fragment twice (here and in the ToF act) and we loose some time.
- ➔ For now this was used to speedup the coding... But as for now neither the ST nor the TW proceeded... We have to rediscuss the strategies today...