

GROMACS - Bug #2646

gmxapi uses major() and minor() which clashes with deprecated glibc macros

09/18/2018 01:28 PM - Mark Abraham

Status:	Closed	
Priority:	Normal	
Assignee:	Eric Irrgang	
Category:	gmxapi	
Target version:	2019	
Affected version - extra info:		Difficulty: uncategorized
Affected version:	git master	

Description

compiler: clang-6.0.0

glibc: 2.27-3ubuntu1

In file included from ../src/api/cpp/version.cpp:38:

```
src/api/cpp/include/gmxapi/version.h:110:26: warning: In the GNU C Library, "major" is defined by
<sys/sysmacros.h>. For historical compatibility, it is currently defined by <sys/types.h> as well,
but we plan to remove this soon. To use "major", include <sys/sysmacros.h> directly. If you did n
ot intend to use a system-defined macro "major", you should undefine it after including <sys/types
.h>. [-W#pragma-messages]
    static version_t major();
                ^
```

```
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:101:21: note: expanded from macro 'major'
#define major(dev) __SYSMACROS_DM (major) gnu_dev_major (dev)
                ^
```

```
# define major(dev) __SYSMACROS_DM (major) gnu_dev_major (dev)
```

```
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:52:32: note: expanded from macro '__SYSMACROS_DM'
#define __SYSMACROS_DM(symbol) __SYSMACROS_DM1 \
                ^
```

```
#define __SYSMACROS_DM(symbol) __SYSMACROS_DM1 \
```

```
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:62:30: note: expanded from macro '__SYSMACROS_DM1'
#define __SYSMACROS_DM1(...) __glibc_macro_warning (#__VA_ARGS__)
                ^
```

```
#define __SYSMACROS_DM1(...) __glibc_macro_warning (#__VA_ARGS__)
```

```
/usr/include/x86_64-linux-gnu/sys/cdefs.h:470:3: note: expanded from macro '__glibc_macro_warning'
__glibc_macro_warning1 (GCC warning message)
^
```

```
__glibc_macro_warning1 (GCC warning message)
```

```
/usr/include/x86_64-linux-gnu/sys/cdefs.h:468:42: note: expanded from macro '__glibc_macro_warning
1'
```

```
# define __glibc_macro_warning1(message) _Pragma (#message)
                ^
```

```
# define __glibc_macro_warning1(message) _Pragma (#message)
```

```
<scratch space>:3:6: note: expanded from here
```

```
GCC warning "In the GNU C Library, \"major\" is defined\n by <sys/sysmacros.h>. For historical co
mpatibility, it is\n currently defined by <sys/types.h> as well, but we plan to\n
remove this soon
. To use \"major\", include <sys/sysmacros.h>\n directly. If you did not intend to use a system-de
fined macro\n \"major\", you should undefine it after including <sys/types.h>."
```

