

JSR 352 Expert Group Meeting
March 2nd, 2012
Minutes

Meeting Time: 9-10:00AM ET

Attendees:

Kevin Connor
Tim Fanelli
Mahesh Kannan
Simon Martinelli
Michael Minella
Joe Pullen
Chris Vignola

References: EGAagenda-2Mar2012 (slides)

Discussion Summary:

- In reviewing readers/writers, Michael asked if the spec would include concrete readers/writers. Chris replied: "No, but the RI/TCK will include samples."
- The group discussed proposed listener annotations. Much discussion ensued. Here is as much of the transcript as I could capture:
 - Joe - seems like we need an eventing model. Should consider CDI events. What is the sequence of invocation onto these callbacks during execution?
 - Michael - seems like these annotations could be used across multiple artifacts, not just the ones presented.
 - Kevin - seems like you have combined lifecycle events and listeners.
 - Michael - we don't need both lifecycle events and listeners . You can build lifecycle events from the listeners.
 - Kevin - the semantics are different - e.g. order is defined for lifecycle, but not necessarily for listeners. You need both.
 - Simon - JPA has a lifecycle event model worth looking at.
 - Joe - seems like the item read/process/write annotations belong on readers/writers as lifecycle events.
 - Michael - actually seems like item read/process/write annotations belong on step...
 - Joe - how do you reuse a step and attach different listeners?
 - Chris - it is clear, I have combined listeners and lifecycle events - they need to be separated.

- Joe - I'm not sure we need all these listeners - they make sense in Spring Batch, but not necessarily here.

The net of the discussion was that it is clear Chris combined presented listeners as lifecycle events, which caused a fair amount of confusion. Chris agreed to separate listeners from lifecycle events and put a proposal in the public forum.

- The group then took a first look at concurrency (parallelization) models. Michael commented that the depiction of thread parallelization, which was intended to following the Spring Batch model, was not accurate - rather than a single thread doing all reads/writes, actually each thread does its own reads/writes per chunk. Michael agreed to check this out and follow up on the public forum, which he did later on March 2nd. Michael was correct - one thread per chunk, each thread doing it's own I/O.
- Michael suggested we need to explore how checkpointing works with parallel threads - especially concurrent threads. Chris agreed and said we'd dig into that next session because we ran out of time on March 2nd.
- The group meets again on Wednesday, 7 March 2012. The main topic of discussion will be repeat, retry, and concurrency.

Submitted,

Chris Vignola

March 5th, 2012