

gp3m2

0.1

Generated by Doxygen 1.8.11

Contents

1	Class Documentation	2
1.1	ActionInitialization Class Reference	2
1.1.1	Detailed Description	2
1.1.2	Constructor & Destructor Documentation	2
1.1.3	Member Function Documentation	2
1.1.4	Member Data Documentation	3
1.2	DetectorConstruction Class Reference	3
1.2.1	Detailed Description	3
1.2.2	Constructor & Destructor Documentation	4
1.2.3	Member Function Documentation	4
1.2.4	Member Data Documentation	5
1.3	PhysicsList Class Reference	6
1.3.1	Detailed Description	6
1.3.2	Constructor & Destructor Documentation	6
1.3.3	Member Function Documentation	7
1.3.4	Member Data Documentation	7
1.4	PrimaryGeneratorAction Class Reference	7
1.4.1	Detailed Description	8
1.4.2	Constructor & Destructor Documentation	8
1.4.3	Member Function Documentation	8
1.4.4	Member Data Documentation	9
1.5	RunAction Class Reference	9
1.5.1	Detailed Description	11
1.5.2	Constructor & Destructor Documentation	11
1.5.3	Member Function Documentation	11
1.5.4	Member Data Documentation	14
1.6	SteppingAction Class Reference	15
1.6.1	Detailed Description	16
1.6.2	Constructor & Destructor Documentation	16
1.6.3	Member Function Documentation	16
1.6.4	Member Data Documentation	16

2	File Documentation	17
2.1	ActionInitialization.cc File Reference	17
2.1.1	Detailed Description	17
2.2	ActionInitialization.hh File Reference	17
2.2.1	Detailed Description	18
2.3	Analysis.hh File Reference	18
2.3.1	Detailed Description	19
2.4	DetectorConstruction.cc File Reference	19
2.4.1	Detailed Description	19
2.5	DetectorConstruction.hh File Reference	20
2.5.1	Detailed Description	20
2.6	gp3m2.cc File Reference	21
2.6.1	Detailed Description	21
2.6.2	Function Documentation	21
2.7	PhysicsList.cc File Reference	21
2.7.1	Detailed Description	22
2.8	PhysicsList.hh File Reference	22
2.8.1	Detailed Description	22
2.9	PrimaryGeneratorAction.cc File Reference	23
2.9.1	Detailed Description	23
2.10	PrimaryGeneratorAction.hh File Reference	23
2.10.1	Detailed Description	24
2.11	RunAction.cc File Reference	24
2.11.1	Detailed Description	24
2.12	RunAction.hh File Reference	25
2.12.1	Detailed Description	25
2.13	SteppingAction.cc File Reference	26
2.13.1	Detailed Description	26
2.14	SteppingAction.hh File Reference	26
2.14.1	Detailed Description	27

1 Class Documentation

1.1 ActionInitialization Class Reference

Instantiate user classes in master or worker threads.

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

Public Member Functions

- [ActionInitialization](#) ()
- virtual [~ActionInitialization](#) ()
- virtual void [BuildForMaster](#) () const
Instantiate objects for the master thread.
- virtual void [Build](#) () const
Instantiate objects for the worker threads.

Private Attributes

- [RunAction](#) * [fMasterRunAction](#)

1.1.1 Detailed Description

Instantiate user classes in master or worker threads.

This class is instantiated only once

1.1.2 Constructor & Destructor Documentation

1.1.2.1 [ActionInitialization::ActionInitialization](#) ()

1.1.2.2 [ActionInitialization::~~ActionInitialization](#) () [virtual]

1.1.3 Member Function Documentation

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

Instantiate objects for the worker threads.

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

Instantiate objects for the master thread.

1.1.4 Member Data Documentation

1.1.4.1 RunAction* ActionInitialization::fMasterRunAction [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](#) ()
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 width_um)
Add a new layer to the target.
- void [SetCommands](#) ()
Set commands to be interpreted with the UI.

Private Attributes

- G4int [fNumberOfLayers](#)
Total number of layers in the target.
- G4bool [fCheckOverlaps](#)
Check if volumes are overlapping.
- G4double [fTargetSizeX](#)
Total target longitudinal size.
- G4double [fTargetSizeYZ](#)
Target transverse size.
- G4LogicalVolume * [fWorldLV](#)
Pointer to the world logical volume.
- G4GenericMessenger * [fMessenger](#)
Pointer to the G4GenericMessenger instance.

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 ()

Initialize pointers, set default values and creates UI commands.

Here is the call graph for this function:



1.2.2.2 DetectorConstruction::~~DetectorConstruction ()

Delete messenger.

1.2.3 Member Function Documentation

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

Add a new layer to the target.

