

GROMACS - Bug #598

dlopen - cmake and dependency issues

10/20/2010 09:32 AM - Mark Abraham

Status: Closed	
Priority: Normal	
Assignee: Mark Abraham	
Category: build system	
Target version:	
Affected version - extra info:	Difficulty: uncategorized
Affected version:	
Description cmake has GMX_DLOPEN ON by default, but makes no attempt to find the libraries. If they're absent, the user doesn't see a problem until compile time. If these should be on by default, then we should check for libraries and turn it off if they are not found	
Related issues: Related to GROMACS - Feature #2082: GMX_USE_PLUGINS=ON is incompatible with G... Closed	

Associated revisions

Revision 347896d9 - 01/07/2012 08:03 PM - Mark Abraham

Made use of VMD plugins more robust

Attempt to use VMD trajectory-reading plugins only if plugin loading functionality exists and shared libraries are being used (else dlopen can be unreliable).

Also, for helping to find those plugins, the user can supply the plug-in path in a CMake cache variable. If not, the code can now fall back on the run-time value of the environment variable VMDDIR, or its value at configuration time. Previously only an explicit run-time environment variable or a hard-coded path was available on non-Windows platforms.

Also printed diagnostic text when GROMACS cannot tell if the trajectory being read by a plug-in might have atom velocities.

Fixes #598

Change-Id: I44267e115267772dae2c1d754895f8ff25c98a6a

Revision 347896d9 - 01/07/2012 08:03 PM - Mark Abraham

Made use of VMD plugins more robust

Attempt to use VMD trajectory-reading plugins only if plugin loading functionality exists and shared libraries are being used (else dlopen can be unreliable).

Also, for helping to find those plugins, the user can supply the plug-in path in a CMake cache variable. If not, the code can now fall back on the run-time value of the environment variable VMDDIR, or its value at configuration time. Previously only an explicit run-time environment variable or a hard-coded path was available on non-Windows platforms.

Also printed diagnostic text when GROMACS cannot tell if the trajectory being read by a plug-in might have atom velocities.

Fixes #598

Change-Id: I44267e115267772dae2c1d754895f8ff25c98a6a

Revision ec923a45 - 01/10/2012 12:04 AM - Mark Abraham

Fixed MSVC compiling with plug-in support

Fixes #598

Change-Id: I8fe274a83c4965466f1241d2237ca3e77b93b038

Revision ec923a45 - 01/10/2012 12:04 AM - Mark Abraham

Fixed MSVC compiling with plug-in support

Fixes #598

Change-Id: I8fe274a83c4965466f1241d2237ca3e77b93b038

History

#1 - 10/21/2010 06:28 AM - Mark Abraham

And while we're there, with dlopen disabled we shouldn't be building vmdio.c and vmddlopen.c. There are no dependencies on them (I hacked them out of src/gmxlib/Makefile, and things built fine), and they create some dependencies.

With autotools and --without-dlopen, vmdio.c #included "glob.h", and compilation of that on some IBM pSeries Linux machine broke in a strange manner.

#2 - 10/28/2010 09:03 AM - Mark Abraham

Further, error messages like the one here

<http://lists.gromacs.org/pipermail/gmx-users/2010-October/055324.html>

shouldn't be possible. Users should have to opt-in for external dependencies, and then they have a chance of understanding what's happening when those dependencies can't be found.

#3 - 11/08/2010 04:56 PM - Rossen Apostolov

I disabled GMX_DLOPEN by default now, but we'll need to write a Find*.cmake script, so the bug stays open.

#4 - 11/08/2010 07:01 PM - Roland Schulz

(In reply to comment #3)

I disabled GMX_DLOPEN by default now, but we'll need to write a Find*.cmake script, so the bug stays open.

I don't think changing the default is a good idea in a bug fix release. Especially because the default is only a problem for a extreme small minority. I think CMAKE_DL_LIBS should contain the dl libs.

(In reply to comment #2)

Further, error messages like the one here

<http://lists.gromacs.org/pipermail/gmx-users/2010-October/055324.html>

shouldn't be possible. Users should have to opt-in for external dependencies, and then they have a chance of understanding what's happening when those dependencies can't be found.

n

I don't understand why the user should have to do an additional step. If the user gives a dcd file as input trajectory, GROMACS should try to read it and let the user know when it is not possible. Adding an additional step would make it less user friendly in my opinion.

What could be improved in my opinion is: Make the error message more clear. Let the user know that the file is not supported directly by GROMACS for trajectory reading and thus it tries to open it with external plug-ins. And only then that opening the external plug-ins failed.

#5 - 11/08/2010 07:57 PM - Roland Schulz

committed better error message. Leave as new because CMAKE_DL_LIBS should be used.

