



# Generic Biasing

**Genproc meeting  
2025/01/29**

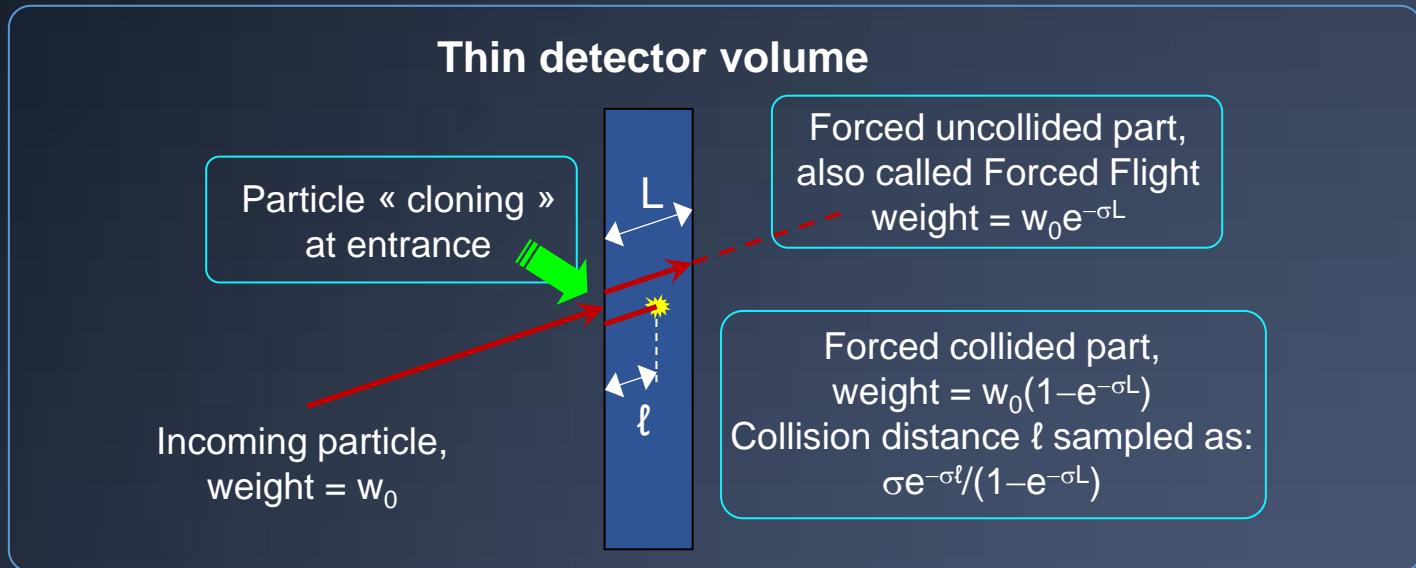
Marc Verderi  
LLR, Ecole polytechnique

# Generic Biasing Scheme

- › **Adopt an “object oriented” (OO) scheme to handle any type of (forward) biasing**
  - It defines an abstract layer, general enough
  - It provides some ready-to-go options based on this layer
  - It is open to users, for customized implementations
- › **It handles:**
  - Physics process biasing → denoted as “physics biasing”
  - Splitting/killing → denoted as “non-physics biasing”
- › **It is aimed at being “modular”:**
  - By defining simple techniques (kind of “primitive of biasing”)
  - That are combined to provide a given biasing scheme

# Modularity ?

- › Example of “force collision” scheme à la MCNP



- › The “particle cloning”, “forced non-collision”, “forced collision” actions can be seen as sort of biasing primitives, we call “biasing operations”
  - They are combined here to provide the “force collision” scheme
  - But they could be used in other schemes
- › Generic Biasing Scheme idea:
  - Define a class for “**biasing operations**”
  - Define a class for “**biasing operators**”, that decide for biasing operations
  - Provide the **interface** of these classes with the tracking

# Generic Biasing Components

## > G4VBiasingOperation

- An abstract class
- Can act on a physics process:
  - > By modifying its interaction law
  - > By modifying its final state generation
- Can act by itself
  - > To split or randomly kill particles

## > G4VBiasingOperator

- An abstract class
- Selects G4VBiasingOperation's
  - > At the beginning of the step
  - > At the final state generation level
- ***This is the entity which is making all the decisions***

## > G4BiasingProcessInterface

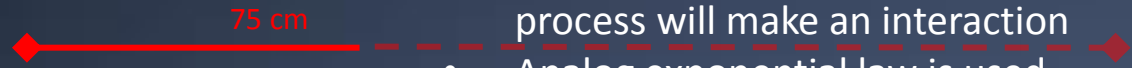
- A concrete class
- Which can wrap a physics process, to bias it
  - > Applying “physics biasing” options
- Which can have no process, to apply splitting & killing techniques
  - > Applying “non-physics biasing” options
- It makes the interface between the biasing classes and the tracking
  - > By collecting and applying the decisions of the current G4VBiasingOperator (if any)

