

GROMACS - Task #642

Proper implementation for FileNameOption/FileNameOptionStorage

01/08/2011 05:06 PM - Teemu Murtola

Status:	Closed	
Priority:	Normal	
Assignee:	Teemu Murtola	
Category:	core library	
Target version:	5.0	
Difficulty:	uncategorized	
Description		
It should be thought about which features of the old implementation in <code>filenm.c</code> is appropriate for new analysis tools, and those should be implemented in the mentioned classes. It should also be possible to wrap the old code in the new classes, but the interface presented by <code>FileNameOption</code> still has to be thought out carefully. At least the interface should be implemented soon, even if all the internal details are not, to allow development of actual tools without having to worry about the need to change them later to adapt to changes in this interface.		
Related issues:		
Related to GROMACS - Task #665: Port existing trajectory analysis tools to us...	New	03/27/2012
Related to GROMACS - Task #666: Improve help printing in the command-line runner	Closed	01/14/2011

Associated revisions

Revision 727418ac - 01/19/2011 05:24 PM - Teemu Murtola

Make default option values more flexible.

Added a new method that only provides default values for options that are set, but for which no value is provided. This makes it possible to conveniently use file name options like they used to work in the old tools, namely that '-ox' without a file name would turn on the output for a certain feature using the default file name. The implementation now supports the same functionality for all types of options.

IssueID #642

Revision 762095c2 - 04/18/2012 06:16 PM - Teemu Murtola

Split `FileNameOption` into a separate file.

IssueID #642 (also helps for some alternatives in IssueID #656)

Change-Id: I61b236fcd59963c8ba5f7a49aba2b5197552d316

Revision 442259a3 - 07/08/2012 07:27 AM - Teemu Murtola

Extension completion for `FileNameOption`.

- Make `FileNameOption` to complete extensions for input files, like the old file options did.
- Add descriptions for all file name options to improve the help output.
- Make `defaultValueIfSet()` really appear in the help output: it worked only in the case the option value was not stored.
- Adjust the reference for command-line help tests to match the new output.

Part of #642.

Change-Id: I4a749e3e82a42a6e64e0a78c91120f96a00381b9

Revision ec1b4a8f - 09/25/2012 03:47 AM - Teemu Murtola

Fine-tuning for `FileNameOption`.

- Change behavior of required options with default values such that the default value is now used without an error if the option is not provided. This is how it used to work with file name options. For

most other cases, this combination doesn't make much sense, but changed it such that it works consistently for all cases.

- Add `FileNameOption::defaultBasename()` as a more intuitive interface for providing a default for file name options. The name makes it clear that no extension is expected, and the option also transparently treats required/optional arguments without the user needing to understand the difference between `defaultValue()` and `defaultValuelSet()`.

Closes #642; there is still some things to be improved, but should now provide about the same features and about the same input options as the old `t_filenm`-based system.

Change-Id: `Ibcaf0667180e0b24b08604869d855baf23476681`

Revision 39086187 - 10/03/2013 02:07 PM - Teemu Murtola

Use `filenm.c` data for `FileNameOption`.

Expose the necessary data (through functions) from `filenm.c` so that `FileNameOption` can use it. Removes the need to duplicate the file name extensions (potentially also other stuff in the future) for the period that both coexist.

Part of #642.

Change-Id: `I0b375da494d6e9c8a0fe6a38d9e1f27078ab9726`

Revision e54b9c33 - 02/09/2014 01:39 PM - Teemu Murtola

Make `FileNameOption` behave more like old `filenm` parser

If a required file name option is provided without a value, then it is a no-op. Previously, `FileNameOption` raised an exception for such usage. Add a test to cover this case.

Related to #642.

Change-Id: `I89d6cc8ee5d5bf2915bb6841317b402ade00b99b`

History

#1 - 01/19/2011 05:46 PM - Teemu Murtola

- *Priority changed from High to Normal*

[727418a](#) introduced functionality that makes it easy to check whether optional file names are provided simply by checking whether a string variable is empty, and to still use default values. From the API point of view, the main remaining issue is to extend the list of file types in <source:src/gromacs/options/optionfiletype.h>. For usability for the user, what should still be implemented is:

1. Completion of default extensions for file names that don't refer to existing files. Here, I think the old functionality should be revised such that if the user provides a file name that refers to an existing file, it should be used as it is, no matter what it's extension is. Usually, the user knows best what he wants to do...
2. Task [#666](#) includes improvements to printing the option summary for the file name options; it can require some adaptation in the actual option implementation as well.

Depending on what other functionality related to file names there is to refactor, it could be clearer to move the file name option stuff to a separate module together with other functionality related to file names.

Now the programming interface of the file name options should provide the major functionality needed in tools, so reducing the priority and removing the dependency to the porting task [#665](#).

#2 - 04/13/2012 05:41 PM - Teemu Murtola

- *Target version set to 5.0*

#3 - 07/05/2012 09:08 PM - Teemu Murtola

- *Status changed from New to In Progress*

- *Assignee set to Teemu Murtola*

#4 - 02/05/2014 09:07 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#642](#).
Uploader: Teemu Murtola (teemu.murtola@gmail.com)

Change-Id: I89d6cc8ee5d5bf2915bb6841317b402ade00b99b
Gerrit URL: <https://gerrit.gromacs.org/3085>

#5 - 02/18/2014 09:04 AM - Teemu Murtola

- *Project changed from Next-generation analysis tools to GROMACS*
- *Category set to core library*
- *Status changed from In Progress to Closed*

Some internal implementation is still necessary to make it behave exactly like the old system, but closing this issue as the interface is reasonable already. New file types can also be added when needed.