gp3m2

0.1

Generated by Doxygen 1.8.11

Contents

1 Class Documentation

1.1 ActionInitialization Class Reference

Instanciate user classes in master or worker threads.

```
#include <ActionInitialization.hh>
```

Public Member Functions

- ActionInitialization (Units ∗units)
- virtual ∼ActionInitialization ()
- virtual void BuildForMaster () const

  Instanciate objects for the master thread.
- virtual void Build () const

  Instanciate objects for the worker threads.

Private Attributes

- Units ∗ fUnits

1.1.1 Detailed Description

Instanciate user classes in master or worker threads.

This class is instanciated only once

1.1.2 Constructor & Destructor Documentation

1.1.2.1 ActionInitialization::ActionInitialization (Units ∗ *units*)

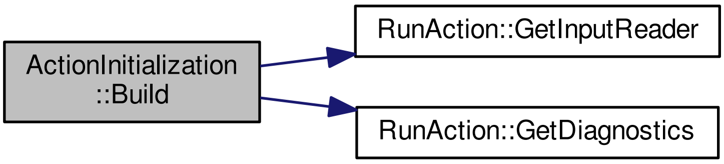
1.1.2.2 ActionInitialization::∼ActionInitialization ( ) [virtual]

1.1.3 Member Function Documentation

1.1.3.1 void ActionInitialization::Build ( ) const [virtual]

Instanciate objects for the worker threads.

Here is the call graph for this function:


ActionInitialization ::Build
RunAction::GetInputReader
RunAction::GetDiagnostics

1.1.3.2 void ActionInitialization::BuildForMaster ( ) const [virtual]

Instanciate objects for the master thread.

1.1.4 Member Data Documentation

1.1.4.1 Units∗ ActionInitialization::fUnits [private]

The documentation for this class was generated from the following files:

- ActionInitialization.hh
- ActionInitialization.cc

1.2 DetectorConstruction Class Reference

Construct geometry.

`#include <DetectorConstruction.hh>`

Public Member Functions

- DetectorConstruction (Units ∗units)

  Initialize pointers, set default values and creates UI commands.
- ∼DetectorConstruction ()

  Delete messenger.
- virtual G4VPhysicalVolume ∗ Construct ()

  Construct the default World volume.
- void AddTargetLayer (G4String materialName, G4double targetWidth)
- void SetTargetRadius (G4double targetRadius)
- void SetPropagationAxis (G4String axis)
- void SetCommands ()

  Set commands to be interpreted with the UI.

Private Attributes

- G4GenericMessenger ∗ fMessenger

  Pointer to the G4GenericMessenger instance.
- G4LogicalVolume ∗ fWorldLV

  Pointer to the world logical volume.
- Units ∗ fUnits

  Pointer to the Units instance.
- G4int fNumberOfLayers

  Total number of layers in the target.
- G4String fPropagationAxis

  Particles propagation axis.
- G4double fTargetSizeLongi

  Total target longitudinal size.
- G4double fTargetRadius

  Target transverse size.
- G4bool fCheckOverlaps

  Check if volumes are overlapping.

1.2.1 Detailed Description

Construct geometry.

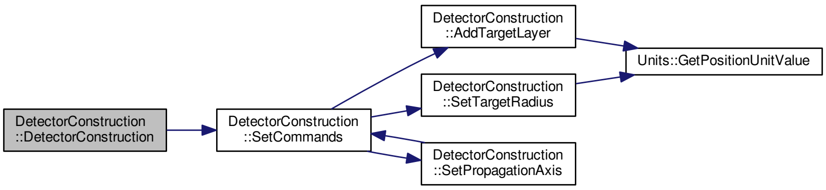
This class is shared and instanciated only once.

1.2.2 Constructor & Destructor Documentation

1.2.2.1 DetectorConstruction::DetectorConstruction (Units * *units*)

Initialize pointers, set default values and creates UI commands.

Here is the call graph for this function:


DetectorConstruction ::DetectorConstruction
DetectorConstruction ::SetCommands
DetectorConstruction ::AddTargetLayer
DetectorConstruction ::SetTargetRadius
DetectorConstruction ::SetPropagationAxis
Units::GetPositionUnitValue

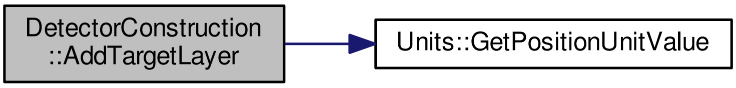
1.2.2.2 DetectorConstruction::∼DetectorConstruction ( )

Delete messenger.

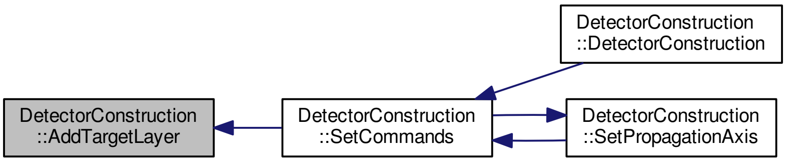
1.2.3 Member Function Documentation

1.2.3.1 void DetectorConstruction::AddTargetLayer (G4String *materialName,* G4double *targetWidth*)

Here is the call graph for this function:


DetectorConstruction ::AddTargetLayer
Units::GetPositionUnitValue

Here is the caller graph for this function:



1.2.3.2 G4VPhysicalVolume ∗ DetectorConstruction::Construct ( ) [virtual]

Construct the default World volume.

The world is defined as a 0.5 x 1 x 1 cm box of G4_Galactic material.

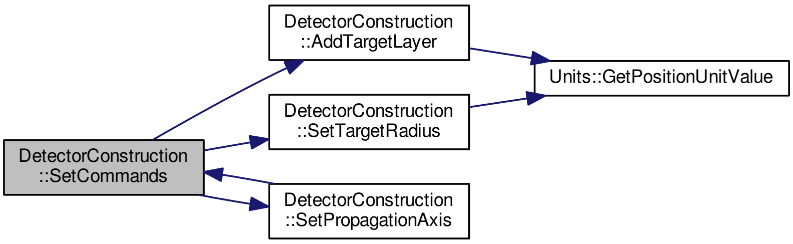
1.2.3.3 void DetectorConstruction::SetCommands ( )

Set commands to be interpreted with the UI.

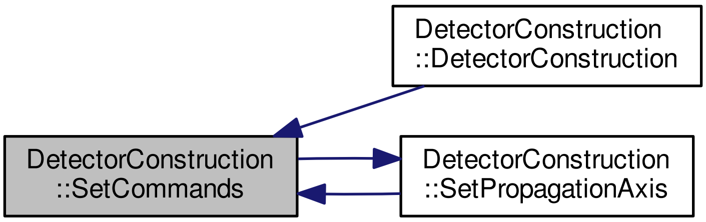
The AddTargetLayer function can be called in UI in the following way :

/target/addLayer materialName width

Here is the call graph for this function:

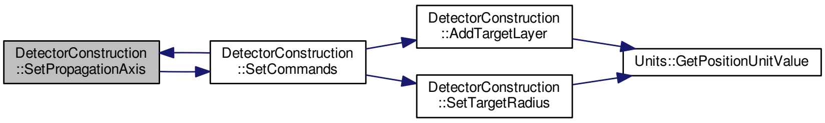


Here is the caller graph for this function:

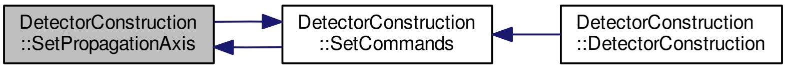


1.2.3.4 void DetectorConstruction::SetPropagationAxis (G4String *axis*) [inline]

Here is the call graph for this function:

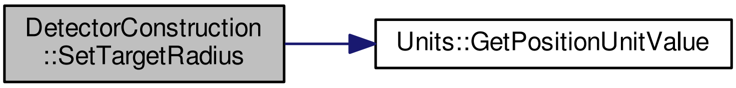


Here is the caller graph for this function:

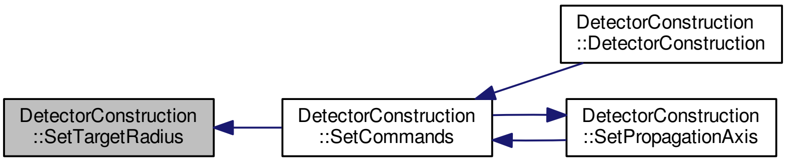


1.2.3.5 void DetectorConstruction::SetTargetRadius (G4double *targetRadius*) [inline]

Here is the call graph for this function:



Here is the caller graph for this function:



1.2.4 Member Data Documentation

1.2.4.1 G4bool DetectorConstruction::fCheckOverlaps [private]

Check if volumes are overlapping.

1.2.4.2 G4GenericMessenger∗ DetectorConstruction::fMessenger [private]

Pointer to the G4GenericMessenger instance.

1.2.4.3 G4int DetectorConstruction::fNumberOfLayers [private]

Total number of layers in the target.

1.2.4.4 G4String DetectorConstruction::fPropagationAxis [private]

Particles propagation axis.

