GROMACS - Bug #1438

gmx mdrun -h broken

02/16/2014 02:29 AM - Mark Abraham

Status: Closed
Priority: Normal
Assignee: Teemu Murtola
Category: mdrun
Target version: 5.0
Affected version - extra info: only dev versions
Affected version: 5.0

Difficulty: uncategorized

Description
Running gmx mdrun -h produces a strange error. gmx -h is ok, gmx editconf -h is OK. I suspect we didn't make a matching code change to mdrun in a recent fix here.

Error was:

GROMACS: gmx mdrun, VERSION 5.0-beta2-dev-20140213-a26a0a6-local
Command line:
   gmx mdrun -h

-------------------------------------
Program: gmx mdrun, VERSION 5.0-beta2-dev-20140213-a26a0a6-local
Source file: ../src/gromacs/options/basicoptions.cpp (line 512)
Function: gmx::StringOptionStorage::StringOptionStorage(const gmx::StringOption&)

API error (bug):
Default enumeration index is out of range

For more information and tips for troubleshooting, please check the GROMACS website at http://www.gromacs.org/Documentation/Errors

(same in single and double)

I noticed it in CMAKE_BUILD_TYPE=Reference, but Release does it also. gmx mdrun -version says

GROMACS: gmx mdrun, VERSION 5.0-beta2-dev-20140213-a26a0a6-local
Command line:
   gmx mdrun -version

Gromacs version: VERSION 5.0-beta2-dev-20140213-a26a0a6-local
GIT SHA1 hash: a26a0a6fe7e13129696cef23555465df9d958db8
Branched from: unknown
Precision: single
Memory model: 64 bit
MPI library: none
OpenMP support: disabled
GPU support: disabled
invsqrt routine: (1.0/sqrt(x))
SIMD instructions: NONE
FFT library: fftpack (built-in)
RDTSCP usage: enabled
C++11 compilation: enabled
TNG support: enabled
Built on: Thu Feb  6 17:04:31 CET 2014

04/05/2020
Built by: mark@amd2 [CMAKE]
Build OS/arch: Linux 3.5.0-45-generic x86_64
Build CPU vendor: AuthenticAMD
Build CPU brand: AMD Opteron(tm) Processor 6376
Build CPU family: 21 Model: 2 Stepping: 0
Build CPU features: aes apic avx cmov cx8 cx16 f16c fma4 htt lahf_lm misalignsse mmx msr
nonstop_tsc pclmulqdq pdpe1gb popcnt pse rdtsscp sse2 sse3 sse4a sse4.1 sse4.2 ssse3 xop
C compiler: /usr/bin/gcc-4.7 GNU 4.7.3
C compiler flags: -Wextra -Wno-missing-field-initializers -Wno-sign-compare -Wall -Wno-unused
-Wunused-value -Wunused-parameter -Wno-unknown-pragmas -O0 -g
C++ compiler: /usr/bin/g++-4.7 GNU 4.7.3
C++ compiler flags: -std=c++0x -Wextra -Wno-missing-field-initializers -Wall -Wno-unused-functi
on -Wno-unused-unknown-pragmas -O0 -g
Boost version: 1.48.0 (internal)

Associated revisions
Revision 64fd20de - 02/16/2014 10:30 AM - Teemu Murtola

Fix default enum values in help output

nenum() does not return an enum value, but an index into the underlying
array, which means that the first element is 1 and not zero like the
code assumed. The help output was off-by-one, and could error out if
some environment variable was set that was affecting the help output.

Fixes #1438.

Change-Id: lec2f05c9cdd55868902069ce59bcb4979a6879f86

History

#1 - 02/16/2014 05:51 AM - Gerrit Code Review Bot
Gerrit received a related patchset `1' for Issue #1438.
Uploader: Teemu Murtola (teemu.murtola@gmail.com)
Change-Id: lec2f05c9cdd55868902069ce59bcb4979a6879f86
Gerrit URL: https://gerrit.gromacs.org/3150

#2 - 02/16/2014 05:55 AM - Teemu Murtola
- Status changed from New to Fix uploaded
- Affected version - extra info set to only dev versions
- Affected version changed from git master to 5.0

I couldn't reproduce it, but noticed that enum defaults were off-by-one in the help output. Probably you have some environment variable like
GMX_VIEW_XVG or GMXTIMEUNIT set which affects the way the defaults are initialized, so that one of them gets selected at the last possible value.
Unrelated lesson learned: when building a source package and generating the man pages etc., such environment variables should not be set.

#3 - 02/16/2014 10:45 AM - Teemu Murtola
- Status changed from Fix uploaded to Resolved
- % Done changed from 0 to 100

Applied in changeset 64fd20de16e5ea975b1064ba0cc89b252e6a5e8.

#4 - 02/16/2014 01:06 PM - Mark Abraham
Teemu Murtola wrote:

I couldn't reproduce it, but noticed that enum defaults were off-by-one in the help output. Probably you have some environment variable like
GMX_VIEW_XVG or GMXTIMEUNIT set which affects the way the defaults are initialized, so that one of them gets selected at the last possible value.
Unrelated lesson learned: when building a source package and generating the man pages etc., such environment variables should not be set.

Thanks - fix works. Indeed, I set GMX_VIEW_XVG=xmgrace -noask' in lieu of not being able to configure that "no ask upon exit" behaviour in
xmgrace.

That's a nasty gotcha, but I am glad we found it early.
#5 - 02/24/2014 10:59 AM - Rossen Apostolov

- Status changed from Resolved to Closed