GROMACS - Task #701

Add symbol visibility macros

02/10/2011 07:43 PM - Teemu Murtola

Status:	New
Priority:	Normal
Assignee:	
Category:	build system
Target version:	future
Difficulty:	uncategorized

Description

See the attached URL for discussion of what this means. This is also required to build shared libraries on Windows. The task itself is straightforward (although support for other compilers than GCC and Visual Studio will require finding out the correct #ifdefs and visibility declarations for those compilers), but it's not so clear what would be the best place for a header like this. Its location should be given some thought and fixed, because it will be included from more or less every single file in the project, so moving it later means some work.

This task also requires some integration to the build system: the build system should define symbol(s) that indicate whether shared libraries are being used and whether a header is being compiled as part of a shared library or as part of another unit that links against that library. Cmake may already define some of these symbols automatically; it may just be a matter of finding out their names and using them in the header file.

Another part (and much larger) of this task is to actually add these symbols to function and class declarations.

After this is done, we still need to add the compiler flags for GCC to change the default visibility. This may be beneficial to do already earlier such that it is easy to enable it for testing (e.g., by setting a GMX_USE_VISIBILITY CMake variable).

URL: http://gcc.gnu.org/wiki/Visibility

Related issues:

Related to GROMACS - Task #988: Definition of "public API"	**New**	
Related to GROMACS - Task #1013: Library division for tools and generic Groma...	**Closed**	**09/30/2012**
Related to GROMACS - Task #2045: API design and language bindings	**New**	
Related to GROMACS - Feature #2896: Python packaging	**Feedback wanted**	
Related to GROMACS - Task #2912: C++ extension module for Python bindings	**Closed**	
Related to GROMACS - Feature #3034: Python gmxapi exception hierarchy	**Closed**	
Has duplicate GROMACS - Bug #999: add MYLIB_EXPORT for public API	**Closed**	**09/03/2012**

Associated revisions

Revision 792f6b06 - 11/27/2012 05:46 PM - Roland Schulz

Add visibility defines

This adds visibility declarations for functions and variables used
outside of a library. This enables shared linking on Windows and
thus makes the binaries much smaller. It doesn't effect any other
build because the defines are set to empty by default.

With GCC the correctness can be tested by adding to CFLAGS:
-fvisibility=hidden -DUSE_VISIBILITY

Related to #701

Change-Id: Ied261586e49bae11c5f8a8dffe377f0b0f02dd9d

Revision 8819cb9d - 12/06/2012 12:34 PM - Teemu Murtola

Fix build of template and other external code.

Installed headers need to use relative include paths.
Broken by Ied26158 (which was part of #701).

Change-Id: I2d9e6ee3878ae1bfa70286996e490abad00f19a0

History

#1 - 02/10/2011 07:50 PM - Teemu Murtola

Added notes about build system integration to the description.

#2 - 09/03/2012 09:37 PM - Roland Schulz

You're right that is a duplicate. Didn't know about this issue.

It seems to me what we want to do is: Use -fvisibility on Unix, add MYLIB_EXPORT to public function/classes, and set MYLIB_EXPORT to *attribute* ((visibility ("default"))) on Unix. On Windows MYLIB_EXPORT would automatically be set correct by cmake.

#3 - 09/04/2012 05:41 AM - Teemu Murtola

- Subject changed from Add header file for symbol visibility macros to Add symbol visibility macros

- Description updated

Updated the description to cover more explicitly #999. We still need to have our own header that declares these export macros; CMake will only automatically define the libgromacs_EXPORTS macro to use in an #ifdef within that header.

With the added clarity of more than a year of development after writing down this issue, I would suggest that we put the declarations in src/gromacs/utility/visibility.h. They could also go into src/gromacs/utility/common.h, which would reduce the number of headers included from a lot of places. But it may be more clear to have them in a separate header (common.h could then be named something else, though).

We would still need to consider what we want to name the export macros: GMX_EXPORT like in the CMake page Roland linked (http://www.cmake.org/Wiki/BuildingWinDLL), or GMX_PUBLIC and GMX_LOCAL like on the GCC visibility page linked from this issue. Also #988 needs to be thought out to decide what we want to export. And it would be nice to also consider the library division a bit: currently, there is still libgromacs and libgmxana. It is not very clear whether we want to keep these separate, and if so, what would be the division (currently, trajectory analysis modules using the new framework are in libgromacs).

#4 - 09/06/2012 07:14 PM - Roland Schulz

I don't care whether we name it GMX_EXPORT or GMX_PUBLIC. But I think it would be better to not add a GMX_LOCAL and make local the default for gcc. We should change the default anyhow to local because it would make it easier to spot when GMX_PUBLIC/EXPORTS are missing. And if that is anyhow the default I don't see the benefit of manually having to GMX_LOCAL.

#5 - 09/06/2012 07:22 PM - Teemu Murtola

I don't have any strong opinion on the GMX_EXPORT vs GMX_PUBLIC issue either. And I agree that hidden visibility should be the default. But from the linked GCC visibility page, the idea of GMX_LOCAL would be that it's possible to hide selected symbols that would be otherwise made public by GMX_PUBLIC. If I interpret that page correctly, then applying GMX_PUBLIC to a class automatically makes all of its members public as well, and GMX_LOCAL would then allow selectively one to hide these.

#6 - 09/06/2012 07:46 PM - Roland Schulz

Missed that about the class member the first time reading it. Makes sense now. +1 for following the gcc recommendation. Regarding naming. Whether we name the first public or export and the second private or hidden I don't really care.

#7 - 09/15/2012 06:42 AM - Teemu Murtola

For the naming, public/private might be a bit confusing names as they are also keywords in C++, and there is no 1-to-1 mapping between the visibility and the C++ accessibility.

#8 - 04/19/2013 03:09 PM - Rossen Apostolov

- Description updated

#9 - 02/15/2014 07:12 PM - Teemu Murtola

- Project changed from Source code reorganization to GROMACS

- Category set to build system

- Target version set to future

#10 - 03/02/2019 01:38 AM - Eric Irrgang

- Related to Task #2045: API design and language bindings added

#11 - 03/02/2019 01:46 AM - Eric Irrgang

This would be nice to resolve for the library and build-system restructuring underway.

#12 - 03/22/2019 11:45 AM - Eric Irrgang

- Related to Feature #2896: Python packaging added

#13 - 03/31/2019 05:25 PM - Eric Irrgang

- Related to Task #2912: C++ extension module for Python bindings added

#14 - 07/12/2019 11:22 AM - Eric Irrgang

- Related to Feature #3034: Python gmxapi exception hierarchy added