

Nov 24, 14 11:09

TrackerHit.hh

Page 1/2

```
//
// *****
// * License and Disclaimer *
// *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// *
// * Neither the authors of this software system, nor their employing *
// * institutes, nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// *****
//

#ifndef TrackerHit_h
#define TrackerHit_h 1

#include "G4VHit.hh"
#include "G4THitsCollection.hh"
#include "G4Allocator.hh"
#include "G4ThreeVector.hh"

class TrackerHit : public G4VHit
{
public:
    TrackerHit();
    TrackerHit(G4ThreeVector, G4ThreeVector, G4double, G4double, G4double, G4int, unsigned int, unsigned int, G4double, G4double);
    ~TrackerHit();
    TrackerHit(const TrackerHit &right);
    const TrackerHit& operator=(const TrackerHit &right);
    G4int operator==(const TrackerHit &right) const;

    inline void *operator new(size_t);
    inline void operator delete(void *aHit);

    void Draw();
    void Print();

    void SetExitPoint(G4ThreeVector aExit);
    void AddEnergyLoss(G4double aEloss);
    inline G4ThreeVector GetEntryPoint(){return theEntryPoint;}
    inline G4ThreeVector GetExitPoint(){return theExitPoint;}
    inline G4double GetMomentum(){return theMomentum;}
    inline G4double GetToF(){return theTimeOfFlight;}
    inline G4double GetELoss(){return theEnergyLoss;}
    inline G4int GetParticleType(){return theParticleType;}
    inline unsigned int GetDetectorId(){return theDetectorId;}
    inline unsigned int GetTrackId(){return theTrackIdInHit;}
    inline G4double GetThetaAtEntry(){return theThetaAtEntry;}
    inline G4double GetPhiAtEntry(){return thePhiAtEntry;}

private:
    G4ThreeVector theEntryPoint;
    G4ThreeVector theExitPoint;
    G4double theMomentum;

```

Nov 24, 14 11:09

TrackerHit.hh

Page 2/2

```
G4double theTimeOfFlight;
G4double theEnergyLoss;
G4int theParticleType;
unsigned int theDetectorId;
unsigned int theTrackIdInHit;
G4double theThetaAtEntry;
G4double thePhiAtEntry;

};

typedef G4THitsCollection<TrackerHit> TrackerHitsCollection;

extern G4Allocator<TrackerHit> TrackerHitAllocator;

inline void* TrackerHit::operator new(size_t)
{
    void *aHit;
    aHit = (void *) TrackerHitAllocator.MallocSingle();
    return aHit;
}

inline void TrackerHit::operator delete(void *aHit)
{
    TrackerHitAllocator.FreeSingle((TrackerHit*) aHit);
}

#endif

```