

GROMACS - Task #2855

Allow compiling GROMACS without C compiler

01/31/2019 01:53 AM - Roland Schulz

Status:	New
Priority:	Normal
Assignee:	
Category:	
Target version:	
Difficulty:	uncategorized
Description	
Primary reason: Would simplify cmake.	
Required subtask: Make TNG compile with C++ compiler. Requires explicit cast for all malloc calls and renaming of "private" variable (2nd is simple sed). Maybe a few other things. Adding cast could be automatized but I can't find existing tooling. Might be pretty easy to create.	
Alternative: Remove the included TNG version and require external TNG or no-TNG.	
Option: Modify cmake before TNG issue is resolved. Don't require C compiler if the user chooses no or external TNG.	

History

#1 - 01/31/2019 05:09 PM - Szilárd Páll

What about GMX_BUILD_OWN_FFTW, fftw requires a C compiler AFAIK?

#2 - 01/31/2019 08:44 PM - Roland Schulz

(copying some discussion from <https://gerrit.gromacs.org/#/c/9058/> to have everything here)
fftw is build using ExternalProject. It has its own configure and looks for its own compiler.

If we want to keep the option to build TNG as part of the GROMACS and keep TNG as C and remove C from GROMACS we have two options:
- Either use ExternalProject for TNG (same as fftw). Note that ExternalProject support a local path/file. So the decision of whether we want to include the TNG source in the GROMACS source (rather than downloading it on demand as with fftw) is independent of the decision of how we build it.
- Or enable the C language as part of the TNG build (e.g. in gm ManageTNG.cmake). That way it is still part of the same cmake process but the c compiler is only needed if internal TNG is chosen by the user. With this approach it is a bit unclear where to set the required cflags. If we remove C from GROMACS we'll want to remove setting of c flags from gmxCFlags.cmake. It'll either has to be set in gm ManageTNG.cmake or in src/external/tng_io somewhere.

Szilard has pointed out: If TNG is to be kept portable, it's perhaps best to keep it C?

My reply: I assume we'll want to keep a C interface so it is usable by non C++ code. But that doesn't require the code to be C++. Do we need more than that?

#3 - 01/10/2020 02:32 AM - Roland Schulz

Any updates what strategy we should use?

#4 - 01/10/2020 12:04 PM - Eric Irrgang

ExternalProject for TNG makes sense to quickly decouple it from the current issue. Are there any down sides? Making TNG build with C++ while providing a C API seems reasonable, but does not need to block this issue.

#5 - 01/10/2020 12:06 PM - Magnus Lundborg

I agree that building TNG using ExternalProject seems to be a reasonable solution. I haven't looked into what CMake changes would need to be made, though.