1.2.4.5 G4double DetectorConstruction::fTargetRadius [private]

Target transverse size.

1.2.4.6 G4double DetectorConstruction::fTargetSizeLongi [private]

Total target longitudinal size.

1.2.4.7 Units∗ DetectorConstruction::fUnits [private]

Pointer to the Units instance.

1.2.4.8 G4LogicalVolume∗ DetectorConstruction::fWorldLV [private]

Pointer to the world logical volume.

The documentation for this class was generated from the following files:

- DetectorConstruction.hh
- DetectorConstruction.cc

1.3 Diagnostics Class Reference

Creates and writes diagnostic output files.

```
#include <Diagnostics.hh>
```

Public Member Functions

- Diagnostics (Units ∗units)

  Retrieve analysis manager instance, initialize diagnostics numbers and call SetCommands.
- ∼Diagnostics ()

  Delete messenger.
- void CreateDiagSurfacePhaseSpace (G4String particle)

  Create a diagnostic that export the particle phase space.
- void FillDiagSurfacePhaseSpace (const G4ParticleDefinition ∗part, const G4StepPoint ∗stepPoint)

  Fill the particle phase space diagnostic at each layer surface.
- void InitializeAllDiags ()

  Initialize diagnostics by opening output files.
- void FinishAllDiags ()

  Write and close output files.
- void SetCommands ()

  Define UI commands.
- G4bool GetDiagSurfacePhaseSpaceActivation ()
- void SetOutputFileBaseName (G4String outputFileBaseName)
- G4double GetLowEnergyLimit ()
- G4double GetHighEnergyLimit ()

Private Attributes

- G4AnalysisManager ∗ fAnalysisManager

  Pointer to the G4AnalysisManager instance.
- G4GenericMessenger ∗ fMessenger

  Pointer to the G4GenericMessenger instance for the output file.
- G4ParticleTable ∗ fParticleTable

  Pointer to the G4ParticleTable instance.
- Units ∗ fUnits

  Pointer to the Units instance.
- G4String fOutputFileBaseName

  Output file base name.
- G4double fLowEnergyLimit

  Lower energy to fill diagnostics.
- G4double fHighEnergyLimit

  Higher energy to fill diagnostics.
- G4bool fDiagSurfacePhaseSpaceActivation
- G4MapCache< const G4ParticleDefinition ∗, G4int > fDiagSurfacePhaseSpaceCounter

1.3.1 Detailed Description

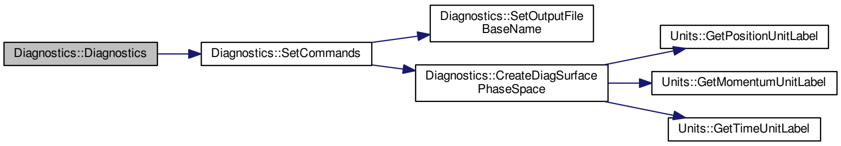
Creates and writes diagnostic output files.

1.3.2 Constructor & Destructor Documentation

1.3.2.1 Diagnostics::Diagnostics (Units ∗ *units*)

Retrieve analysis manager instance, initialize diagnostics numbers and call SetCommands.

Here is the call graph for this function:


Diagnostics::Diagnostics
Diagnostics::SetCommands
Diagnostics::SetOutputFile BaseName
Diagnostics::CreateDiagSurface PhaseSpace
Units::GetPositionUnitLabel
Units::GetMomentumUnitLabel
Units::GetTimeUnitLabel

1.3.2.2 Diagnostics::∼Diagnostics ( )

Delete messenger.

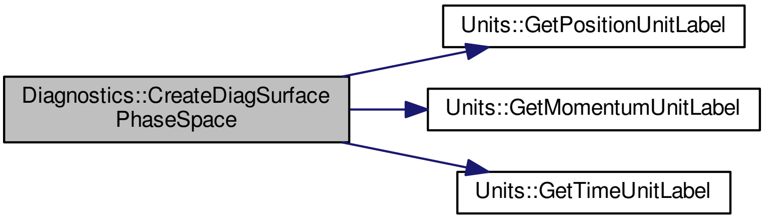
1.3.3 Member Function Documentation

1.3.3.1 void Diagnostics::CreateDiagSurfacePhaseSpace (G4String *particle*)

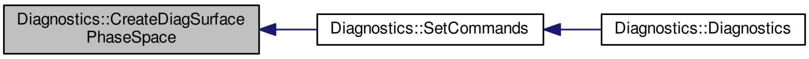
Create a diagnostic that export the particle phase space.

The diagnostic is activated, and the correspondance between the given particle and its Ntuple id is stored in the fDiagSurfacePhaseSpaceCounter map.

Here is the call graph for this function:


Diagnostics::CreateDiagSurface PhaseSpace
Units::GetPositionUnitLabel
Units::GetMomentumUnitLabel
Units::GetTimeUnitLabel

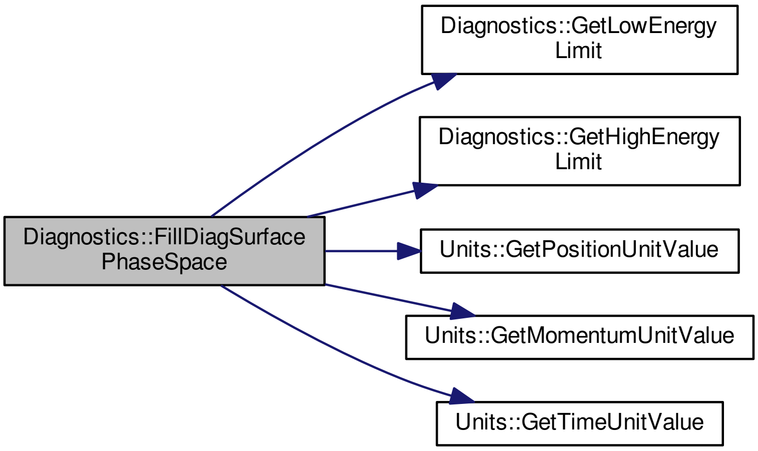
Here is the caller graph for this function:



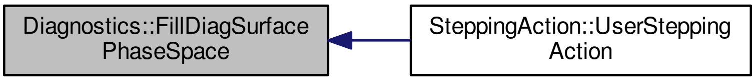
1.3.3.2 void Diagnostics::FillDiagSurfacePhaseSpace (const G4ParticleDefinition ∗ *part*, const G4StepPoint ∗ *stepPoint*)

Fill the particle phase space diagnostic at each layer surface.

Here is the call graph for this function:


Diagnostics::FillDiagSurface PhaseSpace
Diagnostics::GetLowEnergy Limit
Diagnostics::GetHighEnergy Limit
Units::GetPositionUnitValue
Units::GetMomentumUnitValue
Units::GetTimeUnitValue

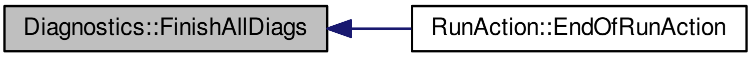
Here is the caller graph for this function:


Diagnostics::FillDiagSurface PhaseSpace
SteppingAction::UserStepping Action

1.3.3.3 void Diagnostics::FinishAllDiags ( )

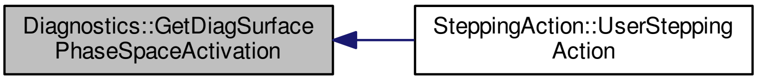
Write and close output files.

Here is the caller graph for this function:



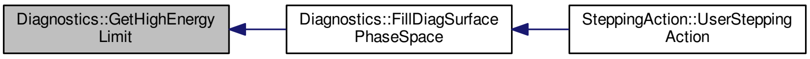
1.3.3.4 G4bool Diagnostics::GetDiagSurfacePhaseSpaceActivation ( ) [inline]

Here is the caller graph for this function:


Diagnostics::GetDiagSurface PhaseSpaceActivation
SteppingAction::UserStepping Action

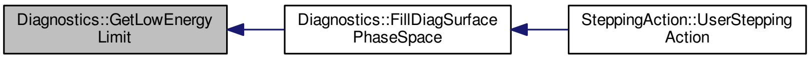
1.3.3.5 G4double Diagnostics::GetHighEnergyLimit ( ) [inline]

Here is the caller graph for this function:


Diagnostics::GetHighEnergy Limit
Diagnostics::FillDiagSurface PhaseSpace
SteppingAction::UserStepping Action

1.3.3.6 G4double Diagnostics::GetLowEnergyLimit ( ) [inline]

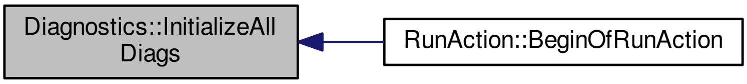
Here is the caller graph for this function:


Diagnostics::GetLowEnergy Limit
Diagnostics::FillDiagSurface PhaseSpace
SteppingAction::UserStepping Action

1.3.3.7 void Diagnostics::InitializeAllDiags ( )

Initialize diagnostics by opening output files.

