GROMACS - Task #666

Improve help printing in the command-line runner

01/14/2011 05:39 PM - Teemu Murtola

Status:	Closed
Priority:	Normal
Assignee:	Teemu Murtola
Category:	core library
Target version:	5.0
Difficulty:	uncategorized

Description

These issues should be considered:

1. Formatting of the help output should be improved. The main issue is in line wrapping, but it should also be thought about how closely we want to follow the help format from the old tools. The new option handling mechanism allows a lot of things that the old one did not, so the format could perhaps also be improved.

2. src/gromacs/options/AsciiHelpWriter.* should be renamed to something like CommandLineHelpWriter, because writing a generic help that would apply to something else than a command-line tool is probably too difficult. Separate help writers can be implemented in other contexts if needed (and common functionality refactored back to src/gromacs/options/ if the need arises).

3. It should be thought about how much of the functionality from the old wman.c is really needed, and that should be ported to the new system. It should also be possible to refactor things such that both implementations use the same underlying routines for text handling.

4. In 4.5, some help is accessible with (using g_select as an example) g_select -h, some with g_select -select help and friends. In 5.0, these become g_ana select -h, g_ana select -select help, and there can additionally be g_ana -h (see #672). It would be good to consolidate these. Also, different tools may provide the selection option under a different name than -select, or may provide multiple options, which may make it confusing to use them for printing help. Suggestions of how it would work best for the user would be welcome. Also, how extensive should the command-line help be? An alternative is to refer to the manual or to a wiki page. If the command-line help contains a lot of information, we may want to add a script that extracts it and inserts it into the manual automatically (see #679).

Related issues:

Related to GROMACS - Task #642: Proper implementation for FileNameOption/File...	**Closed**	**01/08/2011**
Related to GROMACS - Task #665: Port existing trajectory analysis tools to us...	**New**	**03/27/2012**
Related to GROMACS - Bug #690: options program needs to be built by default	**Closed**	**01/28/2011**
Related to GROMACS - Task #672: Improve command-line trajectory analysis appl...	**Closed**	**01/19/2011**
Related to GROMACS - Task #679: g_select documentation should be included in ...	**Closed**	**01/22/2011**
Related to GROMACS - Feature #685: Wrapper binary	**Closed**	**01/26/2011**
Related to GROMACS - Feature #969: Generating man pages, html help etc. from ...	**Closed**	**07/12/2012**

Associated revisions

Revision 8b8f4566 - 04/18/2012 05:21 PM - Teemu Murtola

Renamed AsciiHelpWriter to CommandLineHelpWriter.

Kept the changes to a minimum to avoid messing the git rename detect.
Will rename/reorganize the code a bit more in a separate commit.
Also install the files, since they are documented as being part of the
public API.

IssueID #666

Change-Id: I96cf743d9364f80624ae3424c99019ae824e1520

Revision 282d5e36 - 04/18/2012 05:39 PM - Teemu Murtola

More reorganization of cmdlinehelpwriter.*.

- Moved classes used in the implementation from the ~~impl.h header to an anonymous namespace in the source file.~~

Removed "Ascii" from the names of the moved classes.
- Changed from prefix underscore to a postfix underscore for member variables.
- Removed unnecessary bit flags.
- Slightly changed the order of the classes.

IssueID #666

Change-Id: Iaedc803ba752753abc8943f1e82427cdc276cd13

Revision 9f9d331b - 04/21/2012 10:36 AM - Teemu Murtola

Split command line parsing to separate directory.

Part of IssueID #666.

Change-Id: Ia4969f636fbf6a3b00a9859d6705fc7375f0eb36

Revision 273c2d6d - 04/26/2012 10:30 AM - Teemu Murtola

Misc. option fixes.

- Fix two bugs introduced by earlier refactoring (one-liners in optionstoragetemplate.h and cmdlinehelpwriter.cpp).
Both noticed when starting to work on #666, subsequent commits will add tests that failed because of these.
- Added isRequired() to OptionInfo and related classes.
Needed for IssueID #666, but doing this in a separate commit to reduce the footprint of bigger subsequent commits.

