

Main

Welcome to OpenJDK 8 Updates!

OpenJDK 8 updates are a [separate project](#) of OpenJDK. [Andrew Haley](#) serves as the Project Lead. The list of Reviewers, Committers, and Authors can be found in the [jdk8u entry](#) of the OpenJDK Census.

Maintainers

- [Andrew Haley](#)
- [Andrew Hughes](#)
- [Severin Gehwolf](#)

General Information

As a preamble, the project lead has established general [guidelines for working on jdk8u](#) and [best practices for OpenJDK 8u backports](#).

OpenJDK 8 updates will be delivered on the same established [quarterly cycle](#) used by Oracle i.e. "the Tuesday closest to the 17th day of January, April, July and October."

Development takes place in the [jdk8u-dev](#) Mercurial repository and should be the primary place for OpenJDK committers to submit their work.

Code from the development repository is regularly tagged and promoted to the master [jdk8u](#) repository, which is used to stabilize and deliver the quarterly releases. Distributors should use this as their primary source for creating OpenJDK builds.

For further process details, you may want to continue reading [here](#).

Contributing

New fixes should first be submitted to the development repository for the current version of OpenJDK, [jdk/jdk](#), first. The vast majority of changes submitted to the OpenJDK 8 project will be *backports* from later OpenJDK versions. The version of OpenJDK closest to 8u should be used to minimise the differences between the two JDKs e.g. if 11u is still maintained and has the patch, it should be backported from that repository, rather than jdk/jdk. Occasional exceptions are made when an issue only applies to 8. In particular, the build system can be quite different from that in later versions, especially as regards HotSpot.

Everybody is encouraged to submit fixes for OpenJDK 8 updates by dropping a mail to the [jdk8u-dev mailing list](#). Established community members will help new developers without commit access in getting their patch reviewed. Should you not be willing or not be able to drive a fix into OpenJDK 8 updates, you can still suggest changes. But by only doing that, you are at the grace of the community to pick up your suggestion.

The suggested process is as follows:

1. Check the bug database for which JDK versions already have the patch applied.
2. When actively starting working on the bug, add the label 'jdk8u-<username>' to the bug, where <username> is your OpenJDK username. This makes others aware that someone is already actively backporting this bug.
3. Take a copy of the patch from the repository of the JDK version closest to 8u to minimise changes.
4. If the patch was developed after the switch to a single repository (during the OpenJDK 10 lifecycle), shuffle the paths using <source jdk>/bin/unshuffle_patch.sh -to9 <original.patch> <9.patch> to get a version that works with the forest of repositories. In some cases, this may mean multiple patches for different repositories.
5. If the patch was developed after the switch to the modular system (during the OpenJDK 9 lifecycle) and is not a HotSpot fix, shuffle the paths using <jdk9>/common/bin/unshuffle_patch.sh <repo> <9.patch> <8.patch>
6. Try to apply the patch by using hg import <patch>. If it applies, go to #7. Otherwise, #6.
7. Fix the patch so it applies. This may require identifying other patches which need to be backported first, in which case you start this process again with that fix.
8. Build the resulting JDK, altering if necessary. Again, this may end up needing dependent backports.
9. You now should have a working patch. If no changes were necessary other than path shuffling, proceed to step #11. Otherwise, step #9.
10. Use [the webrev tool](#) to create a webrev. Post a mail to jdk8u-dev@openjdk.java.net with the subject "RFR: <bug ID> <bug description>", asking for a review. Explain the motivation and what technical changes were necessary to make the patch work with 8u. Include a link to the bug and to your webrev uploaded on cr.openjdk.java.net. If you can't upload to cr.openjdk.java.net, ask someone with OpenJDK authorship status or above to do it on your behalf.
11. Update the bug with a link to the RFR on the mailing list and add the label 'jdk8u-needs-review'.
12. Wait for a successful review from an [OpenJDK 8u reviewer](#).
13. Go to the bug in JIRA and replace the 'jdk8u-needs-review' label with the 'jdk8u-fix-request' label. Write a comment preceded with "Fix Request [8u]". Explain the motivation for the fix, and either explain that it applies cleanly to 8u with patch shuffling, or link to the successful review thread on the mailing list. If you don't have bug database access, someone will need to do this on your behalf.
14. Wait for an 8u maintainer to add jdk8u-fix-yes to the bug.
15. The change can now be committed & pushed to the appropriate [jdk8u-dev repository](#). If you don't have committer or above status, someone will need to do so on your behalf. Patches that apply cleanly or only need a few minor changes which don't alter the code (e.g. copyright header fixes, same changes in a different context) should use the original author & reviewers for the commit. If the fix was reviewed, those reviewers should be appended to the end of the list. If substantial code changes were needed to create the 8u fix, authorship should go to the backporter and reviewers should only list those who reviewed the altered patch.
16. Once the patch is pushed to the repositories, remove your 'jdk8u-<username>' label from the bug.
17. Pat yourself on the back, having successfully got a patch into OpenJDK 8u.