Here is the caller graph for this function:

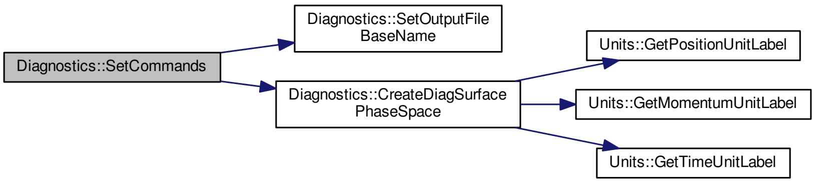


1.3.3.8 void Diagnostics::SetCommands ( )

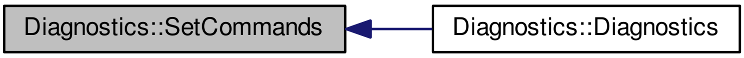
Define UI commands.

The input file name can be changed by using /diags/setFileBaseName baseName /diags/setLowEnergyLimit value unit /diags/setHighEnergyLimit value unit /diags/createDiagSurfacePhaseSpace particleName

Here is the call graph for this function:

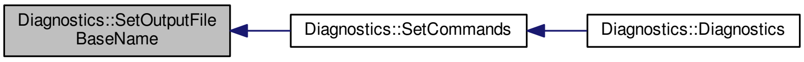

Diagnostics::SetCommands
Diagnostics::SetOutputFile BaseName
Diagnostics::CreateDiagSurface PhaseSpace
Units::GetPositionUnitLabel
Units::GetMomentumUnitLabel
Units::GetTimeUnitLabel

Here is the caller graph for this function:



1.3.3.9 void Diagnostics::SetOutputFileBaseName (G4String *outputFileBaseName*) [inline]

Here is the caller graph for this function:



1.3.4 Member Data Documentation

1.3.4.1 G4AnalysisManager∗ Diagnostics::fAnalysisManager [private]

Pointer to the G4AnalysisManager instance.

1.3.4.2 G4bool Diagnostics::fDiagSurfacePhaseSpaceActivation [private]

1.3.4.3 G4MapCache<const G4ParticleDefinition*, G4int> Diagnostics::fDiagSurfacePhaseSpaceCounter [private]

1.3.4.4 G4double Diagnostics::fHighEnergyLimit [private]

Higher energy to fill diagnostics.

1.3.4.5 G4double Diagnostics::fLowEnergyLimit [private]

Lower energy to fill diagnostics.

1.3.4.6 G4GenericMessenger* Diagnostics::fMessenger [private]

Pointer to the G4GenericMessenger instance for the output file.

1.3.4.7 G4String Diagnostics::fOutputFileBaseName [private]

Output file base name.

1.3.4.8 G4ParticleTable* Diagnostics::fParticleTable [private]

Pointer to the G4ParticleTable instance.

1.3.4.9 Units* Diagnostics::fUnits [private]

Pointer to the Units instance.

The documentation for this class was generated from the following files:

- Diagnostics.hh
- Diagnostics.cc

1.4 InputReader Class Reference

Read input file and interact with input macro-particles.

`#include <InputReader.hh>`

Public Member Functions

- InputReader (Units *units)

  Call the SetCommands method.
- ~InputReader ()

  Delete analysis manager.
- void ReadInputFile ()

  Read the input phase space, and save macro-particles characteristics into arrays.
- void NormalizeMacroParticlesWeights (G4int NumberOfEventsToBeProcessed)

  Normalize macro-particles weights in order to conserve total number of particles.
- G4double GetMacroParticleWeight (G4int id) const
- G4ThreeVector GetMacroParticlePosition (G4int id) const
- G4ThreeVector GetMacroParticleMomentum (G4int id) const
- G4double GetMacroParticleTime (G4int id) const
- G4int GetNumberOfMacroParticles () const
- G4String GetParticleName ()
- void SetInputFileName (G4String inputFileName)
- void SetParticleName (G4String particleName)
- void SetCommands ()

  Define UI commands.

Private Attributes

- G4GenericMessenger $*$ fMessenger

  Pointer to the G4GenericMessenger instance for the input file.
- Units $*$ fUnits

  Pointer to the Units instance.
- G4String fInputFileName

  Input file name.
- G4String fParticleName

  Input particle name.
- std::vector$<$ G4double $>$ fW
- std::vector$<$ G4double $>$ fX
- std::vector$<$ G4double $>$ fY
- std::vector$<$ G4double $>$ fZ
- std::vector$<$ G4double $>$ fPx
- std::vector$<$ G4double $>$ fPy
- std::vector$<$ G4double $>$ fPz
- std::vector$<$ G4double $>$ fT

  Arrays containing input macro-particles characteristics.

1.4.1 Detailed Description

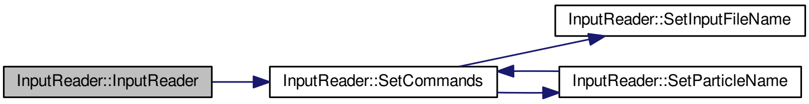
Read input file and interact with input macro-particles.

1.4.2 Constructor & Destructor Documentation

1.4.2.1 InputReader::InputReader (Units $*$ *units*)

Call the SetCommands method.

Here is the call graph for this function:



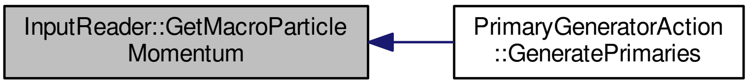
1.4.2.2 InputReader::∼InputReader ( )

Delete analysis manager.

1.4.3 Member Function Documentation

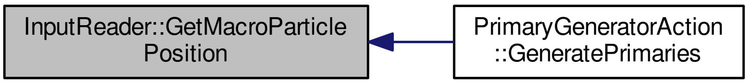
1.4.3.1 G4ThreeVector InputReader::GetMacroParticleMomentum (G4int *id*) const [inline]

Here is the caller graph for this function:


InputReader::GetMacroParticle Momentum
PrimaryGeneratorAction ::GeneratePrimaries

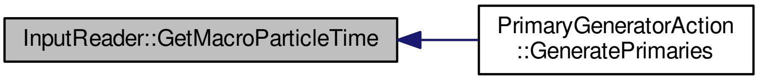
1.4.3.2 G4ThreeVector InputReader::GetMacroParticlePosition (G4int *id*) const [inline]

Here is the caller graph for this function:


InputReader::GetMacroParticle Position
PrimaryGeneratorAction ::GeneratePrimaries

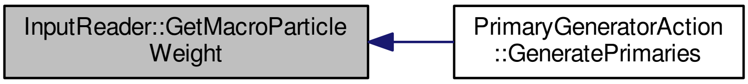
1.4.3.3 G4double InputReader::GetMacroParticleTime (G4int *id*) const [inline]

Here is the caller graph for this function:


InputReader::GetMacroParticleTime
PrimaryGeneratorAction ::GeneratePrimaries

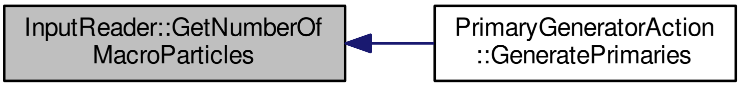
1.4.3.4 G4double InputReader::GetMacroParticleWeight (G4int *id*) const [inline]

Here is the caller graph for this function:


InputReader::GetMacroParticle Weight
PrimaryGeneratorAction ::GeneratePrimaries

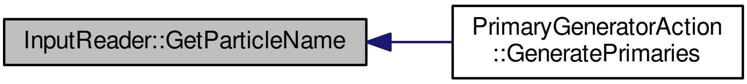
1.4.3.5 G4int InputReader::GetNumberOfMacroParticles ( ) const [inline]

Here is the caller graph for this function:



1.4.3.6 G4String InputReader::GetParticleName ( ) [inline]

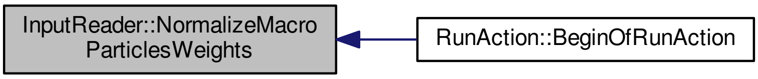
Here is the caller graph for this function:



1.4.3.7 void InputReader::NormalizeMacroParticlesWeights (G4int *NumberOfEventsToBeProcessed*)

Normalize macro-particles weights in order to conserve total number of particles.

Here is the caller graph for this function:



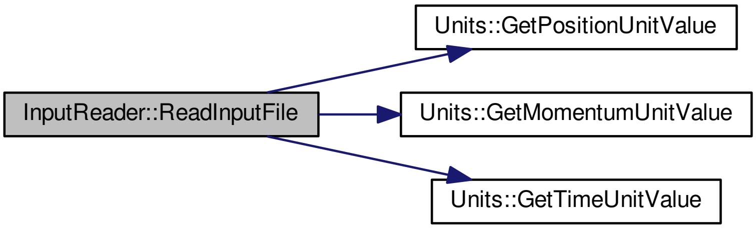
1.4.3.8 void InputReader::ReadInputFile ( )

Read the input phase space, and save macro-particles characteristics into arrays.