Change-Id: Ifee906fe589fd93ff259a02166c41c9b1568439c

Revision e898bbc3 - 04/26/2012 11:17 AM - Teemu Murtola

Add text formatting classes.

Added classes for wrapping lines and for formatting tables, and tests for those.

Needed for IssueID #666.

Change-Id: I7d217a01ff43d2a79aa8bc0b9ae5db6a43b14868

Revision 0a03fb48 - 04/26/2012 11:17 AM - Teemu Murtola

Move string array concatenation to format.h.

- Moved code from Options::setDescription() that concatenated strings from an array into format.h to make it more generally usable.
- Changed the code such that instead of requiring the array to be NULL-terminated, it can also deduce the length of the array passed as an argument.
- Made Options::setDescription() to take a simple string to allow using the alternative overload, and also to help in writing test code (for #666).

Change-Id: I96f45577fc6a4258a1c993983cc1d1a1a0846bdc

Revision b514fd27 - 05/03/2012 06:32 AM - Teemu Murtola

Rewrote command line help writer.

- Rewrote printing code for options using TextTableFormatter. Should now produce at least as good output as the old code invoked by parse_common_args() while keeping most of the layout, and in some cases does significantly better (partly because of the item below).
- Split selection options into a separate table with its own formatting to make more space for long values that they typically have.
- Added test code that exercises different parts of the formatting code and allows one to easily see the output (with ~~stdout cmd-line parameter to the test executable). Without the -stdout parameter, each test is marked as failed if anything in the output changes, allowing them to work as regression tests.~~
Added line wrapping for descriptions.
- Markup substitution ([TT], [PAR] etc.) is not yet done. Will add that

separately, since it requires changes to the line wrapping code to strip some whitespace...

There are some TODOs in the code, but they mostly concern better handling of corner or more complex cases.

IssueID #666

Change-Id: I538fbaac27572a518d24dff7b0f30f8746a540af

Revision 6aee5224 - 05/03/2012 06:32 AM - Teemu Murtola

Implement markup substitution in command-line help.

Use code from wman.h, and adjust the line wrapper to strip extra whitespace from beginning and end of lines (these are easily produced with the combination of concatenateLines() and, e.g., [PAR] markup).

IssueID #666

Change-Id: I5413578c5fc51239b9f4dfc74387e21ac1d1c5ca

Revision 5ef19f95 - 05/03/2012 05:57 PM - Teemu Murtola

Add "replace all" functionality to format.h.

Use the new function in File::readToString() to make the Windows line end handling more robust.

Related to #666.

Change-Id: I24fc8e0dfd8af7ac08c94f2e9392fd418fe463c8

Revision c09ab7ec - 05/03/2012 05:57 PM - Teemu Murtola

Replace "%t" in option descriptions with time unit.

Not sure how useful this is, but there was such functionality in the old option listing, and it was easy to implement.

IssueID #666

Change-Id: Ib9538211fefcde7a89dceb7e1b30a81ab03d9e76

Revision 4db1c85a - 05/03/2012 05:57 PM - Teemu Murtola

Remove replace.* and use replaceAll() in wman.c.

Remove old O(N^2) implementation of replacing. Instead, use the C++ version also in wman.c (switched the file to C++ compilation).
Removed unused variables from wman.c, since they give warnings with our C++ compiler flags.

This is the first commit that introduces a real dependency from the pre-5.0 C code to new C++ code.

Related to #666.

Change-Id: I89db3badc51775cd5c683ac7e8b1040efac16447

Revision 4cce7c9d - 05/07/2012 05:56 AM - Teemu Murtola

Print default file names for command-line help.

Print the value set with defaultValueIfSet() for a file name option in the command-line help if no other value is provided for the option.

- The value is formatted by the AbstractOptionStorage::formatValue() pure virtual method when a special index is provided.
- OptionStorageTemplate implements handling of this special index, and provides a new pure virtual method formatSingleValue() to do the actual formatting, which no longer needs to know where the value comes from.
- Adjust the concrete option storage classes to the changes.

IssueID #666.

Change-Id: I8b51262042415f314bd5d4c8da51e6e31cfe3b21

Revision ab300758 - 05/07/2012 12:16 PM - Teemu Murtola

Generic handling for multiple cmd-line modules.