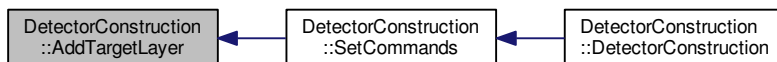
Parameters

<i>materialName</i>	: Material name of the new layer
<i>width_um</i>	: Width of the new layer (in um)

The transverse size of the new layer is always the same default value.

The new width is added to `fTargetSizeX` and `fNumberOfLayers` is incremented.

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.

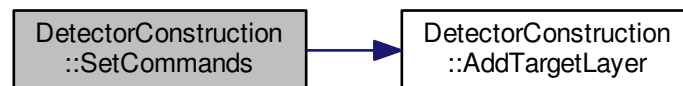
1.2.3.3 void DetectorConstruction::SetCommands ()

Set commands to be interpreted with the UI.

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

```
/target/addLayer materialName width_um
```

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 G4double DetectorConstruction::fTargetSizeX [private]

Total target longitudinal size.

1.2.4.5 G4double DetectorConstruction::fTargetSizeYZ [private]

Target transverse size.

1.2.4.6 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 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 instantiate 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.

Private Attributes

- G4VPhysicsConstructor * [fEmPhysicsList](#)
Pointer to the used pre-packaged [PhysicsList](#).

1.3.1 Detailed Description

Define particles and processes to consider in the simulation.

1.3.2 Constructor & Destructor Documentation

1.3.2.1 PhysicsList::PhysicsList ()

Set default cut values, verbosity and instantiate desired pre-packages [PhysicsList](#).

1.3.2.2 PhysicsList::~~PhysicsList ()

Delete pre-packaged [PhysicsList](#) instance.

1.3.3 Member Function Documentation

1.3.3.1 void PhysicsList::ConstructParticle () [virtual]

Construct the particles to consider in the simulation.

The list of particles is taken from choosen pre-packaged [PhysicsList](#).

1.3.3.2 void PhysicsList::ConstructProcess () [virtual]

Construct the physics processes to use in the simulation.

The list of processes is taken from choosen pre-packaged [PhysicsList](#).

1.3.3.3 void PhysicsList::SetCuts () [virtual]

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

?

1.3.4 Member Data Documentation

1.3.4.1 G4VPhysicsConstructor* PhysicsList::fEmPhysicsList [private]

Pointer to the used pre-packaged [PhysicsList](#).

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

- [PhysicsList.hh](#)
- [PhysicsList.cc](#)

1.4 PrimaryGeneratorAction Class Reference

Generate primary particles.

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

Public Member Functions

- [PrimaryGeneratorAction](#) ([RunAction](#) *masterRunAction)
Instantiate G4ParticleGun and define default primary particles properties.
- [~PrimaryGeneratorAction](#) ()
Do nothing.
- virtual void [GeneratePrimaries](#) (G4Event *)
Generate primary particles.
- void [SetCommands](#) ()
Define UI commands.

Private Attributes

- G4String `fParticleName`
- G4ParticleTable * `fParticleTable`
- RunAction * `fMasterRunAction`

1.4.1 Detailed Description

Generate primary particles.

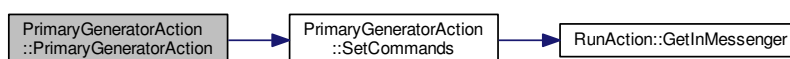
This class is instantiated in each worker thread.

1.4.2 Constructor & Destructor Documentation

1.4.2.1 PrimaryGeneratorAction::PrimaryGeneratorAction (RunAction * *masterRunAction*)

Instantiate G4ParticleGun and define default primary particles properties.

Here is the call graph for this function:



1.4.2.2 PrimaryGeneratorAction::~~PrimaryGeneratorAction ()

Do nothing.

1.4.3 Member Function Documentation

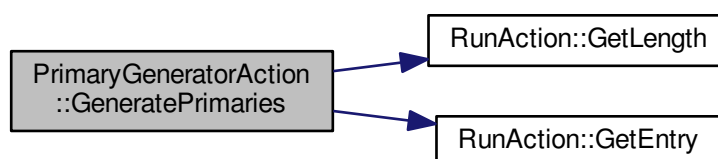
1.4.3.1 void PrimaryGeneratorAction::GeneratePrimaries (G4Event * *anEvent*) [virtual]

Generate primary particles.

A random configuration is taken from the input file, and the primary particle properties are defined with this configuration.

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

Here is the call graph for this function:

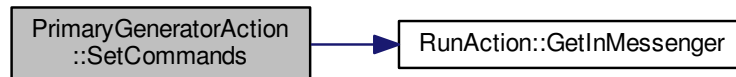


1.4.3.2 void PrimaryGeneratorAction::SetCommands ()

Define UI commands.

The primary particle type can be changed by using /input/setParticle particleName

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 `RunAction* PrimaryGeneratorAction::fMasterRunAction` [private]

1.4.4.2 `G4String PrimaryGeneratorAction::fParticleName` [private]

1.4.4.3 `G4ParticleTable* PrimaryGeneratorAction::fParticleTable` [private]

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

- [PrimaryGeneratorAction.hh](#)
- [PrimaryGeneratorAction.cc](#)

1.5 RunAction Class Reference

Creates and writes diagnostic output files.

