

GROMACS - Bug #2211

gmx writes normal output to stderr

06/26/2017 06:27 PM - Daniel Bauer

Status: New	
Priority: Low	
Assignee:	
Category:	
Target version: future	
Affected version - extra info:	Difficulty: uncategorized
Affected version: N/A	
Description	
Hello, I recently noticed that all gmx commands write their output to stderr instead of stdout. Wouldnt it be better for logging etc to have normal output in stdout? Best regards, Daniel	
Related issues:	
Related to GROMACS - Task #1505: improve handling of logging	New

History

#1 - 06/27/2017 07:39 PM - Mark Abraham

Daniel Bepunkt wrote:

Hello,

I recently noticed that all gmx commands write their output to stderr instead of stdout. Wouldnt it be better for logging etc to have normal output in stdout?

I wouldn't say "all" ;-) The only consistent thing about GROMACS in this regard is inconsistency.

IMO the "normal output" should go to stdout, and anything that helps reacting to a terminating error condition should go to stderr. So gmx pdb2gmx -h writes the usage to stdout, but gmx pdb2gmx -broken writes the usage to stderr. Progress reports (e.g. which step or frame the tool is currently handling) should be written only if the terminal is interactive (e.g. using isatty()), and in that case, to stdout.

This is merely one aspect of the long-overdue overhaul of how gmx tools do logging... Teemu wrote some nice infrastructure, but we haven't taken much advantage of it, yet.

Relevant discussion at <https://unix.stackexchange.com/questions/331611/do-progress-reports-logging-information-belong-on-stderr-or-stdout>

#2 - 06/27/2017 07:39 PM - Mark Abraham

- Related to Task #1505: improve handling of logging added

#3 - 06/29/2017 11:55 AM - Daniel Bauer

Thanks for your response!

Great to see that I am not the only one who thinks that this is in need of an overhaul!

#4 - 06/30/2017 01:18 AM - Mark Abraham

Daniel Bepunkt wrote:

Thanks for your response!

Great to see that I am not the only one who thinks that this is in need of an overhaul!

Ah, but do you have the stomach to help out? ;-)

#5 - 07/05/2017 04:20 PM - Szilárd Páll

Side-note: some (most?) of the use of messages to stderr in mdrun was motivated by console message aggregation with MPI. Not sure to what extent have things changed recently, but stuff like LINCS or settle warnings that are sometimes precursors can be followed of an (expected) mdrun crash, even with a well-implemented console message aggregation, messages could be lost if MPI does not gather them.

#6 - 03/05/2018 02:00 PM - Mark Abraham

Szilárd Páll wrote:

Side-note: some (most?) of the use of messages to stderr in mdrun was motivated by console message aggregation with MPI. Not sure to what extent have things changed recently, but stuff like LINCS or settle warnings that are sometimes precursors can be followed of an (expected) mdrun crash, even with a well-implemented console message aggregation, messages could be lost if MPI does not gather them.

If e.g. a (multi-sim) master rank would play the role of aggregator, then e.g. for terminal-mode mdrun we could have a single message that the n simulations (a,b,c) had lincs issues (maybe on ranks x,y,z). That doesn't provide any guarantees that the message gets handled and output made before the runtime dies from a process segfaulting, but in general there's no promise that the MPI runtime will have propagated the existing terminal messages, either.