G4BiasingProcessInterface  
G4GammaConversion

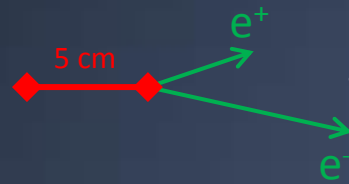
G4BiasingProcessInterface  
(no process)

# “physics biasing”

G4PhotoelectricEffect



G4GammaConversion



G4ComptonScattering



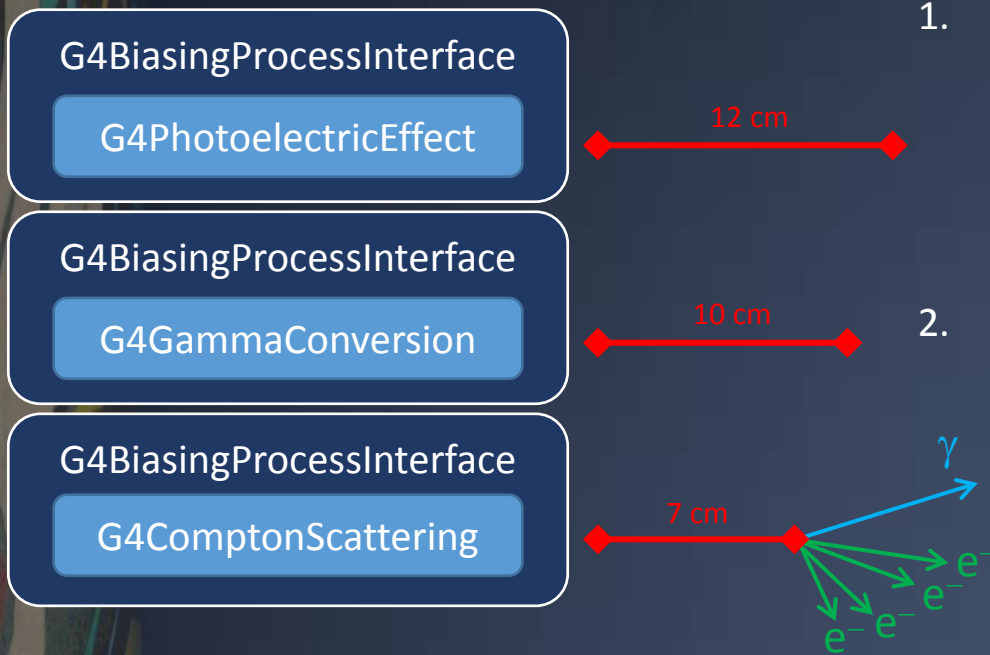
## 1. GetPostStepPhysicalInteractionLength()

- Returns the distance at which the process will make an interaction
- Analog exponential law is used

## 2. PostStepDoIt()

- Called if the process has responded the shortest of the interaction distances
- Generate final state, according to specific process analog physical law

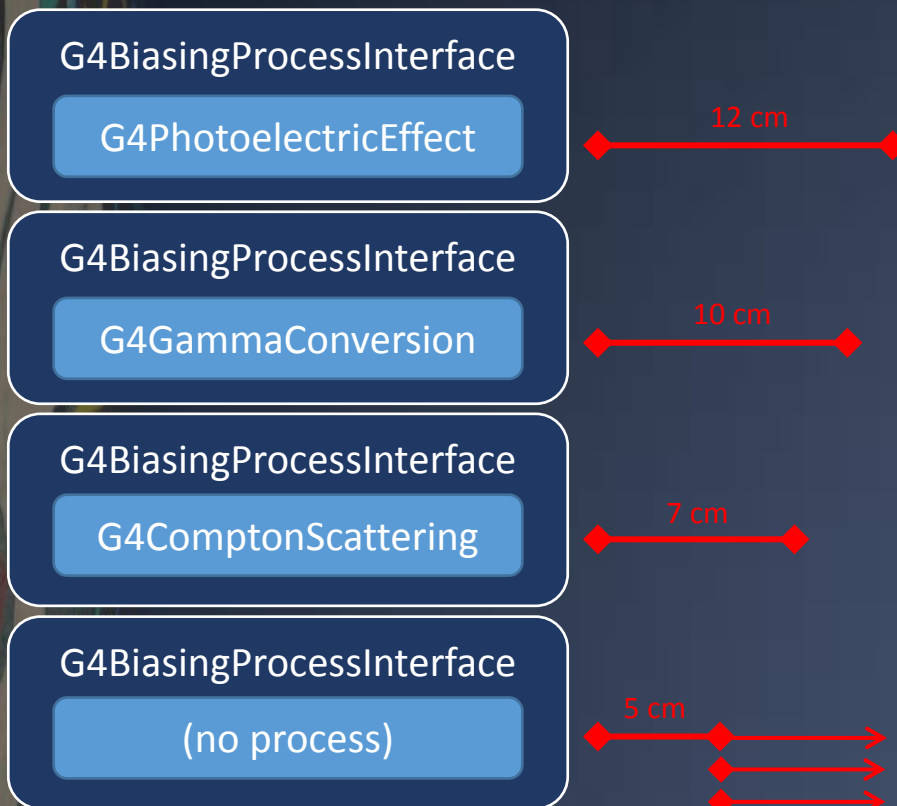
# “physics biasing”



