Scoring - 2

Igor Semeniouk
LLR, CNRS – Ecole Polytechnique

Credits:
I. Hrivnacova (IJCLAB), M. Asai (SLAC), G. Folger (CERN)
and others

Geant4 CNRS Tutorial,
16 - 20 May 2022, Orsay

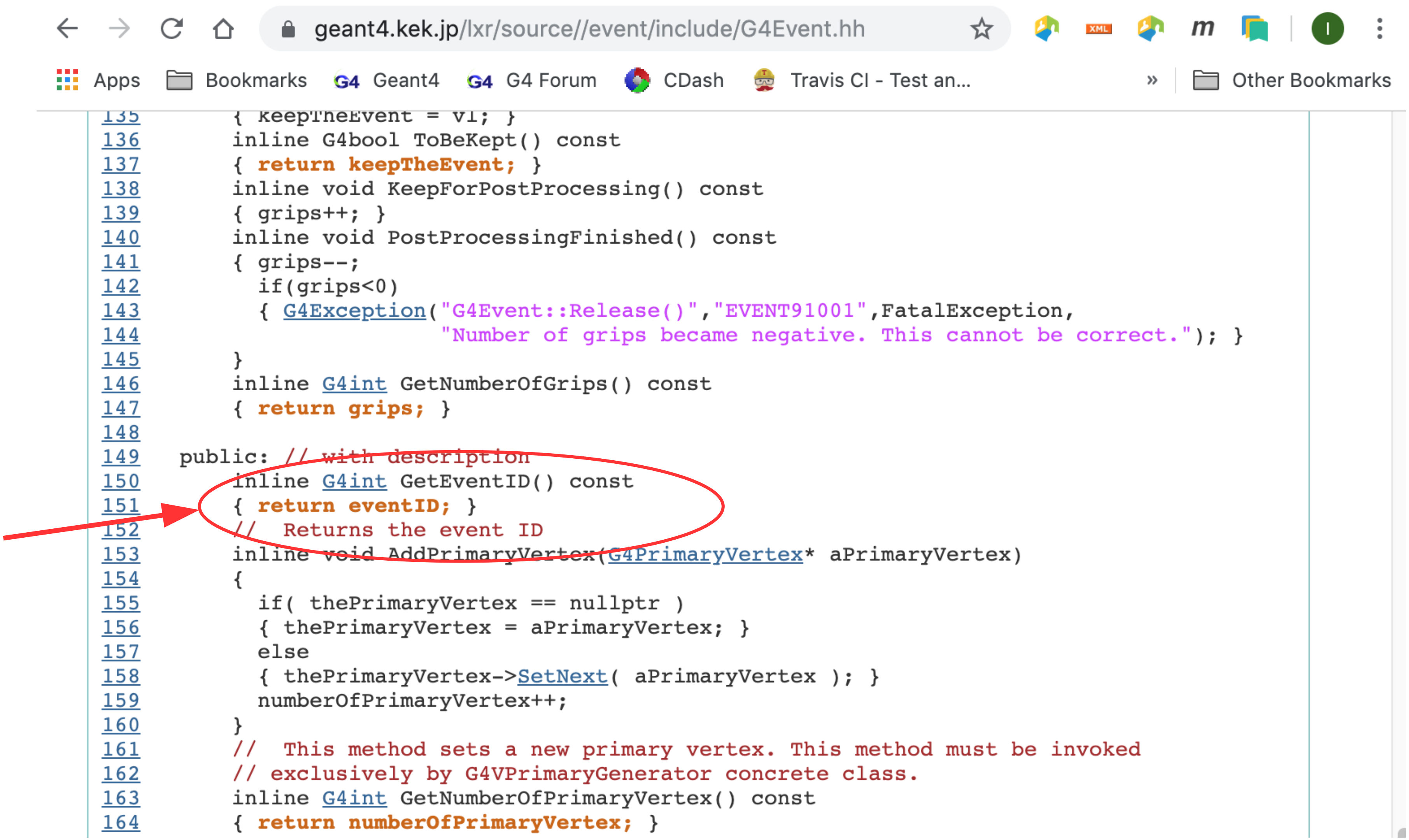
Accessing information from Geant4 objects

Getting Information from Geant4 Objects

- At each phase of run processing user can access the corresponding Geant4 objects:
  - G4Run, G4Event, G4Track, G4Step
  - Note that the objects are provided via constant pointer and so they cannot be modified in the user functions
- An overview of available “Get” functions is provided on the following slides
- The up-to-date information (for each Geant4 version) can be accessed via Geant4 LXR code browser:
  - https://geant4.kek.jp/LXR/
- Alternatively via Doxygen documentations:
  - https://geant4.kek.jp/Reference/v11.0.1/

