

GROMACS - Feature #2846

Set rpath in FindLibStdC.cpp.cmake

01/28/2019 08:50 PM - Roland Schulz

Status:	New
Priority:	Low
Assignee:	
Category:	
Target version:	
Difficulty:	uncategorized

Description

If rpath is enabled the rpath should be set for libstdc++ the same way it is set for other libraries.

Possible solutions:

- manually add rpath to linker flags (similar as we currently do in releng)
- use `target_link_libraries` to add it as library so that cmake automatically does the right thing. Does it matter that the library is added by both cmake and the compiler?
- only add it with `target_link_libraries` (using e.g. `-nodefaultlibs` as compiler flag)

History

#1 - 02/01/2019 02:28 PM - Mark Abraham

Roland Schulz wrote:

If rpath is enabled the rpath should be set for libstdc++ the same way it is set for other libraries.

Preferably.

Possible solutions:

- manually add rpath to linker flags (similar as we currently do in releng)

Undesirable for user installations (e.g. see <https://wiki.debian.org/RpathIssue>), but sufficient for our CI needs.

- use `target_link_libraries` to add it as library so that cmake automatically does the right thing. Does it matter that the library is added by both cmake and the compiler?

I think this is best. It should work fine, because currently we set a path with `-L` to find libstdc++ and the compiler is OK with that. If there's a problem, it would only be an unsuitable ordering of the linking commands.

- only add it with `target_link_libraries` (using e.g. `-nodefaultlibs` as compiler flag)

If necessary.

#2 - 02/01/2019 06:24 PM - Roland Schulz

Another option might be to set `CMAKE_CXX_IMPLICIT_LINK_DIRECTORIES` or `CMAKE_PLATFORM_IMPLICIT_LINK_DIRECTORIES`. Haven't tested yet. Just read about it and leaving a comment here to not forget about this option.

#3 - 02/02/2019 08:09 AM - Roland Schulz

For me with cmake 3.13.3 `CMAKE_INSTALL_RPATH_USE_LINK_PATH` doesn't work at all. Maybe I'm missing something. But it doesn't have any effect for any library I'm testing. Does it work for other or do we have something in our gromacs cmake files which prevent it from working? Or is it broken in 3.13.3?

We could add it to `CMAKE_INSTALL_RPATH` (and if we want `CMAKE_BUILD_RPATH`) that way the rpath is still not set if the user sets `CMAKE_SKIP_RPATH`. That addresses the Debian issue, right? Do we prefer the automatic way through `CMAKE_INSTALL_RPATH_USE_LINK_PATH` (if we can get it to work) or does it not matter?