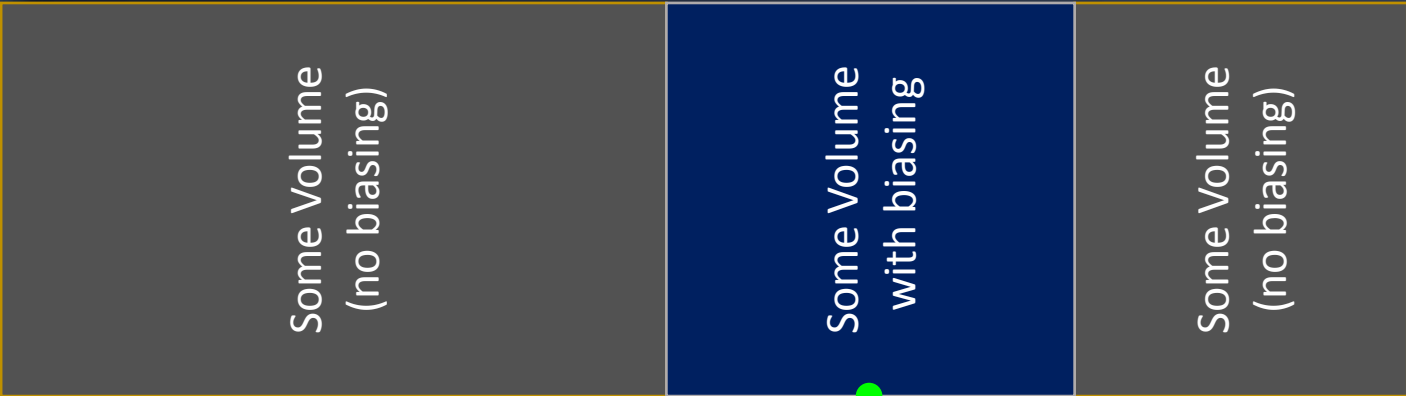
1. `GetPostStepPhysicalInteractionLength()`
  - Returns the distance at which the process will make an interaction
  - ~~Analog exponential law is used~~
  - **A biased interaction law is used (if wished)**
2. `PostStepDoIt()`
  - Called if the process has responded the shortest of the interaction distances
  - ~~Generate final state, according to specific process analog physical law~~
  - **Generate final state, according to biased law (if wished)**



# “physics biasing” + “non-physics biasing”



1. `GetPostStepPhysicalInteractionLength()`
    - Returns the distance at which the process will make an interaction
    - ~~Analog exponential law is used~~
    - **A biased interaction law is used (if wished)**
  2. `PostStepDoIt()`
    - Called if the process has responded the shortest of the interaction distances
    - ~~Generate final state, according to specific process analog physical law~~
    - **Generate final state, according to biased law (if wished)**
- “non-physics biasing” competes with other processes (as any process)
    - Here, winning the race
    - Here, applying a splitting by 3



G4BiasingProcessInterface  
(no process)

G4BiasingProcessInterface  
G4PhotoelectricEffect

G4BiasingProcessInterface  
G4GammaConversion

G4BiasingProcessInterface  
G4ComptonScattering

MyBiasingOperator

G4FlatForce-InteractionOperation

G4ExponentialForce-InteractionOperation

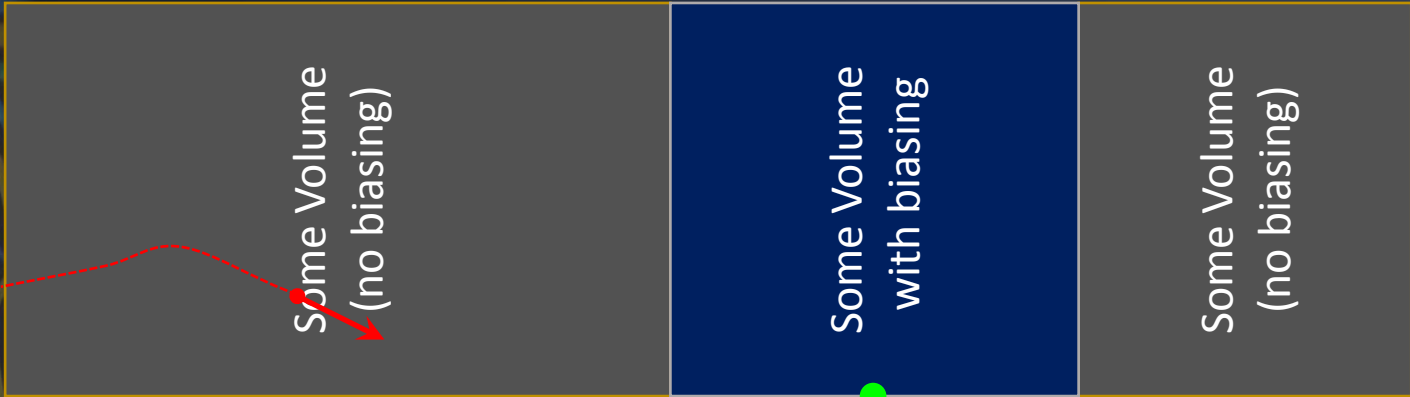
G4ForceFreeFlight-InteractionOperation

G4SplittingOperation

G4WeightWindow-Operation

G4MCNPForce-CollisionOperator





G4BiasingProcessInterface  
(no process)

