GMXRC install is broken in master

05/31/2013 11:51 PM - Mark Abraham

**Teemu's c92076c0f438aa0fb1caba549f21a7c65f11b2 changed the behaviour of cmake .. -DCMAKE_INSTALL_PREFIX=/whatever/you/need such that GMX_INSTALL_PREFIX defaults to an empty string. GMXRC.cmakein relies on GMX_INSTALL_PREFIX being set to an appropriate path, but this no longer occurs. I've no idea what is required for the relocatable binaries people have been trying to facilitate. Anyway, with current master branch, you cannot source GMXRC because "bin/GMXRC.bash" is useless because it uses that empty string.

Setting cmake -DGMX_INSTALL_PREFIX=/whatever/you/need/ works (but you need the final slash), but this seems inconsistent with the description of this variable in the cache string.

How do we want this to work?

Associated revisions

Revision 0b234494 - 10/15/2013 04:59 PM - Teemu Murtola
Fix/improve installation directory logic.

With the move to relative install paths (recommended by CMake), logic that relied on the _INSTALL_DIR variables for something else than locations for the CMake install() command got broken. Explicitly added CMAKE_INSTALL_PREFIX there.

Also improve the approach for customizing the installation directories:
- GMXLIB is now GMX_LIB_INSTALL_DIR, and a proper cache variable.
- Added a GMX_DATA_INSTALL_DIR to customize the directory under which the data files get installed under share/. Use this also in the code that searches for the data directory instead of hardcoding several guesses.

Related, moved the logic of falling back to a hard-coded library directory into get_libdir(); it was duplicated in two places.

Resolves part of #1271: creating a binary package with CPack and installing it somewhere else than in the CMAKE_INSTALL_PREFIX used to create it still breaks all the logic touched here. Also using DESTDIR with 'make install' breaks it (but that has never worked).

Change-Id: I0271a8152f87dd59a229c9d0eca976404f974ea8

History

#1 - 06/01/2013 12:41 AM - Mark Abraham
- Description updated

#2 - 06/01/2013 05:14 AM - Teemu Murtola
If you want it to work like it used to, you can simply add CMAKE_INSTALL_PREFIX to the scripts before the various _INSTALL_DIR variables in the scripts. This also needs to be prefixed to GMXLIBDIR in the main CMakeFiles.txt. But both of these get broken in multiple ways if you use either DESTDIR together with "make install", or create a CPack binary archive and extract it, e.g., under your home directory. While fixing the first could be possible by creating the GMXRC scripts during "make install" and using the real installation path there, having static paths in the GMXRC file (or...
hardcoded into the binary, like with GMXLIBDIR) cannot work when extracted from a tarball into an arbitrary directory. Best would be if the scripts would deduce the base directory from the directory they are in, somewhat like what the executables do.

#3 - 08/11/2013 02:14 PM - Teemu Murtola
- Status changed from New to Accepted

https://gerrit.gromacs.org/#/c/2548/ restores functionality to how it was before while still keeping the relative install paths. But it does not solve the problem for relocatable binaries (for which it has never worked).

#4 - 08/12/2013 08:14 PM - Teemu Murtola
The discussion in https://gerrit.gromacs.org/#/c/2548/ seems to lead towards what is really necessary in GMXRC. Since that has an effect on how to best fix this issue for relocatable binaries (if those need to be supported), moving the general discussion here. Here's a scenario that should contain most of the problematic cases, while still being somehow feasible to actually appear with users.

Let's assume a user has the following:

- System-wide Gromacs installation, e.g., in /usr/local/ (somewhere in the default search path).
- Local Gromacs installation (different version) in home directory /home/user/gromacs/.
- Self-written tool, with build directory in /home/user/source/tool/ (assume this is built using the CMakeLists.txt.template that we provide).
- After building the tool, they've copied the binary to /home/user/bin/g_tool, which is in their PATH.

Now, when the user executes g_tool, how does it work? Can it work without a GMXRC? There are basically three issues:

1. How does the build system for the custom tool find Gromacs? This is probably a minor issue, and it is always possible to cater for such a scenario in the template CMakeLists.txt. But the pkg-config file, as it is now, does not work. Does pkg-config support specifying relative paths in the pkg-config files?

2. How does the g_tool executable find the correct libraries? Does CMake add the correct RPATH there automatically, or do we need to add some logic in CMakeLists.txt to do that? Or do we need LD_LIBRARY_PATH/equivalent for this?

3. How does libgromacs loaded by g_tool find its data files? With the current logic, it will actually use the system-wide share/gromacs/ directory, which may or may not work correctly, depending on how different the versions are.

#5 - 10/06/2013 05:40 PM - Mark Abraham
I'm unsure of the answers to any of the above. I gather relocatable binaries are a useful thing for supporting package creation, but don't know anything further. Hopefully we will hear more from package maintainers in the beta phase.

In the meantime, it is unacceptable that "cmake, make, install, source GMXRC, mdrun" does not work (or worse, picks up some mdrun from another path). I think https://gerrit.gromacs.org/#/c/2548/1 makes that standard workflow work. If there's other problems for less common use cases, we'll have to solve them later.

#6 - 01/01/2014 07:03 AM - Teemu Murtola
- Status changed from Accepted to Resolved
- Assignee changed from Mark Abraham to Teemu Murtola
- Affected version - extra info set to only 5.0 development versions

The issue reported in the original issue description was fixed by the linked commit. The issue of hardcoded paths not working for relocatable binaries is a different issue, and no one seems particularly interested in solving that.

#7 - 01/24/2014 07:16 PM - Teemu Murtola
- Status changed from Resolved to Closed