

Jakarta EE TCK Process 1.0 - Draft

Specification Projects under the Eclipse Foundation Specification Process MUST produce a Technology Compatibility Kit (TCK) that delivers on the promise of enabling multiple compatible implementations.

This document defines:

- Materials a TCK MUST possess to be considered suitable for delivering portability
- Process for challenging tests and how these challenges are resolved
- Means of excluding released TCK tests from certification requirements
- Policy on improving TCK tests for released specifications
- Process for self-certification

Materials for a TCK Release

Artifacts:

- Projects MUST produce the project licensed (EPL, Apache, etc.) version for distribution to Maven Central or other open source channels, usable for automated, non-official testing, and implementation. No compatibility claims can be made on the basis of this TCK.
- Projects MUST produce a final candidate binary that includes the [EFTL](#) license. The Jakarta EE Specification Committee will sign and promote project TCK binary for distribution via Eclipse infrastructure on final approval. This is the TCK binary usable for self-certification when one desires to make a claim of compatibility, allowing for the use of the Jakarta brands.
- Both TCK binaries MUST contain the following
 - User guide outlining
 - Software requirements
 - Installation and configuration
 - How to run the tests
 - Where to file challenges and bug reports
 - A statement that the [Certification of Compatibility](#) process must be followed before a claim of compatibility can be made.
 - TCK specific rules not covered in this process guide
 - Instructions describing how to run the compatible implementation(s) that are being used to validate the TCK.
 - A top-level README.md document pointing to each of the preceding documents
- TCK binaries MAY contain
 - Test coverage document
 - Test assertion document

- release available via a release on project GitHub releases page(or equivalent)
 - Final releases under the EFTL MUST be hosted on download.eclipse.org.

Ratifying a Final TCK

- Projects will submit the EFTL proposed final binary of the TCK for approval to the Specification Committee.
- The Specification Committee will vote to approve or reject the TCK binary.
- Approved binaries will be signed with the GPG key of the Jakarta Specification Committee, and then published on download.eclipse.org along with the digital signature of the SHA-256 hash of the final binary, and the SHA-256 hash of the binary as the fingerprint of the TCK.
- Consumers can use the GPG key of the Jakarta Specification Committee to verify the authenticity of that or any TCK binary.

Challenges

Specifications are the sole source of truth and considered overruling to the TCK in all senses. In the course of implementing a specification and attempting to pass the TCK, implementations may come to the conclusion that one or more tests or assertions do not conform to the specification, and therefore MUST be excluded from the certification requirements.

Requests for tests to be excluded are referred to as Challenges. This section identifies who can make challenges to the TCK, what challenges to the TCK may be submitted, how these challenges are submitted, how and to whom challenges are addressed.

Who can file a challenge?

Any implementor may submit a challenge to one or more tests in the TCK as it relates to their implementation. Implementor means the entity as a whole in charge of producing the final certified release. **Challenges filed MUST represent the consensus of that entity.**

Valid Challenges

Any test case (e.g., test class, @Test method), test case configuration (e.g., deployment descriptor), test beans, annotations, and other resources considered part of the TCK may be challenged.

The following scenarios are considered in scope for test challenges:

- Claims that a test assertion conflicts with the specification.
- Claims that a test asserts requirements over and above that of the specification.
- Claims that an assertion of the specification is not sufficiently implementable.
- Claims that a test is not portable or depends on a particular implementation.

Invalid Challenges

The following scenarios are considered out of scope for test challenges and will be immediately closed if filed:

- Challenging an implementation's claim of passing a test. Certification is an honor system and these issues **MUST** be raised directly with the implementation.
- Challenging the usefulness of a specification requirement. The challenge process cannot be used to bypass the specification process and raise in question the need or relevance of a specification requirement.
- Claims the TCK is inadequate or missing assertions required by the specification. See the Improvement section, which is outside the scope of test challenges.
- Challenges that do not represent a consensus of the implementing community will be closed. If agreement is later reached by the implementing community, the issue can be reopened. The test challenge process is not the place for implementations to initiate their own internal discussions.
- Challenges to tests that are already excluded for any reason.
- Challenges that an excluded test should not have been excluded and **SHOULD** be re-added **MUST** be opened as a new enhancement request

Filing a Challenge

Challenges **MUST** be filed via the specification project's TCK issue tracker using the label ``challenge`` and include the following information:

- The relevant specification version and section number(s)
- The coordinates of the challenged test(s)
- The exact TCK version
- The implementation being tested, including name and company
- A full description of why the test is invalid and what the correct behavior is believed to be
- Any supporting material; debug logs, test output, test logs, run scripts, etc.

Challenge Resolution

Challenges can be resolved by a specification project lead, or a project challenge triage team, after a consensus of the specification project committers is reached or attempts to gain consensus fails. Specification projects may exercise lazy consensus, voting or any practice that follows the principles of [Eclipse Foundation Development Process](#).

Active Resolution

The failure to resolve a Challenge might prevent an implementation from going to market; Challenges SHOULD be given a high priority by the specification project and resolved in a timely manner. Two weeks or less SHOULD be considered the ideal period of time to resolve a challenge. Challenges may go longer as needed, but as a rule SHOULD avoid months.

