GROMACS - Bug #1147

gromacs 4.6 builds shared libraries with the wrong compatibility version on darwin

02/12/2013 03:27 PM - Jack Howarth

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Christoph Junghans</td>
</tr>
<tr>
<td>Category:</td>
<td>build system</td>
</tr>
<tr>
<td>Target version:</td>
<td>4.6.1</td>
</tr>
<tr>
<td>Affected version - extra info:</td>
<td>4.6</td>
</tr>
<tr>
<td>Affected version:</td>
<td>4.6</td>
</tr>
<tr>
<td>Difficulty:</td>
<td>uncategorized</td>
</tr>
</tbody>
</table>

Description
Gromacs 4.6 builds its shared libraries with the soversion of 6 but incorrectly uses the compatibility of 6 as well. The previous 4.5.6 release, which also used the soversion of 6, properly set the compatibility version to 7.

From gromacs 4.6...

1. otool -L libgmx.6.dylib
   libgmx.6.dylib:
   /sw/lib/libgmx.6.dylib (compatibility version 6.0.0, current version 0.0.0)
   /System/Library/Frameworks/Accelerate.framework/Versions/A/Accelerate (compatibility version 1.0.0, current version 4.0.0)
   /sw/lib/libfftw3f.3.dylib (compatibility version 7.0.0, current version 7.2.0)
   /usr/lib/libSystem.B.dylib (compatibility version 1.0.0, current version 125.2.11)
   /sw/lib/gcc4.7/lib/libgcc_s.1.dylib (compatibility version 1.0.0, current version 1.0.0)

Associated revisions
Revision 78bd4a2 - 02/15/2013 11:23 PM - Mark Abraham
Bump shared object version to 7

Almost certainly the ABI is binary incompatible for the 4.5.5 -> 4.5.6 transition, but we didn't think of handling the issue at the time.
So we can do so now.

Refs #1147
Change-Id: l15f978547d7bccc9a72454067825c71c0ff459a

Revision cdaf7223 - 02/20/2013 11:34 AM - Christoph Junghans
Bump shared object version to 8

libs are not compatible with the ones from Gromacs 4.5.x, which had so version 6 through 4.5.6, and now so version 7 for 4.5.7.

Fixes #1147
Change-Id: l196fd044b0a99eb4ba8376f8a0a2ecbea37894a

Revision 33bd761 - 06/13/2013 09:45 AM - Christoph Junghans
Bump shared object version to 8

libs are not compatible with the ones from Gromacs 4.5.x, which had so version 6 through 4.5.6, and now so version 7 for 4.5.7.

Fixes #1147
Change-Id: l196fd044b0a99eb4ba8376f8a0a2ecbea37894a

Revision 7ee98fad - 12/29/2013 02:36 AM - Teemu Murtola
Reset SOVERSION for libgromacs

Since 5.0 will be the first version with the library name libgromacs, there is no reason to start the numbering from 8. Also set the VERSION for completeness.

Related to #1147

Change-Id: ld048adcb85cf833bc947ac73e8d7bbdb5d600815

History

#1 - 02/12/2013 05:22 PM - Jack Howarth
The problem is that gromacs isn't setting the VERSION property.

#2 - 02/13/2013 03:00 PM - Jack Howarth
File gromacs.patch added

Okay, I see now that the compatibility version is valid but deviates from the libtool convention. Cmake has adopted using the explicit major, minor, patch numbering for the compatibility version whereas libtool uses just major+1. There still is one deviation from the configure build of gromacs that should be addressed. The VERSION should be set so that lib*.6.0.0.dylib files are created for the shared libraries as with gromacs 4.5.6. The attached patch implements that change.

#3 - 02/13/2013 03:03 PM - Jack Howarth
Were there any changes to the shared libraries that merited a patch level change? That is should gromacs 4.6 have had...

set(VERSION 6.0.1)
in the top level CMakeLists.txt?

#4 - 02/15/2013 12:35 AM - Szilárd Páll
Target version changed from 4.6 to 4.6.1

#5 - 02/15/2013 01:02 AM - Christoph Junghans
I think, we just forgot to bump the soversion, Gromacs 4.6 was supposed to be lib*.so.7 (and lib*.dylib.7). Setting the version to 6.0.1 is not a solution as libgmx.so.6.0.1 will get symlinked to libgmx.so.6 and libgmx.so.6 (from Gromacs 4.5) is not compatible with libgmx.so.6 (from Gromacs 4.6).

#6 - 02/15/2013 07:17 PM - Roland Schulz
Status changed from New to In Progress
Assignee set to Christoph Junghans

Suggested solution for 4.6: https://gerrit.gromacs.org/#/c/2170

AFAIK 4.5.5 and 4.5.6 is also not compatible and I forgot to upgrade the soversion there too. Should it be for 4.5.7 6.1 or should it be 7 and thus 8 for 4.6.1? What about master? It is currently also 6. The main library has a new name but it still also has libgmxana. Should we in the future always update the SO version as soon as necessary and not just right before a release?

#7 - 02/15/2013 07:29 PM - Mark Abraham
What drives the need for an SO change?