#6 - 11/09/2010 04:07 AM - Mark Abraham

(In reply to comment #4)

(In reply to comment #2)

Further, error messages like the one here

<http://lists.gromacs.org/pipermail/gmx-users/2010-October/055324.html>

shouldn't be possible. Users should have to opt-in for external dependencies, and then they have a chance of understanding what's happening when those dependencies can't be found.

n

I don't understand why the user should have to do an additional step. If the user gives a dcd file as input trajectory, GROMACS should try to read it and let the user know when it is not possible. Adding an additional step would make it less user friendly in my opinion.

Upon reflection, the problem is really that GROMACS appears to be using dlopen as a fallback for any file type that it doesn't recognize for the flag (caveat - I haven't looked at the code!). So when the user supplied a .tpr file where an actual trajectory file was required, GROMACS assumed it was something suitable for dlopen, and only then did they run into the secondary problem with dlopen that obscured the real issue.

We need comparable sophistication with matching arguments to file types when using dlopen as we have when we are not using dlopen. Should this be a separate enhancement-request bugzilla?

What could be improved in my opinion is: Make the error message more clear. Let the user know that the file is not supported directly by GROMACS for trajectory reading and thus it tries to open it with external plug-ins. And only then that opening the external plug-ins failed.

That's superior to the old message, but it's still a kludge. There can't be so many file types that GROMACS can't be error-checking for the user even with dlopen.

#7 - 11/09/2010 04:51 AM - Roland Schulz

(In reply to comment [#6](#))

What could be improved in my opinion is: Make the error message more clear. Let the user know that the file is not supported directly by GROMACS for trajectory reading and thus it tries to open it with external plug-ins. And only then that opening the external plug-ins failed.

That's superior to the old message, but it's still a kludge. There can't be so many file types that GROMACS can't be error-checking for the user even with dlopen.

The message I added to the code is: =====

The file format of %s is not a known trajectory format to GROMACS.

Please make sure that the file is a trajectory!

GROMACS will now assume it to be a trajectory and will try to open it using the VMD plug-ins.

This will only work in case the VMD plugins are found and it is a trajectory format supported by VMD. =====

The problem is that the list of supported file types using dlopen is read from the VMD libraries. That of course is only possible after you have opened the library with dlopen.

Adding a list of supported file types is not a very nice solution because that depends on how the user has compiled VMD and what version of VMD. It is thus not a static list.

#8 - 11/09/2010 05:18 AM - Mark Abraham

(In reply to comment [#7](#))

The message I added to the code is: =====

The file format of %s is not a known trajectory format to GROMACS.

Please make sure that the file is a trajectory!

GROMACS will now assume it to be a trajectory and will try to open it using the VMD plug-ins.

This will only work in case the VMD plugins are found and it is a trajectory format supported by VMD. =====

The problem is that the list of supported file types using dlopen is read from the VMD libraries. That of course is only possible after you have opened the library with dlopen.

Adding a list of supported file types is not a very nice solution because that depends on how the user has compiled VMD and what version of VMD. It is thus

not a static list.

Hmm, that situation is uglier than I appreciated. It seems to me that the current situation is the best that we can do.

An ugly solution would be to have FindDlopen.cmake probe for valid trajectory file formats and embed that list in the GROMACS C code somehow... but that requires a rebuild if the VMD or dlopen library version changes.

Ideally, GMXRC could take care of maintaining the LD_RUN_PATH for this and other external dependencies, but that sounds like a maintenance nightmare.

#9 - 01/06/2011 11:47 PM - Mark Abraham

- Assignee deleted (Erik Lindahl)

Rossen Apostolov wrote:

I disabled GMX_DLOPEN by default now, but we'll need to write a Find*.cmake script, so the bug stays open.

Note that [eb8b905f](#) re-enabled GMX_DLOPEN. I'm not sure why.

#10 - 01/07/2011 01:21 AM - Roland Schulz

Mark Abraham wrote:

Rossen Apostolov wrote:

I disabled GMX_DLOPEN by default now, but we'll need to write a Find*.cmake script, so the bug stays open.

Note that [eb8b905f](#) re-enabled GMX_DLOPEN. I'm not sure why.

I explained this above (comment [#4](#)):

(In reply to comment #3)

I don't think changing the default is a good idea in a bug fix release. Especially because the default is only a problem for a extreme small minority.

I think CMAKE_DL_LIBS should contain the dl libs.

Thus I re-enabled it with Rossen's OK.

#11 - 02/23/2011 03:12 AM - Szilárd Páll

- Category changed from mdrun to build system

- Target version deleted (git master)

#12 - 03/10/2012 02:32 AM - Mark Abraham

- Status changed from New to Closed

- Assignee set to Mark Abraham

#13 - 11/28/2016 03:19 PM - Mark Abraham

- Related to Feature #2082: GMX_USE_PLUGINS=ON is incompatible with GMX_PREFER_STATIC_LIBS=ON -- cmake should fail but does not added