

## GROMACS - Feature #1437

### Online help formatting improvements

02/15/2014 06:03 AM - Teemu Murtola

<b>Status:</b>	New
<b>Priority:</b>	Normal
<b>Assignee:</b>	
<b>Category:</b>	documentation
<b>Target version:</b>	future
<b>Difficulty:</b>	uncategorized

#### Description

Various suggestions to improve the man pages/HTML help/output of `-h` or the process that generates them have been raised (some in [#969](#), some elsewhere). Collected them here and grouped somewhat.

Formatting of man pages:

- **OPTIONS:** AFAIK all `-foo/--foo-long` switches and options typically go to this section which means everything currently in "FILES" belongs to "OPTIONS". **Done for 5.0** (there is a single OPTIONS section for all the command-line options).
- **FILES:** seems to be for the file-arguments, not the options with file arguments. We don't have many (any?) plain file arguments, I think. **Done for 5.0** (see above)
- **SYNOPSIS:** file arguments are normally not listed with the default values, but in a generic way, right? Also, the current formatting is somewhat off of what the synopsis typically looks like: lacks the `[]` and `|` hints, the different forms of invocation would be nice to separate (MD, EM, etc.). Moving toward a more standard SYNOPSIS section would be useful not only for the sake of standardization, but also because IMO it'd make it more readable and suggestive. **Partially done for 5.0** (the synopsis uses a bit more standard notation)
- There are a few stray spaces at the beginning or end of lines (e.g. `man gmx-mdrun DESCRIPTION 1st par 1st line, 2nd par 2nd line` both beginning and end). Would it be easy to strip these away? **Done for 5.0**
- Because of limitations in the markup commands available, multiple tools have resorted to hardcoded formatting like below. This looks horrible in particular in the HTML pages, but also not very good on man pages.

```
Header [BR]
----- [PAR]
 * Itemized list. [BR]
 * Another entry. [BR]
```

**Done for 5.1** (some reStructuredText paragraph formatting is supported to get rid of all [BR] tags in the input)

Console help:

- It would be useful to have a *brief* `-h` help output with usage and brief option description; e.g. see `git commit -h` vs `git commit --help`: 45 vs 450 lines.
- Additionally, formatting like the above example doesn't work currently very nicely for console output either, because it strips leading whitespace from each output line. **Done for 5.1**

Build system for man pages:

- Man pages should be produced and installed by default also for builds from git. The problem is that this requires executing the compiled binaries, which does not work in cross-compilation cases, and we currently have no reliable means to detect when this is the case. **Obsolete in 5.1** (`GMX_BUILD_HELP=AUTO` did something for this, but it already got superseded by using Sphinx for generating the man pages, which makes it difficult to produce and install the man pages by default)
- Current approach of generating uniform headers and footers for the HTML pages is horrible, with a mixture of C++ and CMake code doing string replacements and cutting and pasting file fragments to get the final files. Also, links on HTML pages that are not automatically generated are hard to maintain and check that they are actually working. **Done for 5.1** (all HTML documentation is produced with Sphinx)
- With `GMX_BUILD_MDRUN_ONLY=ON` we should install a `share/man/man1/mdrun.1` (or perhaps symlink to `gmx-mdrun.1`?)

Content of man pages:

- How/where should we document relevant env. vars - often there is a separate section for this (see e.g. `man git`)?
- Could be nice to add in the future a few more sections like "FURTHER DOCUMENTATION", "REPORTING BUGS", etc. (got inspired by `man git`). This will require code to format new sections, right?

- Just realized that the main binary is called gmx, but we install share/man/man7/gromacs.7. As one would typically expect to have a man page corresponding to the binary's name, perhaps we should add a gmx.7 -> gromacs.7 symlink. **Done for 5.1** (we now install gmx.1 instead of gromacs.7; the content may need improving, sitll).

Our current support for nroff formatting is very limited. The new formatter produces slightly nicer-looking man pages than the old one, but they are still ugly. However, nroff is ancient, and it appears that typically people generate man pages using some tools from a more user-friendly format. If we want to put some effort into cleaning them up, we should also invest in choosing such a tool; ideally, it would also allow us to generate the whole HTML online reference from the same input format, including the headers/footers and links between the pages. For example, the mentioned git man pages appear to be generated from asciidoc. **Done for 5.1** (now the man pages are generated using Sphinx)

Some preceding discussion is in [#969](#), on the possible alternative formats and on the man page improvements.