LXR Browser

- Hyperlinked source code
- Including comments in the code

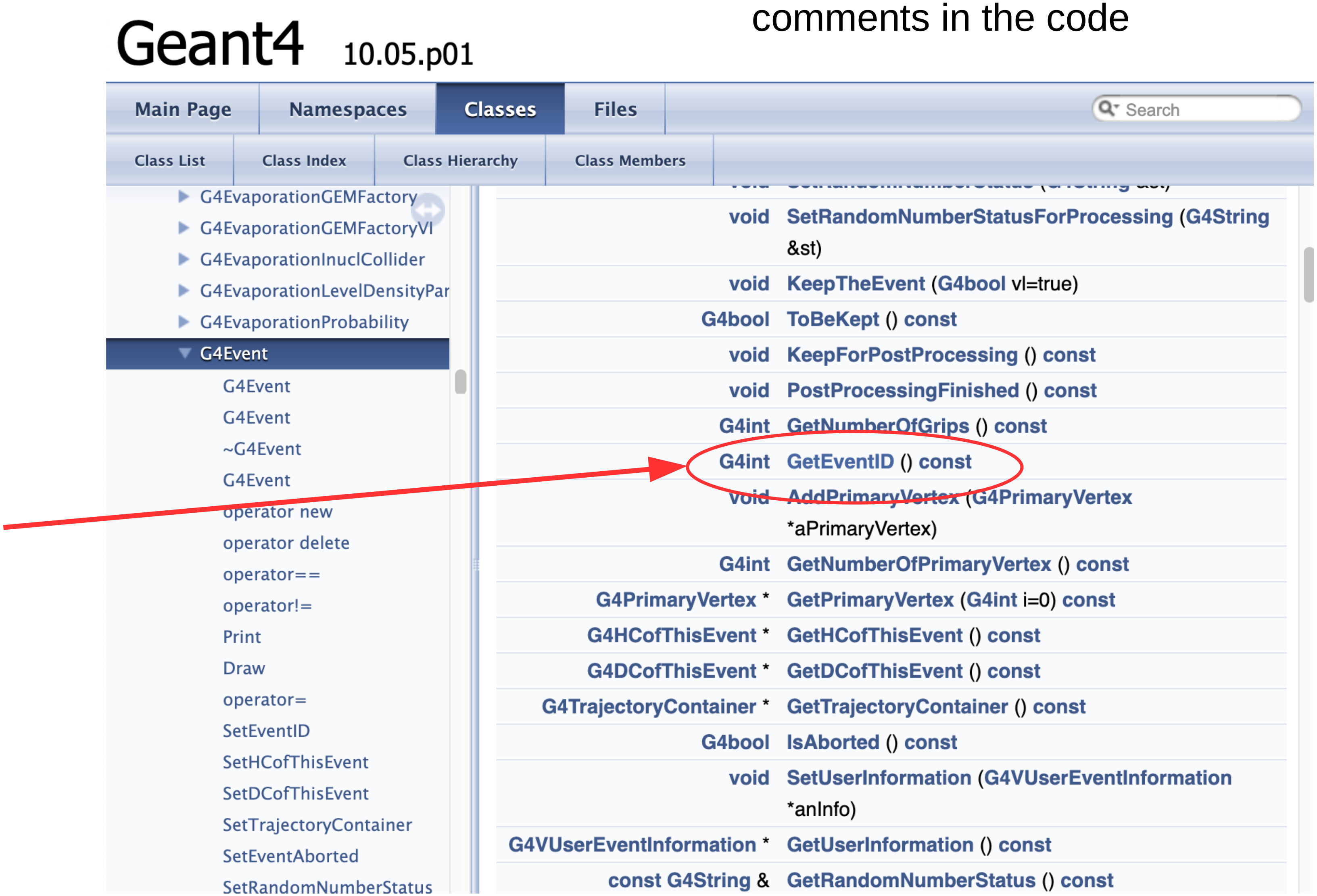


```
      { keepTheEvent = v1; }
      inline G4bool ToBeKept() const
      { return keepTheEvent; }
      inline void KeepForPostProcessing() const
      { grips++; }
      inline void PostProcessingFinished() const
      { grips--;
        if(grips<0)
        { G4Exception("G4Event::Release()","EVENT91001",FatalException,
                      "Number of grips became negative. This cannot be correct."); }
      }
      inline G4int GetNumberOfGrips() const
      { return grips; }

  public: // with description
      inline G4int GetEventID() const
      { return eventID; }
      //  Returns the event ID
      inline void AddPrimaryVertex(G4PrimaryVertex* aPrimaryVertex)
      {
        if( thePrimaryVertex == nullptr )
        { thePrimaryVertex = aPrimaryVertex; }
        else
        { thePrimaryVertex->SetNext( aPrimaryVertex ); }
        numberOfPrimaryVertex++;
      }
      //  This method sets a new primary vertex. This method must be invoked
      // exclusively by G4VPrimaryGenerator concrete class.
      inline G4int GetNumberOfPrimaryVertex() const
      { return numberOfPrimaryVertex; }
```

Doxygen

- Lists of classes, class member functions etc.
- No access to complete code, comments in the code



Geant4 10.05.p01

- G4EvaporationGEMFactory
- G4EvaporationGEMFactoryVI
- G4EvaporationInuclCollider
- G4EvaporationLevelDensityPar
- G4EvaporationProbability
- G4Event
  - G4Event
  - G4Event
  - ~G4Event
  - G4Event
  - operator new
  - operator delete
  - operator==
  - operator!=
  - Print
  - Draw
  - operator=
  - SetEventID
  - SetHCofThisEvent
  - SetDCofThisEvent
  - SetTrajectoryContainer
  - SetEventAborted
  - SetRandomNumberStatus

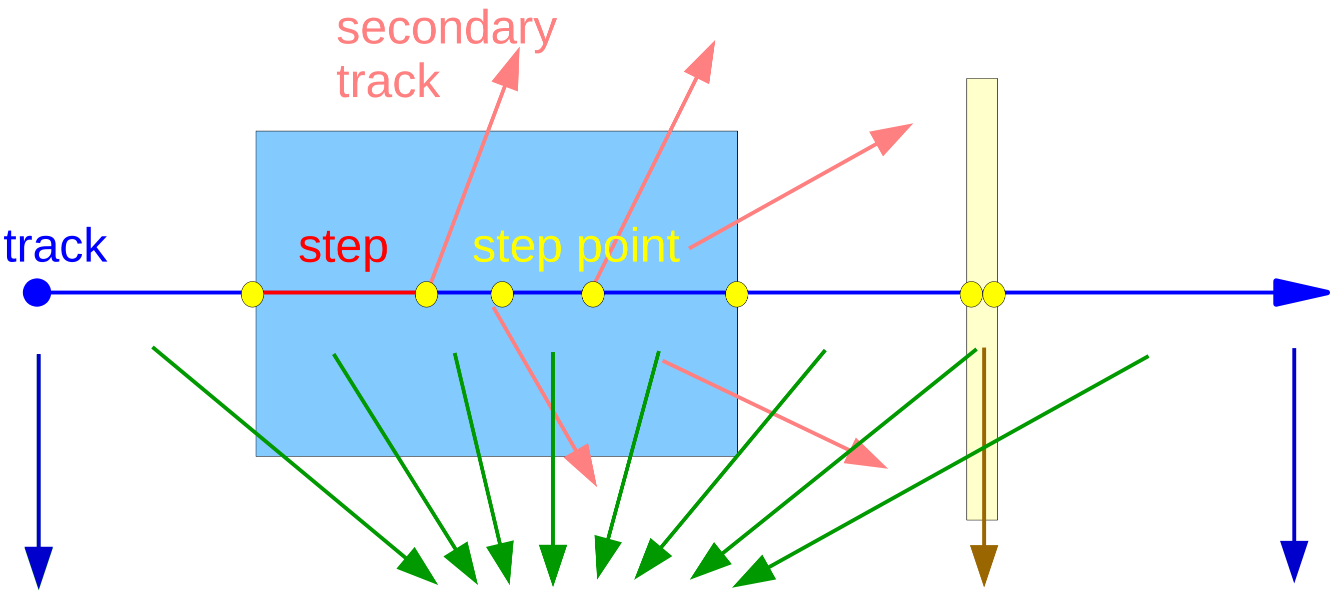
void	SetRandomNumberStatusForProcessing (G4String &st)
void	KeepTheEvent (G4bool vl=true)
G4bool	ToBeKept () const
void	KeepForPostProcessing () const
void	PostProcessingFinished () const
G4int	GetNumberOfGrips () const
G4int	GetEventID () const
void	AddPrimaryVertex (G4PrimaryVertex *aPrimaryVertex)
G4int	GetNumberOfPrimaryVertex () const
G4PrimaryVertex *	GetPrimaryVertex (G4int i=0) const
G4HCofThisEvent *	GetHCofThisEvent () const
G4DCofThisEvent *	GetDCofThisEvent () const
G4TrajectoryContainer *	GetTrajectoryContainer () const
G4bool	IsAborted () const
void	SetUserInformation (G4VUserEventInformation *anInfo)
G4VUserEventInformation *	GetUserInformation () const
const G4String &	GetRandomNumberStatus () const