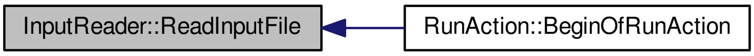
The input file format must be a list of macro-particles: w x y z px py pz t w x y z px py pz t w x y z px py pz t w x y z px py pz t

with separators being spaces.

Here is the call graph for this function:


Units::GetPositionUnitValue
InputReader::ReadInputFile
Units::GetMomentumUnitValue
Units::GetTimeUnitValue

Here is the caller graph for this function:

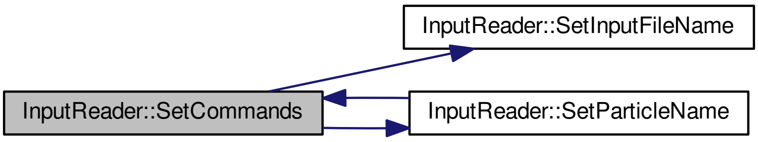


1.4.3.9 void InputReader::SetCommands ( )

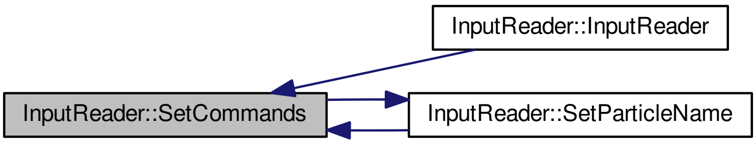
Define UI commands.

The input file name can be changed by using /input/setFileName fileName /input/setParticle particleName

Here is the call graph for this function:

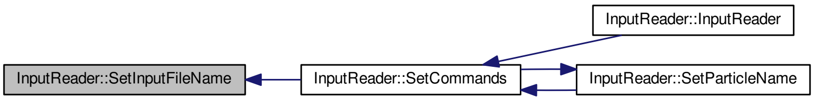


Here is the caller graph for this function:



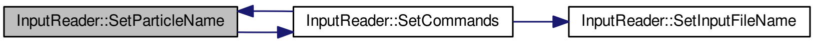
1.4.3.10 void InputReader::SetInputFileName (G4String *inputFileName*) [inline]

Here is the caller graph for this function:

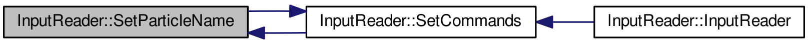


1.4.3.11 void InputReader::SetParticleName (G4String *particleName*) [inline]

Here is the call graph for this function:



Here is the caller graph for this function:



1.4.4 Member Data Documentation

1.4.4.1 G4String InputReader::fInputFileName [private]

Input file name.

1.4.4.2 G4GenericMessenger* InputReader::fMessenger [private]

Pointer to the G4GenericMessenger instance for the input file.

1.4.4.3 G4String InputReader::fParticleName [private]

Input particle name.

1.4.4.4 std::vector< G4double > InputReader::fPx [private]

1.4.4.5 std::vector< G4double > InputReader::fPy [private]

1.4.4.6 std::vector< G4double > InputReader::fPz [private]

1.4.4.7 std::vector< G4double > InputReader::fT [private]

Arrays containing input macro-particles characteristics.

1.4.4.8 Units* InputReader::fUnits [private]

Pointer to the Units instance.

1.4.4.9 std::vector< G4double > InputReader::fW [private]

1.4.4.10 std::vector< G4double > InputReader::fX [private]

1.4.4.11 std::vector< G4double > InputReader::fY [private]

1.4.4.12 std::vector< G4double > InputReader::fZ [private]

The documentation for this class was generated from the following files:

- InputReader.hh
- InputReader.cc

1.5 PhysicsList Class Reference

Define particles and processes to consider in the simulation.

```
#include <PhysicsList.hh>
```

Public Member Functions

- PhysicsList ()

  Set default cut values, verbosity and instanciate desired pre-packages PhysicsList.
- ∼PhysicsList ()

  Delete pre-packaged PhysicsList instance.
- virtual void ConstructParticle ()

  Construct the particles to consider in the simulation.
- virtual void ConstructProcess ()

  Construct the physics processes to use in the simulation.
- virtual void SetCuts ()

  Call base class method to set cuts which default value can be modified via /run/setCut/_ commands.

Protected Member Functions

- void SetPhysicsList (G4String name)
- void SetCommands ()

  Set commands to be interpreted with the UI.

Private Attributes

- G4VPhysicsConstructor ∗ fPhysicsList

  Pointer to the used pre-packaged PhysicsList.
- G4GenericMessenger ∗ fMessenger

  Pointer to the G4GenericMessenger instance.

1.5.1 Detailed Description

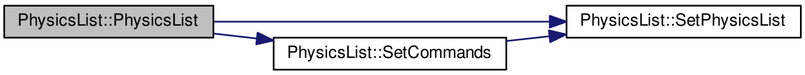
Define particles and processes to consider in the simulation.

1.5.2 Constructor & Destructor Documentation

1.5.2.1 PhysicsList::PhysicsList ( )

Set default cut values, verbosity and instanciate desired pre-packages PhysicsList.

Here is the call graph for this function:



1.5.2.2 PhysicsList::∼PhysicsList ( )

Delete pre-packaged PhysicsList instance.

1.5.3 Member Function Documentation

1.5.3.1 void PhysicsList::ConstructParticle ( ) [virtual]

Construct the particles to consider in the simulation.

The list of particles is taken from choosen pre-packaged PhysicsList.

1.5.3.2 void PhysicsList::ConstructProcess ( ) [virtual]

Construct the physics processes to use in the simulation.

The list of procresses is taken from choosen pre-packaged PhysicsList.

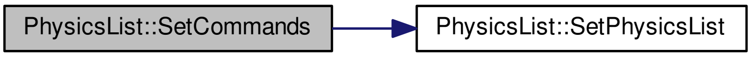
1.5.3.3 void PhysicsList::SetCommands ( ) [protected]

Set commands to be interpreted with the UI.

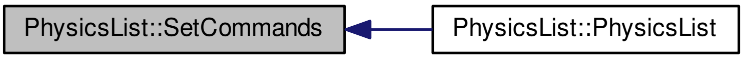
The setPhysicsList function can be called in UI in the following way :

/physics/setPhysicsList str

Here is the call graph for this function:



Here is the caller graph for this function:



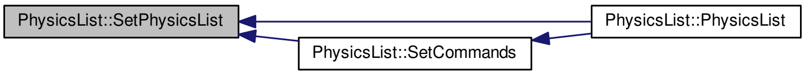
1.5.3.4 void PhysicsList::SetCuts ( ) [virtual]

Call base class method to set cuts which default value can be modified via /run/setCut/_ commands.

?

1.5.3.5 void PhysicsList::SetPhysicsList (G4String *name*) [protected]

Here is the caller graph for this function:



1.5.4 Member Data Documentation

1.5.4.1 G4GenericMessenger∗ PhysicsList::fMessenger [private]

Pointer to the G4GenericMessenger instance.

1.5.4.2 G4VPhysicsConstructor∗ PhysicsList::fPhysicsList [private]

Pointer to the used pre-packaged PhysicsList.

The documentation for this class was generated from the following files:

- PhysicsList.hh
- PhysicsList.cc

1.6 PrimaryGeneratorAction Class Reference

Generate primary particles.

```
#include <PrimaryGeneratorAction.hh>
```

Public Member Functions

- PrimaryGeneratorAction (InputReader ∗inputReader)

  Retrieve the G4ParticleTable instance.
- ∼PrimaryGeneratorAction ()

  Do nothing.
- virtual void GeneratePrimaries (G4Event ∗)

  Generate primary particles.

Private Attributes

- G4ParticleTable ∗ fParticleTable
- InputReader ∗ fInputReader

1.6.1 Detailed Description

Generate primary particles.

This class is instanciated in each worker thread.

1.6.2 Constructor & Destructor Documentation

1.6.2.1 PrimaryGeneratorAction::PrimaryGeneratorAction (InputReader ∗ *inputReader*)

Retrieve the G4ParticleTable instance.

1.6.2.2 PrimaryGeneratorAction::∼PrimaryGeneratorAction ( )

Do nothing.

1.6.3 Member Function Documentation

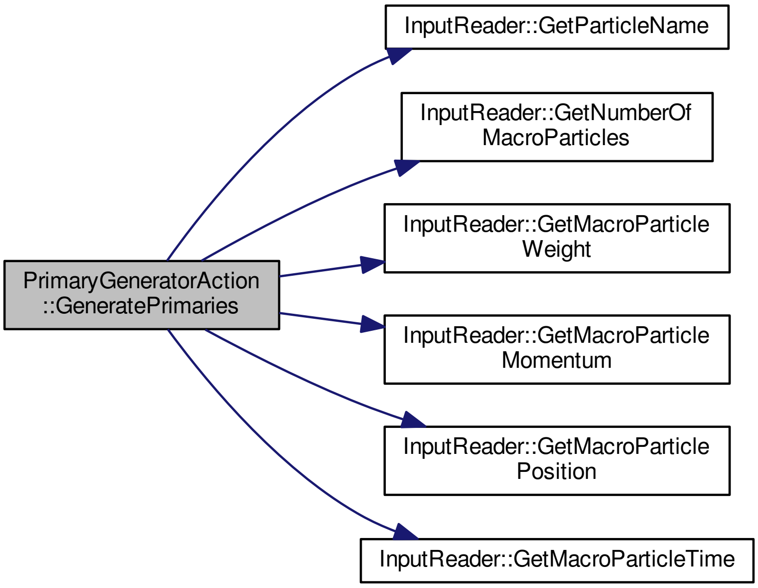
1.6.3.1 void PrimaryGeneratorAction::GeneratePrimaries (G4Event ∗ *anEvent*) [virtual]

Generate primary particles.