G4BiasingProcessInterface  
G4PhotoelectricEffect

G4BiasingProcessInterface  
G4GammaConversion

G4BiasingProcessInterface  
G4ComptonScattering

MyBiasingOperator

G4FlatForce-InteractionOperation

G4ExponentialForce-InteractionOperation

G4ForceFreeFlight-InteractionOperation

G4SplittingOperation

G4WeightWindow-Operation

G4MCNPForce-CollisionOperator



G4BiasingProcessInterface  
(no process)

G4BiasingProcessInterface  
G4PhotoelectricEffect

G4BiasingProcessInterface  
G4GammaConversion

G4BiasingProcessInterface  
G4ComptonScattering

MyBiasingOperator

G4FlatForce-InteractionOperation

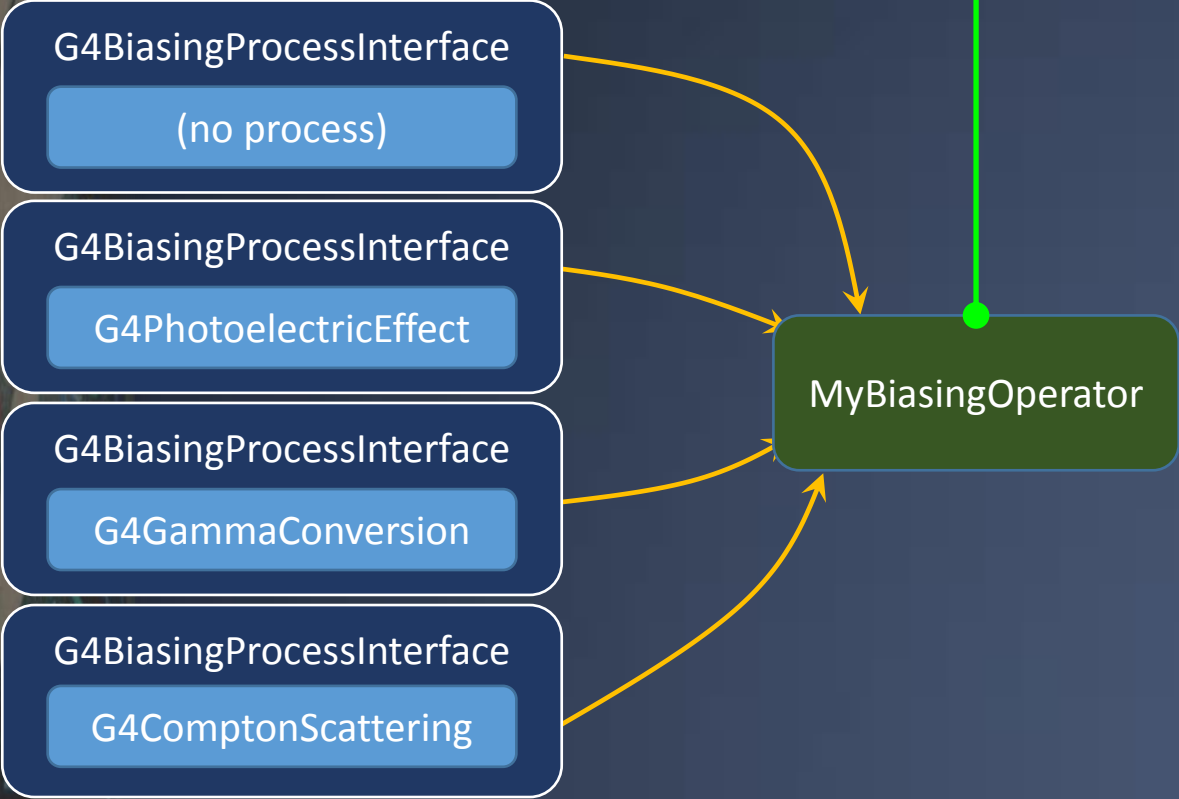
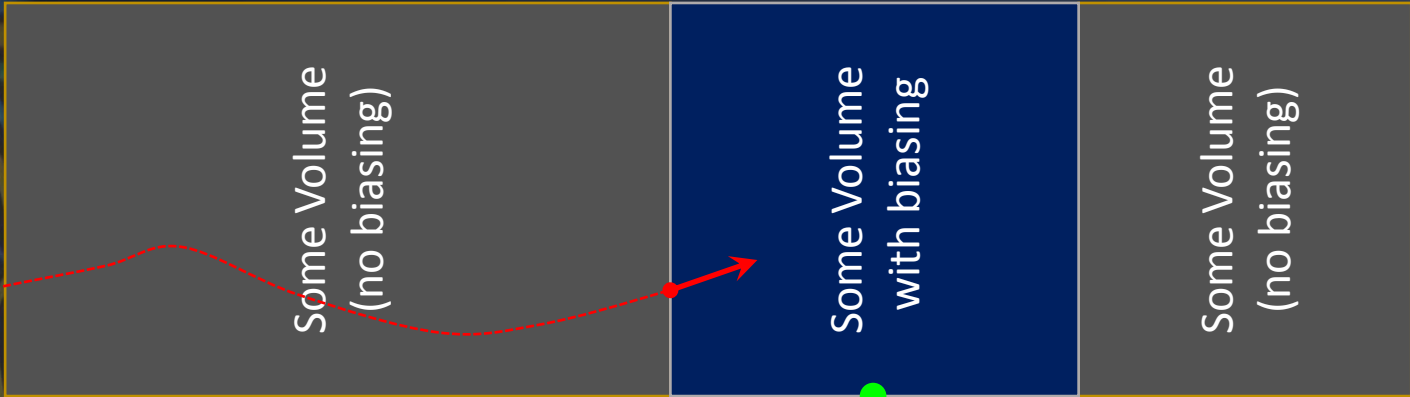
G4ExponentialForce-InteractionOperation