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

Public Member Functions

- [RunAction](#) ()
Create analysis manager, Ntuples and set UI commands.
- [~RunAction](#) ()
Delete analysis manager.
- virtual void [BeginOfRunAction](#) (const G4Run *aRun)
Read input file & open output file.
- virtual void [EndOfRunAction](#) (const G4Run *)
Write and close output file.
- void [FillData](#) (const G4ParticleDefinition *part, G4double weight, G4ThreeVector position, G4ThreeVector momentum, G4double time)
Add a new line to the corresponding Ntuple.
- void [SetCommands](#) ()
Define UI commands.
- void [ReadInput](#) ()
- G4double [GetEntry](#) (G4String variable, G4int id) const
- G4int [GetLength](#) () const
- G4GenericMessenger * [GetInMessenger](#) ()
- G4GenericMessenger * [GetOutMessenger](#) ()
- G4double [GetLowEnergyLimit](#) ()

Private Attributes

- G4AnalysisManager * [fAnalysisManager](#)
Pointer to the G4AnalysisManager instance.
- G4String [fOutFileName](#)
Output file name.
- G4String [fInFileName](#)
Input file name.
- const G4ParticleDefinition * [fGamma](#)
Gamma particle definition.
- const G4ParticleDefinition * [fElectron](#)
Electron particle definition.
- const G4ParticleDefinition * [fPositron](#)
Positron particle definition.
- G4GenericMessenger * [fOutMessenger](#)
Pointer to the G4GenericMessenger instance for the output file.
- G4GenericMessenger * [fInMessenger](#)
Pointer to the G4GenericMessenger instance for the input file.
- G4double [fLowEnergyLimit](#)
- G4double [fNormW](#)
- 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](#)

1.5.1 Detailed Description

Creates and writes diagnostic output files.

1.5.2 Constructor & Destructor Documentation

1.5.2.1 RunAction::RunAction ()

Create analysis manager, Ntuples and set UI commands.

Here is the call graph for this function:



1.5.2.2 RunAction::~~RunAction ()

Delete analysis manager.

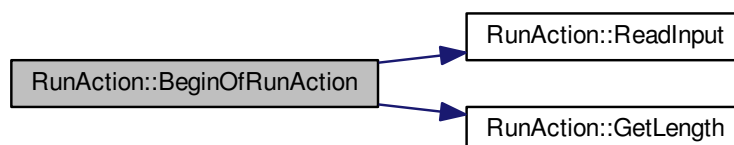
1.5.3 Member Function Documentation

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

Read input file & open output file.

This user code is executed at the beginning of each run

Here is the call graph for this function:



1.5.3.2 void RunAction::EndOfRunAction (const G4Run *) [virtual]

Write and close output file.

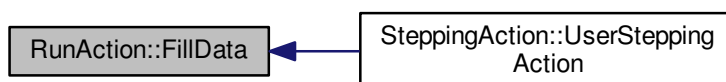
This user code is executed at the end of each run

1.5.3.3 void RunAction::FillData (const G4ParticleDefinition * *part*, G4double *weight*, G4ThreeVector *position*, G4ThreeVector *momentum*, G4double *time*)

Add a new line to the corresponding Ntuple.

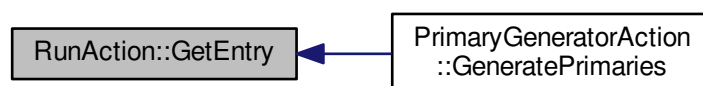
All the quantities must be given in code units.

Here is the caller graph for this function:



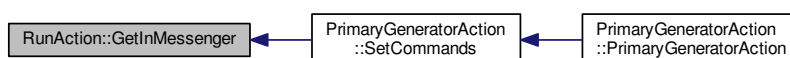
1.5.3.4 G4double RunAction::GetEntry (G4String *variable*, G4int *id*) const [inline]

Here is the caller graph for this function:



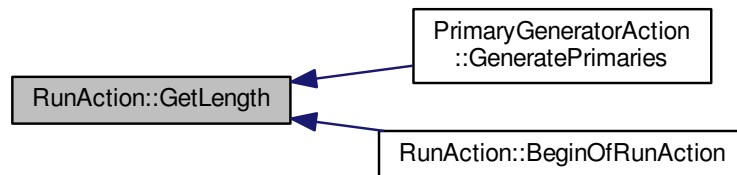
1.5.3.5 G4GenericMessenger* RunAction::GetInMessenger () [inline]

Here is the caller graph for this function:



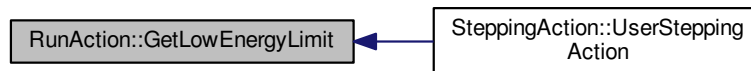
1.5.3.6 G4int RunAction::GetLength () const [inline]

Here is the caller graph for this function:



1.5.3.7 G4double RunAction::GetLowEnergyLimit () [inline]

Here is the caller graph for this function:



1.5.3.8 G4GenericMessenger* RunAction::GetOutMessenger () [inline]

1.5.3.9 void RunAction::ReadInput ()

Here is the caller graph for this function:



1.5.3.10 void RunAction::SetCommands ()

Define UI commands.