The primary particle is defined with properties of a random input macro-particle.

This virtual function is called at the begining of each event.

Here is the call graph for this function:


InputReader::GetParticleName
InputReader::GetNumberOf MacroParticles
InputReader::GetMacroParticle Weight
PrimaryGeneratorAction ::GeneratePrimaries
InputReader::GetMacroParticle Momentum
InputReader::GetMacroParticle Position
InputReader::GetMacroParticleTime

1.6.4 Member Data Documentation

1.6.4.1 InputReader∗ PrimaryGeneratorAction::fInputReader [private]

1.6.4.2 G4ParticleTable∗ PrimaryGeneratorAction::fParticleTable [private]

The documentation for this class was generated from the following files:

- PrimaryGeneratorAction.hh
- PrimaryGeneratorAction.cc

1.7 RunAction Class Reference

Deal with input file reading and diagnostic creation.

```
#include <RunAction.hh>
```

Public Member Functions

- RunAction (Units ∗units)
- ∼RunAction ()

    Delete.
- virtual void BeginOfRunAction (const G4Run ∗aRun)

    Read input file & initialize diagnostics.
- virtual void EndOfRunAction (const G4Run ∗)

    Write and close output files.
- InputReader ∗ GetInputReader ()
- Diagnostics ∗ GetDiagnostics ()

Private Attributes

- Units ∗ fUnits

    Pointer to the Units instance.
- InputReader ∗ fInputReader

    Pointer to the InputReader instance.
- Diagnostics ∗ fDiagnostics

    Pointer to the Diagnostics instance.

1.7.1 Detailed Description

Deal with input file reading and diagnostic creation.

1.7.2 Constructor & Destructor Documentation

1.7.2.1 RunAction::RunAction (Units ∗ *units*)

1.7.2.2 RunAction::∼RunAction ( )

Delete.

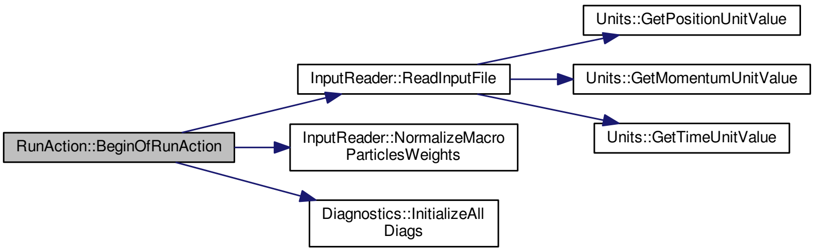
1.7.3 Member Function Documentation

1.7.3.1 void RunAction::BeginOfRunAction (const G4Run ∗ *aRun*) [virtual]

Read input file & initialize diagnostics.

This user code is executed at the beginning of each run

Here is the call graph for this function:

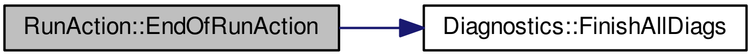

RunAction::BeginOfRunAction
InputReader::ReadInputFile
InputReader::NormalizeMacroParticlesWeights
Diagnostics::InitializeAllDiags
Units::GetPositionUnitValue
Units::GetMomentumUnitValue
Units::GetTimeUnitValue

1.7.3.2 void RunAction::EndOfRunAction (const G4Run ∗) [virtual]

Write and close output files.

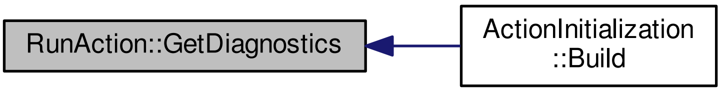
This user code is executed at the end of each run

Here is the call graph for this function:



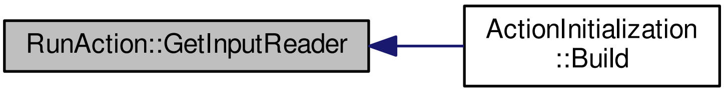
1.7.3.3 Diagnostics∗ RunAction::GetDiagnostics ( ) [inline]

Here is the caller graph for this function:



1.7.3.4 InputReader∗ RunAction::GetInputReader ( ) [inline]

Here is the caller graph for this function:



1.7.4 Member Data Documentation

1.7.4.1 Diagnostics∗ RunAction::fDiagnostics [private]

Pointer to the Diagnostics instance.

1.7.4.2 InputReader∗ RunAction::fInputReader [private]

Pointer to the InputReader instance.

1.7.4.3 Units∗ RunAction::fUnits [private]

Pointer to the Units instance.

The documentation for this class was generated from the following files:

- RunAction.hh
- RunAction.cc

1.8 SteppingAction Class Reference

Export particles phase-space at each geometry boundary.

```
#include <SteppingAction.hh>
```

Public Member Functions

- SteppingAction (Diagnostics ∗diagnostics)

  Save pointer to the current Diagnostics instance.
- ∼SteppingAction ()

  Do nothing.
- virtual void UserSteppingAction (const G4Step ∗)

  Call the Diagnostics FillDiagXXX methods if they are activated.

Private Attributes

- Diagnostics ∗ fDiagnostics

  Pointer to the Diagnostics instance of the current thread.

1.8.1 Detailed Description

Export particles phase-space at each geometry boundary.

This class is instanciated in each worker thread

1.8.2 Constructor & Destructor Documentation

1.8.2.1 SteppingAction::SteppingAction (Diagnostics ∗ *diagnostics*)

Save pointer to the current Diagnostics instance.

1.8.2.2 SteppingAction::∼SteppingAction ( )

Do nothing.

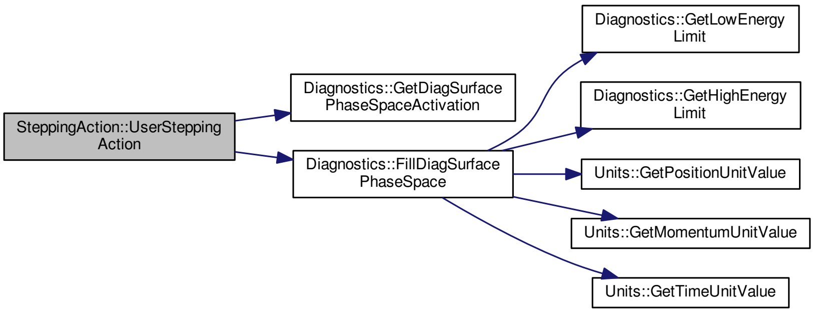
1.8.3 Member Function Documentation

1.8.3.1 void SteppingAction::UserSteppingAction (const G4Step ∗ *aStep*) [virtual]

Call the Diagnostics FillDiagXXX methods if they are activated.

This virtual method is called at each Step ends.

Here is the call graph for this function:


SteppingAction::UserStepping Action
Diagnostics::GetDiagSurface PhaseSpaceActivation
Diagnostics::FillDiagSurface PhaseSpace
Diagnostics::GetLowEnergy Limit
Diagnostics::GetHighEnergy Limit
Units::GetPositionUnitValue
Units::GetMomentumUnitValue
Units::GetTimeUnitValue

1.8.4 Member Data Documentation

1.8.4.1 Diagnostics∗ SteppingAction::fDiagnostics [private]

Pointer to the Diagnostics instance of the current thread.

The documentation for this class was generated from the following files:

- SteppingAction.hh
- SteppingAction.cc

1.9 Units Class Reference

Define units of the input and output files.

```
#include <Units.hh>
```

Public Member Functions

- Units ()

  Create analysis manager, Ntuples and set UI commands.
- ∼Units ()

  Delete analysis manager.
- G4String GetPositionUnitLabel ()
- G4String GetMomentumUnitLabel ()
- G4String GetTimeUnitLabel ()
- G4double GetPositionUnitValue ()
- G4double GetMomentumUnitValue ()
- G4double GetTimeUnitValue ()
- void SetPositionUnit (G4String positionUnitLabel)
- void SetMomentumUnit (G4String momentumUnitLabel)
- void SetTimeUnit (G4String timeUnitLabel)
- void SetCommands ()

  Define UI commands.