#8 - 02/15/2013 08:09 PM - Christoph Junghans
The main problem is that ld.so will load whatever shared object with the right name it finds first. ldd/otool will show the details. It is only an issue on system where we cannot use LD_LIBRARY_PATH, which we set in GMXRC.

master is not an issue as the library got rename and ld.so is looking for libgromacs.

From the release notes of 4.5.6 I don't see anything, which is immediately obvious to break the API and as 4.5.X will receive less attention in the future anyhow, I would risk it to not change the so number.

In VOTCA we fix that issue by replacing LD_LIBRARY_PATH by DYLD_LIBRARY_PATH on darwin and LIBPATH on AIX, see <https://code.google.com/p/votca/source/browse/scripts/CMakeLists.txt?repo=tools>
Are there any other packages, outside of gromacs, which build against the gromacs shared libraries? If so, you probably should pay attention to ABI changes. In that case, if you are only adding new calls to the shared lib, a minor or patch number bump would suffice. However if you are changing the ABI such that binaries built under the old shared libraries will break when those are run against the new shared libraries, this would require a bump in the major number. Also, are the gromacs shared libraries and ABI used outside of gromacs or are those really considered to be private ABIs for gromacs alone? If they private, perhaps gromacs really shouldn't be installing include/gromacs in 'make install'.

VOTCA uses libgmx, but we got used to the ABI breaking in 4.5.x and have an internal workaround.

Roland Schulz wrote:

Suggested solution for 4.6: https://gerrit.gromacs.org/#/c/2170

AFAIK 4.5.5 and 4.5.6 is also not compatible and I forgot to upgrade the soversion there too. Should it be for 4.5.7 6.1 or should it be 7 and thus 8 for 4.6.1? What about master? It is currently also 6. The main library has a new name but it still also has libgmxana. Should we in the future always update the SO version as soon as necessary and not just right before a release?

I'd be amazed if there was ABI compatibility for the 4.5.5->4.5.6 transition, but I'm pretty sure it's not worth taking much trouble over either. There are not going to be many installs of 4.5[67], given that 4.6 available, so worrying about 4.5.x SO compatibility isn't really worth much to anyone. But an SO version bump is very cheap... let's bump SOVERSION for 4.5.7, 4.6.1 and 5.0 (so 7, 8 and 9, respectively)

I'm not aware of any other projects that call the GROMACS ABI. We are trying to play nicely with people who might want to do so, but there's a lot of things we'd have to do to achieve that properly. Chicken-and-egg problem, of course. Most "derivative" works are forks, plugins or include our source. That said, one of our most useful ABI features are our low-level kernels, which changed radically for GROMACS 4.6, so a major bump is definitely in order there. Unfortunately, we didn't think of it until Jack brought it up, but we can do it for the patch-level 4.6.1 release due out Monday.

Given our plans, 5.0 will obviously require another SOVERSION bump.

Mark Abraham wrote:

Given our plans, 5.0 will obviously require another SOVERSION bump.

No, as the library name has changed that is not necessary.
For Gromacs 5.0 with should start with libgromacs.so.0 (SOVERSION = 0) if want to do anything at all.

Christoph Junghans wrote:

Given our plans, 5.0 will obviously require another SOVERSION bump.

No, as the library name has changed that is not necessary.
For Gromacs 5.0 with should start with libgromacs.so.0 (SOVERSION = 0) if want to do anything at all.

Fair enough, that sounds better.

Does 4.6.1 resolve this suitably, Jack?
Why aren't we setting VERSION to 8.0.0 as well? Your current approach is unconventional in that it doesn't use versioning to cope with the situation where new calls are added to the shared libraries but the existing calls don't change their ABI. If you had adopted the patch I proposed, gromacs 4.6.1 would have created shared libraries named lib*.8.0.0.dylib with install names (as seen with 'otool -L') of lib*.8.dylib. Not doing this limits your flexibility to add new calls without being forced to unnecessarily bump the SOVERSION number of the shared libraries. You really want to avoid doing that, which brings up the question why are we jumping from SOVERSION 6.0 to directly to 8.0?

Updated patch for gromacs 4.6.1 to add proper versioning support to gromacs. Note that gromacs 4.6.2 can remain with VERSION set to 8.0.0 as long as you don't change or add to the shared library ABIs.

As long as

1. it hasn't been defined what constitutes the public API/ABI of Gromacs (the contents of include/gromacs/ are more or less a random combination of headers that are technically necessary in the current code organization), and
2. there are no guidelines or controls for how can the API/ABI change between releases (I doubt that people pay any attention to whether they are changing the ABI when fixing bugs),

I think it is purely academic discussion to consider how to best version the libraries. Because of the above points, it is only a pure coincidence if the ABI (assuming the whole of include/gromacs/) does not break between each and every release, so preparing to bump the major version each time sounds reasonable to me.

I'm happy to incorporate Jack's patch into 4.6.2, not least so that we continue to have the visual reminder to make a decision about it. Possibly 5.0 will have a defined ABI and we can be less heavy-handed. But first we have to start writing something :)

Files
gromacs.patch 3.38 KB 02/13/2013 Jack Howarth