Subject: Compiling a box generator (Geant3)
Posted by Prometeusz Jasinski on Thu, 29 Nov 2012 08:07:47 GMT

I wanted to share and document modifications needed to compile a box generator with Luminosity detector, beam pipe, Dipole and Solenoid.

To make it clear: no increase in speed was observed, as expected. But it might be helpful some time when macros might not work anymore due to changes in root.

Apart from the more or less trivial CMakelists entries (will be uploaded soon to macro/lmd/Promme) I encountered root related problems.

When running a macro, root environment is already loaded.

If you compile it, static constructors of some root objects crash during initialization. When I found first evil root calls and I opened a ticket

https://subversion.gsi.de/trac/fairroot/ticket/95#comment:3 [/link]

But there was more.

FairRuntimeDb contains a static TList contFactories. I made it a member of FairRuntimeDb. Therefore further changes were needed:

FairBasecontFact
PndPassiveContFact
PndFieldContFact
PndLmdContFact
PndSensorNameContFact

all contained a global variable (behaving as it would be static) gFairBasecontFact and so on. I changed all of them to singleton classes:

in the cxx files


```
PndSensorNameContFact* PndSensorNameContFact::pPndSensorNameContFact = NULL;

//static PndSensorNameContFact gPndSensorNameContFact;


PndSensorNameContFact& PndSensorNameContFact::gPndSensorNameContFact(){
 if (!pPndSensorNameContFact) {
  pPndSensorNameContFact = new PndSensorNameContFact();
 }
 return *pPndSensorNameContFact;
}
```

and in the header files

```
class PndSensorNameContFact : public FairContFact {
public:
...
  static PndSensorNameContFact& gPndSensorNameContFact();
private:
  static PndSensorNameContFact* pPndSensorNameContFact;
...
};
```

Finally one had to call the instance prior to run any task with

```
  PndSensorNameContFact& sensornamecontfact =
PndSensorNameContFact::gPndSensorNameContFact();
  FairBaseContFact& basecontfact = FairBaseContFact::gFairBaseContFact();
  PndFieldContFact& fieldcontfact = PndFieldContFact::gPndFieldContFact();
  PndPassiveContFact& passivecontfact = PndPassiveContFact::gPndPassiveContFact();
```

in the main code.

I will try to make an executable without those modifications but calling for gRoot in a static way like

TROOT& _root = *gROOT; or similar to force root creating an environment.

And in addition I would like to mention that Geant4 VMC does not work since there is a macro being executed not compatible with a compiled version. It gives the error:

```
Error: Symbol cout is not defined in current scope
/home/jasinski/bin/fairroot/share/geant4_vmc/macro/g4libs.C:104:
Error: << Illegal operator for pointer 3
/home/jasinski/bin/fairroot/share/geant4_vmc/macro/g4libs.C:104:
Error: Symbol endl is not defined in current scope
/home/jasinski/bin/fairroot/share/geant4_vmc/macro/g4libs.C:104:
*** Interpreter error recovered ***
Error: Symbol TG4RunConfiguration is not defined in current scope
/home/jasinski/bin/pandaroot/gconfig/g4Config.C:29:
Error: Symbol TG4RunConfiguration is not defined in current scope
/home/jasinski/bin/pandaroot/gconfig/g4Config.C:29:
Error: type TG4RunConfiguration not defined
FILE:/home/jasinski/bin/pandaroot/gconfig/g4Config.C LINE:29
Error: Invalid type 'TG4RunConfiguration*' in declaration of 'runConfiguration'
/home/jasinski/bin/pandaroot/gconfig/g4Config.C:29:
*** Interpreter error recovered ***
Error: Symbol cout is not defined in current scope
/home/jasinski/bin/pandaroot/gconfig/SetCuts.C:9:
```

Error: << Illegal operator for pointer 3 /home/jasinski/bin/pandaroot/gconfig/SetCuts.C:9:
Error: Symbol endl is not defined in current scope
/home/jasinski/bin/pandaroot/gconfig/SetCuts.C:9:
*** Interpreter error recovered ***

 *** Break *** segmentation violation

I will keep you up to date within this post at least for my self.