# GROMACS - Task #2706

## Rework classic QM/MM interface

10/23/2018 08:59 AM - Paul Bauer

| | | |
|---|---|---|
| **Status:** | Accepted | |
| **Priority:** | Normal | |
| **Assignee:** | | |
| **Category:** | core library | |
| **Target version:** | 2020 | |
| **Difficulty:** | hard | |

**Description**

As reported by Carsten Kutzner on the mailing list, building with gaussian fails because of

```
QMMrec=fr->qr is an invalid use of incomplete type in qm_gaussian.cpp, line 435.
```

**Related issues:**

| | |
|---|---|
| Related to GROMACS - Bug #1934: QMMM with ORCA: memory leaks, buffer overflow... | **New** |
| Related to GROMACS - Bug #1899: Unable to compile QMMM (MOPAC) | **Closed** |

### Associated revisions

**Revision e26e468f - 10/24/2018 09:07 PM - Paul Bauer**

Update QM gaussian interface

The files had not been touched due to being hidden beneath
the QMMM flag and have thus started to rot.

Updated files and changed functions to reflect current status of the
rest of the code.

Fixes #2706

Change-Id: I2ecd24f3f85f36f704bbd5ba5df3c5faa6a7d6e5

**Revision 21f509ef - 11/05/2018 02:28 PM - Paul Bauer**

Make QM/MM code always compile

Brought all the old interfaces back to a state where they can always
compile regardless of the build configuration, and give fatal errors
if used from a configuration that didn't support the method.

When configured, this should work as before, but we have no ability to
test that in Jenkins.

Added some necessary const correctness.

Did QM/MM preparation all in the same place, to simplify runner.cpp

Added deprecation status to release notes.

Refs #2706, #2569

Change-Id: I4a6566c60bfbf27a7b1916be1874b36987fb7da5

### History

**#1 - 10/23/2018 12:22 PM - Gerrit Code Review Bot**

Gerrit received a related patchset '1' for Issue #2706.
Uploader: Paul Bauer (paul.bauer.q@gmail.com)
Change-Id: gromacs~release-2019~I2ecd24f3f85f36f704bbd5ba5df3c5faa6a7d6e5
Gerrit URL: https://gerrit.gromacs.org/8590

**#2 - 10/24/2018 09:19 AM - Paul Bauer**

There are additional issues with the remaining qm interfaces that should also be fixed, but at least the gamess one causes link errors because functions are not available.

#### #3 - 10/24/2018 09:21 AM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #2706.
Uploader: Paul Bauer (paul.bauer.q@gmail.com)
Change-Id: gromacs~release-2019~I4a6566c60bfbf27a7b1916be1874b36987fb7da5
Gerrit URL: https://gerrit.gromacs.org/8594

#### #4 - 10/24/2018 09:15 PM - Paul Bauer

*- Status changed from New to Resolved*

Applied in changeset e26e468fadff8489cf69d7e8793c558181938b79.

#### #5 - 10/25/2018 09:34 AM - Mark Abraham

*- Related to Bug #1934: QMMM with ORCA: memory leaks, buffer overflows and much more added*

#### #6 - 10/25/2018 09:34 AM - Mark Abraham

*- Related to Bug #1899: Unable to compile QMMM (MOPAC) added*

#### #7 - 10/25/2018 09:40 AM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #2706.
Uploader: Mark Abraham (mark.j.abraham@gmail.com)
Change-Id: gromacs~master~I4a6566c60bfbf27a7b1916be1874b36987fb7da5
Gerrit URL: https://gerrit.gromacs.org/8596

#### #8 - 10/25/2018 01:42 PM - Gerrit Groenhof

Yes, indeed apart from qm_gaussian the codes have not been touched in a decade and I think no-one is using them.

Besides, it is very simple to use the gaussian interface to call other programs as well, which is how we work with other QM programs now, such as Terachem, or Gamess-US.

Perhaps, rather than fixing these issues, we could just ditch the codes?

#### #9 - 10/30/2018 11:59 AM - Mark Abraham

Gerrit Groenhof wrote:

> Yes, indeed apart from qm_gaussian the codes have not been touched in a decade and I think no-one is using them.
>
> Besides, it is very simple to use the gaussian interface to call other programs as well, which is how we work with other QM programs now, such as Terachem, or Gamess-US.
>
> Perhaps, rather than fixing these issues, we could just ditch the codes?

@Arthur Might that approach suit your ORCA and MOPAC use cases?

#### #10 - 10/30/2018 12:42 PM - Arthur Zalevsky

While low-level routines are still present (which are more or less the same in case of every interface) I think it's ok to ditch old codes. It wouldn't affect our setup.

Though I'd prefer different - plugin-like setup for calling qm software. We've discussed it in brief with Paul in Riga. The basic idea, to have general, implementation-independent functions for qmmm (like pass coordinates and receive gradients) and a set of dynamic libraries for every package. So you're able to switch qm package almost on the fly without the need to recompile the whole gromacs. This is particularly useful when you have to compare different levels of theory which can be implemented in different packages (which is quite typical for qm). Another benefit of this approach is the ability to process data with python code, compiled with cython into C libraries (some qm outputs can be very laborious to parse with pure C/C++).

We already have a working prototype and I will definitely open a ticket later to discuss it.

#### #11 - 11/05/2018 10:06 AM - Paul Bauer

*- Tracker changed from Bug to Task*

*- Subject changed from Beta does not build with -DGMX_QMMM_PROGRAM=gaussian to Rework classic QM/MM interface*

*- Status changed from Resolved to Accepted*

*- Target version changed from 2019-beta2 to 2020*

*- Affected version - extra info deleted (likely also master)*

*- Affected version deleted (2019-beta1)*

*- Difficulty hard added*

*- Difficulty deleted (uncategorized)*

I changed this to a task because the general rework will be more involved than simply fixing the bug first mentioned here.