

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](#)

Releases

Latest GA release: 8u312

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:

- 8u312-b07 (GA), October 19th 2021 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 8u302-b08 (GA), July 20th 2021 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 8u292-b10 (GA), April 20th 2021 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 8u282-b08 (GA), January 19th 2021 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 8u275-b01 (GA), November 5th 2020 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 8u272-b10 (GA), October 20th 2020 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 8u265-b01 (GA), July 14th 2020 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 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)

Repository Status

[jdk8u-dev](#): Closed for rampdown of 8u322 and transition to monorepo. Once re-opened, pushes for OpenJDK 8u332 after [jdk8u-fix-yes](#) approval. Check [here](#) for clearance.

[jdk8u](#): Closed for rampdown of 8u322 and transition to monorepo. Once re-opened, pushes for OpenJDK 8u322 after [jdk8u-critical-yes](#) approval. Check [here](#) for clearance.

Timelines

Dates may be subject to change.

OpenJDK 8u322

- Thursday, September 2nd 2021: jdk8u-dev forest open (tag: jdk8u322-b00)
- Friday, October 29th 2021: First build promotion jdk8u-devjdk8u (tag: jdk8u322-b01)
- Friday, November 5th 2021: Second build promotion jdk8u-dev jdk8u (tag: jdk8u322-b02)
- Friday, November 12th 2021: Third build promotion jdk8u-dev jdk8u (tag: jdk8u322-b03)
- Friday, November 19th 2021: Build promotion skipped; no changes
- Friday, November 26th 2021: Fourth build promotion jdk8u-dev jdk8u (tag: jdk8u322-b04) & start of Rampdown phase
- Friday, December 3rd 2021: First jdk8u build promotion (tag: jdk8u322-b06)
- Friday, December 10th 2021: Second jdk8u build promotion (tag: jdk8u322-b07)
- Friday, December 17th 2021: Third jdk8u build promotion (tag: jdk8u322-b08)
- Friday, December 24th 2021: Final jdk8u build promotion (tag: jdk8u322-b09)
- Tuesday, January 18th 2022: GA; OpenJDK 8u322 released (tag: jdk8u322-ga, likely to be jdk8u322-b10)

OpenJDK 8u332

- Monday, November 29th 2021: jdk8u-dev forest open (tag: jdk8u332-b00)
- Friday, January 28th 2022: First build promotion jdk8u-devjdk8u (tag: jdk8u332-b01)
- Friday, February 4th 2022: Second build promotion jdk8u-dev jdk8u (tag: jdk8u332-b02)
- Friday, February 11th 2022: Third build promotion jdk8u-dev jdk8u (tag: jdk8u332-b03)
- Friday, February 18th 2022: Fourth build promotion jdk8u-dev jdk8u (tag: jdk8u332-b04)
- Friday, February 25th 2022: Fifth build promotion jdk8u-dev jdk8u (tag: jdk8u332-b05) & start of Rampdown phase
- Friday, March 4th 2022: First jdk8u build promotion (tag: jdk8u332-b06)
- Friday, March 11th 2022: Second jdk8u build promotion (tag: jdk8u332-b07)
- Friday, March 18th 2022: Third jdk8u build promotion (tag: jdk8u332-b08)
- Friday, March 25th 2022: Final jdk8u build promotion (tag: jdk8u332-b09)
- Tuesday, April 19th 2022: GA; OpenJDK 8u332 released (tag: jdk8u332-ga, likely to be jdk8u332-b10)

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

We are currently in the process of planning the [transition of the 8u repositories from Mercurial to git](#), via a monolithic single Mercurial repository. The preliminary schedule for this is:

1. Both 8u & 8u-dev will become monolithic Mercurial repositories some time after the rampdown of 8u322 on November 26th, 2021. Read-only git mirrors will be made available at this time. **COMPLETED**
2. The 8u-dev git repository will become live for development after the rampdown of 8u332 on February 25th, 2022. See [SKARA-1260](#).
3. The 8u git repository will receive build promotions directly after the release of 8u332 on April 19th, 2022 (first expected on April 29th, 2022). See [SKARA-1261](#).