Reminder

Geant4 and User Application

Event Processing

A special user class, sensitive detector, can be attached to (a) selected volume(s) and then called during event processing



User Application	PreTracking Action	Stepping Action	Sensitive Detector	PostTracking Action
Geant4	**G4Track**	**G4Step**	**G4Step**	**G4Track**

Reminder

Create a Hit

- A hit can be created when a step takes place in a sensitive logical volume, in a user sensitive detector function ProcessHits(..)
- In this function we have access to G4Step

MySD.cc

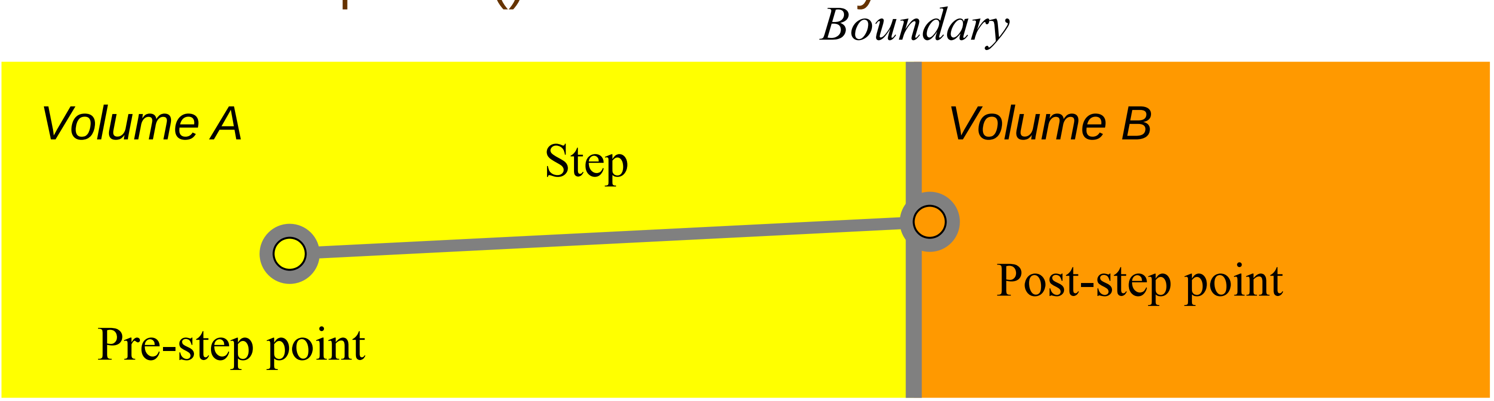
```
G4bool MySD::ProcessHits(G4Step* step,
                         G4TouchableHistory* /*history*/)
{
  MyHit* newHit = new MyHit();
  // Get some properties from G4Step and set them to the hit
  // newHit->SetXYZ();
  G4double edep = step->GetTotalEnergyDeposit();
  newHit->SetEdep(edep);
  // ...
}
```

Track, Step & Step Point

- The G4Track and G4Step objects give access to all properties of the tracked particles
- The track properties which can be different at start and end of step have to be accessed by G4StepPoint class

```
G4ThreeVector position
  = step->GetPreStepPoint()->GetPosition();
```

- The post step point may be 0 if track is leaving world (a test of the GetPostStepPoint() call result may be needed



G4Track

- The G4Track object can be accessed via G4Step:

```
G4Track* track = step->GetTrack();
```

- Some track properties can be accessed via the objects associated to a track
  - G4ParticleDefinition: static particle properties

```
G4double pdgCode
  = track->GetParticleDefinition()→GetPDGEncoding();
```

  - G4DynamicParticle: dynamic particle properties

```
G4double ...
  = track->GetDynamicParticle()→Get...();
```

Track Status

- At the end of each step, according to the processes involved, the state of a track may be changed.
  - It can be accessed via:

```
G4TrackStatus status = track->GetTrackStatus();
```

- The G4TrackStatus is `enum` which can take the following values
  - fAlive – track continues the tracking.
  - fStopButAlive - he track has come to zero kinetic energy, but still AtRest process to occur.
  - fStopAndKill - the track no longer exists - it has decayed, interacted or gone out of the world boundary. Secondaries will be pushed to the stack.
- The user can also change the status in UserSteppingAction with the following values
  - fKillTrackAndSecondaries, fSuspend, fPostponeToNextEvent

Step Status

- Step status is attached to G4StepPoint to indicate why that particular step was determined.
  - Use "PostStepPoint" to get the status of this step, "PreStepPoint" has the status of the previous step.

```
G4StepStatus status
  = step->GetPostStepPoint()->GetStepStatus();
```

- The G4StepStatus is `enum` which can take the following values
  - `fWorldBoundary` - step reached the world boundary
  - `fGeomBoundary` - step is limited by a volume boundary except the world
  - `fAtRestDoItProc, fAlongStepDoItProc, fPostStepDoItProc` - step is limited by a physical process
  - `fUserDefinedLimit, fExclusivelyForcedProc, fUndefined`
- To identify *a track entering [exiting] a volume*, pick `fGeomBoundary` status in PreStep [PostStep]

Overview of Geant4 Classes used in Scoring

G4Run, G4Event

G4Run useful functions:

G4int GetRunID() const;

G4int GetNumberOfEvent() const;

G4Event useful functions:

G4int GetEventID() const;

G4HCofThisEvent* GetHCofThisEvent() const;

- This function gives an access to all registered hits collections

G4DCofThisEvent* GetDCofThisEvent() const;