Private Attributes

- G4GenericMessenger ∗ fMessenger
- G4String fPositionUnitLabel
- G4String fMomentumUnitLabel
- G4String fTimeUnitLabel

1.9.1 Detailed Description

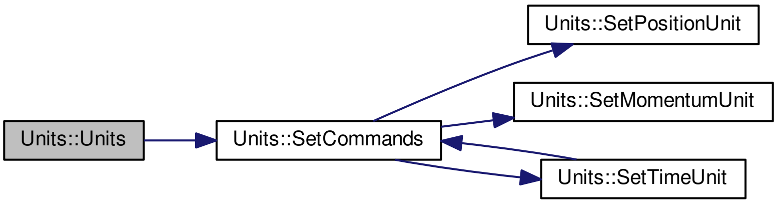
Define units of the input and output files.

1.9.2 Constructor & Destructor Documentation

1.9.2.1 Units::Units ( )

Create analysis manager, Ntuples and set UI commands.

Here is the call graph for this function:



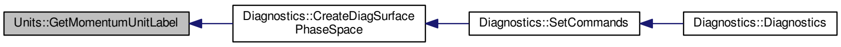
1.9.2.2 Units::∼Units ( )

Delete analysis manager.

1.9.3 Member Function Documentation

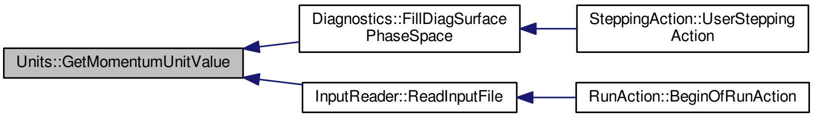
1.9.3.1 G4String Units::GetMomentumUnitLabel ( ) [inline]

Here is the caller graph for this function:


Units::GetMomentumUnitLabel
Diagnostics::CreateDiagSurface PhaseSpace
Diagnostics::SetCommands
Diagnostics::Diagnostics

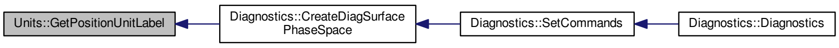
1.9.3.2 G4double Units::GetMomentumUnitValue ( ) [inline]

Here is the caller graph for this function:


Units::GetMomentumUnitValue
Diagnostics::FillDiagSurface PhaseSpace
SteppingAction::UserStepping Action
InputReader::ReadInputFile
RunAction::BeginOfRunAction

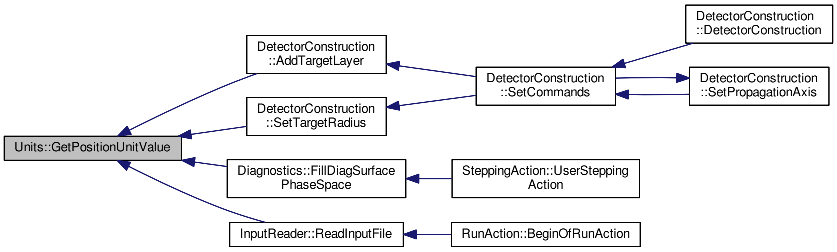
1.9.3.3 G4String Units::GetPositionUnitLabel ( ) [inline]

Here is the caller graph for this function:


Units::GetPositionUnitLabel
Diagnostics::CreateDiagSurface PhaseSpace
Diagnostics::SetCommands
Diagnostics::Diagnostics

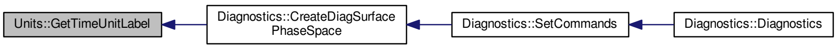
1.9.3.4 G4double Units::GetPositionUnitValue ( ) [inline]

Here is the caller graph for this function:


Units::GetPositionUnitValue
DetectorConstruction ::AddTargetLayer
DetectorConstruction ::SetTargetRadius
Diagnostics::FillDiagSurface PhaseSpace
InputReader::ReadInputFile
DetectorConstruction ::SetCommands
DetectorConstruction ::DetectorConstruction
DetectorConstruction ::SetPropagationAxis
SteppingAction::UserStepping Action
RunAction::BeginOfRunAction

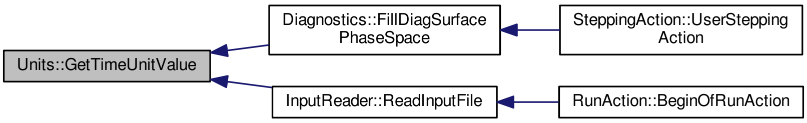
1.9.3.5 G4String Units::GetTimeUnitLabel ( ) [inline]

Here is the caller graph for this function:



1.9.3.6 G4double Units::GetTimeUnitValue ( ) [inline]

Here is the caller graph for this function:

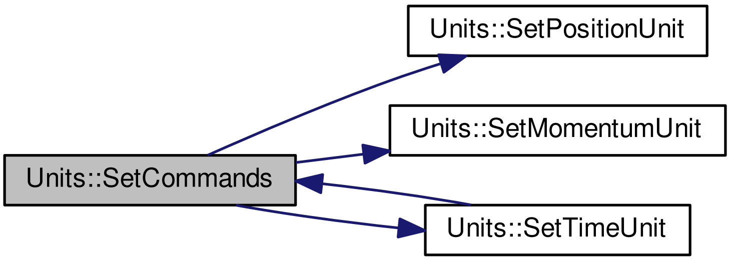

Units::GetTimeUnitValue
Diagnostics::FillDiagSurface PhaseSpace
InputReader::ReadInputFile
SteppingAction::UserStepping Action
RunAction::BeginOfRunAction

1.9.3.7 void Units::SetCommands ( )

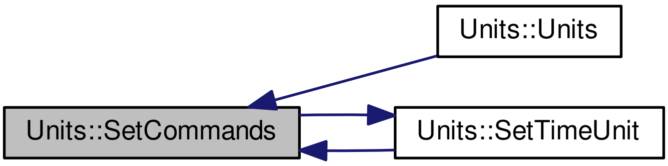
Define UI commands.

The input file name can be changed by using

Here is the call graph for this function:

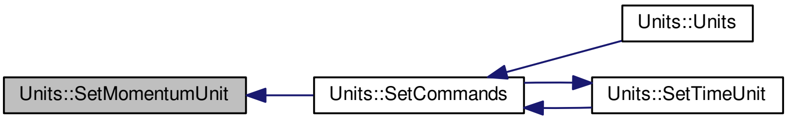


Here is the caller graph for this function:



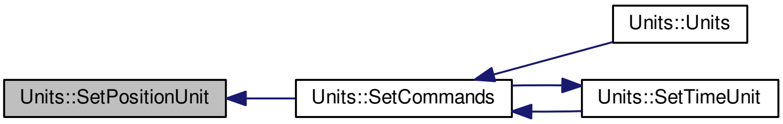
1.9.3.8 void Units::SetMomentumUnit (G4String *momentumUnitLabel*) [inline]

Here is the caller graph for this function:



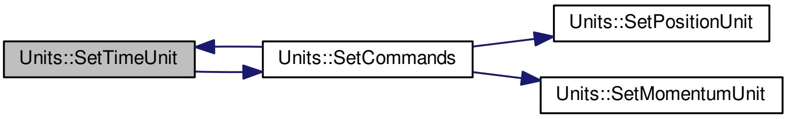
1.9.3.9 void Units::SetPositionUnit (G4String *positionUnitLabel*) [inline]

Here is the caller graph for this function:

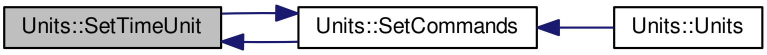


1.9.3.10 void Units::SetTimeUnit (G4String *timeUnitLabel*) [inline]

Here is the call graph for this function:



Here is the caller graph for this function:



1.9.4 Member Data Documentation

1.9.4.1 G4GenericMessenger∗ Units::fMessenger [private]

1.9.4.2 G4String Units::fMomentumUnitLabel [private]

1.9.4.3 G4String Units::fPositionUnitLabel [private]

1.9.4.4 G4String Units::fTimeUnitLabel [private]

The documentation for this class was generated from the following files:

- Units.hh
- Units.cc

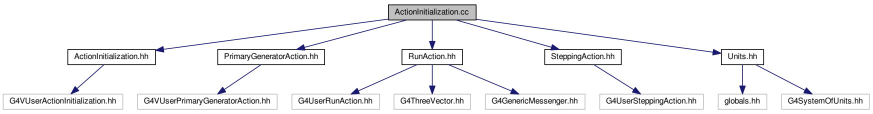
2 File Documentation

2.1 ActionInitialization.cc File Reference

Implementation of the ActionInitialization class.

```
#include "ActionInitialization.hh"
#include "PrimaryGeneratorAction.hh"
#include "RunAction.hh"
#include "SteppingAction.hh"
#include "Units.hh"
```

Include dependency graph for ActionInitialization.cc:



2.1.1 Detailed Description

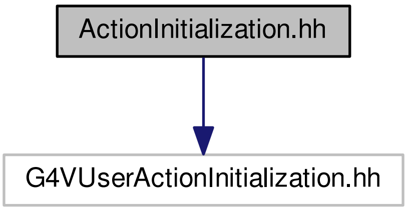
Implementation of the ActionInitialization class.

