

JDK 7 Update Project Proposal Q & A

Why create a new Project?

The two main reasons for a new, separate JDK 7 Update Project are

1. to allow work on updates to proceed outside of a big [JDK Release Project](#), and
2. to ensure that the materials associated with the JDK 7 Release Project remain available long-term rather than get buried under JDK 7 Update materials.

Is this a JDK Release Project?

No.

The [final draft](#) of the OpenJDK Community Bylaws defines a [JDK Release Projects](#) as "New releases of Java SE Platform implementations".

Update releases are not "new releases".

Will this Project serve as the basis for Oracle JDK 7 Update releases?

Yes.

To quote from Joe Darcy's FOSDEM [blog post on OpenJDK 6](#):

In particular, there will not be the same dichotomy between the OpenJDK 7 code base and the 7 update code base as there is between OpenJDK 6 and the 6 update train.

Why start this Project now?

This is the first JDK Update release being used as the basis for Oracle JDK update releases that's being done in the open, in OpenJDK. So we would like to start the work on JDK 7 Update in OpenJDK as soon as we can, from the beginning.

Will this Project be run exactly like OpenJDK 6?

The dichotomy between OpenJDK 6 and the 6 update train allowed for some experimentation with very lightweight release management processes. Since this Project will serve as the basis for Oracle JDK 7 Update releases, it will likely need to develop more formal release processes than OpenJDK 6 did for code review, putback approval, repository management and bug management, the details of which are TBD on the Project mailing list.

Will the 7 Update Project receive security fixes?

Yes.

As with OpenJDK 6, security fixes are first kept confidential and applied to a private forest before being pushed to the public forest as part of the general synchronized publication of the fix to affected JDK release trains. In addition, they will not go through the public code review and putback approval process, and their corresponding issues will not be publicly visible.

What Mercurial repositories will be available?

We will have a mainline JDK 7 repository, that will always be open for putbacks. In addition, for each release, a repository will be cloned from the mainline JDK 7 repository, with the applicable changesets being cloned back and forth between the mainline and release repositories, until the GA date for the release.

How are fixes going to get into the repositories?

All fixes require the approval of the release manager and may require additional technical review and approval at the discretion of the release manager. The Project's moderator can appoint a release manager for one or more repositories. Fixes must go into JDK 8 first, unless they are JDK 7-specific, and be accompanied by an issue in the bug database.

How are stabilization forests managed and what is phase2?

See the [phase 2 process](#) page.

Which bug database will this Project use?

For now we'll use the [Java bug database](#). We'll eventually migrate to the new OpenJDK bug system once that's available.

Help, I can't create issues in the bug database!

If you need an issue created in order to be able to commit into a repository, let the Project's moderator know on the Project's mailing list.

How do I get a change into JDK 8?

For processes regarding JDK 8, please see the [JDK 8 Project page](#).

I have an idea for a new feature for Java! Who should I talk to?

In general, the [JDK 9 Project](#) is the right place to discuss new features.

In this Project, we're a lot more concerned with fixes for bugs, regressions, and significant issues with JDK 7 Update releases.