# ND280 v10r11p3

/ oaAnalysis/ v5r13/ src/ TGRooTrackerVtx.hxx

Versions: [ v5r3 ] [ v6r3p9 ] [ v7r21p9 ] [ v8r3 ] [ v8r5p3 ] [ v8r5p11 ] [ v9r7 ] [ v9r7p9 ] [ v9r11p9 ] [ v10r3 ] [ v10r9p3 ] [ v10r11p3 ]

```
//
//
#ifndef GRooTrackerVtx_hh_seen
#define GRooTrackerVtx_hh_seen

#include <iostream>

#include "TObject.h"
#include "TBits.h"
#include "TObjString.h"

#include "TJNuBeamFlux.hxx"

using std::ostream;

///
/// This is a simple event class which is essentially an objectified version
/// of the GENIE gRooTracker output format (based on StdHep format).
/// This can be used with the GRooTracker utils class in AnalysisTools as well
/// as GENIE reweighting tools.
///

namespace ND {

  const int kGStdHepNPmax = 100;
  const int kGStdHepIdxPx = 0;
  const int kGStdHepIdxPy = 1;
  const int kGStdHepIdxPz = 2;
  const int kGStdHepIdxE  = 3;
  const int kGStdHepIdxX  = 0;
  const int kGStdHepIdxY  = 1;
  const int kGStdHepIdxZ  = 2;
  const int kGStdHepIdxT  = 3;

  class GRooTrackerVtx : public JNuBeamFlux {

  public:
    GRooTrackerVtx();
   ~GRooTrackerVtx();

    void Reset   (void);
    void Init    (void);

    // Using methods from TObject to remove 'hidden' compiler warnings
    using TObject::Copy;
    void Copy    (const GRooTrackerVtx * event);
    void Print   (const Option_t* option = "") const;

    // Define the output rootracker tree branches
    TObjString* EvtCode;                         // generator-specific string with 'event code'
    int         EvtNum;                          // event num.
    double      EvtXSec;                         // cross section for selected event (1E-38 cm2)
    double      EvtDXSec;                        // cross section for selected event kinematics (1E-38 cm2 /{K^n})
    double      EvtWght;                         // weight for that event
    double      EvtProb;                         // probability for that event (given cross section, path lengths, etc)
    double      EvtVtx[4];                       // event vertex position in detector coord syst (SI)
    int         StdHepN;                         // number of particles in particle array
    //
    //  stdhep-like particle array
    //
    int         StdHepPdg   [kGStdHepNPmax];     // pdg codes (& generator specific codes for pseudoparticles)
    int         StdHepRescat[kGStdHepNPmax];     // generator-specific status code
    int         StdHepStatus[kGStdHepNPmax];     // generator-specific status code
    double      StdHepX4    [kGStdHepNPmax][4];  // 4-x (x, y, z, t) of particle in hit nucleus frame (fm)
    double      StdHepP4    [kGStdHepNPmax][4];  // 4-p (px,py,pz,E) of particle in LAB frame (GeV)
    double      StdHepPolz  [kGStdHepNPmax][3];  // polarization vector
    int         StdHepFd    [kGStdHepNPmax];     // first daughter
    int         StdHepLd    [kGStdHepNPmax];     // last  daughter
    int         StdHepFm    [kGStdHepNPmax];     // first mother
    int         StdHepLm    [kGStdHepNPmax];     // last  mother
    //
    // neutrino parent info (passed-through from the beam-line MC / quantities in 'jnubeam' units)
    // is defined in TJNuBeamFlux from which this class inherits.
    //

    //
    // etc
    //
    int         G2NeutEvtCode;                   // NEUT-like reaction code for the GENIE event
    TObjString* GeomPath;                        //

    // Some pass through info
    TObjString* GeneratorName;                   //
    TObjString* OrigFileName;                    //
    TObjString* OrigTreeName;                    //
    int OrigEvtNum;
    int OrigTreeEntries;
    double OrigTreePOT;
    double TimeInSpill;

    int TruthVertexID;

    ClassDef(ND::GRooTrackerVtx, 2);
  };

} // nd280 namespace
#endif
```