GROMACS - Feature #3034

Python gmxapi exception hierarchy

07/12/2019 11:22 AM - Eric Irrgang

Status:	Closed
Priority:	Normal
Assignee:	Eric Irrgang
Category:	gmxapi
Target version:	2020
Difficulty:	uncategorized

Description

libgmxapi can throw exceptions derived from gmxapi::Exception, defined in the gmxapi/exceptions.h installed header. These exceptions should have bindings in the gmxapi Python package (in the _gmxapi C++ extension module) so that they are easily catchable from Python.

Should C++ exceptions defined in the _gmxapi module should derive from gmxapi::Exception?

Is it a priority that users of the Python package should be able to use gmxapi.exceptions.Error as a catch-all base exception, even for exceptions originating in the _gmxapi extension module or in the core GROMACS library?

Developer documentation should describe how exceptions propagate and how they are catchable in both directions between Python and C++, with explanations of caveats for exceptions originating in different binaries or Python packages.

Also reference prior discussion at https://github.com/kassonlab/gmxapi/issues/125

Related issues:

Related to GROMACS - Task #701: Add symbol visibility macros	**New**
Related to GROMACS - Task #988: Definition of "public API"	**New**
Related to GROMACS - Task #2045: API design and language bindings	**New**

Associated revisions

Revision 6b9c1965 - 08/13/2019 11:04 AM - Eric Irrgang

Map gmxapi C++ exceptions to Python exceptions.

Create a base exception and several derived exceptions in the
gmxapi._gmxapi C++ extension module for the Python package. In addition
to mapping the few existing exceptions from gmxapi/exceptions.h, we add
handlers for unmapped exceptions derived from gmxapi::Exception and for
unknown exceptions, such as exceptions originating in the core library
or uncaught stdlib exceptions. gmxapi._gmxapi.Exception derives from
gmxapi.exceptions.Error.

Refs #3034

Change-Id: I270908216271876bcb8ef6e83b78ca333042c336

History

#1 - 07/12/2019 11:22 AM - Eric Irrgang

- Related to Task #701: Add symbol visibility macros added

#2 - 07/12/2019 11:22 AM - Eric Irrgang

- Related to Task #988: Definition of "public API" added

#3 - 07/12/2019 11:22 AM - Eric Irrgang

- Related to Task #2045: API design and language bindings added

#4 - 07/12/2019 11:24 AM - Eric Irrgang

- Description updated

#5 - 07/12/2019 05:40 PM - Eric Irrgang

- Subject changed from Python bindings for gmxapi exceptions to Python gmxapi exception hierarchy

- Description updated

We may not want an actual inheritance relationship between the base exception in gmxapi._gmxapi and the base exception in gmxapi.exceptions because that would mean either we have a potentially troublesome circular dependency (if gmxapi._gmxapi depends on gmxapi.exceptions) or we can't raise gmxapi.exceptions.Error until gmxapi._gmxapi is imported successfully. Unfortunately, the exception matching in try: ... except expressions doesn't seem to use the Python virtual inheritance hooks.

We could try to catch exceptions from gmxapi._gmxapi.Exception and wrap them in gmxapi.exception.X exceptions wherever possible in the Python package, but that would mean that all gmxapi._gmxapi objects would need to be wrapped in pure Python objects, and I don't think we want to do that.

We can also try minimizing the exceptions that can escape from gmxapi._gmxapi and instead develop the data model to embed the error reporting in return values, but we need to be cautious to be sure that errors can't go unnoticed and that failures occur as early as possible.

#6 - 07/25/2019 05:46 PM - Eric Irrgang

- Description updated

The more I think about it, the more comfortable I am with letting the CPython gmxapi._gmxapi.Exception subclass the pure Python gmxapi.exceptions.Error. The gmxapi._gmxapi module is an implementation detail of the gmxapi Python package, and in the real world it is essentially inaccessible without explicitly or implicitly importing the pure Python package, which imports gmxapi.exceptions very early. If we want to be rigorously certain that we are subclassing the gmxapi.exceptions.Error class that has already been imported by the parent package, the import machinery should allow us to do that explicitly within the _gmxapi module initialization code, rather than trusting py::module::import("gmxapi")

#7 - 09/25/2019 01:38 PM - Eric Irrgang

- Status changed from New to Resolved

gmxapi._gmxapi.Exception is the base exception in the C++ extension module, and is a subclass of gmxapi.exceptions.Error.

#8 - 10/13/2019 12:21 PM - Eric Irrgang

- Status changed from Resolved to Closed