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

The output file name can be changed by using /output/setFileName fileName

Here is the caller graph for this function:



1.5.4 Member Data Documentation

1.5.4.1 G4AnalysisManager* RunAction::fAnalysisManager [private]

Pointer to the G4AnalysisManager instance.

1.5.4.2 const G4ParticleDefinition* RunAction::fElectron [private]

Electron particle definition.

1.5.4.3 const G4ParticleDefinition* RunAction::fGamma [private]

Gamma particle definition.

1.5.4.4 G4String RunAction::fInFileName [private]

Input file name.

1.5.4.5 G4GenericMessenger* RunAction::fInMessenger [private]

Pointer to the G4GenericMessenger instance for the input file.

1.5.4.6 G4double RunAction::fLowEnergyLimit [private]

1.5.4.7 G4double RunAction::fNormW [private]

1.5.4.8 G4String RunAction::fOutFileName [private]

Output file name.

1.5.4.9 `G4GenericMessenger* RunAction::fOutMessenger` [private]

Pointer to the `G4GenericMessenger` instance for the output file.

1.5.4.10 `const G4ParticleDefinition* RunAction::fPositron` [private]

Positron particle definition.

1.5.4.11 `std::vector<G4double> RunAction::fPx` [private]

1.5.4.12 `std::vector<G4double> RunAction::fPy` [private]

1.5.4.13 `std::vector<G4double> RunAction::fPz` [private]

1.5.4.14 `std::vector<G4double> RunAction::fT` [private]

1.5.4.15 `std::vector<G4double> RunAction::fW` [private]

1.5.4.16 `std::vector<G4double> RunAction::fX` [private]

1.5.4.17 `std::vector<G4double> RunAction::fY` [private]

1.5.4.18 `std::vector<G4double> RunAction::fZ` [private]

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

- [RunAction.hh](#)
- [RunAction.cc](#)

1.6 SteppingAction Class Reference

Export particles phase-space at each geometry boundary.

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

Public Member Functions

- [SteppingAction](#) ([RunAction](#) *runAction)
Save pointer to the current [RunAction](#) instance.
- [~SteppingAction](#) ()
Delete pointer to the current [RunAction](#) instance.
- virtual void [UserSteppingAction](#) (const `G4Step` *)
Add a new line to the corresponding Ntuple if the particle is crossing a boundary.

Private Attributes

- [RunAction](#) * [fRunAction](#)
Pointer to the [RunAction](#) instance of the current thread.

1.6.1 Detailed Description

Export particles phase-space at each geometry boundary.

This class is instanciated in each worker thread

1.6.2 Constructor & Destructor Documentation

1.6.2.1 SteppingAction::SteppingAction (RunAction * runAction)

Save pointer to the current [RunAction](#) instance.

1.6.2.2 SteppingAction::~~SteppingAction ()

Delete pointer to the current [RunAction](#) instance.

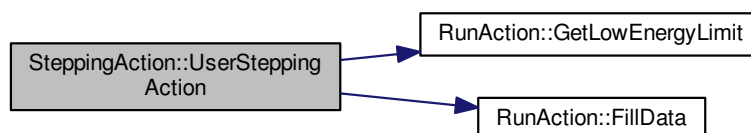
1.6.3 Member Function Documentation

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

Add a new line to the corresponding Ntuple if the particle is crossing a boundary.

This virtual method is called at each Step ends.

Here is the call graph for this function:



1.6.4 Member Data Documentation

1.6.4.1 RunAction* SteppingAction::fRunAction [private]

Pointer to the [RunAction](#) instance of the current thread.

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

- [SteppingAction.hh](#)
- [SteppingAction.cc](#)

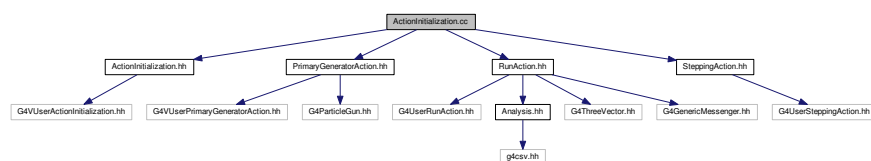
2 File Documentation

2.1 ActionInitialization.cc File Reference

Implementation of the [ActionInitialization](#) class.

