

GROMACS - Task #652

Change selection method implementation to use C++

01/09/2011 04:40 PM - Teemu Murtola

Status:	Blocked, need info
Priority:	Normal
Assignee:	
Category:	selections
Target version:	future
Difficulty:	
Description	
<p>This would make it easier to publish the API for creating new selection methods in the C++ framework, and it would also make it more robust and easier to use and to maintain.</p> <p>The current implementation should map quite straightforwardly to C++. <code>gmx_ana_selmethod_t</code> should be made into a base class or an interface with virtual methods for each function pointer currently used, and all places where it is used should be adapted accordingly. Implementations of individual keywords in <code>sm_*.cpp</code> should be changed to be subclasses of it. The main difficulty is in handling options (that currently use an array of <code>gmx_ana_selparam_t</code> structures). It should be possible to use the options implementation in <code>src/gromacs/options/</code>, but it will need a non-trivial extension.</p>	
Subtasks:	
Task # 653: Extend options module for use in the selection engine	Closed
Related issues:	
Related to GROMACS - Task #655: Improve selection error reporting / switch to...	Closed 01/09/2011
Blocked by GROMACS - Task #651: Increase coverage of selection unit tests	Closed 01/09/2011
Blocked by GROMACS - Task #880: Exception handling in selection parsing	Closed 02/05/2012

Associated revisions

Revision 28c1cfd7 - 08/28/2012 05:45 AM - Teemu Murtola

Enable C++ use in selection methods.

The method that frees the internal data structure for a selection method now also has the responsibility of freeing the data structure itself. This makes it possible to allocate the data structure using operator `new` in the initialization function, and then correctly free it.

Prerequisite for using C++ types in selection methods before #652 is implemented.

Change-Id: I99e5a907cbdc4975d20becb0fb1d33e0026c135c

History

#1 - 12/29/2012 01:56 PM - Teemu Murtola

- Target version set to future

#2 - 04/30/2013 05:45 AM - Teemu Murtola

- Status changed from New to Accepted

#3 - 04/30/2013 05:46 AM - Teemu Murtola

- Status changed from Accepted to In Progress

#4 - 04/30/2013 05:46 AM - Teemu Murtola

- Status changed from In Progress to Blocked, need info

Waiting for [#653](#).

#5 - 05/22/2013 06:01 AM - Mark Abraham

- Target version changed from future to 5.0

#6 - 05/22/2013 08:09 PM - Teemu Murtola

Is it realistic/necessary to plan to do this for 5.0? This is a large effort, with very little user-visible effects. Even the API doesn't change, unless we expose a way to add custom selection keywords (which this task would enable). Unless some extra refactoring is also done (haven't created any Redmine task for that yet), it may also make the C/C++ mixture in the selection code even messier...

#7 - 05/22/2013 10:40 PM - Mark Abraham

- *Target version changed from 5.0 to future*

Sorry, I was assuming it was part of plans in the area. Do feel free to re-target if it seems appropriate. :-)

#8 - 06/10/2014 02:59 PM - Teemu Murtola

- *Project changed from Next-generation analysis tools to GROMACS*

- *Category set to selections*