G4ForceFreeFlight-InteractionOperation

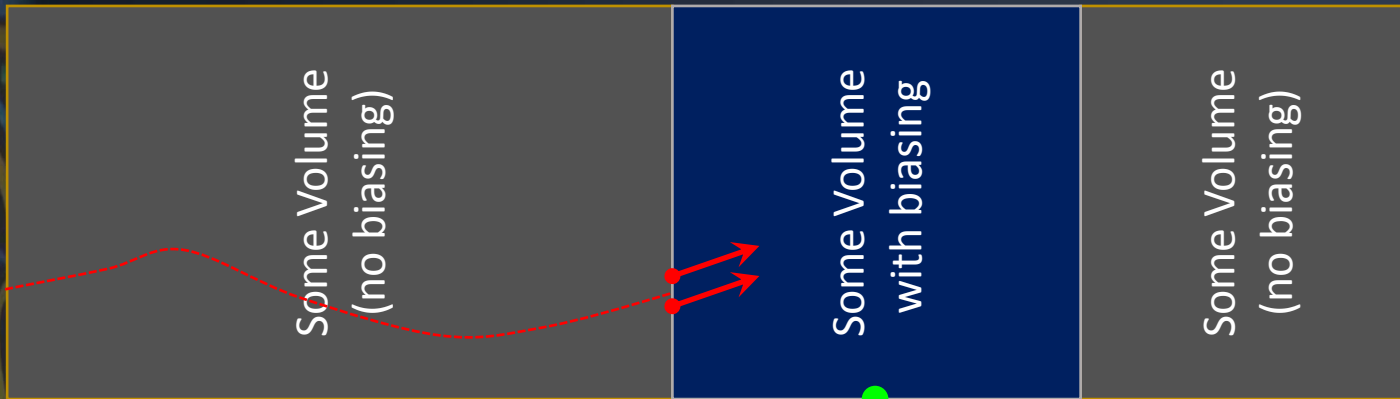
G4SplittingOperation

G4WeightWindow-Operation

G4MCNPForce-CollisionOperator



- G4FlatForce-InteractionOperation
- G4ExponentialForce-InteractionOperation
- G4ForceFreeFlight-InteractionOperation
- G4SplittingOperation
- G4WeightWindow-Operation
- G4MCNPForce-CollisionOperator



G4BiasingProcessInterface  
(no process)

G4BiasingProcessInterface  
G4PhotoelectricEffect

G4BiasingProcessInterface  
G4GammaConversion

G4BiasingProcessInterface  
G4ComptonScattering

MyBiasingOperator

G4FlatForce-InteractionOperation

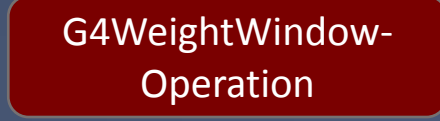
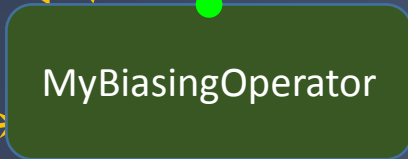
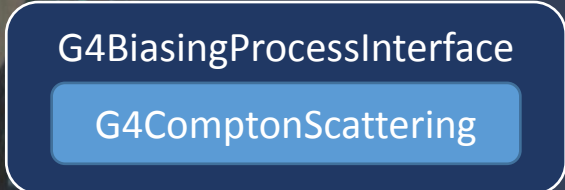
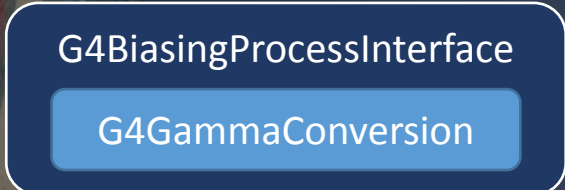
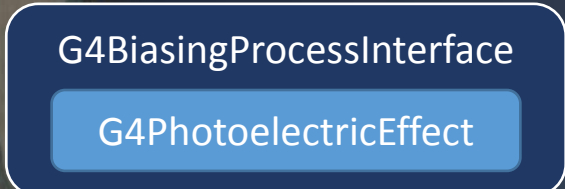
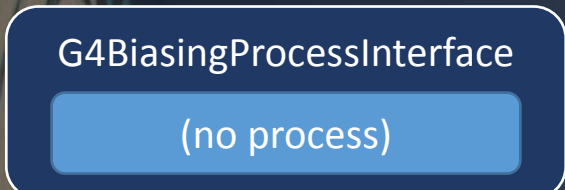
G4ExponentialForce-InteractionOperation

