Subject: FairMCEventHeader
Posted by Volker Friese on Mon, 11 May 2009 15:46:30 GMT
View Forum Message <> Reply to Message

The fairbase defines the class FairMCEventHeader, storing information on the event level, which is to be filled by the respective generator classes.

Now, it may not be good an idea to define such a data class in the base directories since the requirements (which information to be stored) may vary from experiment to experiment. For instance, we now want to store an additional variable (the event plane angle).

Is it possible to have a derived, experiment-specific class? As far as I can see, the class is instantiated (and the branch registered) in CbmMCApplication and otherwise used only by FairPrimaryGenerator. So, I think it would be easy to shift the instantiation and filling to the experiment specific part of the software.

A possible solution would be:

Implement an experiment-specific data class, e.g. CbmMCEventHeader (TNamed)
Introduce a virtual method Init() in CbmPrimaryGenerator, to be called from FairMCApplication::Init(). Here, the event class array has to be instantiated and its branch registered.
Make CbmPrimaryGenerator::GenerateEvent virtual, so the proper filling of the event class can be done in an experiment-specific class derived from FairPrimaryGenerator (e.g. CbmPrimaryGenerator) and/or by the concrete generator classes themselves.

Subject: Requirements
Posted by Volker Friese on Tue, 12 May 2009 00:12:29 GMT
View Forum Message <> Reply to Message

OK, I implemented that and it works. What is required from the framework is:

1. Introduce a method virtual void Init() { }; in FairPrimaryGenerator.h

2. Declare the private members of FairPrimaryGenerator protected (in fact, this is needed only for fStack).

3. Declare the method FairPrimaryGenerator::GenerateEvent virtual.

4. Add if(fEvGen)fEvGen->Init(); somewhere in FairMCApplication::Init

Subject: Re: Requirements
Posted by Mohammad Al-Turany on Wed, 20 May 2009 14:23:28 GMT
View Forum Message <> Reply to Message

Hi,
It is done and seems to work! please test it and let me know.

Mohammad

Subject: Re: Requirements
Posted by Mikhail Ryzhinskiy on Thu, 21 May 2009 06:18:08 GMT
View Forum Message <> Reply to Message

It seems that #include <stdlib.h> is missing in FairGeoLoader.cxx.
Otherwise it is not compiled.

Subject: Re: Requirements
Posted by Florian Uhlig on Thu, 21 May 2009 13:56:33 GMT
View Forum Message <> Reply to Message

Hi Mikhail

Thanks for the info. I added the missing library to the includes.
This error seems to occur only for new versions of gcc (4.3?).
Which version of gcc and linux do you use?

Ciao

Florian

Subject: Re: Requirements
Posted by Mikhail Ryzhinskiy on Thu, 21 May 2009 17:04:02 GMT
View Forum Message <> Reply to Message

Hello Florian
I use gcc version 4.3.2 on OpenSUSE 11.1
Regards,
Mikhail.

Subject: FairPrimaryGenerator::Init()
Posted by Volker Friese on Mon, 25 May 2009 15:49:38 GMT
View Forum Message <> Reply to Message

Dear Mohammad,

thanks for applying the changes. Only one detail: You call fGen->Init() in FairMCApplication::InitMC. This seems to have no effect on the output tree; it is possibly too late.

If you put this call to the method FairMCApplication::InitGeometry(), it works fine. There, the possibility to register branches to the cbmsim tree still exists.

Subject: Re: FairPrimaryGenerator::Init()
Posted by Mohammad Al-Turany on Mon, 25 May 2009 19:19:12 GMT
View Forum Message <> Reply to Message

Hi Volker,

I moved it to InitGeometry just before registering the branches and event headers, it should work now!

Mohammad

Subject: Thanks
Posted by Volker Friese on Mon, 25 May 2009 21:36:26 GMT
View Forum Message <> Reply to Message

Thanks a lot, Mohammad, it is fine now!