In file included from ../src/api/cpp/version.cpp:38:

```
src/api/cpp/include/gmxapi/version.h:116:26: warning: In the GNU C Library, "minor" is defined by
<sys/sysmacros.h>. For historical compatibility, it is currently defined by <sys/types.h> as well,
but we plan to remove this soon. To use "minor", include <sys/sysmacros.h> directly. If you did n
ot intend to use a system-defined macro "minor", you should undefine it after including <sys/types
.h>. [-W#pragma-messages]
    static version_t minor();
                ^
```

```
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:102:21: note: expanded from macro 'minor'
#define minor(dev) __SYSMACROS_DM (minor) gnu_dev_minor (dev)
                ^
```

```
# define minor(dev) __SYSMACROS_DM (minor) gnu_dev_minor (dev)
```

```
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:52:32: note: expanded from macro '__SYSMACROS_DM'
#define __SYSMACROS_DM(symbol) __SYSMACROS_DM1 \
                ^
```

```
#define __SYSMACROS_DM(symbol) __SYSMACROS_DM1 \
```

```
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:62:30: note: expanded from macro '__SYSMACROS_DM1'
#define __SYSMACROS_DM1(...) __glibc_macro_warning (#__VA_ARGS__)
                ^
```

```

#define __SYSMACROS_DM1(...) __glibc_macro_warning (__VA_ARGS__)
^
/usr/include/x86_64-linux-gnu/sys/cdefs.h:470:3: note: expanded from macro '__glibc_macro_warning'
__glibc_macro_warning1 (GCC warning message)
^
/usr/include/x86_64-linux-gnu/sys/cdefs.h:468:42: note: expanded from macro '__glibc_macro_warning
1'
# define __glibc_macro_warning1(message) _Pragma (#message)
^
<scratch space>:8:6: note: expanded from here
GCC warning "In the GNU C Library, \"minor\" is defined\n by <sys/sysmacros.h>. For historical co
mpatibility, it is\n currently defined by <sys/types.h> as well, but we plan to\n remove this soon
. To use \"minor\", include <sys/sysmacros.h>\n directly. If you did not intend to use a system-de
fined macro\n \"minor\", you should undefine it after including <sys/types.h>."
^
../src/api/cpp/version.cpp:43:20: warning: In the GNU C Library, "major" is defined by <sys/sysmac
ros.h>. For historical compatibility, it is currently defined by <sys/types.h> as well, but we pla
n to remove this soon. To use "major", include <sys/sysmacros.h> directly. If you did not intend t
o use a system-defined macro "major", you should undefine it after including <sys/types.h>. [-W#pr
agma-messages]
version_t Version::major()
^
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:101:21: note: expanded from macro 'major'
# define major(dev) __SYSMACROS_DM (major) gnu_dev_major (dev)
^
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:52:32: note: expanded from macro '__SYSMACROS_DM'
#define __SYSMACROS_DM(symbol) __SYSMACROS_DM1 \
^
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:62:30: note: expanded from macro '__SYSMACROS_DM1'
#define __SYSMACROS_DM1(...) __glibc_macro_warning (__VA_ARGS__)
^
/usr/include/x86_64-linux-gnu/sys/cdefs.h:470:3: note: expanded from macro '__glibc_macro_warning'
__glibc_macro_warning1 (GCC warning message)
^
/usr/include/x86_64-linux-gnu/sys/cdefs.h:468:42: note: expanded from macro '__glibc_macro_warning
1'
# define __glibc_macro_warning1(message) _Pragma (#message)
^
<scratch space>:13:6: note: expanded from here
GCC warning "In the GNU C Library, \"major\" is defined\n by <sys/sysmacros.h>. For historical co
mpatibility, it is\n currently defined by <sys/types.h> as well, but we plan to\n remove this soon
. To use \"major\", include <sys/sysmacros.h>\n directly. If you did not intend to use a system-de
fined macro\n \"major\", you should undefine it after including <sys/types.h>."
^
../src/api/cpp/version.cpp:48:20: warning: In the GNU C Library, "minor" is defined by <sys/sysmac
ros.h>. For historical compatibility, it is currently defined by <sys/types.h> as well, but we pla
n to remove this soon. To use "minor", include <sys/sysmacros.h> directly. If you did not intend t
o use a system-defined macro "minor", you should undefine it after including <sys/types.h>. [-W#pr
agma-messages]
version_t Version::minor()
^
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:102:21: note: expanded from macro 'minor'
# define minor(dev) __SYSMACROS_DM (minor) gnu_dev_minor (dev)
^
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:52:32: note: expanded from macro '__SYSMACROS_DM'
#define __SYSMACROS_DM(symbol) __SYSMACROS_DM1 \
^
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:62:30: note: expanded from macro '__SYSMACROS_DM1'
#define __SYSMACROS_DM1(...) __glibc_macro_warning (__VA_ARGS__)
^
/usr/include/x86_64-linux-gnu/sys/cdefs.h:470:3: note: expanded from macro '__glibc_macro_warning'
__glibc_macro_warning1 (GCC warning message)
^
/usr/include/x86_64-linux-gnu/sys/cdefs.h:468:42: note: expanded from macro '__glibc_macro_warning
1'
# define __glibc_macro_warning1(message) _Pragma (#message)
^

