

JSR 352 Expert Group Meeting  
February 29th, 2012  
Minutes

Meeting Time: 9-10:00AM ET

Attendees:

Kevin Connor  
Mahesh Kannan  
Wayne Lund  
Simon Martinelli  
Joe Pullen  
Chris Vignola

References: EGAagenda-29Feb2012 (slides)

Discussion Summary:

Group walked through the referenced slides. The following points were made:

- The batch parameter (or property) discussion has been moved to the public mailing list. The latest iteration, which proposed the `@Parameter` annotation, appears to be gaining traction. Expect continued refinement of this topic.
- The group reviewed the revised batch container interfaces, that separate actions into "launch" and "management" categories, resulting in two distinct interfaces.
- The group discussed initial proposal about theoretical batch container topologies and batch application archive format. At this high level of detail, there was no disagreement in the general approach depicted in the slides.
- Joe raised a question about APIs vs SPIs, asking where SPIs fit into the scheme of things when it comes to things like the job repository. Chris responded the approach proposed thus far has SPIs to influence the batch container behavior and that we could validate this concept further once we get to slide 7. Chris further said, when we get to slide 7 and talk about container plugins, we can ask ourselves the question: "is this the appropriate placement of function?".
- The group further discussed the batch container plugin concept. With this concept, the batch container invokes replaceable plugins to carry such actions as transaction management, repository management, thread management, and so forth. The implementing of each plugin governs how each discipline is carried out. For example, a Java EE TransactionManagement plugin implementation would use JTA; a Java SE TransactionManagement plugin

implementation would be a no-op. A config.xml would be used to control which plugin implementations are actually used. These plugins are considered SPIs.

- Mahesh asked what is the boot-strap and lifecycle behavior of these plugins. Chris responded that is a good question, but we will have to wait until we start writing up the actual specification before we see that level of detail. Chris further stressed that at this (high) level of detail we are seeking only to establish general agreement on broad principles.
- Chris raised a question about synchronous vs asynchronous job execution. This topic briefly came up in a previous expert group meeting. Chris pointed out that an "execution handler" plugin would allow a container configuration to choose between either synchronous execution or asynchronous execution and asked "is that sufficient to accommodate both execution styles or do we need sync vs async execution choice in the Launcher interface itself?". Joe responded that the overwhelming use case is for asynchronous execution. Chris agreed. (Editorial: the strongest use case for sync execution of a batch job is a command line launch).
- The group took an initial look at proposed reader/writer annotations. The proposal essentially suggests the Spring Batch concepts of ItemReader or ItemWriter be combined with ItemStream semantics. The group seemed ok with this, but Joe and Simon stressed the importance of ensuring the batch specification is DI friendly (e.g. CDI or Spring DI) and asked how CDI (for example) @Inject and @<qualifier> would work. Chris pointed out that if @Reader, @Writer were moved from class level to field level that CDI would work naturally. Joe pointed out it would be a good idea to mock this up to confirm. Chris said he already had and it does, but invites others to do so as well. (Editorial: Chris should share his workspace that demonstrates this).
- The group meets again on Wednesday, 29 February 2012. The main topic of discussion will be listeners and concurrency.

Submitted,

Chris Vignola

February 29th, 2012