```
⊢ P Events;1
      EventAuxiliary
      imb::MCParticlesimb::MCTruthvoidart::Assns | largeant | PDSIMCHAINREF.
      art::TriggerResults_TriggerResults_PDSIMCHAINREF.
      art::RNGsnapshots rns PDSIMCHAINREF.
       sim::OpDetBacktrackerRecords PDFastSim pvs PDSIMCHAINREF.
   Sim::SimEnergyDeposits_IonAndScint__PDSIMCHAINREF.
          🗽 sim::SimEnergyDeposits_IonAndScint__PDSIMCHAINREF.present
          🗽 sim::SimEnergyDeposits_IonAndScint__PDSIMCHAINREF.rangeSetID

☐ ✓ Kim::SimEnergyDeposits_IonAndScint__PDSIMCHAINREF.obj

             sim::SimEnergyDeposits_IonAndScint__PDSIMCHAINREF.obj.numPhotons
             🔖 sim::SimEnergyDeposits_IonAndScint__PDSIMCHAINREF.obj.numElectrons
             sim::SimEnergyDeposits_IonAndScint__PDSIMCHAINREF.obj.scintYieldRatio
             sim::SimEnergyDeposits_lonAndScint__PDSIMCHAINREF.obj.edep
             🔖 sim::SimEnergyDeposits_IonAndScint__PDSIMCHAINREF.obj.startPos.fCoordinates.fX
             🔖 sim::SimEnergyDeposits_IonAndScint__PDSIMCHAINREF.obj.startPos.fCoordinates.fY
             🔖 sim::SimEnergyDeposits_IonAndScint__PDSIMCHAINREF.obj.startPos.fCoordinates.fZ
             🔖 sim::SimEnergyDeposits_IonAndScint__PDSIMCHAINREF.obj.endPos.fCoordinates.fX
             sim::SimEnergyDeposits_IonAndScint__PDSIMCHAINREF.obj.endPos.fCoordinates.fY
             sim::SimEnergyDeposits_IonAndScint__PDSIMCHAINREF.obj.endPos.fCoordinates.fZ
             💸 sim::SimEnergyDeposits_IonAndScint__PDSIMCHAINREF.obj.startTime
             💸 sim::SimEnergyDeposits_IonAndScint__PDSIMCHAINREF.obj.endTime
             💸 sim::SimEnergyDeposits_IonAndScint__PDSIMCHAINREF.obj.trackID
             💸 sim::SimEnergyDeposits_IonAndScint__PDSIMCHAINREF.obj.pdgCode

☐ ... / sim::SimEnergyDeposits_largeant_LArG4DetectorServicevoITPCActive_PDSIMCHAINREF.

      Xim::SimEnergyDeposits_largeant_LArG4DetectorServicevoITPCActiveOuter_PDSIMCHAINREF.
      sim::SimPhotonsLites_PDFastSim_pvs_PDSIMCHAINREF.
      simb::MCParticles_largeant__PDSIMCHAINREF.
      # art::TriggerResults_TriggerResults_SinglesGen.
      art::RNGsnapshots_rns__SinglesGen.
       Ksimb::MCTruths generator SinglesGen.
```

```
SimEnergyDeposit(int np = 0,
         int ne = 0,
         double sy = 0,
         double e = 0.,
         geo::Point t start = \{0.,0.,0.\},
         geo::Point t end = \{0.,0.,0.\},
         double t0 = 0.
         double t1 = 0.,
         int id = 0,
         int pdg = 0)
  : numPhotons(np)
  , numElectrons(ne)
  , scintYieldRatio(sy)
  , edep(e)
  , startPos(start)
  , endPos(end)
  , startTime(t0)
  , endTime(t1)
  , trackID(id)
  , pdqCode (pdq)
  « end SimEnergyDeposit »
// Note that even if we store a value as float, we return
// it as double so the user doesn't have to think about
// precision issues.
int NumPhotons() const { return numPhotons; }
int NumFPhotons() const { return round(numPhotons * scintYieldRatio); }
int NumSPhotons() const { return round(numPhotons * (1.0 - scintYieldRatio)); }
int NumElectrons() const { return numElectrons; }
double ScintYieldRatio() const { return scintYieldRatio;}
double Energy() const { return edep; }
geo::Point t Start() const { return { startPos.X(), startPos.Y(), startPos.Z() }; }
geo::Point t End() const { return { endPos.X(), endPos.Y(), endPos.Z() }; }
and a second second
```

```
BEGIN PROLOG
         IonAndScint:
                                                                                                                    ScintTimeLAr
              module type:
                                               "IonAndScint"
              SimulationLabel:
                                               "largeant:LArG4DetectorServicevolTPCActive"
                                                                                                                         tool type: ScintTimeLAr
                ISCalcAlg:
                                                 "NEST"
                                                                                                                         LogLevel:
              ISCalcAlg:
                                               "Separate"
                                                                                                                         FastRisingTime:
                                                                                                                                                0.0
                                                                                                                         FastDecayTime:
                                                                                                                                                6.0
                                                                                                                         SlowRisingTime:
                                                                                                                                                0.0
         PDFastSim:
                                                                                                                         SlowDecavTime:
                                                                                                                                                1600.0
             module type:
                                               "PDFastSimPVS"
                                                 "PDFastSimPAR"
               module type:
                                                                                                                    ScintTimeXeDopedLAr:
             SimulationLabel:
                                               "IonAndScint"
                                                                                                                         tool type: ScintTimeXeDopedLAr
             DoSlowComponent:
                                               true
                                                                                                                         LogLevel:
             ScintTimeTool:
                                               @local::ScintTimeLAr
                                                                                                                    END PROLOG
                                                                               namespace phot
                                                                                  class ScintTimeLAr : public ScintTime
                                                art class tools
namespace phot
                                                                                  public:
                                                                                     explicit ScintTimeLAr(fhicl::ParameterSet const& pset);
   class ScintTime
                                                                                     void GenScintTime(bool is fast, CLHEP::HepRandomEngine& engine);
   public:
                                                                                  private:
      ScintTime();
                                                                                     int
                                                                                                 LogLevel;
      virtual void GenScintTime(bool is fast, CLHEP::HepRandomEngine& engine) = 0;
                                                                                     // parameters for the shape of argon scinitllation light time distribution
      double GetScintTime() const {return timing;}
                                                                                     double
                                                                                                 SRTime
                                                                                                                          // PureLAr: rising time of slow LAr scinitllation;
                                                                                     double
                                                                                                 SDTime;
                                                                                                                          // PureLAr: decay time of slow LAr scintillation;
                                                                                     double
                                                                                                 FRTime;
                                                                                                                          // PureLAr: rising time of fast LAr scinitllation;
   protected:
                                                                                     double
                                                                                                 FDTime;
                                                                                                                          // PureLAr: decay time of fast LAr scintillation;
       double timing;
   };
                                                                                     // general functions
                                                                                     double single exp(double t, double tau2);
                                                                                     double bi exp(double t, double tau1, double tau2);
                                                                                  };
```