

Streamlining JCP for Java SE

Next steps

Brian Goetz, Java Language Architect

JCP Executive Committee Meeting, June 2018

ORACLE®



The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract.

It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Background

Radical shift in release cycle for Java SE

- Moved from 2-4 year feature-driven cycle to 6 month time-driven cycle
 - Keep platform vibrant
 - More frequent opportunities to deliver value to users
 - Focus on continuous innovation, not release planning
- Previously ran MR of JCP process to fit UJSR in a 6-month cycle
 - Shortened review periods, streamlined milestone deliveries
- By Sept, will have delivered two releases on this process
 - Everything has gone smoothly so far
 - But could lower overhead further

Background

Current SE Release Cycle

- Six month development window (Jan-Jun, Jul-Dec)
- At end of development window ("freeze"), fork codeline for stabilization
 - Next development window begins immediately
 - Three month stabilization period begins immediately
- Features are developed under OpenJDK "JEP" process
 - Features are started without regard for release version
 - When a feature is *done*, owner proposes to target (PTT) a specific release
 - Decision made at PTT whether to target as preview or permanent feature
 - We try to target riskier features early in the window

Background

JCP history

- JCP arose in a context when development was done in the closed
 - The world has changed since then
 - Java SE developed in the open for last ~10 years
- Significant aspects of the process are around injecting transparency
 - Early Draft Review was usually the public's first look at a feature design
- Would like to further streamline JCP process
 - Reduce process friction
 - Reflect how development is actually done today

Reducing Overhead

Milestone publication

- JCP process calls for milestone builds to be published *on the JCP site*
 - Significant manual overhead of preparing, staging, checking
 - Significant time lag to publish a milestone
 - Introduces delays in getting to next step
 - Duplication of effort for little added value
 - Milestone contents already published on OpenJDK under JCP-friendly license
- No need for having a separate publication process
 - Just link to OpenJDK spec/RI builds from JCP site

Reducing Overhead

Early Draft Review

- JCP process calls for an Early Draft Release
 - Historically, was the public's first peek at the technology
 - Now, EDRs are generally stale by the time they are published
- Reality: Discrete EDR is obsolete
 - We don't want one, we want dozens
 - And we have that now – we produce weekly builds on OpenJDK
 - We get almost no technical feedback on EDR via JCP channels
- EDR should be continuous, over the whole development window

Reducing Overhead

JSR Inception

- There is significant overhead to proposing, submitting, voting on, approving, and creating the EG for a JSR
 - Doing this every six months is getting a little silly
 - The JSR submission text changes not at all (except dates and version)
 - The EG is largely constant
 - "Another version of Java SE" shouldn't be controversial at this point...
- Can streamline by either
 - Having one long-running Platform JSR that produces multiple releases
 - Have an simpler "renewal" process for Platform JSRs

Reducing Overhead

Public Review

- Reviews in the JCP have two purposes
 - Gathering technical feedback
 - Gathering IP-related feedback
- The former was necessary when development was closed
 - OpenJDK now provides ample channels for technical feedback
 - In reality, we receive almost no technical feedback from the JCP channels, other than from the EG
- With a three-month stabilization period, running PR + PFD + FAB back-to-back starts to look a little silly
 - And has a lot of overhead
 - Eventually, we'd like to *shorten* the stabilization window

Proposals

Streamline and consolidate

- OpenJDK is the locus for technical information, discussions and feedback around Java SE
 - Builds of RI and Spec done on OpenJDK, under JCP-friendly T&C
 - Technical feedback goes through OpenJDK channels
- Streamline Platform JSR lifecycle
 - Auto-renewal, or long-running platform JSR with multiple release
- Streamline milestones
 - Replace EDR with a continuous review coinciding with dev window
 - PR is just a "promoted" EDR draft
 - Consolidate PR/PFD/FAB with a streamlined process of review, revised spec, and ballot