- This function gives an access to a digits collection (not presented in this course)

G4TrajectoryContainer* GetTrajectoryContainer() const

G4Track

Useful functions:

```
G4int Get[Track,Parent]ID() const;
const G4DynamicParticle* GetDynamicParticle() const;
const G4ParticleDefinition* GetParticleDefinition() const;
const G4VProcess* GetCreatorProcess() const;
const G4ThreeVector& Get[Vertex]Position() const;
G4double Get[Global,Local,Proper]Time() const;
G4double Get[Vertex][Kinetic,Total]Energy() const;
const G4ThreeVector[&] Get[Vertex]Momentum[Direction]() const;
G4double GetVelocity() const;
const G4ThreeVector& GetPolarization() const;
G4double GetWeight() const;
```

G4Track (2)

Useful functions:

- G4double GetTrackLength() const;
- [const G4Step*, G4int, G4double] Get[Current]Step[Length,Number]() const;
- G4VPhysicalVolume* Get[Next]Volume() const;
- G4Material* Get[Next]Material() const;
- const G4MaterialCutsCouple* Get[Next]MaterialCutsCouple() const;
- const [G4VTouchable*,G4TouchableHandle&] Get[Next,Origin]Touchable[Handle]() const;
- G4TrackStatus GetTrackStatus() const;

G4Step

Useful functions:

```
G4Track* GetTrack() const;
G4StepPoint* GetPreStepPoint() const;
G4StepPoint* GetPostStepPoint() const;
G4double GetStepLength() const;
G4double Get[Total,NonIonizing]EnergyDeposit() const;
G4ThreeVector GetDeltaPosition() const;
G4double GetDeltaTime() const;
```

G4StepPoint

Useful functions:

```
const G4VProcess* GetProcessDefinedStep() const;
* const G4ThreeVector& GetPosition() const;
* G4double Get[Global,Local,Proper]Time() const;
* G4double Get[Kinetic,Total]Energy() const;
* const G4ThreeVector[&] GetMomentum[Direction]() const;
* G4double GetVelocity() const;
  G4double Get[Beta,Gamma]() const;
* const G4ThreeVector& GetPolarization() const;
G4double GetMass() const;
G4double GetCharge() const;
G4double GetMagneticMoment() const;
* G4double GetWeight() const;
```

*The functions preceded with * exist also for G4Track*

G4StepPoint (2)

Useful functions:

G4VSensitiveDetector* GetSensitiveDetector() const;

* G4VPhysicalVolume* GetPhysicalVolume() const;

* G4Material* GetMaterial() const;

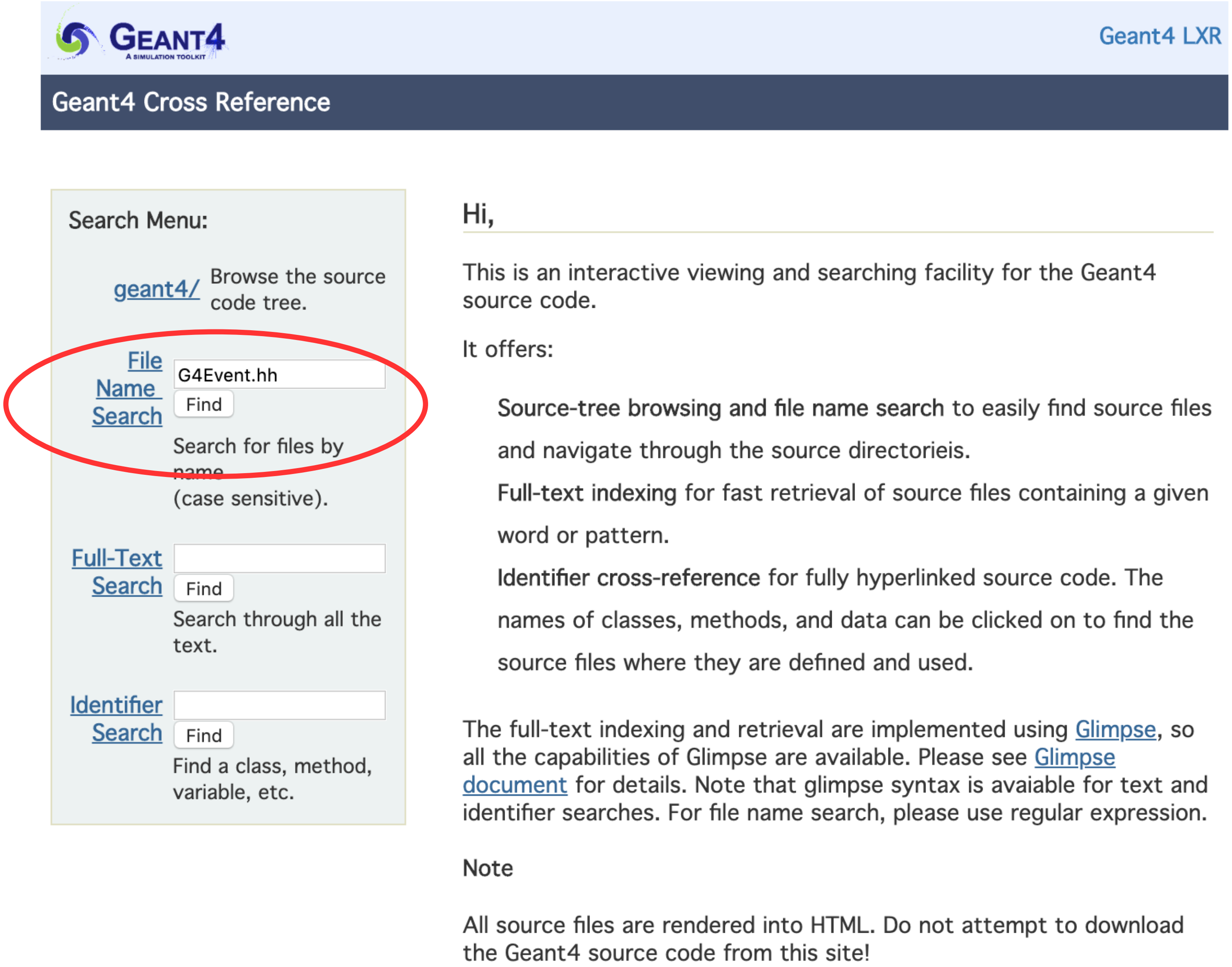
* const G4MaterialCutsCouple* Get[Next]MaterialCutsCouple() const;

* const [G4VTouchable*,G4TouchableHandle&] GetTouchable[Handle]() const;

G4StepStatus GetStepStatus() const;

Searching in Geant4 Source Code Documentation

LXR Browser (1)



Geant4 Cross Reference

Search Menu:

geant4/ Browse the source code tree.