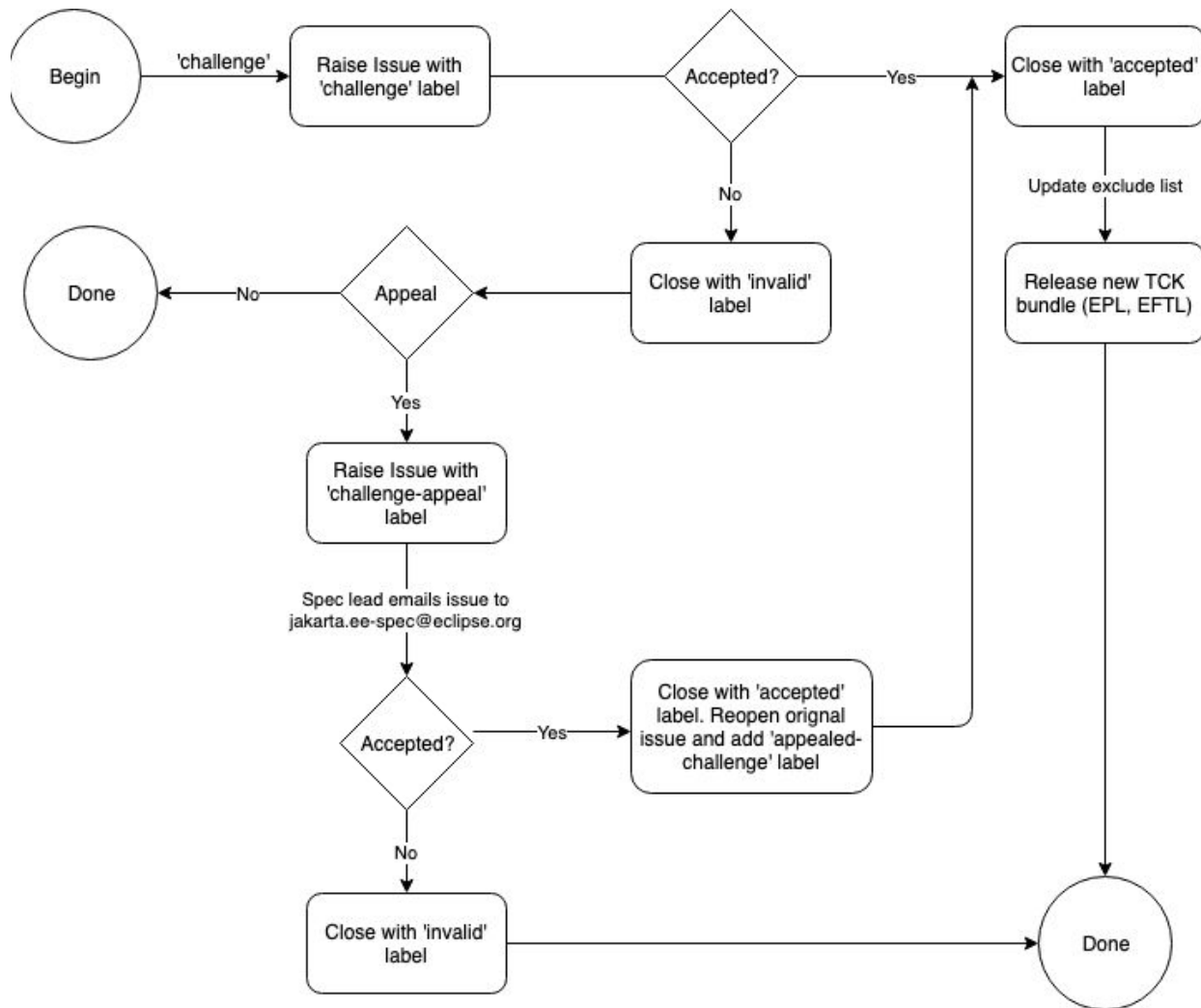
If consensus cannot be reached by the specification project for a prolonged period of time, the default recommendation is to exclude the tests and address the dispute in a future revision of the specification.

Accepted Challenges

A consensus that a test produces invalid results will result in the exclusion of that test from certification requirements, and an immediate update and release of an official distribution of the TCK including the new exclude list. The associated ``challenge`` issue MUST be closed with an ``accepted`` label to indicate it has been resolved.

Rejected Challenges and Remedy

When a ``challenge`` issue is rejected, it MUST be closed with a label of ``invalid`` to indicate it has been rejected. The appeal process for challenges rejected on technical terms is outlined in [Escalation Appeal](#). If, however, an implementer feels the TCK challenge process was not followed, an appeal issue MUST be filed with the specification project's TCK issue tracker using the label ``challenge-appeal``. A project lead MUST escalate the issue with the Jakarta EE Specification Committee via email (jakarta.ee-spec@eclipse.org). The committee will evaluate the matter purely in terms of due process. If the appeal is accepted, the original TCK challenge issue will be reopened and a label of ``appealed-challenge`` added, along with a discussion of the appeal decision, and the ``challenge-appeal`` issue will be closed. If the appeal is rejected, the ``challenge-appeal`` issue MUST be closed with a label of ``invalid``.