Added generic methods for multiple independent modules within a single command-line tool, and modified g_ana to use this.

By itself, does not change the external behavior significantly, but makes it easier to implement parts of #666 and #672.
The wrapper binary in #685 should also be easy to implement using this approach.

Change-Id: I3058bf5db2cbb7c7a1f9e4c87dd2db8f9727fe82

Revision 2ea25c76 - 05/08/2012 07:35 AM - Teemu Murtola

Print list of modules with 'g_ana help'.

IssueIDs #666 and #672.

Change-Id: I001966ab3ba44c59b0186456a6467322b3b3d26d

Revision 8aef6a7c - 05/29/2012 03:56 PM - Teemu Murtola

Separated string formatting used only for help.

Moved string formatting routines used only in help printing to src/gromacs/onlinehelp/ from src/gromacs/utility/format.*.
Also moved one function from cmdlinehelpwriter.cpp to the new file.
Subsequent commits will add more content to the new directory and will use the same string formatting routines.

Added NOMINMAX define for Windows to avoid min/max macros.

Related to #666.

Change-Id: I9b52012df2fc718bc3d68c697140661a241a1467

Revision bfdfd536 - 05/29/2012 03:56 PM - Teemu Murtola

Additional features to TextTableFormatter.

Add possibility to print out a table without a header (if there are no column titles) and possibility to indent the whole table.
Use the new features for formatting the list of commands in CommandLineModuleManager.

IssueID #666

Change-Id: I2d5aaceaf1c5545ff3be215d27ec30df38fbacde

Revision 1440526f - 05/29/2012 03:56 PM - Teemu Murtola

Generic handling of online help topics.

- Added a few simple classes for handling a tree of help topics (src/gromacs/onlinehelp/).
- Rewrote the selection help to use the new classes. Required changing the help parser slightly to pass all requested help topics in a single call to _gmx_sel_handle_help_cmd(). Although this changes the usage of the help slightly, did not yet update the help texts.
- Added a way to specify the bison and flex binaries to use for regenerate_parser.sh (useful, e.g., if multiple versions are installed).

Related to #666.

Change-Id: I318fc0fc42e4af39734a5aa65a503582302ecdd8

Revision 7c771f0a - 05/29/2012 03:56 PM - Teemu Murtola

Improve 'g_ana help'.

- 'g_ana help' now internally uses the new online help classes.
Without any additional arguments, it still prints out the list of
commands, and now also list of additional help topics.
- 'g_ana help <command>' prints help for the given command (the same as
'g_ana <command> ~~h' for the trajectory analysis modules).~~
'g_ana help selections <subtopic>' can be used to access the general
help for selections.

IssueID #666.

Change-Id: Id34044842b297591923bef242a93b4ae1ae9e14e

Revision 45d74f45 - 06/13/2012 11:52 PM - Teemu Murtola

Updates to selection online help.

Some of the help topics were outdated after changes introduced by the
linked Redmine issues (some of them even before that). Improved the
situation a bit and also did some misc. other documentation fixes.

Related to #656 and #666.

Change-Id: I7f18ca4ad8ce2aa5cd808281bab044253ca77f56

History

#1 - 01/17/2011 06:19 AM - Mark Abraham

Teemu Murtola wrote:

> There are three issues:
>
> 1. Formatting of the help output should be improved. The main issue is in line wrapping, but it should also be thought about how closely we want to follow the help format from the old tools. The new option handling mechanism allows a lot of things that the old one did not, so the format could perhaps also be improved.

The old format seems fine as g_tool -h, and man g_tool, and in the manual (LaTeX via g_tool -h).

> 2. src/gromacs/options/AsciiHelpWriter.* should be renamed to something like CommandLineHelpWriter, because writing a generic help that would apply to something else than a command-line tool is probably too difficult. Separate help writers can be implemented in other contexts if needed (and common functionality refactored back to src/gromacs/options/ if the need arises).

Seems fair.

> 3. It should be thought about how much of the functionality from the old wman.c is really needed, and that should be ported to the new system. It should also be possible to refactor things such that both implementations use the same underlying routines for text handling.