Fix Approvals

In general, we follow the [common rules for the jdk-updates project](#).

If the backport does not apply to the 8u tree via the automated shuffling described above, it should first be submitted for review.

Push approval for a fix is then requested by setting the [jdk8u-fix-request](#) label on the original [JBS](#) bug. The maintainer will either approve this by setting [jdk8u-fix-yes](#) or reject it by setting [jdk8u-fix-no](#). Outstanding approvals can be monitored [here](#). If, and only if, the fix is approved, it may be pushed to the appropriate [jdk8u-dev](#) repositories. Approved fixes show up in this [JBS filter](#) (login required).

During the later stages of a release cycle, the release enters *rampdown*. The master jdk8u repositories contain the latest version of that release, while the jdk8u-dev repositories are used to start work on the next release. If a change needs to be pushed to a release in rampdown, push approval can still be requested using the [jdk8u-critical-request](#) label. As the name of this tag suggests, this process is intended for fixes such as major regressions that must make the release. More minor bugs and new features should go in the next release being developed in jdk8u-dev. The maintainers may approve with [jdk8u-critical-yes](#), defer to [jdk8u-dev](#) or reject altogether. Outstanding approvals for critical fixes can be monitored [here](#). If, and only if, the fix gets approved with [jdk8u-critical-yes](#), it may be pushed to the [jdk8u](#) repository. Approved critical fixes show up in this [JBS filter](#) (login required).

At the end of the month prior to the release month, the jdk8u repository is declared frozen, so embargoed security fixes can be added in private during the final few weeks. On release day, the final version will be pushed to the jdk8u repository and source bundles made available.

Repository Status

[jdk8u-dev](#): Open for development of 8u272. Pushes after [jdk8u-fix-yes](#) approval. Check [here](#) for clearance.

[jdk8u](#): Open for weekly maintainer build promotions of 8u272.

Timelines

OpenJDK 8u272

- Monday, June 1st 2020: jdk8u-dev forest open (tag: jdk8u272-b00)
- Friday, July 24th 2020: First build promotion jdk8u-devjdk8u (tag: jdk8u272-b01)
- Friday, July 31st 2020: Second build promotion jdk8u-devjdk8u (tag: jdk8u272-b02)
- Friday, August 7th 2020: Third build promotion jdk8u-dev jdk8u (tag: jdk8u272-b03)
- Friday, August 14th 2020: Fourth build promotion jdk8u-dev jdk8u (tag: jdk8u272-b04)
- Friday, August 21st 2020: Fifth build promotion jdk8u-dev jdk8u (tag: jdk8u272-b05)
- Friday, August 28th 2020: Sixth build promotion jdk8u-dev jdk8u (tag: jdk8u272-b06) & start of Rampdown phase
- Friday, September 4th 2020: First jdk8u build promotion (tag: jdk8u272-b07)
- Friday, September 11th 2020: Second jdk8u build promotion (tag: jdk8u272-b08)
- Friday, September 18th 2020: Third jdk8u build promotion (tag: jdk8u272-b09)
- Friday, September 25th 2020: Final jdk8u build promotion & tag before code freeze (tag: jdk8u272-b10)
- Tuesday, October 20th 2020: GA; OpenJDK 8u272 released (tag: jdk8u272-ga, likely to be jdk8u272-b11)

OpenJDK 8u282

- Monday, August 31st 2020: jdk8u-dev forest open (tag: jdk8u282-b00)
- Friday, October 30th 2020: First build promotion jdk8u-devjdk8u (tag: jdk8u282-b01)
- Friday, November 6th 2020: Second build promotion jdk8u-devjdk8u (tag: jdk8u282-b02)
- Friday, November 13th 2020: Third build promotion jdk8u-dev jdk8u (tag: jdk8u282-b03)
- Friday, November 20th 2020: Fourth build promotion jdk8u-dev jdk8u (tag: jdk8u282-b04)
- Friday, November 27th 2020: Fifth build promotion jdk8u-dev jdk8u (tag: jdk8u282-b05) & start of Rampdown phase
- Friday, December 4th 2020: First jdk8u build promotion (tag: jdk8u282-b06)
- Friday, December 11th 2020: Second jdk8u build promotion (tag: jdk8u282-b07)
- Friday, December 25th 2020: Third jdk8u build promotion (tag: jdk8u282-b08)
- Friday, December 25th 2020: Final jdk8u build promotion & tag before code freeze (tag: jdk8u282-b09)
- Tuesday, January 19th 2021: GA; OpenJDK 8u282 released (tag: jdk8u282-ga, likely to be jdk8u282-b10)

Older releases can be found in the [archive](#).