```

```

<scratch space>:2:6: note: expanded from here
GCC warning "In the GNU C Library, \"minor\" is defined\n by <sys/sysmacros.h>. For historical co
mpatibility, it is\n currently defined by <sys/types.h> as well, but we plan to\n remove this soon
. To use \"minor\", include <sys/sysmacros.h>\n directly. If you did not intend to use a system-de
fined macro\n \"minor\", you should undefine it after including <sys/types.h>."
^
./src/api/cpp/version.cpp:75:18: warning: In the GNU C Library, "major" is defined by <sys/sysmac
ros.h>. For historical compatibility, it is currently defined by <sys/types.h> as well, but we pla
n to remove this soon. To use "major", include <sys/sysmacros.h> directly. If you did not intend t
o use a system-defined macro "major", you should undefine it after including <sys/types.h>. [-W#pr
agma-messages]
    if (Version::major() < major)
        ^
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:101:21: note: expanded from macro 'major'
# define major(dev) __SYSMACROS_DM (major) gnu_dev_major (dev)
        ^
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:52:32: note: expanded from macro '__SYSMACROS_DM'
#define __SYSMACROS_DM(symbol) __SYSMACROS_DM1 \
        ^
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:62:30: note: expanded from macro '__SYSMACROS_DM1'
#define __SYSMACROS_DM1(...) __glibc_macro_warning (#__VA_ARGS__)
        ^
/usr/include/x86_64-linux-gnu/sys/cdefs.h:470:3: note: expanded from macro '__glibc_macro_warning'
__glibc_macro_warning1 (GCC warning message)
^
/usr/include/x86_64-linux-gnu/sys/cdefs.h:468:42: note: expanded from macro '__glibc_macro_warning
1'
# define __glibc_macro_warning1(message) _Pragma (#message)
        ^
<scratch space>:7:6: note: expanded from here
GCC warning "In the GNU C Library, \"major\" is defined\n by <sys/sysmacros.h>. For historical co
mpatibility, it is\n currently defined by <sys/types.h> as well, but we plan to\n remove this soon
. To use \"major\", include <sys/sysmacros.h>\n directly. If you did not intend to use a system-de
fined macro\n \"major\", you should undefine it after including <sys/types.h>."
^
./src/api/cpp/version.cpp:79:18: warning: In the GNU C Library, "major" is defined by <sys/sysmac
ros.h>. For historical compatibility, it is currently defined by <sys/types.h> as well, but we pla
n to remove this soon. To use "major", include <sys/sysmacros.h> directly. If you did not intend t
o use a system-defined macro "major", you should undefine it after including <sys/types.h>. [-W#pr
agma-messages]
    if (Version::major() > major)
        ^
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:101:21: note: expanded from macro 'major'
# define major(dev) __SYSMACROS_DM (major) gnu_dev_major (dev)
        ^
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:52:32: note: expanded from macro '__SYSMACROS_DM'
#define __SYSMACROS_DM(symbol) __SYSMACROS_DM1 \
        ^
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:62:30: note: expanded from macro '__SYSMACROS_DM1'
#define __SYSMACROS_DM1(...) __glibc_macro_warning (#__VA_ARGS__)
        ^
/usr/include/x86_64-linux-gnu/sys/cdefs.h:470:3: note: expanded from macro '__glibc_macro_warning'
__glibc_macro_warning1 (GCC warning message)
^
/usr/include/x86_64-linux-gnu/sys/cdefs.h:468:42: note: expanded from macro '__glibc_macro_warning
1'
# define __glibc_macro_warning1(message) _Pragma (#message)
        ^
<scratch space>:12:6: note: expanded from here
GCC warning "In the GNU C Library, \"major\" is defined\n by <sys/sysmacros.h>. For historical co
mpatibility, it is\n currently defined by <sys/types.h> as well, but we plan to\n remove this soon
. To use \"major\", include <sys/sysmacros.h>\n directly. If you did not intend to use a system-de
fined macro\n \"major\", you should undefine it after including <sys/types.h>."
^
./src/api/cpp/version.cpp:83:18: warning: In the GNU C Library, "minor" is defined by <sys/sysmac
ros.h>. For historical compatibility, it is currently defined by <sys/types.h> as well, but we pla
n to remove this soon. To use "minor", include <sys/sysmacros.h> directly. If you did not intend t