Don't know what that functionality was :-)

#2 - 01/17/2011 07:33 AM - Teemu Murtola

Mark Abraham wrote:

> The old format seems fine as g_tool -h, and man g_tool, and in the manual (LaTeX via g_tool -h).

You can see at least some of the issues with the current formatting by running src/programs/g_ana/g_ana select -h from the master branch, and comparing the output to g_select -h from the 4.5 series. It is possible to replicate the old output exactly, but as said, the old output is not ideal for, e.g., the option -select, that can have a very long string value. The new system can also support options that take an arbitrary number of values, or can be specified more than once for different effects, or for which the order matters.

The main reason is that I had the choice of either doing a proper C++ implementation of the option handling for the new framework, or doing a pile of hacks on top of the old implementation, and I chose to first one to make it more future-proof. One option would be to still convert the new options to the old ones and feed them into the old help system, but that's easily as much work as writing the formatting from scratch. I just wanted to raise some discussion on whether the old format is the best one before me or someone else puts in a lot of effort of writing all that string manipulation code.

> Don't know what that functionality was :-)

It's all the functionality that's used to write the help in various formats, including console output, Latex, man, XML, wiki, ... The console output is still the one that takes the most work (point 1 in the description), because in everything else the formatting is markup-based, and so there is no need for

complicated string manipulation.

#3 - 01/28/2011 09:18 AM - Mark Abraham

Teemu Murtola wrote:

> Mark Abraham wrote:
>
> > The old format seems fine as g_tool -h, and man g_tool, and in the manual (LaTeX via g_tool -h).
>
> You can see at least some of the issues with the current formatting by running src/programs/g_ana/g_ana select -h from the master branch, and comparing the output to g_select -h from the 4.5 series.

Yeah that is ugly. I don't know if it suggests that -selrpos and -seltype should be replaced by several enum options of shorter lengths. Either way, wrapping to a new line offset by 10 characters or so isn't the world's hardest thing to do. I'd aim for clarity for the tool in its current form more than exact consistency.

> It is possible to replicate the old output exactly, but as said, the old output is not ideal for, e.g., the option -select, that can have a very long string value.

Even if a full 80 columns was available, there'd have to be code to do line-wrapping. Other than sorting such options to the end of the list, I'm not sure there's anything special to do.

> The new system can also support options that take an arbitrary number of values, or can be specified more than once for different effects, or for which the order matters.
>
> The main reason is that I had the choice of either doing a proper C++ implementation of the option handling for the new framework, or doing a pile of hacks on top of the old implementation, and I chose to first one to make it more future-proof. One option would be to still convert the new options to the old ones and feed them into the old help system, but that's easily as much work as writing the formatting from scratch. I just wanted to raise some discussion on whether the old format is the best one before me or someone else puts in a lot of effort of writing all that string manipulation code.

I'd certainly not demand that all option descriptions fit into the right-hand column. Wrapping to a longer line will often look right, unless the second line is so short it could fit into just a second row of the right-hand-column.

> Don't know what that functionality was :-)
>
> It's all the functionality that's used to write the help in various formats, including console output, Latex, man, XML, wiki, ... The console output is still the one that takes the most work (point 1 in the description), because in everything else the formatting is markup-based, and so there is no need for complicated string manipulation.

Minimising code duplication seems like it has to be right.

#4 - 01/28/2011 05:47 PM - Teemu Murtola

Mark Abraham wrote:

> Teemu Murtola wrote:
>
> > You can see at least some of the issues with the current formatting by running src/programs/g_ana/g_ana select -h from the master branch, and comparing the output to g_select -h from the 4.5 series.
>
> Yeah that is ugly. I don't know if it suggests that -selrpos and -seltype should be replaced by several enum options of shorter lengths. Either way, wrapping to a new line offset by 10 characters or so isn't the world's hardest thing to do. I'd aim for clarity for the tool in its current form more than exact consistency.