File Name Search: G4Event.hh

Search for files by name (case sensitive).

Full-Text Search

Search through all the text.

Identifier Search

Find a class, method, variable, etc.

Hi,

This is an interactive viewing and searching facility for the Geant4 source code.

It offers:

Source-tree browsing and file name search to easily find source files and navigate through the source directorieis.

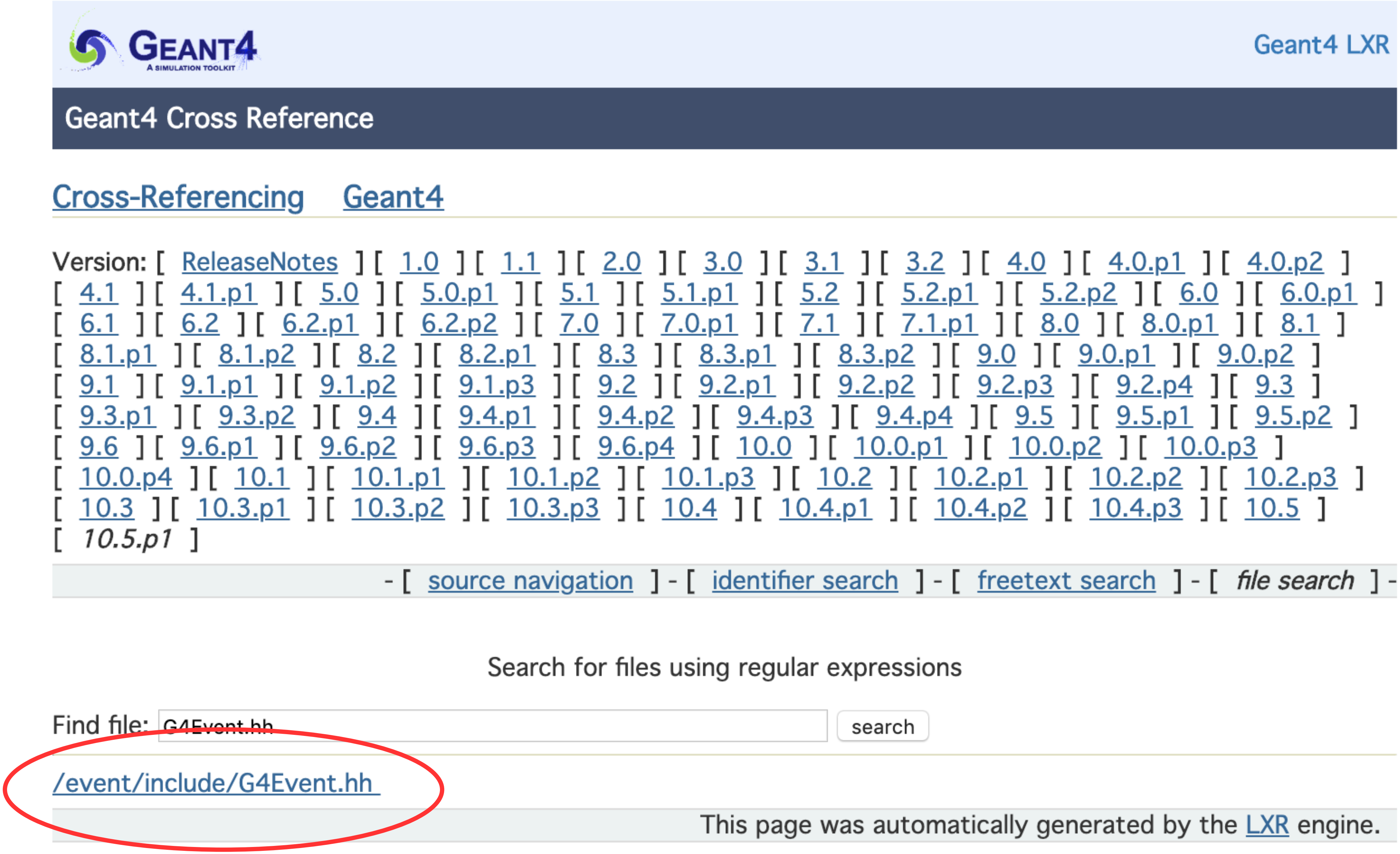
Full-text indexing for fast retrieval of source files containing a given word or pattern.

Identifier cross-reference for fully hyperlinked source code. The names of classes, methods, and data can be clicked on to find the source files where they are defined and used.

The full-text indexing and retrieval are implemented using Glimpse, so all the capabilities of Glimpse are available. Please see Glimpse document for details. Note that glimpse syntax is avaiable for text and identifier searches. For file name search, please use regular expression.

Note

All source files are rendered into HTML. Do not attempt to download the Geant4 source code from this site!



Geant4 LXR

Geant4 Cross Reference

Cross-Referencing Geant4

Version: [ReleaseNotes] [1.0] [1.1] [2.0] [3.0] [3.1] [3.2] [4.0] [4.0.p1] [4.0.p2]
[4.1] [4.1.p1] [5.0] [5.0.p1] [5.1] [5.1.p1] [5.2] [5.2.p1] [5.2.p2] [6.0] [6.0.p1]
[6.1] [6.2] [6.2.p1] [6.2.p2] [7.0] [7.0.p1] [7.1] [7.1.p1] [8.0] [8.0.p1] [8.1]
[8.1.p1] [8.1.p2] [8.2] [8.2.p1] [8.3] [8.3.p1] [8.3.p2] [9.0] [9.0.p1] [9.0.p2]
[9.1] [9.1.p1] [9.1.p2] [9.1.p3] [9.2] [9.2.p1] [9.2.p2] [9.2.p3] [9.2.p4] [9.3]
[9.3.p1] [9.3.p2] [9.4] [9.4.p1] [9.4.p2] [9.4.p3] [9.4.p4] [9.5] [9.5.p1] [9.5.p2]
[9.6] [9.6.p1] [9.6.p2] [9.6.p3] [9.6.p4] [10.0] [10.0.p1] [10.0.p2] [10.0.p3]
[10.0.p4] [10.1] [10.1.p1] [10.1.p2] [10.1.p3] [10.2] [10.2.p1] [10.2.p2] [10.2.p3]
[10.3] [10.3.p1] [10.3.p2] [10.3.p3] [10.4] [10.4.p1] [10.4.p2] [10.4.p3] [10.5]
[*10.5.p1*]