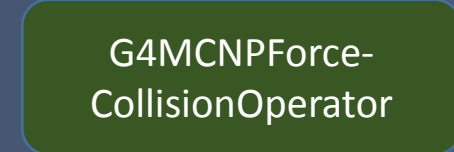
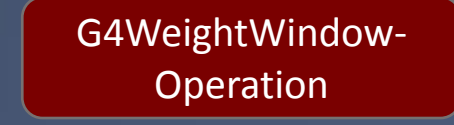
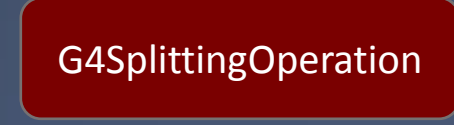
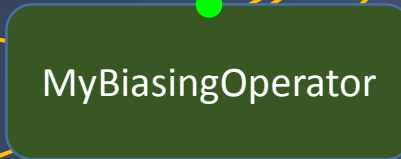
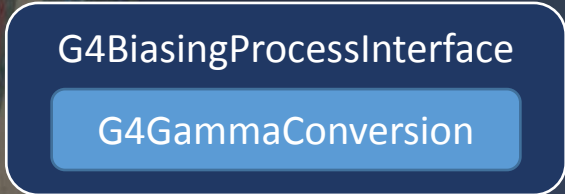
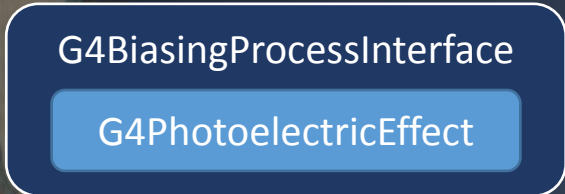
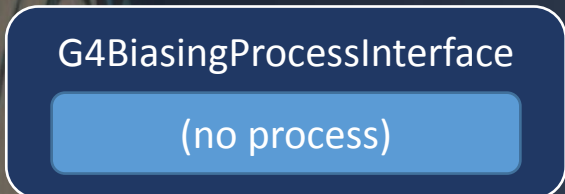
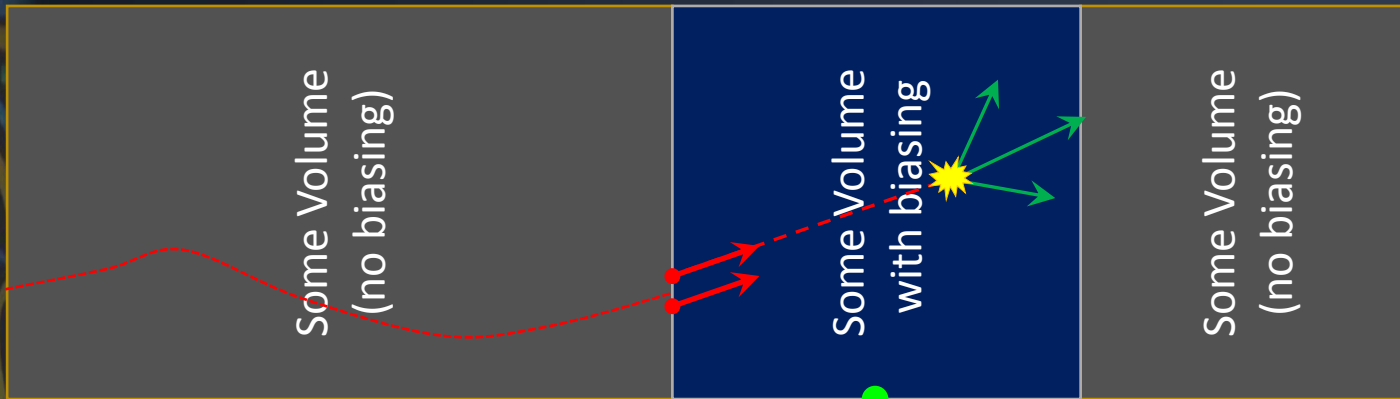
G4ForceFreeFlight-InteractionOperation