```

```

o use a system-defined macro "minor", you should undefine it after including <sys/types.h>. [-W#pr
agma-messages]
    if (Version::minor() < minor)
        ^
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:102:21: note: expanded from macro 'minor'
# define minor(dev) __SYSMACROS_DM (minor) gnu_dev_minor (dev)
        ^
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:52:32: note: expanded from macro '__SYSMACROS_DM'
#define __SYSMACROS_DM(symbol) __SYSMACROS_DM1 \
        ^
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:62:30: note: expanded from macro '__SYSMACROS_DM1'
#define __SYSMACROS_DM1(...) __glibc_macro_warning (#__VA_ARGS__)
        ^
/usr/include/x86_64-linux-gnu/sys/cdefs.h:470:3: note: expanded from macro '__glibc_macro_warning'
__glibc_macro_warning1 (GCC warning message)
    ^
/usr/include/x86_64-linux-gnu/sys/cdefs.h:468:42: note: expanded from macro '__glibc_macro_warning
1'
# define __glibc_macro_warning1(message) _Pragma (#message)
        ^
<scratch space>:17:6: note: expanded from here
GCC warning "In the GNU C Library, \"minor\" is defined\n by <sys/sysmacros.h>. For historical co
mpatibility, it is\n currently defined by <sys/types.h> as well, but we plan to\n remove this soon
. To use \"minor\", include <sys/sysmacros.h>\n directly. If you did not intend to use a system-de
fined macro\n \"minor\", you should undefine it after including <sys/types.h>."
    ^
../src/api/cpp/version.cpp:87:18: warning: In the GNU C Library, "minor" is defined by <sys/sysmac
ros.h>. For historical compatibility, it is currently defined by <sys/types.h> as well, but we pla
n to remove this soon. To use "minor", include <sys/sysmacros.h> directly. If you did not intend t
o use a system-defined macro "minor", you should undefine it after including <sys/types.h>. [-W#pr
agma-messages]
    if (Version::minor() > minor)
        ^
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:102:21: note: expanded from macro 'minor'
# define minor(dev) __SYSMACROS_DM (minor) gnu_dev_minor (dev)
        ^
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:52:32: note: expanded from macro '__SYSMACROS_DM'
#define __SYSMACROS_DM(symbol) __SYSMACROS_DM1 \
        ^
/usr/include/x86_64-linux-gnu/sys/sysmacros.h:62:30: note: expanded from macro '__SYSMACROS_DM1'
#define __SYSMACROS_DM1(...) __glibc_macro_warning (#__VA_ARGS__)
        ^
/usr/include/x86_64-linux-gnu/sys/cdefs.h:470:3: note: expanded from macro '__glibc_macro_warning'
__glibc_macro_warning1 (GCC warning message)
    ^
/usr/include/x86_64-linux-gnu/sys/cdefs.h:468:42: note: expanded from macro '__glibc_macro_warning
1'
# define __glibc_macro_warning1(message) _Pragma (#message)
        ^
<scratch space>:4:6: note: expanded from here
GCC warning "In the GNU C Library, \"minor\" is defined\n by <sys/sysmacros.h>. For historical co
mpatibility, it is\n currently defined by <sys/types.h> as well, but we plan to\n remove this soon
. To use \"minor\", include <sys/sysmacros.h>\n directly. If you did not intend to use a system-de
fined macro\n \"minor\", you should undefine it after including <sys/types.h>."
    ^
8 warnings generated.

```

Offhand, we should probably rename to majorVersion() et al.

Associated revisions

Revision 7569e5f1 - 09/19/2018 02:29 PM - Eric Irgang

Change names of gmxapi::Version functions.

major() and minor() member functions clash with a historical macro, producing compiler warnings. Change names to majorVersion(), etc.

Fixes #2646

Change-Id: I568ab1dc216b200b182b34f76016fbfa54015dbd

History

#1 - 09/18/2018 01:29 PM - Mark Abraham

- *Category set to gmxapi*

#2 - 09/18/2018 04:54 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#2646](#).

Uploader: M. Eric Irrgang (ericirrgang@gmail.com)

Change-Id: gromacs~master~I568ab1dc216b200b182b34f76016fbfa54015dbd

Gerrit URL: <https://gerrit.gromacs.org/8376>

#3 - 09/18/2018 05:11 PM - Eric Irrgang

- *Status changed from New to Fix uploaded*

Is this a build configuration in the Jenkins matrix tests? How should we confirm that the patch I submitted is a solution?

#4 - 09/19/2018 01:27 PM - Mark Abraham

Eric Irrgang wrote:

Is this a build configuration in the Jenkins matrix tests? How should we confirm that the patch I submitted is a solution?

I tested it manually on the same machine and voted verify +1 on Gerrit

#5 - 09/19/2018 02:30 PM - Eric Irrgang

- *Status changed from Fix uploaded to Resolved*

Applied in changeset [7569e5f1d90cff383c89d16e2f6dfe937ed16ea5](#).

#6 - 09/19/2018 03:03 PM - Mark Abraham

- *Status changed from Resolved to Closed*