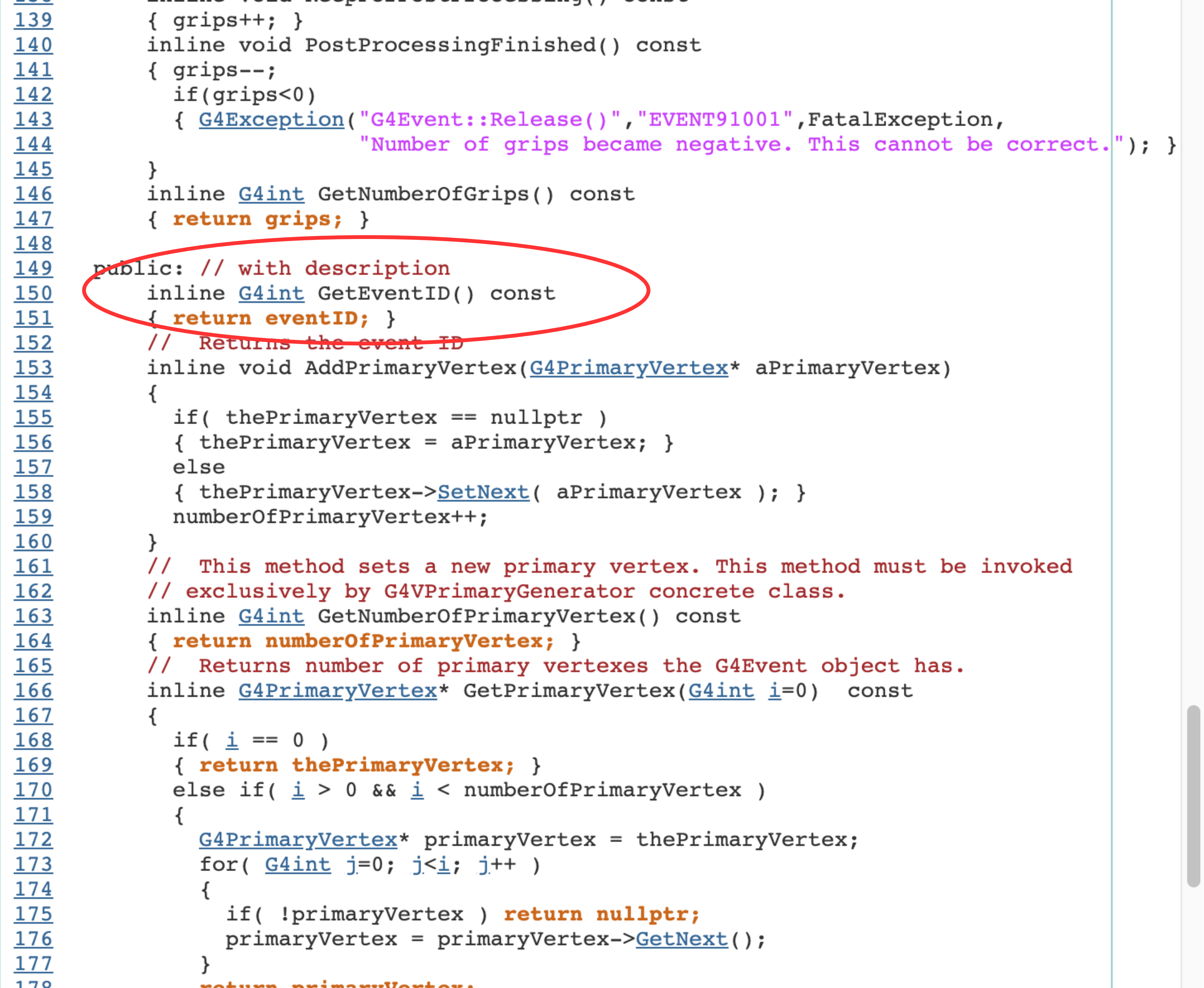
- [source navigation] - [identifier search] - [freetext search] - [*file search*] -

Search for files using regular expressions

Find file: G4Event.hh search

/event/include/G4Event.hh

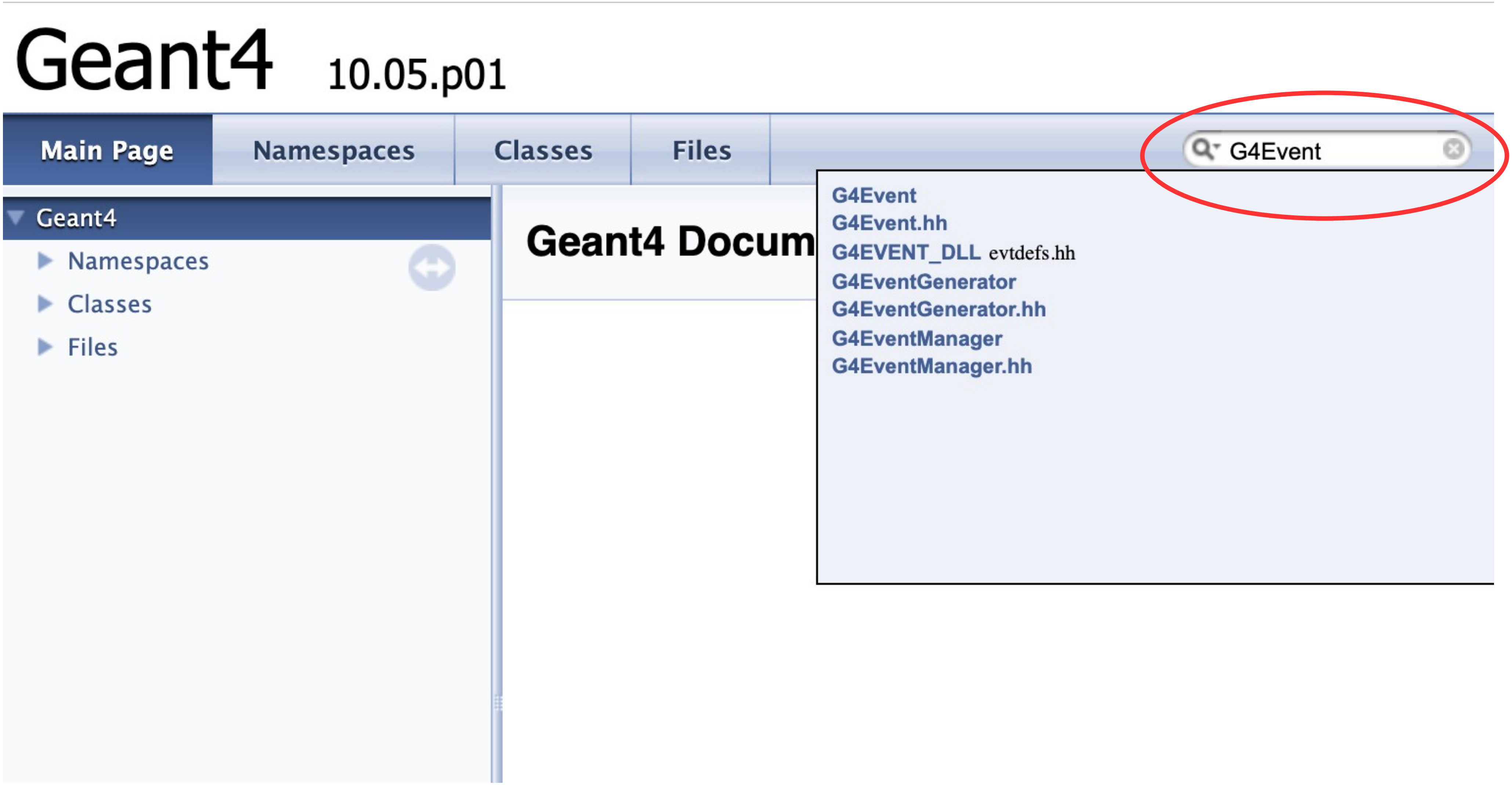
This page was automatically generated by the LXR engine.



