Subject: [FIXED] LmdFit does not compile on MAC
Posted by StefanoSpataro on Thu, 02 May 2013 10:36:50 GMT
View Forum Message <> Reply to Message

Dear all,
I noticed that the package LmdFit does not compile on my mac:

```
/.../trunk/lmd/LmdFit/ModelPar.h:28:17: error: no member named 'shared_ptr' in namespace
'std'

  std::set<std::shared_ptr<ModelPar>> connections;
           ~~~~~^
```

This is due mainly to the usage of shared_ptr... it seems that under linux it is inside std::shared_ptr, while under mac it sits on std::tr1::shared_ptr. If I modify the call adding tr1 then it works on mac, but not on linux.

Another strange thing I found is that in ModelParSet the following function is private:

```
class ModelParSet {
  private:
    /**
     * Small structure defining the comparison operator used in the map
     * #model_par_map. Will return true only if this model parameter is equal to
     * the parameter. It is used when concatenating multiple ModelPar objects to
     * avoid having multiple instances of the same parameter. The check is based
     * on a name comparison.
     */
    struct stringcomp {
        bool operator()(const std::string& lhs, const std::string& rhs) const {
          return lhs.compare(rhs) < 0;
        }
    };
```

but it is used in ModelParameterHandler.cxx. In this case one has to move the function to public in order to compile.

Another problem under MAC:

```
[ 57%] Building CXX object lmd/CMakeFiles/Model.dir/LmdFit/Model.cxx.o

cc1plus: error: unrecognized command line option "-std=c++0x"
```

Is is really needed the flag c++0x? All the other packages do not have it. What's the reason?

Is it possible to fix this package so that it compiles also under Mac?

However, it seems the package does not follow the pandaroot coding rules. If this is only for group anayses maybe it would be better to put it under development so that the main compilation is not affected.

Subject: Re: LmdFit does not compile on MAC
Posted by Stefan Pflueger on Thu, 02 May 2013 14:39:22 GMT
View Forum Message <> Reply to Message

Hi Stefano,

sry for this... basically everything you describe is related to the fact that the mac compiler does not support c++11 standard yet and I wanted to use that standard as shared_ptr are in the std namespace (ok for the gnu compiler I needed the . But I will switch everything to use tr1 then it should compile just fine. If not I will jump back to the boost namespace.
Regarding the private/public problem: this compiles just fine for me, because I'm not actually using it just laying out the structure for the iterator.

I'll let you know when I commited the new version (prob in a few hours as I'm working on a rework aswell).

Best regards,

Stefan

Subject: Re: LmdFit does not compile on MAC
Posted by Florian Uhlig on Thu, 02 May 2013 14:47:24 GMT
View Forum Message <> Reply to Message

Hi Stefano,

shared_ptr is a new functionality of the C++11 standard. So if you want to use it you must tell the compiler to use the new standard.

The real problem is that not all compilers support already the new standard. To make the mess even bigger depending on the compiler and compiler version different features of the new standard are supported. So you can't say easily the gnu compiler collection supports C++11, but you have to check which features are already supported for the used compiler version. The Clang compiler team announced last week that they now fully support all new features.

The only way to use C++11 features I see in the moment is to force the users to use a defined (by the Panda computing coordinator) compiler and compiler version.

Ciao

Florian

Subject: Re: LmdFit does not compile on MAC
Posted by Stefan Pflueger on Thu, 02 May 2013 16:27:58 GMT
View Forum Message <> Reply to Message

Hi again,

ok I temporarily took out the lmdfit libary from the pandaroot building process. I will have to finish the work tomorrow and let you know.

Best regards,

Stefan

Subject: Re: LmdFit does not compile on MAC
Posted by StefanoSpataro on Thu, 02 May 2013 16:32:33 GMT

View Forum Message <> Reply to Message

Many thanks,
maybe one IFDEF could help to understand if the correct code is present, check done by cmake, and in such a case to switch between diffrent "using std...".
Not sure if this could help.

Subject: Re: LmdFit does not compile on MAC
Posted by Stefan Pflueger on Fri, 03 May 2013 17:05:24 GMT

View Forum Message <> Reply to Message

Hi,

ok the compiling problems with shared_ptr etc should be resolved (i only tested it on my linux machine atm, and will test it on my macbook later). Right now its using the tr1/memory namespace. if more problems occur due to this problem i could move things to the boost namespace which is included in the external packages anyways, but would be an unnecessary requirement.

Best regards,

Stefan

Subject: Re: LmdFit does not compile on MAC
Posted by Ralf Kliemt on Mon, 06 May 2013 07:16:29 GMT

View Forum Message <> Reply to Message

Hi Stefan,

I have a bad feeling about using C++11 in Panda. It will force people to upgrade systems with a compatible compiler. It was always a statement that we can compile on "the laptop", "the batch farm" and "the old terminal". To give an example: At GSI(!) the default system does not support C++11.

Would there be a neat way around C++11?

Greetings.
Ralf

Subject: Re: LmdFit does not compile on MAC
Posted by Stefan Pflueger on Mon, 06 May 2013 07:39:10 GMT
View Forum Message <> Reply to Message

Hi Ralf,

at the moment I'm using things from the tr1, which I believe is not c++11, but is only supported by "newer" compilers (gcc 4.1 or later if I'm not mistaken). I guess since Fairroot already has the dependency of the boost libraries, the best way for the largest compatibility for older compilers/systems would be to use the boost shared_ptrs.
The reason for using c++11 was just that I started to develop my own modeling/fitting framework for arbitrary functions and wanted as few dependencies as possible, hence the shared_ptrs from std.
Ultimately I would like to have this modeling framework to be an external link to the svn (like genfit). I could use a boost version here and then develop a separate version in parallel, but that's kind of work overhead to keep these things synced.

Best regards,

Stefan