
Subject: FTF generator

Posted by [Aida Galoyan](#) on Wed, 04 Mar 2015 15:27:37 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Ralf,

why do you replace files with ".c" extension
(Test30Materials.c, Test30HadronProduction.c and so on)
on the files *.cxx .
After your changing, FTFmake is not working. It crashes.

If I return *.c, FTFmake works and FTFGen exe-file is created.

Aida

Subject: Re: FTF generator

Posted by [Ralf Kliemt](#) on Wed, 04 Mar 2015 16:09:26 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Aida.

The issue is, that with the *.c extension our CMake based compilation fails. When modifying the standard_FORTRAN.mk to read the *.cxx files, it will try to compile PndFtfDirect, which is not desirable as all settings from CMake are not available in the standalone version.

What I did now (please update): There is the executable <build>/bin/FTFGen in the standard CMake compilation.

Another solution will be to manually specify the relevant .cxx files in the standard_FORTRAN.mk. I'm not sure how to do that correctly.

Kind regards.

Ralf

Subject: Re: FTF generator

Posted by [Aida Galoyan](#) on Wed, 04 Mar 2015 16:52:08 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Ralf,

CMake based compilation can work with file.cc .
So, I assume, if you replace PndFtfDirect.cxx on PndFtfDirect.cc
CMAke will compile.

In this case, I will perform corresponding changes (*.c on *.cxx) in *.mk files,
to work FTFmake.

I think, it is solution for work of both: standalone FTFmake and FTFdirect.

Aida

Subject: Re: FTF generator

Posted by [Stefano Spataro](#) on Wed, 04 Mar 2015 18:47:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

The last change break the build, please fix it.

In any case PndFtfDirect should have the priority over the compiled version. If the bin does not compile with .cxx it means that the makefile needs some changed, most probably by substituting .c with .cxx.

Subject: Re: FTF generator

Posted by [Ralf Kliemt](#) on Thu, 05 Mar 2015 10:09:08 GMT

[View Forum Message](#) <> [Reply to Message](#)

Stefano Spataro wrote on Wed, 04 March 2015 19:47: The last change break the build, please fix it.

It's OK for me - both old and new fairsoft. Freshly checked out.

Stefano Spataro wrote on Wed, 04 March 2015 19:47: If the bin does not compile with .cxx it means that the makefile needs some changed, most probably by substituting .c with .cxx. Exactly, however there is the issue that it then tries to compile also FtfDirect (also a.cxx). I'll play around with positioning FtfDirect.

Ralf

Subject: Re: FTF generator

Posted by [Aida Galoyan](#) on Thu, 05 Mar 2015 10:26:36 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Ralf and Stefano,

I don't understand, why PndFtfDirect.cxx is directly implemented in /FtfEvtGen directory.

I created this directory for standalone FTF calculations.

For example, earlier, PndDpmDirect.cxx is implemented in /pgenerators directory.

How I understand, when we run FTFmake, PndFtfDirect.cxx is conflicted with other files in the directory, for example, with main.cc.

I think, the solution will be following.

If you move PndFtfDirect.cxx to /pgenerators directory, similar PndDpmDirect.cxx, I will perform corresponding changes: to remove *.c on *.cxx in needed makefiles.

If you have better solution, I will be glad.
In any case, it is needed to have normal working FTFmake.

Aida

Subject: Re: FTF generator
Posted by [Stefano Spataro](#) on Thu, 05 Mar 2015 11:26:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

This could make sense, and since it worked for dpm it should work also for ftf.

Subject: Re: FTF generator
Posted by [Ralf Kliemt](#) on Thu, 05 Mar 2015 12:19:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello.

The binary (FTFGen) is automatically compiled by Pandaroot and is installed in your build/bin directory, next to DPMGen.

I am able to put FtfDirect into the PGen library (now that I sorted out some lib issues).

However, I have trouble to compile with the standalone makefile. The Problem is that the include files are not found. Which environment setup should one have (e.g. Geant4?) to compile Ftf?

Lastly: Is it really necessary to have the standalone compilation? I would agree for a package `_outside_` of our svn repository.

Cheers!
Ralf

Subject: Re: FTF generator
Posted by [Ralf Kliemt](#) on Thu, 05 Mar 2015 13:32:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Aida,

it should be OK for you now.

Ralf

Subject: Re: FTF generator
Posted by [Stefano Spataro](#) on Thu, 05 Mar 2015 15:52:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

From the dashboard the build is still crashing.

Subject: Re: FTF generator

Posted by [Stefano Spataro](#) on Fri, 06 Mar 2015 14:05:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

Now maybe it is fixed in Linux, but in Mac I can see the following:

Linking CXX executable ../../bin/FTFGen

Undefined symbols for architecture x86_64:

"G4Evaporation::SetGEMChannel()", referenced from:
 _main in main.cc.o
"G4Evaporation::SetDefaultChannel()", referenced from:
 _main in main.cc.o
"G4Evaporation::SetCombinedChannel()", referenced from:
 _main in main.cc.o
"G4Evaporation::G4Evaporation()", referenced from:
 _main in main.cc.o
"G4ProductionCuts::G4ProductionCuts()", referenced from:
 _main in main.cc.o
"G4BGGPionElasticXS::G4BGGPionElasticXS(G4ParticleDefinition const*)", referenced from:
 _main in main.cc.o
"G4ChipsComponentXS::G4ChipsComponentXS()", referenced from:
 _main in main.cc.o
"G4PreCompoundModel::UseNGB()", referenced from:
 _main in main.cc.o
"G4PreCompoundModel::UseSCO()", referenced from:
 _main in main.cc.o
"G4PreCompoundModel::UseSICB()", referenced from:
 _main in main.cc.o
"G4PreCompoundModel::SetOPTxs(int)", referenced from:
 _main in main.cc.o
"G4PreCompoundModel::UseCEMtr()", referenced from:
 _main in main.cc.o
"G4PreCompoundModel::G4PreCompoundModel(G4ExcitationHandler*)", referenced from:
 _main in main.cc.o
"G4ExcitationHandler::SetParameters()", referenced from:
 G4ExcitationHandler::SetOPTxs(int) in main.cc.o
 G4ExcitationHandler::UseSICB() in main.cc.o
"G4ExcitationHandler::SetEvaporation(G4VEvaporation*)", referenced from:
 _main in main.cc.o
"G4ExcitationHandler::SetMinEForMultiFrag(double)", referenced from:
 _main in main.cc.o
"G4ExcitationHandler::SetMaxAandZForFermiBreakUp(int, int)", referenced from:
 _main in main.cc.o
"G4ExcitationHandler::G4ExcitationHandler()", referenced from:
 _main in main.cc.o
"G4BGGPionInelasticXS::G4BGGPionInelasticXS(G4ParticleDefinition const*)", referenced
from:
 _main in main.cc.o

```

"G4MaterialCutsCouple::G4MaterialCutsCouple(G4Material const*, G4ProductionCuts*)",
referenced from:
    _main in main.cc.o
"G4BGGNucleonElasticXS::G4BGGNucleonElasticXS(G4ParticleDefinition const*)",
referenced from:
    _main in main.cc.o
"G4HadronCrossSections::GetInelasticCrossSection(G4DynamicParticle const*, int, int)",
referenced from:
    _main in main.cc.o
"G4HadronCrossSections::Instance()", referenced from:
    _main in main.cc.o
"G4ProductionCutsTable::GetProductionCutsTable()", referenced from:
    _main in main.cc.o
"G4HadronElasticDataSet::G4HadronElasticDataSet(G4String const&)", referenced from:
    _main in main.cc.o
"G4IonsShenCrossSection::G4IonsShenCrossSection()", referenced from:
    _main in main.cc.o
"G4TripathiCrossSection::G4TripathiCrossSection()", referenced from:
    _main in main.cc.o
"G4VCrossSectionDataSet::ComputeCrossSection(G4DynamicParticle const*, G4Element
const*, G4Material const*)", referenced from:
    G4VCrossSectionDataSet::GetCrossSection(G4DynamicParticle const*, G4Element
const*, G4Material const*) in main.cc.o
"G4BGGNucleonInelasticXS::G4BGGNucleonInelasticXS(G4ParticleDefinition const*)",
referenced from:
    _main in main.cc.o
"G4PiNuclearCrossSection::G4PiNuclearCrossSection()", referenced from:
    _main in main.cc.o
"G4HadronInelasticDataSet::G4HadronInelasticDataSet(G4String const&)", referenced from:
    _main in main.cc.o
"G4TripathiLightCrossSection::G4TripathiLightCrossSection()", referenced from:
    _main in main.cc.o
"G4ComponentAntiNuclNuclearXS::G4ComponentAntiNuclNuclearXS()", referenced from:
    _main in main.cc.o
"G4ProtonInelasticCrossSection::G4ProtonInelasticCrossSection()", referenced from:
    _main in main.cc.o
"G4NeutronInelasticCrossSection::G4NeutronInelasticCrossSection()", referenced from:
    _main in main.cc.o
ld: symbol(s) not found for architecture x86_64
clang: error: linker command failed with exit code 1 (use -v to see invocation)
make[2]: *** [bin/FTFGen] Error 1
make[1]: *** [pgenerators/FtfEvtGen/CMakeFiles/FtfEvtGen_exe.dir/all] Error 2
make: *** [all] Error 2

```

There is still something missing, since before (27139) it was compiling.

Subject: Re: FTF generator
 Posted by [Ralf Kliemt](#) on Mon, 09 Mar 2015 10:22:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Stefano,

I have no access to a mac with a running pandaroot. My first guess is missing lib dependencies, so I added some in Pgen. Maybe, stefano, you could provide me a sandbox account to look at the issue?

Cheers
Ralf

Subject: Re: FTF generator
Posted by [Aida Galoyan](#) on Mon, 09 Mar 2015 15:52:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Stefano,

In GSI Linux, FTF works in new version of PandaRoot.
Just now, I have installed and checked.

In Mac, to overcome errors
please, add in /FtfEvtGen/common.mk
in line
LDLIBS += \$(PANDALIBS) -L\$(GEANT4_LIBRARY_DIR) -IG4clhep -IG4global
-IG4geometry -IG4physicslists \$(GLIBS)

-IG4processes

before \$(GLIBS)

Good luck
Aida

Subject: Re: FTF generator
Posted by [Stefano Spataro](#) on Tue, 10 Mar 2015 01:18:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,
a G4processes was missing in the CMakeLists.txt, in target_link_libraries for "Darwin". Now it is fixed.