These dates are not yet confirmed. Developers should be aware that this may lead to delays in the schedules of 8u322 & 8u332, particularly with regard to step 1, the conversion to monolithic repositories.

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 a label of the form 'jdk8u-<username>', to the bug - where <username> is your OpenJDK username - to indicate that you are creating a backport.
3. Take a copy of the patch from the repository of the JDK version closest to 8u to minimise changes.
4. 8u is now a single repository, so there should be a 1:1 relationship between patches to later JDKs and 8u patches, rather than multiple patches for individual sub-repositories. File locations may however be different on 8u, especially with changes for the modular JDK in later versions. 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>`. An updated version of this script is maintained in Andrew Hughes' [jdk9u-updated branch on github](#).
5. Try to apply the patch by using `hg import <patch>`. If it applies, go to #8. Otherwise, #7.
6. 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.
7. Build the resulting JDK and/or run any new or modified tests, altering if necessary. Again, this may end up needing dependent backports to fix issues that arise.
8. You now should have a working patch. If no changes were necessary other than path shuffling, proceed to step #13. Otherwise, step #10.
9. 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.
10. Update the bug with a link to the RFR on the mailing list and add the label 'jdk8u-needs-review'.
11. Wait for a successful review from an [OpenJDK 8u reviewer](#).
12. 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. In the case that the bug is inaccessible (the page displays "You can't view this issue"), please make the request by e-mail to jdk8u-dev@openjdk.java.net with the subject "RFA: <bug ID> <bug description>" and the same motivation in the body of the e-mail as would have been included in the request comment. It may also be worth replying to the original review thread, asking for the bug to be made public.

13. Wait for an 8u maintainer to add `jdk8u-fix-yes` to the bug.
14. The change can now be committed & pushed to the [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.
15. Pat yourself on the back, having successfully got a patch into OpenJDK 8u.

Backport bugs will be automatically created on push by the hgupdater system. If a backport bug needs to be explicitly created - for example, for a Compatibility and Specification Review (CSR) - then please apply labels to that bug to avoid the need to work on two different bugs for the one issue. The fix version should be set to 'openjdk8ux' where x is the current version of 8u being developed. Please avoid using 'openjdk8u' as the hgupdater will not resolve such bugs automatically. Maintainers should double-check this fix version is correct when approving.

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.

JBS Filters

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

Backports Needing Review

[\[All Requests\]](#)

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 8u302

[\[Open Downports Oracle -> OpenJDK\]](#)

Filters for Release 8u292

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

Filters for Release 8u282

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

Filters for Release 8u272

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

Source code

The jdk8u-dev tree for ongoing development can be cloned using this command: `hg clone http://hg.openjdk.java.net/jdk8u/monojdk8u-dev`

The corresponding master tree jdk8u can be cloned using this command: `hg clone http://hg.openjdk.java.net/jdk8u/monojdk8u`

If you just want a read-only copy of the sources, you may also use git: `git clone https://github.com/openjdk/jdk8u`

Recent space activity



Andrew Hughes

[Main](#) updated Dec 01, 2021 • [view change](#)

[Archive](#) updated Nov 02, 2021 • [view change](#)



Severin Gehwolf

[Main](#) updated Jul 22, 2021 • [view change](#)

[Archive](#) updated Jan 26, 2021 • [view change](#)



Andrew Hughes

[Detailed Process Description](#) updated Sep 15, 2020 • [view change](#)

Space contributors

- [Andrew Hughes](#) (2 days ago)
- [Severin Gehwolf](#) (29 days ago)
- [Christoph Langer](#) (501 days ago)
- [Aleksey Shipilev](#) (920 days ago)
- [Andrew Haley](#) (921 days ago)
- ...