

Goal / Motivation

Check-lists (and questions) for both the Specification Documents and TCK User Guides.

Requesting the Spec. committee to review and perhaps we can have a discussion in the Specification Committee on July 8.

- *My preference would be that we arrive at some minimal list (may or may not be short) so that we can all perform roughly similar levels of minimum review. Certainly, this isn't intended to be a limiting list, just something to provide a common starting place. Ideally, we can resolve or decide to eliminate items on the Question lists. They would either be removed (perhaps just deferred to a later specification version), or added to the checklist.*

General

These issues occurred to me after reviewing: Jakarta Mail Specification Document, Jakarta Mail TCK User Guide; Jakarta Annotations TCK User Guide. All review drafts were built locally from the default repo-branch. I also looked at the Jakarta EE Annotations RC1 Specification ([here](#)).

General Questions:

- What level of errors can we tolerate?
- How do we report issues we find?

Specification Documents

Document what you reviewed (docs posted to specifications repo; read from GH Repo; repo-clone and built locally; something else?) Date and Branch -- whatever is needed to identify what you are reading.

Check the following

- Name, version, status -- should be as anticipated (in EE 9, probably .0.0 and status should be Final).
- License is properly EFSL
- PDF and HTML are both available
- Copyright in footer should be <previous date>, 2020

Scan the content and look for any of the following:

- Look for any obvious formatting problems

- javax -> jakarta -- package includes and properties (N.B. some are legitimate, e.g. [Annotation Spec. section 3.3.1](#))
- Look for any obvious image / chart problems (Maybe just write issues if these are noticed?)
- Check a few links from the TOC to be sure they point to the right target
- Note any Acronyms and possible proprietary marks. Some are okay (i.e. JavaBeans, but what about JAF in Jakarta Mail?)
- Check that dependent and reference specifications renamed as necessary
- Check that outbound links are updated and consistent (spot-check okay)
- Are properties renamed (from javaNN...NN to jakartaNN...NN)?
- Are code-samples within the document updated (generally, just check javax to jakarta)?
 - Check the appendices

Questions

- Should there be some text at the front-matter that says, to the effect, all trademark names are the property of the holder ... or something like that? Should we keep a list of appropriate/approved cases of these?
- If property names are not going to be renamed (say, for compatibility), should we keep a list or compile anything on these? (if for no other reason than to limit other notifications of these strings (Jakarta Mail Spec)
- Most specifications include a History section. These are frequently in the form of Acknowledgements. Should there be text that clarifies legacy versions and contributions were to Java EE?
 - Similar question if the spec. contains an Expert Group member list. Recommend this at least be labeled as a legacy or original list. Should be updated to list all current committers to the API Project.
- Should outbound references be updated to a Jakarta EE specification?
 - Certainly, there should be a preference to update, unless there is something specific I'd recommend they be revised (e.g. [Jakarta Annotations. Section 4. References](#)) -- Otherwise the documents may be behind a Java Specification License.
- Should we allow CI as an abbreviation for Compatible Implementation?
- Is there a "platform list" (i.e. Windows 10, Linux (some flavor, some version)? Specific versions of Java?
- If internal links are broken, is this a cause to fail the Spec? Just write an issue? I'd suggest we just spot-check a few. If those are satisfactory, that's probably good enough.
- Should the specifications refer to the EFSP, or the JESP? (Obviously, the license text comes from the EFSL, but how does a reader know they should read the JESP?)
- JavaHome -- I presume it's okay to intermingle installed artifacts. Do we need a JakartaHome property?

- Probably minor but there are some places where white-space is inserted inappropriately (see Annotation Spec, example in section [3.6 jakarta.annotation.PreDestroy](#)). This is certainly minor but flag it?

TCK User Guide Review

Document what you reviewed (docs posted to specifications repo; read from GH Repo; repo-clone and built locally; something else?) Date and Branch -- whatever is needed to identify what you are reading.

- Confirm Name, version, status -- should be as anticipated (in EE 9, probably .0.0 and status should be Final).
- Confirm License is properly [EFSL](#) or [EF TCK License](#)? (Assume the latter?)
- Confirm PDF and HTML are both available?
- Confirm footer copyright

Scan the content and look for any of the following:

- Any obvious formatting problems.
 - javax -> jakarta -- package includes and properties
 - reference implementation replaced with compatible implementation
- Check a few links from the Table of Contents to verify they point to the intended location
- Acronyms and possible proprietary marks. Some are okay (i.e. JavaBeans).
- Are dependent and reference specifications renamed as necessary?
- Are outbound links updated? (Spot check probably OK)
- For Jakarta EE 9, the TCK rules should not appreciably be changed from EE 8. If there are changes, they should be consistent with any changes identified in the Release Plan.

Questions

- Currently TCK docs (at least the few I looked at) state that it requires Java SE 8. Is that correct? Will we allow SE 8 and/or SE 11?
 - Is there a "platform matrix" we should consult?
- Should the specification include a firm reference to a compatible implementation? Are the release details necessary? (As an example, it is difficult to tell (in the doc) what is actually tested compatible for Jakarta Annotations? (e.g. in Jakarta EE 8, it was annotation-api.jar v1.3.5) Perhaps just refer to the specification ballot page?)
- For distribution through Maven, I think we concluded the content should be dual licensed. We need advice for that. (TCK User Guide is included in the TCK .zip). Is EPL compatible with the EFTckL?
- Should we cross check the included material with the IP log?

- I don't think most teams check the TCK run in GUI Mode. Maybe we just ignore this?
(i.e. not known to be broken.)