G4SplittingOperation

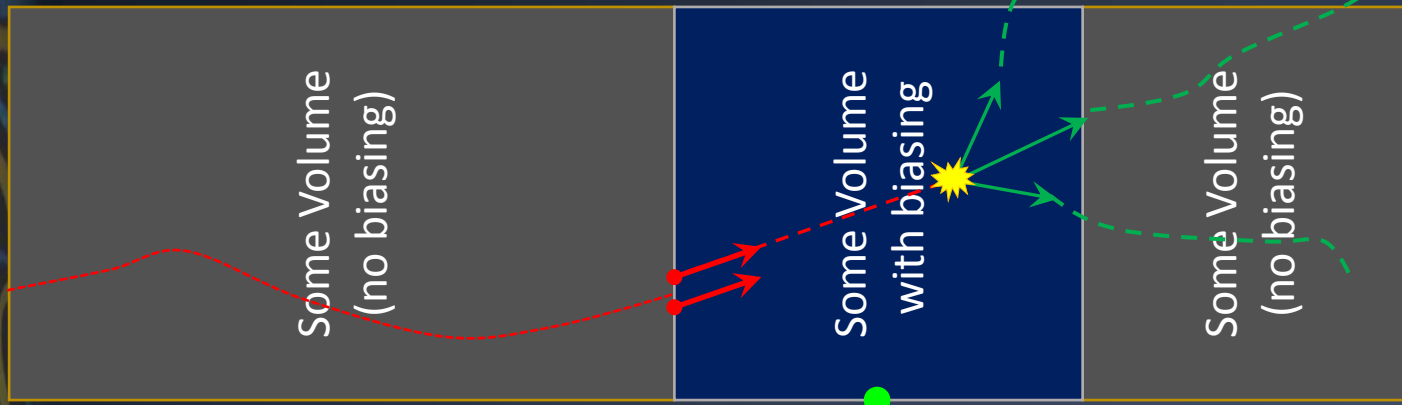
G4WeightWindow-Operation

G4MCNPForce-CollisionOperator





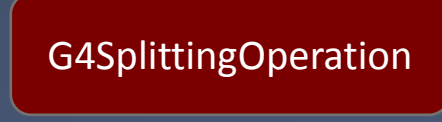
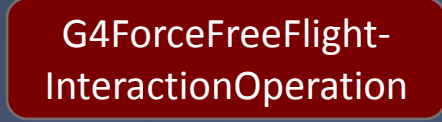
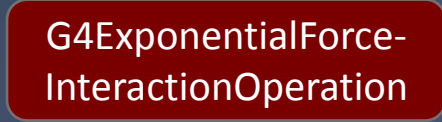
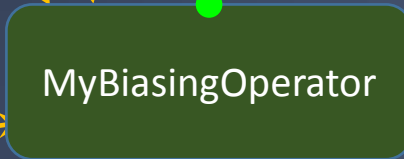
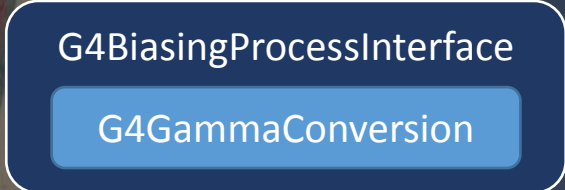
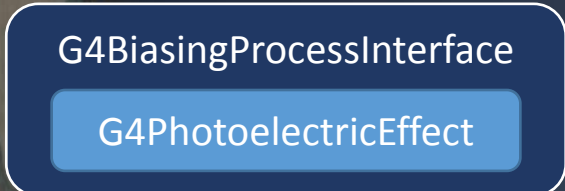
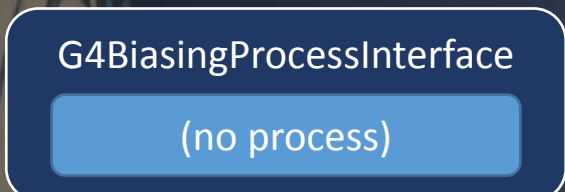
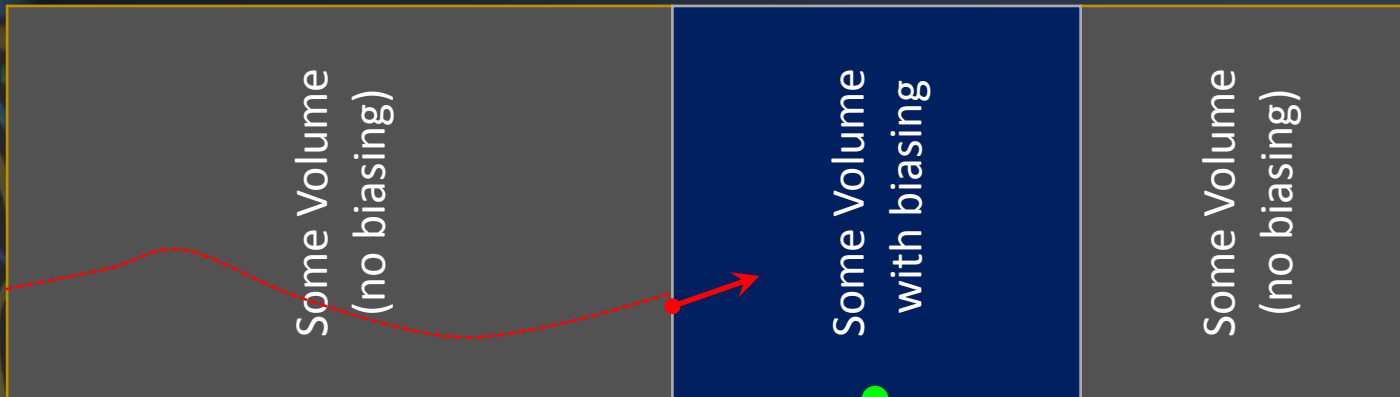