2.2 ActionInitialization.hh File Reference

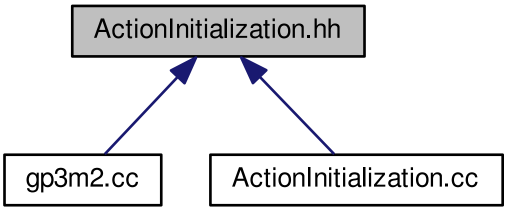
Definition of the ActionInitialization class.

```
#include "G4VUserActionInitialization.hh"
```

Include dependency graph for ActionInitialization.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class ActionInitialization

  Instanciate user classes in master or worker threads.

2.2.1 Detailed Description

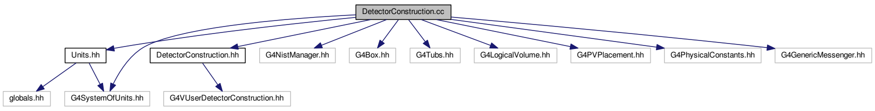
Definition of the ActionInitialization class.

2.3 DetectorConstruction.cc File Reference

Implementation of the DetectorConstruction class.

```
#include "Units.hh"
#include "DetectorConstruction.hh"
#include "G4NistManager.hh"
#include "G4Box.hh"
#include "G4Tubs.hh"
#include "G4LogicalVolume.hh"
#include "G4PVPlacement.hh"
#include "G4SystemOfUnits.hh"
#include "G4PhysicalConstants.hh"
#include "G4GenericMessenger.hh"
```

Include dependency graph for DetectorConstruction.cc:



2.3.1 Detailed Description

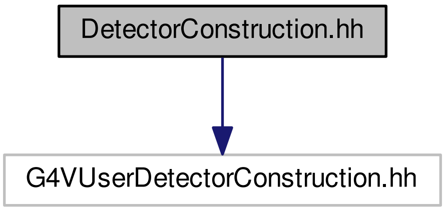
Implementation of the DetectorConstruction class.

2.4 DetectorConstruction.hh File Reference

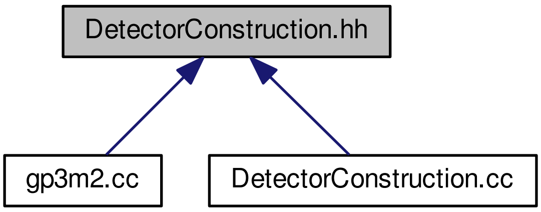
Definition of the DetectorConstruction class.

```
#include "G4VUserDetectorConstruction.hh"
```

Include dependency graph for DetectorConstruction.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class DetectorConstruction

  Construct geometry.

2.4.1 Detailed Description

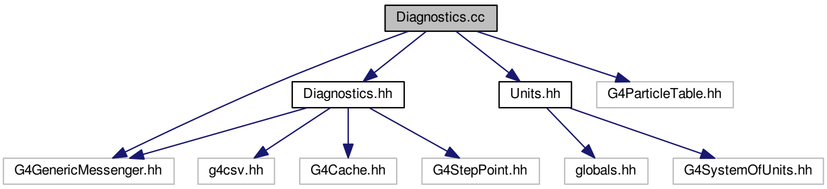
Definition of the DetectorConstruction class.

2.5 Diagnostics.cc File Reference

Implementation of the Diagnostics class.

```
#include "Diagnostics.hh"
#include "Units.hh"
#include "G4GenericMessenger.hh"
#include "G4ParticleTable.hh"
```

Include dependency graph for Diagnostics.cc:



2.5.1 Detailed Description

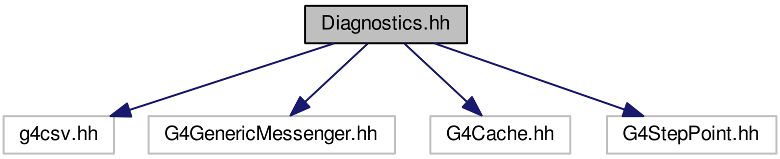
Implementation of the Diagnostics class.

2.6 Diagnostics.hh File Reference

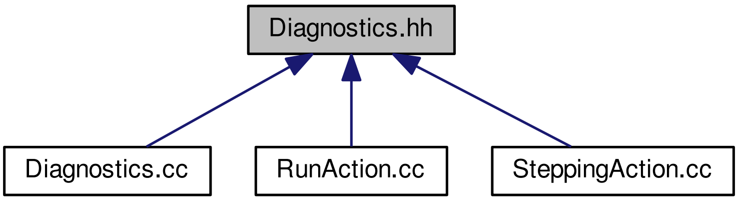
Definition of the Diagnostics class.

```
#include "g4csv.hh"
#include "G4GenericMessenger.hh"
#include "G4Cache.hh"
#include "G4StepPoint.hh"
```

Include dependency graph for Diagnostics.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class Diagnostics

  Creates and writes diagnostic output files.

2.6.1 Detailed Description

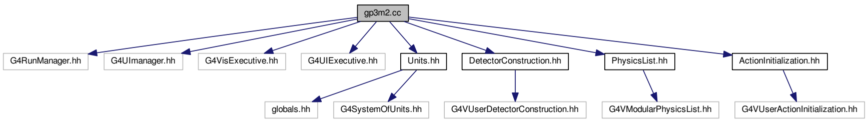
Definition of the Diagnostics class.

2.7 gp3m2.cc File Reference

Main program of gp3m2.

```
#include "G4RunManager.hh"
#include "G4UImanager.hh"
#include "G4VisExecutive.hh"
#include "G4UIExecutive.hh"
#include "Units.hh"
#include "DetectorConstruction.hh"
#include "PhysicsList.hh"
#include "ActionInitialization.hh"
```

Include dependency graph for gp3m2.cc:



Functions

- int main (int argc, char $**$argv)

2.7.1 Detailed Description

Main program of gp3m2.

2.7.2 Function Documentation

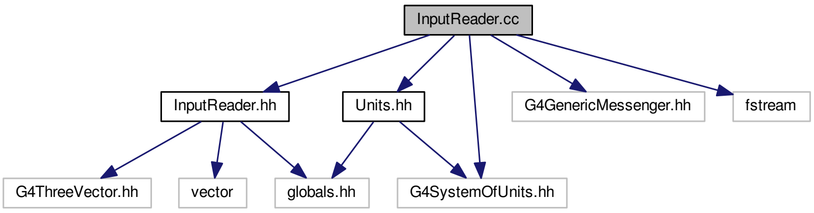
2.7.2.1 int main (int *argc,* char ∗∗ *argv*)

2.8 InputReader.cc File Reference

Implementation of the InputReader class.

```
#include "InputReader.hh"
#include "Units.hh"
#include "G4GenericMessenger.hh"
#include "G4SystemOfUnits.hh"
#include <fstream>
```

Include dependency graph for InputReader.cc:



2.8.1 Detailed Description

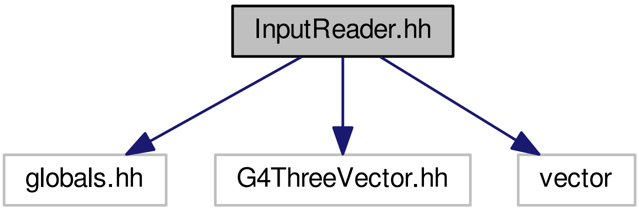
Implementation of the InputReader class.

2.9 InputReader.hh File Reference

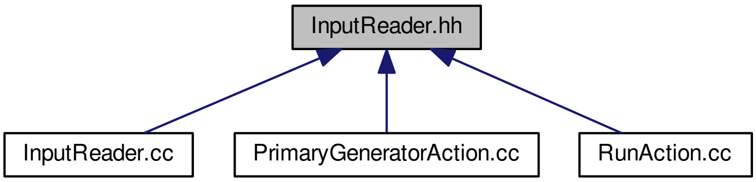
Definition of the InputReader class.

```
#include "globals.hh"
#include "G4ThreeVector.hh"
#include <vector>
```

Include dependency graph for InputReader.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class InputReader

    Read input file and interact with input macro-particles.

2.9.1 Detailed Description

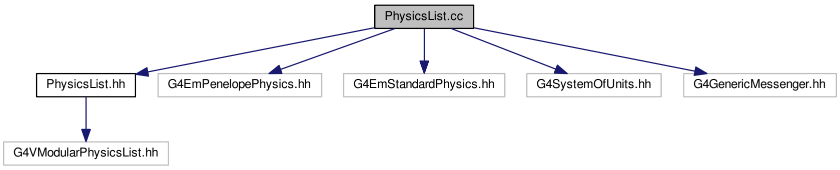
Definition of the InputReader class.

2.10 PhysicsList.cc File Reference

Implementation of the PhysicsList class.

```
#include "PhysicsList.hh"
#include "G4EmPenelopePhysics.hh"
#include "G4EmStandardPhysics.hh"
#include "G4SystemOfUnits.hh"
#include "G4GenericMessenger.hh"
```

Include dependency graph for PhysicsList.cc:



2.10.1 Detailed Description

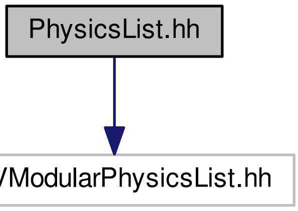
Implementation of the PhysicsList class.

