For GROMACS 2019, gmxapi doxygen documentation is built with separate targets:

- `gmxapi_cppdocs` for installed headers, and
- `gmxapi_cppdocs_dev` for implementation details and verbosity on extension interfaces

HTML output appears in separate build/docs/api-user and build/docs/api-dev directories, respectively. The documentation should be better integrated during GROMACS 2020 development to be discoverable, helpful, and appropriately cross-referenced. This will require some decisions about intended user experience and appropriate technological dependencies.

Establish policies and layout for

- external (installed) API documentation
- extension API for plug-in developers
- developer documentation for API and library implementation levels.

Integrate with previous webpage-sphinx and doxygen targets and output.

Plan TBD.

Considerations

Extension-friendly software documentation usually seems to be most helpful when provided in both a prose / manual / tutorial form as well as an information-dense well-cross-referenced API reference. Both could be provided with the webpage build using Sphinx and auto-documentation. It is also reasonable to leave the API reference in standard doxygen output form, but we could use breathe to extract documentation from the Doxygen XML for building in the Sphinx-based documentation. gmxapi is an external interface that warrants at least some documentation in the same context as the command line interface or file-based input/output. As Python front-end and API documentation begins to be incorporated, it could be appropriate to provide C++ and Python documentation in parallel for the equivalent functionality.

We should note that the target audience of the gmxapi API documentation (researchers, power-users) is different from the target audience for library internal documentation (GROMACS contributors, GROMACS maintainers). It should be very clear which documentation is which, and probably difficult to stumble from one into the other. We don't have to make it quite as hard as in some other software projects to find the library documentation for the implementation of a feature described in the high-level API docs. But as we sharpen the distinction between internal and external interfaces, the distinction in documentation for API levels will necessarily be sharpened, too.

Subtasks:

- Feature # 2985: Python package documentation (Closed)

Related issues:

- Related to GROMACS - Task #988: Definition of "public API" (New)

Associated revisions

- Revision aa6a516e - 07/01/2019 07:10 AM - Eric Irrgang
  - gmxapi Python package documentation infrastructure.
    - Begin a documentation tree in python_packaging/documentation.
    - Create a docker image for quick rebuild and review of Sphinx docs.
Revision 17779a30 - 07/10/2019 09:52 AM - Eric Irrgang
gmxapi Python package documentation layout.

Refs #2698

Revision 41ceffda - 09/15/2019 02:51 PM - Eric Irrgang
Import gmxapi docs to main GROMACS docs directory.
This change defers removal of the documentation staged in python_packaging so to allow rearrangement of the docs in the new location without risking loss of content or the ability to generate the previous documentation.

Refs #2698

Revision f29a1758 - 10/03/2019 11:57 AM - Eric Irrgang
Build gmxapi documentation in main GROMACS documentation.
- Update gmxapi Python package installation instructions and remove quickstart.rst
- Minor updates to main installation guide document.

Refs #2698 #2985

Revision 9124029b - 10/31/2019 09:42 AM - Eric Irrgang
Deduplicate gmxapi documentation.
`docs/gmxapi` seems to have settled down and does not have substantial layout changes from the documentation in python_packaging/documentation. Compartmentalized doc builds, e.g. with docs.dockerfile, still warrant a separate simple conf.py, so that much is retained. Otherwise, it looks like documentation for the standalone gmxapi package or development can be maintained in a unified way with the GROMACS project documentation, so this change removes the transitonally duplicated content.
- Remove most of python_packaging/documentation
- Update python_packaging/docker/docs.dockerfile

Refs #2698 #2985

Change-Id: I0c23f6526894ec1ee4c910463c8c9d08a7315f6

History
#1 - 03/02/2019 01:31 AM - Eric Irrgang
- Parent task changed from #2585 to #2045

#2 - 03/02/2019 01:31 AM - Eric Irrgang
- Related to Task #988: Definition of "public API" added

#3 - 01/29/2020 03:59 PM - Eric Irrgang
- Status changed from New to In Progress
- Target version set to 2021-infrastructure-stable