Subject: Problem with loading TVector3 in R3BRoot
Posted by C. A. Douma on Tue, 30 Oct 2018 14:27:39 GMT
View Forum Message <> Reply to Message

Dear all,

I am trying to build an R3BRoot class with a TVector3 member, but when I try to use the class in
a macro, I am getting the following error:

cling::DynamicLibraryManager::loadLibrary():
/home/christiaan/Desktop/FAIR/R3BRoot_Build/lib/libTest.so.SOVERSION: undefined symbol:
_ZTV8TVector3
Error in <TInterpreter::TCling::AutoLoad>: failure loading library Test.so for R3BTest

I have produced a minimal working example. The included .zip-file contains a directory Test, which
should be placed in the R3BRoot source directory. Subsequently, a command
add_subdirectory(Test)
should be added to the master CMakeLists.txt near line 370 and then R3BRoot should be re-build.

Afterwards, running the macro TestClass.cpp in R3BRoot/Test/Macros/ should produce the error.
However, When I comment out lines 7 and 13 in R3BTest.cxx in R3BRoot/Test/SubDir/ the error is gone.

Can anyone explain to me what caused the error and how I can fix this?

Thanks in advance!
Christiaan Douma.

File Attachments
1) Test.zip, downloaded 544 times

Subject: Re: Problem with loading TVector3 in R3BRoot
Posted by Dmytro Kresan on Wed, 31 Oct 2018 08:28:10 GMT
View Forum Message <> Reply to Message

TVector3 class is defined in libPhysics library of ROOT. You need to add "Physics" to the DEPENDENCIES variable inside of CMakeLists.txt.

Best regards,
Dima

Subject: Re: Problem with loading TVector3 in R3BRoot
Posted by Jan Mayer on Wed, 31 Oct 2018 09:21:35 GMT

I would also like to add the following:

You should not use a pointer for a member that the class owns (= is responsible for), with few exceptions.

Here, R3BTest class can simply own it naturally, like the TString one line above. Just think simple. No need to initialize and delete it, nothing can go wrong, the code is much shorter and easier to understand.

If the class needs to own a polymorphic class, there are unique and shared ptrs.
A class may include pointer members to stuff it is not responsible for.

See also https://isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines#S-resource