2.11 PhysicsList.hh File Reference

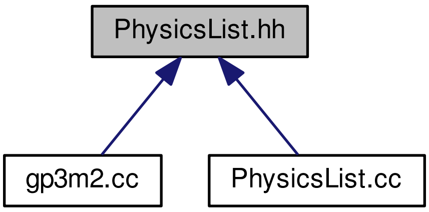
Definition of the PhysicsList class.

```
#include "G4VModularPhysicsList.hh"
```

Include dependency graph for PhysicsList.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class PhysicsList

  Define particles and processes to consider in the simulation.

2.11.1 Detailed Description

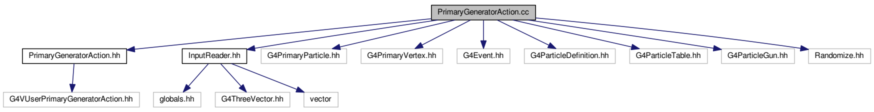
Definition of the PhysicsList class.

2.12 PrimaryGeneratorAction.cc File Reference

Implementation of the PrimaryGeneratorAction class.

```
#include "PrimaryGeneratorAction.hh"
#include "InputReader.hh"
#include "G4PrimaryParticle.hh"
#include "G4PrimaryVertex.hh"
#include "G4Event.hh"
#include "G4ParticleDefinition.hh"
#include "G4ParticleTable.hh"
#include "G4ParticleGun.hh"
#include "Randomize.hh"
```

Include dependency graph for PrimaryGeneratorAction.cc:



2.12.1 Detailed Description

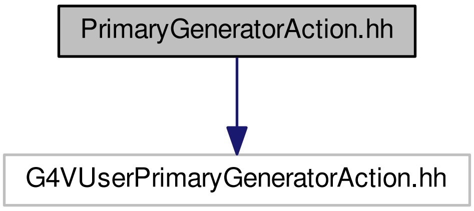
Implementation of the PrimaryGeneratorAction class.

2.13 PrimaryGeneratorAction.hh File Reference

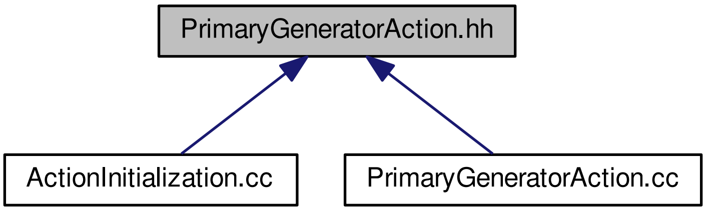
Definition of the PrimaryGeneratorAction class.

```
#include "G4VUserPrimaryGeneratorAction.hh"
```

Include dependency graph for PrimaryGeneratorAction.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class PrimaryGeneratorAction

  Generate primary particles.

2.13.1 Detailed Description

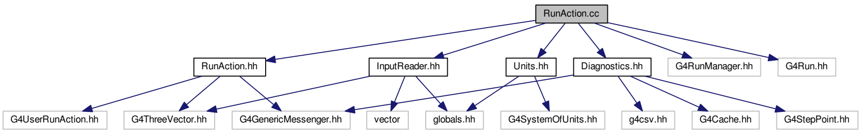
Definition of the PrimaryGeneratorAction class.

2.14 RunAction.cc File Reference

Implementation of the RunAction class.

```
#include "RunAction.hh"
#include "G4RunManager.hh"
#include "G4Run.hh"
#include "Units.hh"
#include "InputReader.hh"
#include "Diagnostics.hh"
```

Include dependency graph for RunAction.cc:



2.14.1 Detailed Description

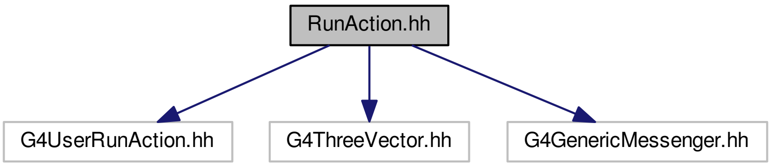
Implementation of the RunAction class.

2.15 RunAction.hh File Reference

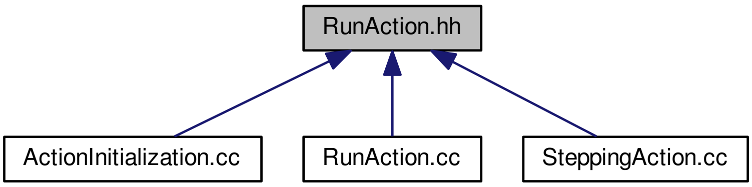
Definition of the RunAction class.

```
#include "G4UserRunAction.hh"
#include "G4ThreeVector.hh"
#include "G4GenericMessenger.hh"
```

Include dependency graph for RunAction.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class RunAction

  Deal with input file reading and diagnostic creation.

2.15.1 Detailed Description

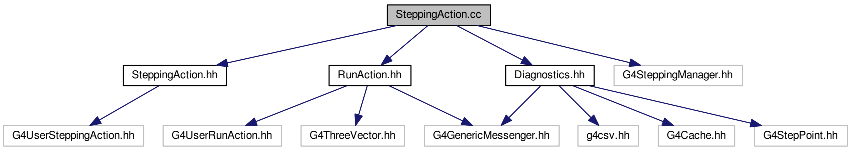
Definition of the RunAction class.

2.16 SteppingAction.cc File Reference

Implementation of the SteppingAction class.

```
#include "SteppingAction.hh"
#include "RunAction.hh"
#include "Diagnostics.hh"
#include "G4SteppingManager.hh"
```

Include dependency graph for SteppingAction.cc:



2.16.1 Detailed Description

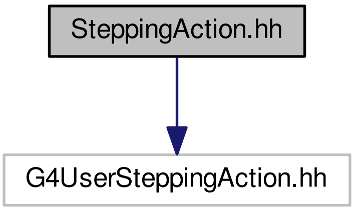
Implementation of the SteppingAction class.

2.17 SteppingAction.hh File Reference

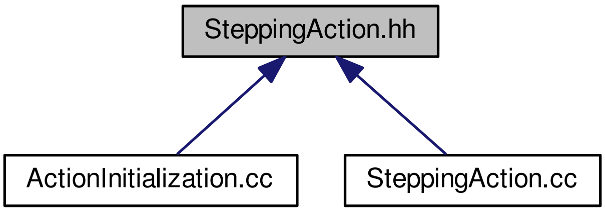
Definition of the SteppingAction class.

```
#include "G4UserSteppingAction.hh"
```

Include dependency graph for SteppingAction.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class SteppingAction

  Export particles phase-space at each geometry boundary.

2.17.1 Detailed Description

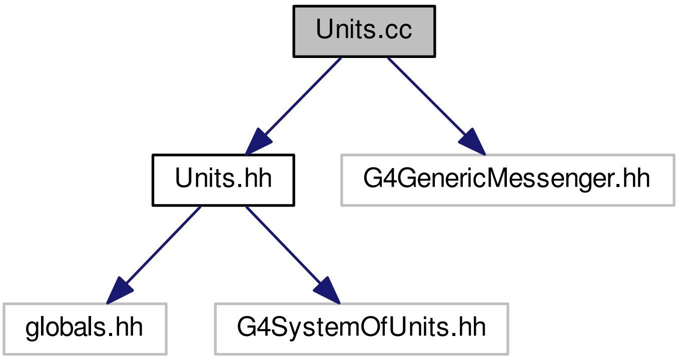
Definition of the SteppingAction class.

2.18 Units.cc File Reference

Implementation of the Units class.

```
#include "Units.hh"
#include "G4GenericMessenger.hh"
```

Include dependency graph for Units.cc:



2.18.1 Detailed Description

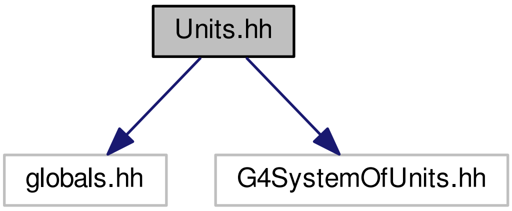
Implementation of the Units class.

2.19 Units.hh File Reference

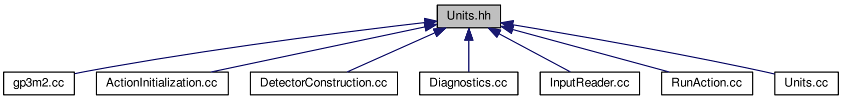
Definition of the Units class.

```
#include "globals.hh"
#include "G4SystemOfUnits.hh"
```

Include dependency graph for Units.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class Units

  Define units of the input and output files.

2.19.1 Detailed Description

Definition of the Units class.

Index