Excludes

Excludes MUST be included in the TCK project release in a format that is compatible with the testing framework in use so that as the excludes are updated, the affected tests are automatically removed from the test suite.

Improvement

Requests for improvement to tests MUST simply be created as issues with a label of `'enhancement'` in the specification project's TCK issue tracker.

Certification of Compatibility

Jakarta EE is a self-certification ecosystem. If you wish to have your implementation listed on the official <https://jakarta.ee> implementations page for the given specification, a certification request as defined in this section is required.

There are additional requirements that **MUST** be met by any organization wishing to use the Jakarta EE logo or jakarta.ee website for promotion. Any request for certification from an organization not meeting the requirements will be held until such time as the requirements are met. See the [Jakarta EE Trademark Guidelines](#) for more information on the complete steps to obtain Jakarta EE branding usage.

An approved certification request is a statement from the Specification Project you have met the intended TCK requirements and is just one of the requirements for logo usage. Logos only apply to platform and profile specifications, but a request to be listed as a compatible implementation can be made for any specification.

Filing a Certification Request

Requests to be acknowledged as a certified implementation **MUST** be filed via the specification project's TCK issue tracker using the label ``certification`` and include the following information:

- Statement of Acceptance of the terms of the EFTL
- Product Name, Version and download URL (if applicable)
- Specification Name, Version and download URL
- TCK Version, digital SHA-256l fingerprint and download URL
- Implementation runtime Version(s) tested
- Public URL of TCK Results Summary
- Any Additional Specification Certification Requirements
- Java runtime used to run the implementation
- Summary of the information for the certification environment, operating system, cloud, ...
- A statement attesting that all TCK requirements have been met, including any compatibility rules

Additional Specification Certification Requirements

Specification projects may require additional items for a Certification Request as defined in their corresponding TCK Documentation under the section labeled "Additional Certification Requirements".

Examples of such additional requirements may include:

- Name and version of Compatible Implementation used for interoperability tests
- Name and version of Databases used for persistence tests
- Name and version of NoSQL implementations used in persistence tests

Public TCK Results Summary

While certification is on your honor, the community **MUST** be able to see your test results summary. At a minimum a results summary **MUST**:

- Be publicly visible with no password protection or sign-up
- Include a Summary Page containing:
 - All information in the above Certification Request
 - The Total number of tests run and passed. If you are running a platform TCK, this means reporting the tests run for each required TCK, which includes the platform TCK and some of the individual TCKs that are not integrated with the platform TCK.

An optional “Test List Page” showing all tests run may be linked from the Summary Page. The Summary Page URL is the URL that **MUST** be included in any Certification Requests.

The following are explicitly not requirements:

- The Ability for the public to run the tests themselves
- Full log output of the TCK

Implementators may supply this information and provide support for how to run a TCK against their implementation, but it is not required.

Certification Resolution

As noted, certification requests do not in themselves grant rights to use the Jakarta EE logo. Approval that the TCK requirements have been met is a prerequisite for obtaining Jakarta EE logo usage for profile certification. The required approval processes is:

- Approval by lazy consensus after a period of two weeks (14 days)
- Approval by a majority vote of the specification project as soon as it happens.
 - The sum of the +1/-1 votes must be greater than 50% of the specification project committers.
- Approval by any top-level specification project lead at any time

All specification project members are encouraged to review the request and associated supporting materials. Reviewers of a certification request MUST carefully check the validity of all required data, in particular:

- the data is complete
- the number of tests passed is consistent with the first implementation used to validate the TCK
- TCK version and digital fingerprint match.
- test results are public and do not require special signup or viewing steps

Any committer on the specification project may vote against the certification request on the basis that the clearly defined requirements of the TCK process have not been met. This means that if there is a (-1) vote, lazy consensus is no longer an option and a majority vote MUST take place.

Accepted Certification Requests

Certification requests that are reviewed and found to meet the requirements will be marked accepted by closing an issue with an ``accepted`` label. This is a requirement for being able to request use of the associated Jakarta EE logo. A pointer/link to the issue MUST then be emailed to tck@eclipse.org to request approval to use the Jakarta EE logo.

Rejected Certification Requests

Certification requests that are reviewed and found to NOT meet the requirements will be marked as such by closing an issue with an ``invalid`` label along with the requirements that were not met. A new certification issue MUST be created with the updated requirements to attempt the certification request again.

Escalation Appeal

If there is a concern that a TCK process issue has not been resolved satisfactorily, the Eclipse Development Process [Grievance Handling](#) procedure SHOULD be followed to escalate the resolution. Note that this is not a mechanism to attempt to handle implementation specific issues.

How Tests May be Added to a TCK

The only time tests may be added to a TCK are in a major or minor release. A service release which updates the exclude list **MUST** not have test additions or changes to tests that affect behavior.