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

Include dependency graph for ActionInitialization.cc:



2.1.1 Detailed Description

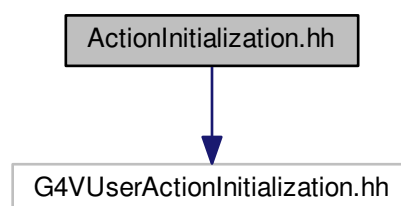
Implementation of the [ActionInitialization](#) class.

2.2 ActionInitialization.hh File Reference

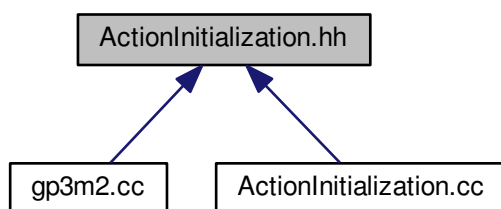
Definition of the [ActionInitialization](#) class.

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

Include dependency graph for ActionInitialization.hh:



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



Classes

- class [ActionInitialization](#)

Instantiate user classes in master or worker threads.

2.2.1 Detailed Description

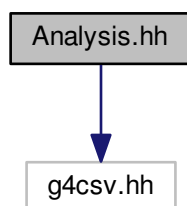
Definition of the [ActionInitialization](#) class.

2.3 Analysis.hh File Reference

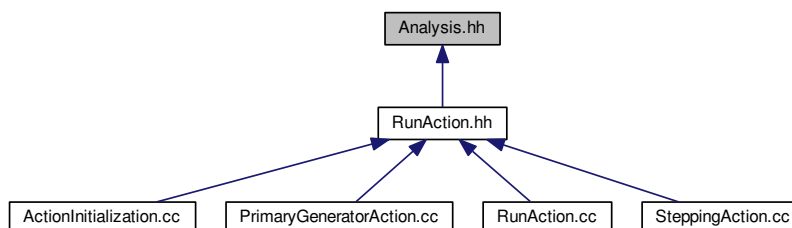
Selection of the analysis technology.

```
#include "g4csv.hh"
```

Include dependency graph for `Analysis.hh`:



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



2.3.1 Detailed Description

Selection of the analysis technology.

2.4 DetectorConstruction.cc File Reference

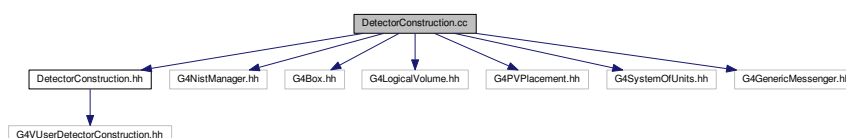
Implementation of the [DetectorConstruction](#) class.

```

#include "DetectorConstruction.hh"
#include "G4NistManager.hh"
#include "G4Box.hh"
#include "G4LogicalVolume.hh"
#include "G4PVPlacement.hh"
#include "G4SystemOfUnits.hh"
#include "G4GenericMessenger.hh"

```

Include dependency graph for DetectorConstruction.cc:



2.4.1 Detailed Description

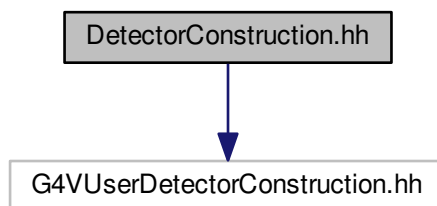
Implementation of the [DetectorConstruction](#) class.