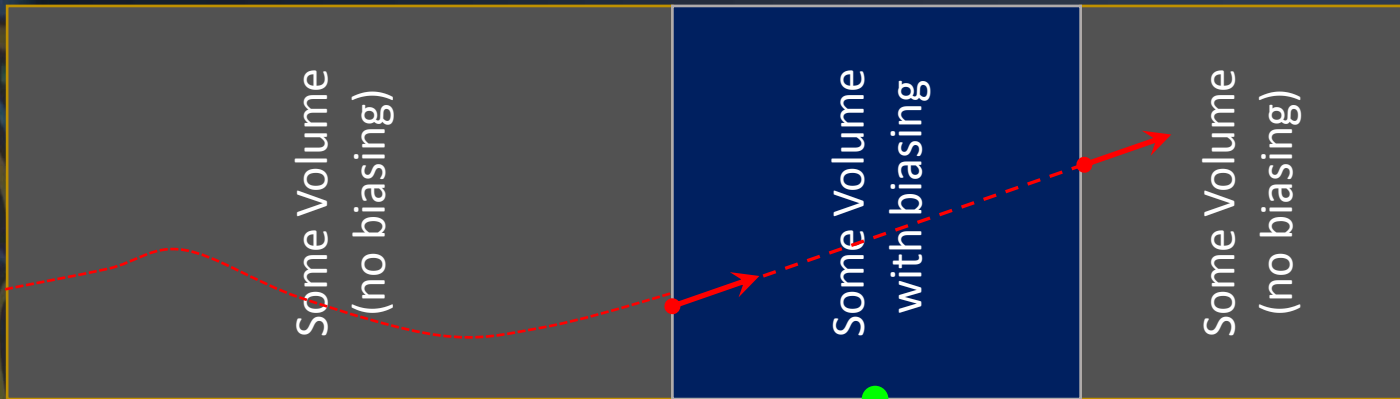


- G4BiasingProcessInterface  
(no process)
- G4BiasingProcessInterface  
G4PhotoelectricEffect
- G4BiasingProcessInterface  
G4GammaConversion
- G4BiasingProcessInterface  
G4ComptonScattering

MyBiasingOperator

- G4FlatForce-InteractionOperation
- G4ExponentialForce-InteractionOperation
- G4ForceFreeFlight-InteractionOperation
- G4SplittingOperation
- G4WeightWindow-Operation
- G4MCNPForce-CollisionOperator





G4BiasingProcessInterface  
(no process)

G4BiasingProcessInterface  
G4PhotoelectricEffect

G4BiasingProcessInterface  
G4GammaConversion

G4BiasingProcessInterface  
G4ComptonScattering

MyBiasingOperator

G4FlatForce-InteractionOperation

G4ExponentialForce-InteractionOperation

G4ForceFreeFlight-InteractionOperation

G4SplittingOperation

G4WeightWindow-Operation

G4MCNPForce-CollisionOperator

# How to use

## > In the main:

```
// -- Select a modular physics list
auto physicsList = new FTFP_BERT;
// -- And augment it with biasing facilities:
auto biasingPhysics = new G4GenericBiasingPhysics();
biasingPhysics->Bias("gamma");
biasingPhysics->Bias("neutron");
physicsList->RegisterPhysics(biasingPhysics);
runManager->SetUserInitialization(physicsList);
```

### – More options exist for the G4GenericBiasingPhysics constructor:

- > Activate biasing for processes only
- > Or for doing splitting/killing only
- > Or to activate biasing for all charged
- > Or all neutral particles
- > Etc.

## > In ConstructSDandField() of detector construction:

```
auto biasingOperator = new MyBiasingOperator();
biasingOperator->AttachTo( logicalVolumeToBias );
```

# Existing functionalities & examples

- › `geant4/examples/extended/biasing/GB01` :
  - Individual process cross-section biasing
  - Implemented for neutral particles
  - Charged particle case requires development
    - › Issue : variation of cross-section during step because of energy loss
- › `geant4/examples/extended/biasing/GB02` :
  - Force collision à la MCNP
  - Implemented for neutral particles (as MCNP)
- › `geant4/examples/extended/biasing/GB03` :
  - Geometry importance + further option
  - Scheme augmented compared to classical geometry importance
    - › Allows kinds of “intermediate” non-integer splitting values
- › `geant4/examples/extended/biasing/GB04` :
  - Re-implementation of a classical Bremsstrahlung splitting
- › `geant4/examples/extended/biasing/GB05` :
  - Illustrates a “splitting by cross-section”
  - An invention (to my knowledge) where the splitting rate is directly adjusted from absorption cross-sections
- › `geant4/examples/extended/biasing/GB06` :
  - Parallel geometries with generic biasing.
- › `geant4/examples/extended/biasing/GB07` :
  - Implement a “particle leading” biasing scheme