#### Related issues:

Related to GROMACS - Feature #969: Generating man pages, html help etc. from ...	<b>Closed</b>	<b>07/12/2012</b>
Related to GROMACS - Task #1687: improve readability of help/man output	<b>Closed</b>	<b>02/13/2015</b>

#### Associated revisions

##### Revision 5ac574b9 - 02/16/2014 10:32 AM - Teemu Murtola

Improve man page formatting slightly

- Don't add extra space whenever [TT]/[BB]/[II] is specified. This may have been necessary 15 years ago, or has always been broken, but currently it adds a lot of forced double spaces and leading spaces onto man pages. The text is much more readable without it.
- Merge all command line options into a single OPTIONS section. Divide it into groups using extra paragraphs in between. Could possibly be formatted nicer, but that would require extra nroff trickery that we might rather get rid of.

Related to #1437.

Change-Id: I4b27b79604a1830914cd5d14d6581918f33bb107

##### Revision 62306b64 - 06/04/2015 07:09 AM - Teemu Murtola

Replace gromacs.7 man page with gmx.1

The description is the same, and the command list is there, but other content is omitted. Useful parts can be added back later, when it is clearer what we want to include here. Also, the man page now contains a synopsis and common options for the gmx binary, as is suitable for gmx.1.

Part of #1437.

Change-Id: I4dbdf542dd2b8a0a7a2e734eebdf13d68483842e

#### History

##### #1 - 02/15/2014 06:09 AM - Teemu Murtola

- *Related to Feature #969: Generating man pages, html help etc. from Options added*

##### #2 - 02/15/2014 06:19 AM - Teemu Murtola

- *Category set to documentation*

Earlier comments from [#969](#) by Szilard:

If I understand correctly, GMX\_BUILD\_HELP triggers building man + HTML + pdf. Could we set GMX\_BUILD\_HELP=ON by default? The only drawback I could see while testing is that the user may get a "Could not find Doxygen" message which is not particularly harmful.

GMX\_BUILD\_HELP currently controls building the man, HTML, and shell completions. Doxygen is unrelated, and so is the PDF manual.

Still, I'd prefer to not have to manually set GMX\_BUILD\_HELP=ON just to get the man pages installed. I think we could come up with a simple solution to test for cross-compile; would it not be enough to try to either build and run a test program or run gmx -h and if it fails we can assume that the build is either broken or it is cross-compiling. A note could reflect this assumption and this could trigger GMX\_BUILD\_HELP=OFF. Does this sound too complicated?

For this, I would like to refer to an older post on gmx-developers, also by Szilard:

[https://mailman-1.sys.kth.se/pipermail/gromacs.org\\_gmx-developers/2012-August/006221.html](https://mailman-1.sys.kth.se/pipermail/gromacs.org_gmx-developers/2012-August/006221.html)

There, the conclusion was that trying to check whether the compiled binaries will run on the build host would perhaps not be feasible, but would be a maintenance headache. If you now think otherwise, you are more than welcome to contribute the check and use it to set the default value for GMX\_BUILD\_HELP.

Also, the suggestion of running `gmx -h` to change GMX\_BUILD\_HELP doesn't sound feasible; that would mean that the configuration pass of CMake would actually need to compile the whole binary... Or alternatively, when the build is already running, we would suddenly need to rerun CMake and restart the build from the beginning.

What we perhaps could do is to make the man, html and completion targets robust in the sense that they only produce a reasonable error message, but would allow the build to finish, and would also not fail during make install even if the output was not produced. But this is non-trivial amount of work as well.

In any case, if we want to create nicely formatted man pages, I don't think that anyone is particularly interested in learning nroff in-depth and figuring out what is the best way to use it to produce such formatting. As stated in the description, most projects seem to generate the nroff text from some other source. If we go that way as well, then GMX\_BUILD\_HELP will also require this external tool, and the benefit of this extra detection code is suddenly a lot smaller. So do we want to spend extra effort on this?

This should now contain all the discussion from [#969](#); please move anything here that I may have missed.

### #3 - 02/15/2014 12:22 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#1437](#).

Uploader: Teemu Murtola ([teemu.murtola@gmail.com](mailto:teemu.murtola@gmail.com))

Change-Id: I4b27b79604a1830914cd5d14d6581918f33bb107

Gerrit URL: <https://gerrit.gromacs.org/3149>

### #4 - 02/15/2014 12:29 PM - Teemu Murtola

- Description updated

Grouped the list of issues in the description into a few groups.