Releases

Latest GA release: 8u262

Latest Generally Available (GA) binary releases of the OpenJDK jdk8u project are available at: <https://adoptopenjdk.net/upstream.html?variant=openjdk8&ga=ga>

Latest Early Access (EA) binary releases of the OpenJDK jdk8u project are available at: <https://adoptopenjdk.net/upstream.html?variant=openjdk8&ga=ea>

Most recent and past release details:

- 8u262-b10 (GA), July 14th 2020 [[Release](#)] [[Tag](#)] [[Binaries](#)] [[Missing changes vs 8u262 of Oracle](#)] (JBS Login required) [[Additional changes vs 8u262 of Oracle](#)] (JBS Login required)
- 8u252-b09 (GA), April 14th 2020 [[Release](#)] [[Tag](#)] [[Binaries](#)] [[Missing changes vs 8u252 of Oracle](#)] (JBS Login required) [[Additional changes vs 8u252 of Oracle](#)] (JBS Login required)

- [8u242-b08](#) (GA), January 19th 2020 [[Release](#)] [[Tag](#)] [[Binaries](#)] [[Missing changes vs 8u242 of Oracle](#)] (JBS Login required) [[Additional changes vs 8u242 of Oracle](#)] (JBS Login required)
- [8u232-b09](#) (GA), October 15th 2019 [[Release](#)] [[Tag](#)] [[Binaries](#)] [[Missing changes vs 8u232 of Oracle](#)] (JBS Login required) [[Additional changes vs 8u232 of Oracle](#)] (JBS Login required)
- [8u222-b09](#) (GA), July 16th 2019 [[Release](#)] [[Tag](#)] [[Binaries](#)] [[Missing changes vs 8u222 of Oracle](#)] (JBS Login required) [[Additional changes vs 8u222 of Oracle](#)] (JBS Login required)
- [8u212-b03](#) (GA), April 16th 2019 [[Release](#)] [[Tag](#)] [[Binaries](#)] [[Missing changes vs 8u212 of Oracle](#)] (JBS Login required) [[Additional changes vs 8u212 of Oracle](#)] (JBS Login required)

JBS Filters

Some filters will only work for users that are logged into JBS.

Standard Fix Requests

[\[All Requests\]](#) [\[Approved requests\]](#) [\[Approved requests without push\]](#) [\[Unapproved requests\]](#)

Critical Fix Requests

[\[Critical requests\]](#) [\[Approved critical requests\]](#) [\[Approved critical requests without push\]](#) [\[Unapproved critical requests\]](#)

Filters for Release 8u272

[\[Open Downports Oracle -> OpenJDK\]](#) [\[Additional commits in OpenJDK vs Oracle\]](#)

Filters for Release 8u282

[\[Open Downports Oracle -> OpenJDK\]](#) [\[Additional commits in OpenJDK vs Oracle\]](#)

Source code

The jdk8u-dev forest for ongoing development can be cloned using this command: `hg clone http://hg.openjdk.java.net/jdk8u/jdk8u-dev; cd jdk8u-dev; sh get_source.sh`

The corresponding master forest jdk8u can be cloned using this command: `hg clone http://hg.openjdk.java.net/jdk8u/jdk8u; cd jdk8u; sh get_source.sh`

Recent space activity



Andrew Hughes

[Main](#) updated Jul 23, 2020 • [view change](#)



Severin Gehwolf

[Main](#) updated Jul 23, 2020 • [view change](#)



Christoph Langer

[Main](#) updated Jul 21, 2020 • [view change](#)



Andrew Hughes

[Archive](#) updated Jul 19, 2020 • [view change](#)

[Detailed Process Description](#) updated Jun 11, 2020 • [view change](#)

Space contributors

- [Andrew Hughes](#) (17 days ago)
- [Severin Gehwolf](#) (17 days ago)
- [Christoph Langer](#) (19 days ago)
- [Aleksey Shipilev](#) (439 days ago)
- [Andrew Haley](#) (439 days ago)
- ...