2.5 DetectorConstruction.hh File Reference

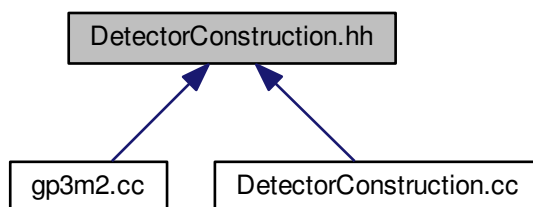
Definition of the [DetectorConstruction](#) class.

```
#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.

2.5.1 Detailed Description

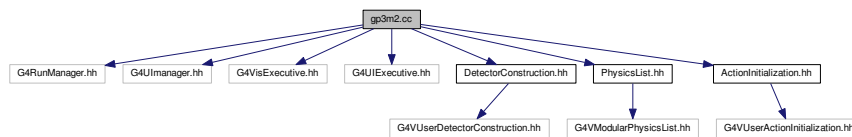
Definition of the [DetectorConstruction](#) class.

2.6 gp3m2.cc File Reference

Main program of gp3m2.

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

Include dependency graph for gp3m2.cc:



Functions

- int [main](#) (int argc, char **argv)

2.6.1 Detailed Description

Main program of gp3m2.

2.6.2 Function Documentation

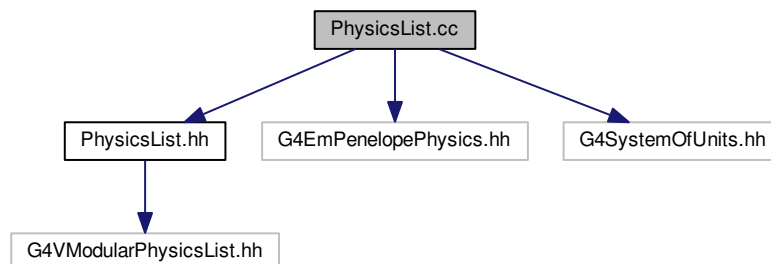
2.6.2.1 int main (int argc, char ** argv)

2.7 PhysicsList.cc File Reference

Implementation of the [PhysicsList](#) class.

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

Include dependency graph for PhysicsList.cc:



2.7.1 Detailed Description

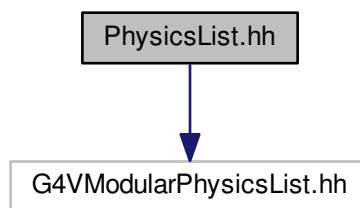
Implementation of the [PhysicsList](#) class.

2.8 PhysicsList.hh File Reference

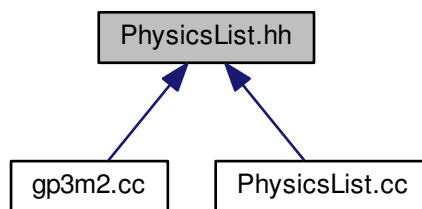
Definition of the [PhysicsList](#) class.

```
#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.

2.8.1 Detailed Description

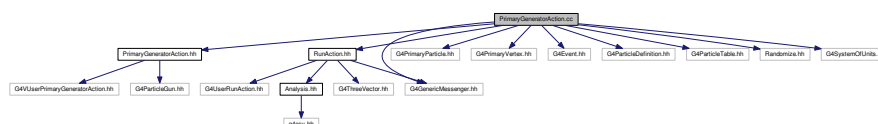
Definition of the [PhysicsList](#) class.

2.9 PrimaryGeneratorAction.cc File Reference

Implementation of the `PrimaryGeneratorAction` class.

```
#include "PrimaryGeneratorAction.hh"
#include "RunAction.hh"
#include "G4PrimaryParticle.hh"
#include "G4PrimaryVertex.hh"
#include "G4Event.hh"
#include "G4ParticleDefinition.hh"
#include "G4ParticleTable.hh"
#include "Randomize.hh"
#include "G4SystemOfUnits.hh"
#include "G4GenericMessenger.hh"
```

Include dependency graph for PrimaryGeneratorAction.cc:



2.9.1 Detailed Description

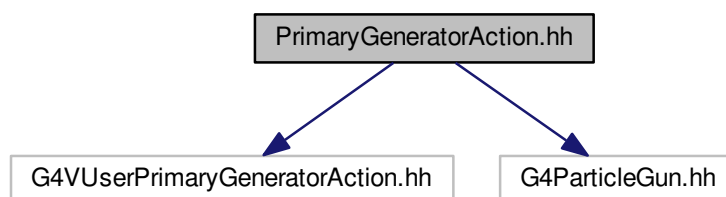
Implementation of the `PrimaryGeneratorAction` class.

2.10 PrimaryGeneratorAction.hh File Reference

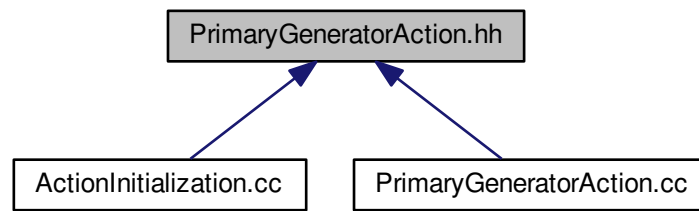
Definition of the `PrimaryGeneratorAction` class.

```
#include "G4VUserPrimaryGeneratorAction.hh"
#include "G4ParticleGun.hh"
```

Include dependency graph for PrimaryGeneratorAction.hh:



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



Classes

- class [PrimaryGeneratorAction](#)
Generate primary particles.

2.10.1 Detailed Description

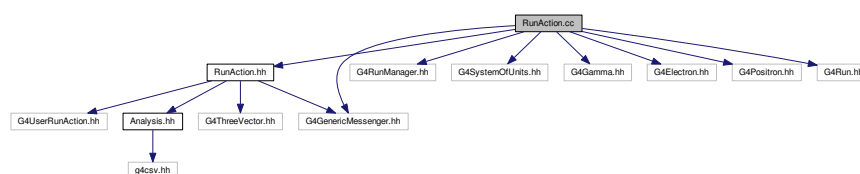
Definition of the [PrimaryGeneratorAction](#) class.