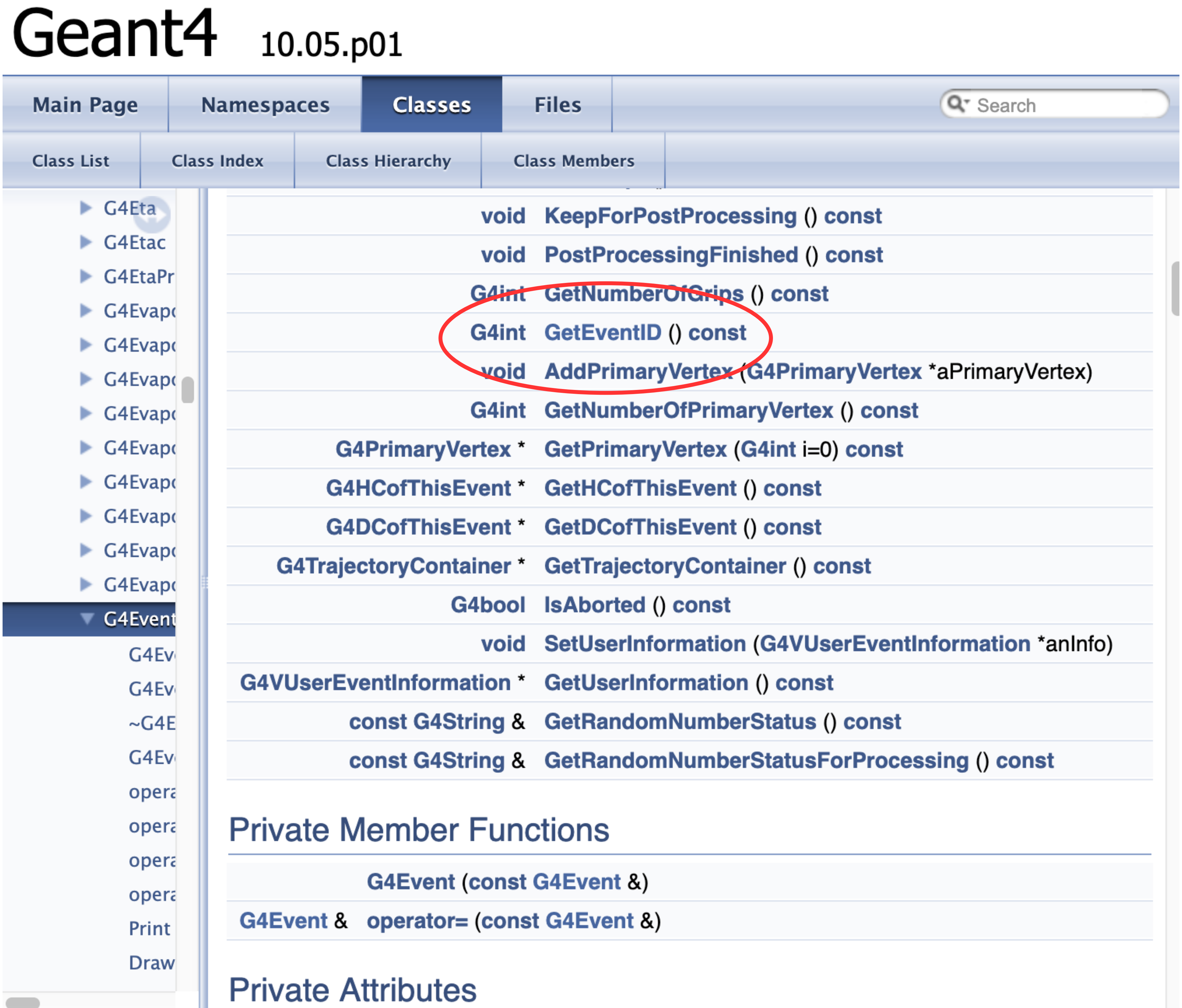
```
    inline void KeepForPostProcessing() const
    { grips++; }
    inline void PostProcessingFinished() const
    { grips--;
      if(grips<0)
      { G4Exception("G4Event::Release()","EVENT91001",FatalException,
            "Number of grips became negative. This cannot be correct."); }
    }
    inline G4int GetNumberOfGrips() const
    { return grips; }

  public: // with description
    inline G4int GetEventID() const
    { return eventID; }
    //  Returns the event ID
    inline void AddPrimaryVertex(G4PrimaryVertex* aPrimaryVertex)
    {
      if( thePrimaryVertex == nullptr )
      { thePrimaryVertex = aPrimaryVertex; }
      else
      { thePrimaryVertex->SetNext( aPrimaryVertex ); }
      numberOfPrimaryVertex++;
    }
    //  This method sets a new primary vertex. This method must be invoked
    // exclusively by G4VPrimaryGenerator concrete class.
    inline G4int GetNumberOfPrimaryVertex() const
    { return numberOfPrimaryVertex; }
    //  Returns number of primary vertexes the G4Event object has.
    inline G4PrimaryVertex* GetPrimaryVertex(G4int i=0)  const
    {
      if( i == 0 )
      { return thePrimaryVertex; }
      else if( i > 0 && i < numberOfPrimaryVertex )
      {
        G4PrimaryVertex* primaryVertex = thePrimaryVertex;
        for( G4int j=0; j<i; j++ )
        {
          if( !primaryVertex ) return nullptr;
          primaryVertex = primaryVertex->GetNext();
        }
        return primaryVertex;
```

