

Hadronic Developer Parameters

KOI, Tatsumi

Dotti, Andrea

Wright, H. Denssi

SLAC National Accelerator Laboratory

Yarba, Julia

Fermi National Accelerator Laboratory

22nd Geant4 collaboration workshop 2017-09-28



What is G4HadronicDeveloperParameters

A utility class for handling hadronic developer parameters

- Should be simple but also provide reasonable functionalities to users
- Set, Get, **DeveloperGet**, SetDefault, GetDefault and Dump
- Allow to set lower and upper limits of a parameter in SetDefault

Hadronic developer parameter is a parameter that a model developer allows changing its value to specific experienced users (testers)

- Model developer gives name of the parameter to users.
- A user changes value of the parameter and checks the impact of the modification in his/her result.
- The user must know the name of parameter.
 - For avoiding abuse usage the functionalities by novice users

Type of parameter can be double, integer or boolean.

Target user of HadronicDeveloperParameters

An experienced user who want to see an impact of model parameter in his/her result.

However, he/her has no interested in actual implementation of the model.

Users (or developers) who have a knowledge of implementation of model are not primary target

- They can easily wrote a code for their purpose

Model developers may use the HadronicDeveloperParameters on top of their management system of parameters

- Easily separate open and close parameter by using the HadronicDeveloperParameters

What need to do by model developer

Determine parameters to be opened

- Consider proper name of parameter
- Tell the name to users (testers)

Must set default values of the parameters

- SetDefault
- Enable to register the lower and upper limits to the parameter

Incorporate values of parameters through DeveloperGet method in the initialization of model

- To be enable to issue a warning message, if the value is modified

What need to do by user (tester)

To get the names of parameters

- HadronicDeveloperParameter does not provide full list of changeable parameters

Set a new value for the parameter

- **Should do this very early in his/her code.**
 - It must be set before initialization of model
- **Changes only allow once**
 - We will discuss this limitation later

Can check default value of parameter

Can dump default and new value of a parameter with the name of parameter

- Dump require a parameter name in argument.

Run the simulation of interest and see the impact of changing parameter

only allow to change parameter values once.

- Incorporation timing of parameter values into model is not well defined in hadronic framework.
- Minimize the risk that change of the parameter is not reflected in the calculation.

dump always requires name of parameter

- To prevent abuse usage by beginners, we do not want to provide a full list of developer parameters for users.

What is “DeveloperGet”

This is the method that a model developer uses for incorporation of parameter values during the initialization of model.

If the value is different from default, then warning message is issued.

- If user changed a value but not having this warning message, then the new value is not used in the simulation by some reasons. Most likely it was too late to set the parameter.
- This helps users to avoid mistake.

Difference between “DeveloperGet” and “Get”

- The later will not issue the warning messages.
- The use case of the later is that a user want to know current parameter value by some reason, for example print out for log.

G4HadronicDeveloperParameters

source/processes/hadronic/util

```
class G4HadronicDeveloperParameters
{
public:
    static G4HadronicDeveloperParameters& GetInstance();
private:
    G4HadronicDeveloperParameters();

public:
    G4bool Set( const std::string name , const G4double );
    G4bool GetDefault( const std::string name , G4double& value );
    G4bool Get( const std::string name , G4double& value );
    G4bool SetDefault( const std::string name , const G4double value , G4double lower_limit = -
DBL_MAX , G4double upper_limit = DBL_MAX );
    G4bool DeveloperGet( const std::string name , G4double& value );
    void Dump( const std::string name );
''''
}
```


Developer side implementation

G4CascadeParameters

```
#include "G4HadronicDeveloperParameters.hh"
#define OLD_RADIUS_UNITS (3.3836/1.2)    // Used with NucModel params
namespace {
  G4HadronicDeveloperParameters& HDP = G4HadronicDeveloperParameters::GetInstance();
  class BERTParameters {
  public:
    BERTParameters(){
      // Define default values
      //HDP.SetDefault("NAME",VALUE,LOWER_LIMIT(default=-DBL_MAX),UPPER_LIMIT(default=DBL_MAX));
      HDP.SetDefault( "BERT_RADIUS_SCALE" , OLD_RADIUS_UNITS );
      HDP.SetDefault( "BERT_XSEC_SCALE" , 1.0 , 0. );
    }
  };
  BERTParameters BP;
}

void G4CascadeParameters::Initialize() {
  ""
  //RADIUS_SCALE = (G4NUCMODEL_RAD_SCALE ? strtod(G4NUCMODEL_RAD_SCALE,0)
  //      : (BEST_PAR?1.0:OLD_RADIUS_UNITS));
  HDP.DeveloperGet("BERT_RADIUS_SCALE",RADIUS_SCALE);
  ""
  //XSEC_SCALE = (G4NUCMODEL_XSEC_SCALE ? strtod(G4NUCMODEL_XSEC_SCALE,0)
  //      : (BEST_PAR?0.1:1.0) );
  HDP.DeveloperGet("BERT_XSEC_SCALE",XSEC_SCALE);
  ""
}
```

User(tester) side implementation

User main (Hadr01)

```
#include "G4HadronicDeveloperParameters.hh"
int main(,,,,) {
//followings should be implemented very beginning of main
G4HadronicDeveloperParameters& HDP =
G4HadronicDeveloperParameters::GetInstance();
    HDP.Set("BERT_RADIUS_SCALE",1.0);
    //HDP.Set("BERT_RADIUS_SCALE",1.0); //cause error
    HDP.Set("BERT_XSEC_SCALE",2.0);
    HDP.Dump("BERT_RADIUS_SCALE");
    HDP.Dump("BERT_XSEC_SCALE");

'''
}
```

In run time, you should have two warning messages
for BERT_RADIUS_SCALE and BERT_XSEC_SCALE

Restriction that parameters can be changed only once in a job

Is this limitation too much strict?

- User may want to change multiple times in a job
- We recommend to separate jobs for each changes.
 - Is there any fundamental reason why it can not be divided into multiple jobs?

Again, hadronic framework does not well define incorporation timing of parameters into simulation (model).

- We cannot guarantee that new values are properly incorporated and used in following calculations.
- Some models may handle them properly but others may not

G4HadronicDeveloperParameters

- Included in reference tags

G4CascadeParameters

- Included in reference tags
- Activation key of “TEST_HDP” is required for use now.
 - Can we delete this activation key and set this as default?

Only “double” and “Boolean” type parameters are tested through BERT-like cascade model and high energy string models

- “integer” type parameter is not yet tested

G4HadronicDeveloperParameters is developed for helping management of model parameters that is allowed changing by experienced users

It is designed as simple as possible but also provide reasonable functionalities for developer and user

In the development, many efforts are made for avoiding improper usage and mistakes

We encourage to use G4HadronicDeveloperParameter for management and sharing such parameters, but this does NOT mean forcing developer to use it.