

## GROMACS - Feature #3149

Task # 2045 (New): API design and language bindings

Feature # 3148 (New): Roadmap for gmxapi filesystem interactions.

### Python user interface for obtaining simulation artifacts as files.

10/15/2019 06:15 PM - Eric Irrgang

<b>Status:</b>	New
<b>Priority:</b>	Normal
<b>Assignee:</b>	Eric Irrgang
<b>Category:</b>	gmxapi
<b>Target version:</b>	
<b>Difficulty:</b>	uncategorized

#### Description

### Scenario

A user wants to run a simulation and acquire the resulting trajectory in a file with a specific name or even a specific filesystem location.

### Remediation

With the resolution of [#3144](#), a user is able to get (a reasonably good guess of) the trajectory file name from an MD operation handle with `md.output.trajectory.result()` (if the operation was assigned to `md`). Tools outside of the API can be used to stash the output where the user would like.

There are several problems. Some of them are:

- If the file is not named or located where the user wants, the user must either copy or move the file. If the file is moved, gmxapi loses access to the simulation output for an operation that was previously considered to have been completed.
- The "trajectory" output of the simulation operation is not intended to be a file path string in the near future. Treating it as one has to be considered an unsupported use case.
- This is completely unintegrated with the trajectory appending semantics.

### Significant questions:

- Under what circumstances should we checkpoint the entire trajectory product of the simulation (retain a complete copy)? I.e. Are trajectory frames a stream of data events that are consumed and then dropped, or is the entire trajectory a single result? My initial thought is that, in the long run, it is a series of data events, and whether they are retained for the entire graph execution is a function of the consuming operation, not of the `mdrun` operation.
- If `write_trajectory()` is a gmxapi operation, what options (if any) should the user have regarding how it is checkpointed? Should completion be a function of the target location, and, if so, what sort of error is encountered if the Context thinks the work is complete, but the output file does not exist or has a different fingerprint than expected? Should the Context retain the ability to re-deliver the file? How might the answer be affected by whether the working directory of the operation is on the same filesystem as the user-named output target (i.e. whether a filesystem "move" operation is a rename versus a data transfer)?

### Proposal

I believe that the best idiom at the high-level interface is to explicitly convert gmxapi operation output to a filesystem artifact with a helper that consumes the output of the simulation operation's trajectory output. I.e.  
`write_trajectory(myfilename, trajectory=md.output.trajectory)`

This allows us flexibility in the details of the trajectory output handle. It also allows us to confine details of trajectory writing semantics to the `write_trajectory` operation. For instance, it is clear when the expected behavior is to produce a complete trajectory in a single file, so it becomes clear what is required to checkpoint or relaunch a partially executed work graph. It also clarifies that we do not need to keep the full trajectory produced by `mdrun` in order to consider an MD node to be complete, as long as the Context is able to support the checkpointing behavior required by the trajectory consumers.

### Additional discussion

Some older relevant discussions should be migrated to the GROMACS issue tracking system, but can be found on GitHub:  
- <https://github.com/kassonlab/gmxapi/issues/190>

#### Related issues:

Related to GROMACS - Bug #3144: gmxapi.mdrun does not clearly expose the outp...	<b>Closed</b>
Related to GROMACS - Bug #3141: gmxapi File placeholders missing from beta re...	<b>New</b>
Related to GROMACS - Task #3139: gmxapi Futures should be subscribable	<b>New</b>
Related to GROMACS - Feature #3147: gmxapi workflow checkpointing	<b>New</b>

#### History

---

##### #1 - 10/15/2019 06:15 PM - Eric Irrgang

- Related to Bug #3144: gmxapi.mdrun does not clearly expose the output trajectory. added

##### #2 - 10/15/2019 06:15 PM - Eric Irrgang

- Related to Bug #3141: gmxapi File placeholders missing from beta release added

##### #3 - 10/15/2019 06:17 PM - Eric Irrgang

- Related to Task #3139: gmxapi Futures should be subscribable added

##### #4 - 10/15/2019 06:18 PM - Eric Irrgang

- Related to Feature #3147: gmxapi workflow checkpointing added

##### #5 - 10/15/2019 06:21 PM - Eric Irrgang

- Description updated