### #5 - 02/15/2014 12:55 PM - Teemu Murtola

Formatting of man pages:

- OPTIONS: AFAIK all `-foo/--foo-long` switches and options typically go to this section which means everything currently in "FILES" belongs to "OPTIONS".
- FILES: seems to be for the file-arguments, not the options with file arguments. We don't have many (any?) plain file arguments, I think.
- There are a few stray spaces at the beginning or end of lines (e.g. `man gmx-mdrun DESCRIPTION 1st par 1st line, 2nd par 2nd line` both beginning and end). Would it be easy to strip these away?

All of this was legacy from the 15+ years old code. These should be fixed in <https://gerrit.gromacs.org/#/c/3149/>.

- SYNOPSIS: file arguments are normally not listed with the default values, but in a generic way, right? Also, the current formatting is somewhat off of what the synopsis typically looks like: lacks the `[]` and `|` hints,

<https://gerrit.gromacs.org/#/c/3079/> should make it relatively standard in notation and format it slightly nicer. Extra effort is required if we want to improve it from there.

the different forms of invocation would be nice to separate (MD, EM, etc.). Moving toward a more standard SYNOPSIS section would be useful not only for the sake of standardization, but also because IMO it'd make it more readable and suggestive.

That requires significant extra effort, unless you'd prefer to maintain all the synopsis stuff manually. Rather than having multiple mutually incompatible ways of invoking the same program, it would be nicer to just split those into separate commands. There can be cases where such multi-function single command could be warranted, but I think the general desire has been to make the commands simpler.

- It would be useful to have a `brief -h` help output with usage and brief option description; e.g. see `git commit -h` vs `git commit --help`: 45 vs 450 lines.

Note that `git commit --help` is exactly equal to `man git-commit`. We could possibly provide something like the shorter option list, but we should first decide how we want things to behave. It would help a lot if there was no way to get the full description as console output, but only as man/HTML pages, as formatting the console output is the part where it is difficult to rely on external tools. We could do as git does, and automatically launch man with some command line option (on Windows, git help opens the HTML pages in a browser). Also, should the short output also have shorter descriptions for the options, or could we just reuse the same descriptions?

- With `GMX_BUILD_MDRUN_ONLY=ON` we should install a `share/man/man1/mdrun.1` (or perhaps symlink to `gmx-mdrun.1`?)

Symlinking should be easy to do, but I don't think we want to add extra complexity of also making it possible to generate the man pages from an mdrun-only build (currently, it is not possible, as the export option is only available for gmx help).

- Just realized that the main binary is called gmx, but we install share/man/man7/gromacs.7. As one would typically expect to have a man page corresponding to the binary's name, perhaps we should add a gmx.7 -> gromacs.7 symlink.

The contents of the current gromacs.7 do not have any direct relation to the use of the gmx binary. The whole man page should be rewritten to actually describe the gmx binary and not Gromacs in general. And the parts that describe the command-line options to gmx, as well as the command lists, should still be auto-generated. And it should be installed as gmx.1, not gmx.7, once it actually describes how to use that binary. Until we have some mechanism to conveniently manage the man pages without directly writing nroff, I think this isn't worth the effort.

- How/where should we document relevant env. vars - often there is a separate section for this (see e.g. man git)?

I think an ideal solution would be a separate man page, e.g., gmx-environment.7 or similar, that is automatically generated from the source code such that it actually contains the environment variables actually used, without requiring constant maintenance.

- Could be nice to add in the future a few more sections like "FURTHER DOCUMENTATION", "REPORTING BUGS", etc. (got inspired by man git). This will require code to format new sections, right?

The code to add new sections by itself should not be difficult to add. But the issue is that the current markup format is extremely limited, so if those should contain anything except plain text (like an environment variable section, for example, would), then a lot of extra effort is required to enhance the formatting engine and/or to add special-purpose code for all these cases. Yet another reason to find a tool that does this with reasonable effort from us... Also, we probably don't want to have identical sections in all the generated man pages, but only have these centrally on the gromacs.7/gmx.1 man page (with possible references from where it is relevant).

#### **#6 - 02/13/2015 07:24 PM - Teemu Murtola**

- Related to Task #1687: improve readability of help/man output added

#### **#7 - 05/29/2015 12:23 PM - Teemu Murtola**

- Description updated

Updated description to reflect the current status.

#### **#8 - 05/30/2015 08:35 PM - Gerrit Code Review Bot**

Gerrit received a related patchset '1' for Issue [#1437](#).  
Uploader: Teemu Murtola ([teemu.murtola@gmail.com](mailto:teemu.murtola@gmail.com))  
Change-Id: I4dbdf542dd2b8a0a7a2e734eebdf13d68483842e  
Gerrit URL: <https://gerrit.gromacs.org/4677>

#### **#9 - 06/04/2015 09:20 PM - Teemu Murtola**

- Description updated

One more status update.