Doxygen



G4Event
G4Event.hh
G4EVENT_DLL evtdefs.hh
G4EventGenerator
G4EventGenerator.hh
G4EventManager
G4EventManager.hh

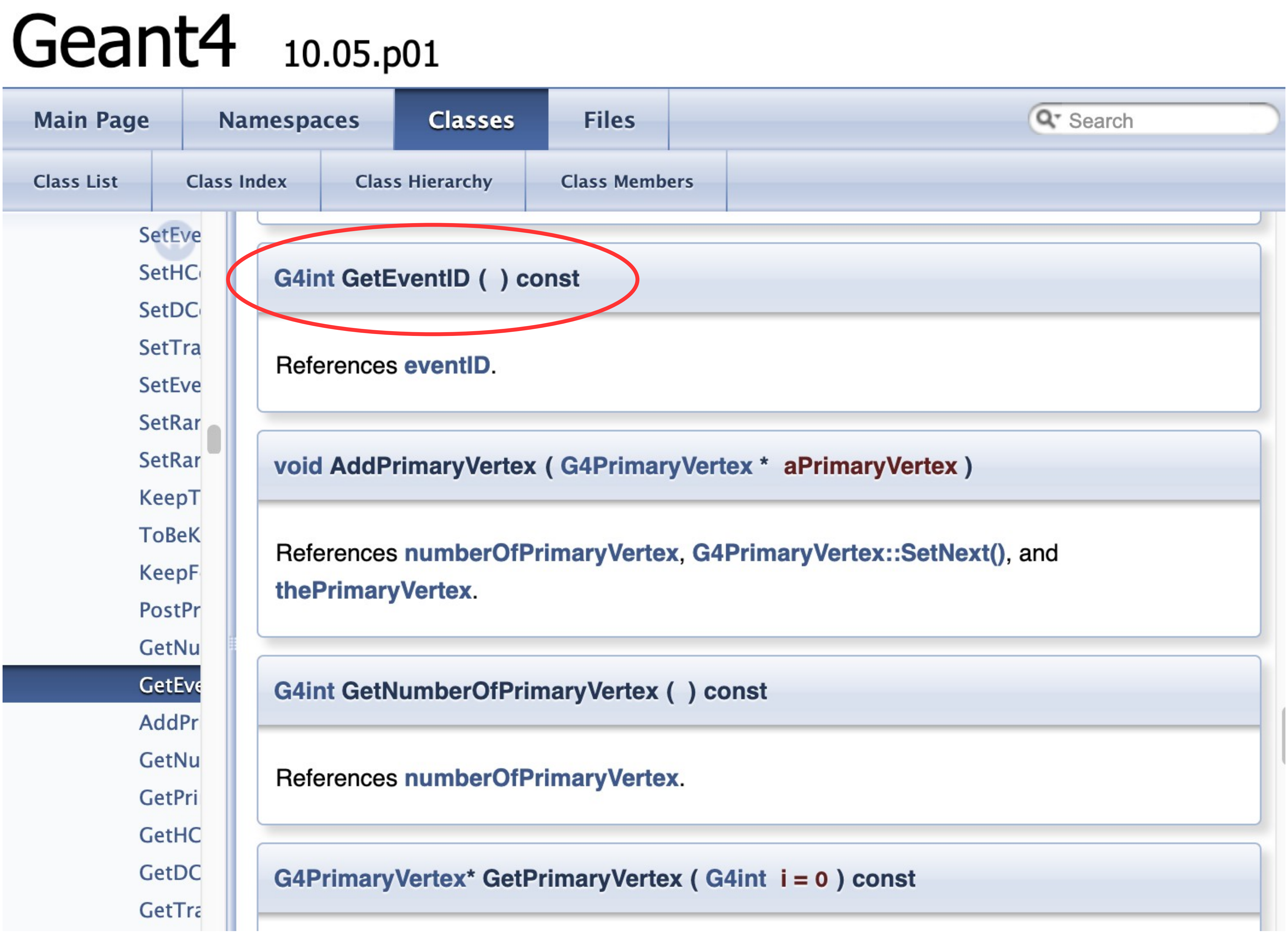


void	KeepForPostProcessing () const
void	PostProcessingFinished () const
G4int	GetNumberOfGrips () const
G4int	GetEventID () const
void	AddPrimaryVertex (G4PrimaryVertex *aPrimaryVertex)
G4int	GetNumberOfPrimaryVertex () const
G4PrimaryVertex *	GetPrimaryVertex (G4int i=0) const
G4HCofThisEvent *	GetHCofThisEvent () const
G4DCofThisEvent *	GetDCofThisEvent () const
G4TrajectoryContainer *	GetTrajectoryContainer () const
G4bool	IsAborted () const
void	SetUserInformation (G4VUserEventInformation *anInfo)
G4VUserEventInformation *	GetUserInformation () const
const G4String &	GetRandomNumberStatus () const
const G4String &	GetRandomNumberStatusForProcessing () const

Private Member Functions

	G4Event (const G4Event &)
G4Event &	operator= (const G4Event &)

Private Attributes



```
G4int GetEventID ( ) const
```

References eventID.

```
void AddPrimaryVertex ( G4PrimaryVertex * aPrimaryVertex )
```

References numberOfPrimaryVertex, G4PrimaryVertex::SetNext(), and thePrimaryVertex.

```
G4int GetNumberOfPrimaryVertex ( ) const
```

References numberOfPrimaryVertex.

```
G4PrimaryVertex* GetPrimaryVertex ( G4int i = 0 ) const
```

Summary

- The physical quantities of interest can be accessed via provided "Get" functions of Geant4 objects available during event processing:
  - G4Run, G4Event, G4Track, G4Step, G4StepPoint
- The complete, up-to date list of all available functions can be found in LXR browser:
  - http://www-geant4.kek.jp/LXR/index.html