Splitting the enums is one option, but perhaps out of scope of this issue. :) If someone wants to do it, I don't have any objections as long as it keeps the same level of modularity that is currently in the code: the same list of strings is used to initialize the enums and the position keywords in the selection engine, and the strings themselves are completely handled by a separate position calculation engine. It's low on my list of priorities, because it works quite nicely as it is (unless one doesn't like the help output), and would probably take several hours to change it properly.

> It is possible to replicate the old output exactly, but as said, the old output is not ideal for, e.g., the option -select, that can have a very long string value.
>
> Even if a full 80 columns was available, there'd have to be code to do line-wrapping. Other than sorting such options to the end of the list, I'm not sure there's anything special to do.

Line wrapping is easy to do, but to achieve clarity will require some thinking beyond just writing the whole option description with values and all that on one line and wrapping that. Keeping the format compact for the default case (simple g_ana select -h with no other options specified) has some value in giving a quick overview of the possible options, and default values are useful as well, but the present way of printing the actual values is harder: how about if you give a couple of 200-character selections to the -select option? How should the wrapping look like to make it understandable?

The main point of this issue is that it's easy to make the help look reasonable in the easiest cases, but if it anyways needs to be rewritten, why not do it right from the beginning?

> Minimising code duplication seems like it has to be right.

That's one side of the coin, but it will easily take more code to use the old code for printing the help than to write it from scratch. And again, if the new code will sooner or later replace the old one, writing hundreds of lines of code to reuse a few hundred lines of code is not really worth it. Some of the low-level string replacement code could probably still be written such that it can be used from both, but that's only a small fraction of the code.

#5 - 02/05/2011 07:38 AM - Teemu Murtola

There's discussion on using an external markup syntax for producing the manuals, HTML help, etc. in issue #690. Those issues should be considered for the third point in the issue description.

#6 - 04/13/2012 05:40 PM - Teemu Murtola

- Target version set to 5.0

#7 - 04/19/2012 08:48 PM - Teemu Murtola

There is now a change pending review in gerrit for the rename part (point 2). However, it might be good to do a bit more reorganization and split the command-line help (and the parser at the same time) out of the options module. Then they could have a dependency on SelectionOptionInfo without introducing a cyclic module-level dependency. This would help in printing more useful help for the selection options.

#8 - 04/21/2012 07:48 PM - Teemu Murtola

- Status changed from New to In Progress

- Assignee set to Teemu Murtola

#9 - 04/26/2012 01:34 PM - Teemu Murtola

- Description updated

https://gerrit.gromacs.org/#/c/815/ adds better formatting for the option tables. With that (and some markup substitution as mentioned in that commit message), it should be acceptable. But it would be good to consider how command-line help in general should work. Updated the description with point 4 to consider. Point 2 from the description is done, and the mentioned change addresses part of point 1. Others remain.

#10 - 04/30/2012 09:30 AM - Teemu Murtola

- Description updated

A change adding markup substitution is now also available in gerrit (https://gerrit.gromacs.org/#/c/819/).

Updated the fourth point in the description with more things to consider. One option for making the situation more consistent would be to enhance the g_ana -h and/or g_ana select -h options to take string arguments to specify what parts of help to produce. The current output could still be produced if no string is provided. But that will take some design and implementation effort, so it would be nice to get some feedback on what people think before doing anything extra.

#11 - 05/05/2012 09:12 AM - Teemu Murtola

Teemu Murtola wrote:

> One option for making the situation more consistent would be to enhance the g_ana -h and/or g_ana select -h options to take string arguments to specify what parts of help to produce. The current output could still be produced if no string is provided. But that will take some design and implementation effort, so it would be nice to get some feedback on what people think before doing anything extra.

Another option could be to use g_ana help instead of g_ana -h for printing the help, which would also be in line with discussion in #685 (g_ana could latter be easily extended to be come the wrapper binary).

#12 - 07/12/2012 04:32 PM - Teemu Murtola

- Status changed from In Progress to Closed

All except the third point has been implemented. I created a separate task (#969) for that point for clarity, so this one can be closed. I still have some

ideas on how the usability of the online help could be improved, and some content could be added, but that can be done outside this issue.

#13 - 02/16/2014 09:02 PM - Teemu Murtola

- Project changed from Next-generation analysis tools to GROMACS

- Category set to core library