

GROMACS - Bug #2722

gmxapi may over-manage RPATH

10/31/2018 02:43 PM - Mark Abraham

Status: Closed	
Priority: Normal	
Assignee: Mark Abraham	
Category: build system	
Target version: 2020-rc1	
Affected version - extra info:	Difficulty: uncategorized
Affected version: 2019-beta1	
Description	
The RPATH settings we have for libgmxapi look either like defaults or things that are artefacts of developing outside the main GROMACS project. We should simplify these, or document what they are intended to achieve.	
Context: https://gitlab.kitware.com/cmake/community/wikis/doc/cmake/RPATH-handling	

Associated revisions

Revision 3fd14b5f - 12/10/2018 05:07 PM - Paul Bauer

Disable gmxapi by default

Due to outstanding issues with the integration testing and tests failing with large number of ranks, the gmxapi default has been changed to not be build. In Jenkins, all supported builds still are still set to build with GMXAPI enabled.

Refs #2765, #2722, #2756

Change-Id: I2cc42c461edc206aaa30be6cac3db0a52ccae991

Revision 18227c07 - 12/10/2019 02:41 PM - Mark Abraham

Simplify gmxapi RPATH handling

Fixes #2722

Change-Id: I697f596285acdb5c7728b7ebc995ea44c82f3ac

Revision 2041753f - 12/14/2019 01:36 PM - Mark Abraham

Simplify gmxapi RPATH handling

Fixes #2722

Change-Id: I697f596285acdb5c7728b7ebc995ea44c82f3ac

History

#1 - 10/31/2018 02:44 PM - Mark Abraham

In particular, when building with clang+cuda+ninja, I get

```
CMake Error at src/api/cpp/CMakeLists.txt:53 (add_library):
  The install of the gmxapi target requires changing an RPATH from the build
  tree, but this is not supported with the Ninja generator unless on an
  ELF-based platform.  The CMAKE_BUILD_WITH_INSTALL_RPATH variable may be set
  to avoid this relinking step.
```

which I can get rid of if I comment out

```
set_target_properties(gmxapi PROPERTIES BUILD_WITH_INSTALL_RPATH FALSE)
```

so unless the later has a clear use, then we should probably drop it.

#2 - 10/31/2018 02:45 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#2722](#).
Uploader: Mark Abraham (mark.j.abraham@gmail.com)
Change-Id: gromacs~release-2019~lc697f596285acdb5c7728b7ebc995ea44c82f3ac
Gerrit URL: <https://gerrit.gromacs.org/8632>

#3 - 11/01/2018 09:53 AM - Eric Irrgang

I had never encountered a problem with changing RPATH between build and install directory before. Good to know.

As long as the RPATH is relative and the relationship between libgmxapi and libgromacs is consistent, removing that line should be fine.

#4 - 11/16/2018 01:50 PM - Eric Irrgang

- Blocked by Task #2756: gmxapi integration testing added

#5 - 12/10/2018 04:03 PM - Gerrit Code Review Bot

Gerrit received a related DRAFT patchset '1' for Issue [#2722](#).
Uploader: Paul Bauer (paul.bauer.q@gmail.com)
Change-Id: gromacs~release-2019~l2cc42c461edc206aaa30be6cac3db0a52ccae991
Gerrit URL: <https://gerrit.gromacs.org/8801>

#6 - 12/17/2018 01:30 PM - Paul Bauer

More likely to happen in 2020

#7 - 12/17/2018 01:31 PM - Paul Bauer

- Target version changed from 2019 to 2020

#8 - 10/16/2019 10:28 AM - Eric Irrgang

- Target version changed from 2020 to 2020-beta3

We should be able to resolve this with less effort when we move to the new testing infrastructure, but the appropriate test coverage is pretty broad.

Note that some of these CMake config options predate the current CMake policy behaviors and GROMACS choices, but originally they could affect both how libgmxapi finds libgromacs and how libgmxapi clients find libgmxapi (or how the libgmxapi ability to find libgromacs could be affected by how libgmxapi is loaded by a client.)

I believe that the behavior we want is that (a) libgmxapi is built so that libgromacs is at a fixed location relative to itself at build through to install, that (b) the installed libgmxapi (and imported CMake target) provide the necessary information for client code to resolve the libgmxapi location via the client RPATH (or equivalent facility). I don't **think** we need to support relocatability of the libgmxapi installation, but if we do, it doesn't seem like much to ask clients linked against the first location to have to relink against the new location. We do need to allow client code to be relocatable without relinking because of how various packaging systems work, which has slight constraints on how the library identifies itself and what RPATH hinting is provided through CMake.

Note that, currently, we do not produce an installable binary package of client software (such as the Python gmxapi package or the sample_restraint MD extension) in the GROMACS build tree. If we did, we would need to consider that the package would have to be built in an environment where its ultimate link targets are not yet available.

Matrix dimensions

Builder

- make
- ninja

Platform

- Linux (multiple binary formats?)
- Mac OS X
- Windows? (please no)

libgmxapi state

Assume libgromacs state == libgmxapi state

- in build tree (BUILD_INTERFACE and library target)
- installed (INSTALLED_INTERFACE and imported target)

Client state

- `_gmxapi` target in build tree (e.g. with `gmxapi_pytest` target)
- Python `gmxapi` package building and linking against GROMACS installation in packaging sandbox.
- Python `gmxapi` package relocated during package installation.
- `sample_restraint` target tested in GROMACS build tree.
- `sample_restraint` package installed in user environment.

#9 - 12/02/2019 09:48 AM - Paul Bauer

- Target version changed from 2020-beta3 to 2020-rc1

bump!

#10 - 12/04/2019 11:32 AM - Mark Abraham

We haven't ever designed for supporting multiple binary/linking formats (e.g. ELF, DWARF) and definitely not mixing them, so the only thing we should consider now is testing with the toolchain provided by the user and that means we have to assume it's capable of correct behavior and issuing a message when it can't.

Since Windows has never been tested with GMXAPI, we should give an error from `src/CMakeLists.txt` if someone tries it (not just default off).

Otherwise, Erik's matrix looks fine. We need scripts that can cover those cases, whether we run them manually or automated. Once we have them, then it's easier to see whether things in `src/api/cpp/CMakeLists.txt` are necessary or not.

#11 - 12/12/2019 07:30 PM - Mark Abraham

- Status changed from *New* to *Resolved*

Applied in changeset [18227c07e94c7f2e0e00faaf0fc224b0fa98eae7](#).

#12 - 12/14/2019 01:45 PM - Mark Abraham

Applied in changeset [2041753fe7d1d3325bbc8eba39179758cc4a4952](#).

#13 - 12/20/2019 08:22 AM - Paul Bauer

- Blocked by deleted (Task #2756: *gmxapi* integration testing)

#14 - 12/20/2019 08:22 AM - Paul Bauer

- Status changed from *Resolved* to *Closed*