2.11 RunAction.cc File Reference

Implementation of the [RunAction](#) class.

```
#include "RunAction.hh"
#include "G4RunManager.hh"
#include "G4SystemOfUnits.hh"
#include "G4Gamma.hh"
#include "G4Electron.hh"
#include "G4Positron.hh"
#include "G4GenericMessenger.hh"
#include "G4Run.hh"
```

Include dependency graph for `RunAction.cc`:



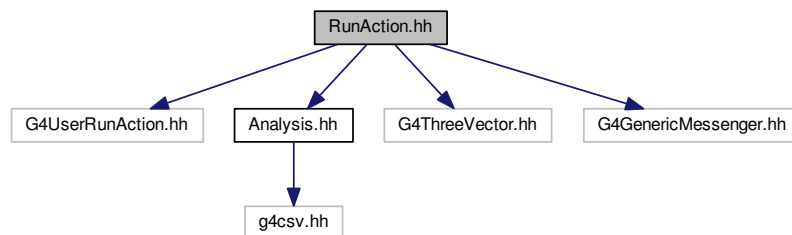
2.11.1 Detailed Description

Implementation of the [RunAction](#) class.

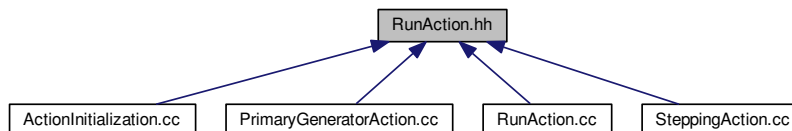
2.12 RunAction.hh File Reference

Definition of the [RunAction](#) class.

```
#include "G4UserRunAction.hh"
#include "Analysis.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](#)

Creates and writes diagnostic output files.

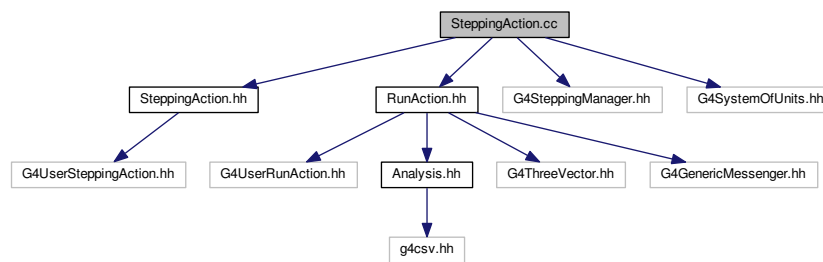
2.12.1 Detailed Description

Definition of the [RunAction](#) class.

2.13 SteppingAction.cc File Reference

Implementation of the [SteppingAction](#) class.

```
#include "SteppingAction.hh"
#include "RunAction.hh"
#include "G4SteppingManager.hh"
#include "G4SystemOfUnits.hh"
Include dependency graph for SteppingAction.cc:
```



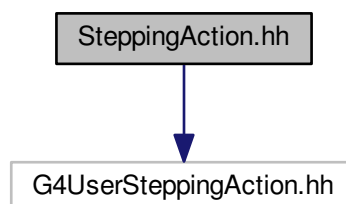
2.13.1 Detailed Description

Implementation of the [SteppingAction](#) class.

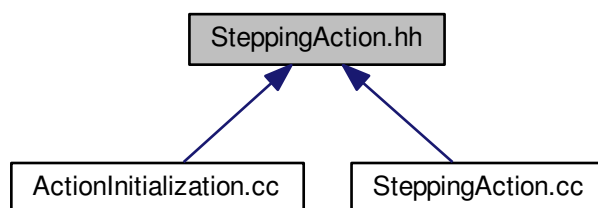
2.14 SteppingAction.hh File Reference

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.

2.14.1 Detailed Description

Definition of the [SteppingAction](#) class.

Index

- ~ActionInitialization
 - ActionInitialization, [2](#)
- ~DetectorConstruction
 - DetectorConstruction, [4](#)
- ~PhysicsList
 - PhysicsList, [6](#)
- ~PrimaryGeneratorAction
 - PrimaryGeneratorAction, [8](#)
- ~RunAction
 - RunAction, [11](#)
- ~SteppingAction
 - SteppingAction, [16](#)
- ActionInitialization, [2](#)
 - ~ActionInitialization, [2](#)
 - ActionInitialization, [2](#)
 - Build, [2](#)
 - BuildForMaster, [2](#)
 - fMasterRunAction, [3](#)
- ActionInitialization.cc, [17](#)
- ActionInitialization.hh, [17](#)
- AddTargetLayer
 - DetectorConstruction, [4](#)
- Analysis.hh, [18](#)
- BeginOfRunAction
 - RunAction, [11](#)
- Build
 - ActionInitialization, [2](#)
- BuildForMaster
 - ActionInitialization, [2](#)
- Construct
 - DetectorConstruction, [4](#)
- ConstructParticle
 - PhysicsList, [7](#)
- ConstructProcess
 - PhysicsList, [7](#)
- DetectorConstruction, [3](#)
 - ~DetectorConstruction, [4](#)
 - AddTargetLayer, [4](#)
 - Construct, [4](#)
 - DetectorConstruction, [4](#)
 - fCheckOverlaps, [5](#)
 - fMessenger, [5](#)
 - fNumberOfLayers, [5](#)
 - fTargetSizeYZ, [5](#)
 - fTargetSizeX, [5](#)
 - fWorldLV, [6](#)
 - SetCommands, [4](#)
- DetectorConstruction.cc, [19](#)
- DetectorConstruction.hh, [20](#)
- EndOfRunAction
 - RunAction, [11](#)
- fAnalysisManager
 - RunAction, [14](#)
- fCheckOverlaps
 - DetectorConstruction, [5](#)
- fElectron
 - RunAction, [14](#)
- fEmPhysicsList
 - PhysicsList, [7](#)
- fGamma
 - RunAction, [14](#)
- fInFileName
 - RunAction, [14](#)
- fInMessenger
 - RunAction, [14](#)
- fLowEnergyLimit
 - RunAction, [14](#)
- fMasterRunAction
 - ActionInitialization, [3](#)
 - PrimaryGeneratorAction, [9](#)
- fMessenger
 - DetectorConstruction, [5](#)
- fNormW
 - RunAction, [14](#)
- fNumberOfLayers
 - DetectorConstruction, [5](#)
- fOutFileName
 - RunAction, [14](#)
- fOutMessenger
 - RunAction, [14](#)
- fParticleName
 - PrimaryGeneratorAction, [9](#)
- fParticleTable
 - PrimaryGeneratorAction, [9](#)
- fPositron
 - RunAction, [15](#)
- fPx
 - RunAction, [15](#)
- fPy
 - RunAction, [15](#)
- fPz
 - RunAction, [15](#)
- fRunAction
 - SteppingAction, [16](#)
- fTargetSizeYZ
 - DetectorConstruction, [5](#)
- fTargetSizeX
 - DetectorConstruction, [5](#)
- fWorldLV
 - DetectorConstruction, [6](#)
- FillData
 - RunAction, [12](#)
- fT
 - RunAction, [15](#)
- fW
 - RunAction, [15](#)

- fX
 - RunAction, 15
- fY
 - RunAction, 15
- fZ
 - RunAction, 15
- GeneratePrimaries
 - PrimaryGeneratorAction, 8
- GetEntry
 - RunAction, 12
- GetInMessenger
 - RunAction, 12
- GetLength
 - RunAction, 12
- GetLowEnergyLimit
 - RunAction, 13
- GetOutMessenger
 - RunAction, 13
- gp3m2.cc, 21
 - main, 21
- main
 - gp3m2.cc, 21
- PhysicsList, 6
 - ~PhysicsList, 6
 - ConstructParticle, 7
 - ConstructProcess, 7
 - fEmPhysicsList, 7
 - PhysicsList, 6
 - SetCuts, 7
- PhysicsList.cc, 21
- PhysicsList.hh, 22
- PrimaryGeneratorAction, 7
 - ~PrimaryGeneratorAction, 8
 - fMasterRunAction, 9
 - fParticleName, 9
 - fParticleTable, 9
 - GeneratePrimaries, 8
 - PrimaryGeneratorAction, 8
 - SetCommands, 8
- PrimaryGeneratorAction.cc, 23
- PrimaryGeneratorAction.hh, 23
- ReadInput
 - RunAction, 13
- RunAction, 9
 - ~RunAction, 11
 - BeginOfRunAction, 11
 - EndOfRunAction, 11
 - fAnalysisManager, 14
 - fElectron, 14
 - fGamma, 14
 - fInFileName, 14
 - fInMessenger, 14
 - fLowEnergyLimit, 14
 - fNormW, 14
 - fOutFileName, 14
 - fOutMessenger, 14
 - fPositron, 15
 - fPx, 15
 - fPy, 15
 - fPz, 15
 - FillData, 12
 - fT, 15
 - fW, 15
 - fX, 15
 - fY, 15
 - fZ, 15
 - GetEntry, 12
 - GetInMessenger, 12
 - GetLength, 12
 - GetLowEnergyLimit, 13
 - GetOutMessenger, 13
 - ReadInput, 13
 - RunAction, 11
 - SetCommands, 13
- RunAction.cc, 24
- RunAction.hh, 25
- SetCommands
 - DetectorConstruction, 4
 - PrimaryGeneratorAction, 8
 - RunAction, 13
- SetCuts
 - PhysicsList, 7
- SteppingAction, 15
 - ~SteppingAction, 16
 - fRunAction, 16
 - SteppingAction, 16
 - UserSteppingAction, 16
- SteppingAction.cc, 26
- SteppingAction.hh, 26
- UserSteppingAction
 - SteppingAction, 16