

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: 8u432

Latest Generally Available (GA) binary releases of the OpenJDK jdk8u project are available at: <https://adoptium.net/temurin/releases/?version=8>

Latest Early Access (EA) binary releases of the OpenJDK jdk8u project are available at: <https://adoptium.net/temurin/nightly/?version=8>

Most recent and past release details:

- 8u432-b06 (GA), October 15th 2024 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 8u422-b05 (GA), July 16th 2024 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 8u412-b08 (GA), April 16th 2024 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 8u402-b06 (GA), January 16th 2024 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 8u392-b08 (GA), October 17th 2023 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 8u382-b05 (GA), July 18th 2023 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 8u372-b07 (GA), April 18th 2023 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 8u362-b09 (GA), January 17th 2023 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 8u352-b08 (GA), October 18th 2022 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 8u345-b01 (GA), August 2nd 2022 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 8u342-b07 (GA), July 19th 2022 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 8u332-b09 (GA), April 22th 2022 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 8u322-b06 (GA), January 30th 2022 [[Release](#)] [[Tag](#)] [[Binaries](#)]
- 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 27th 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](#) (git): Pushes for OpenJDK 8u442 after [jdk8u-fix-yes](#) approval. Check [here](#) for clearance.

[jdk8u](#) (git): Weekly build promotions for OpenJDK 8u442.

Timelines

Dates may be subject to change

OpenJDK 8u442

- Wednesday, September 4th 2024 : jdk8u-dev forest [open](#) (tag: [jdk8u442-b00](#))
- Friday, October 25th 2024: First build promotion jdk8u-devjdk8u (tag: [jdk8u442-b01](#))
- Friday, November 1st 2024: Second build promotion jdk8u-devjdk8u (tag: [jdk8u442-b02](#))
- Friday, November 8th 2024: Third build promotion jdk8u-devjdk8u (tag: [jdk8u442-b03](#))

- Friday, November 15th 2024: Fourth build promotion jdk8u-dev jdk8u (tag: jdk8u442-b04)
- Friday, November 22nd 2024: Fifth build promotion jdk8u-dev jdk8u (tag: jdk8u442-b05)
- Friday, November 29th 2024: Sixth build promotion jdk8u-dev jdk8u (tag: jdk8u442-b06) & start of Rampdown phase
- Friday, December 6th 2024: First jdk8u build promotion (tag: jdk8u442-b07)
- Friday, December 13th 2024: Second jdk8u build promotion (tag: jdk8u442-b08)
- Friday, December 20th 2024: Third jdk8u build promotion (tag: jdk8u442-b09)
- Friday, December 27th 2024: Fourth and final jdk8u build promotion (tag: jdk8u442-b10)
- Tuesday, January 21st 2025: GA; OpenJDK 8u442 released (tag: jdk8u442-ga, likely to be jdk8u442-b11)

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

The [transition of the 8u repositories from Mercurial to git](#) was completed after the April 2022 release, on Monday, April 25th.

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."

Announcements and discussion take place on the [jdk8u-dev mailing list](#). Development takes place in the [jdk8u-dev](#) git repository and should be the primary place for OpenJDK 8u committers to submit their work.

Code from the development repository is regularly tagged and promoted to the main [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 using the [SKARA](#) processes now common to all OpenJDK projects. 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. However, file locations are highly likely to be different on 8u, especially with changes for the modular JDK in later versions, so automated [backports via SKARA](#) won't work. 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 `git cherry-pick` as documented in the [Skara backporting process](#). 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. Create a pull request on GitHub as described in the [Skara documentation](#). The pull request should use the title "Backport <x>" where <x> is the hash of the original commit. This ensures that the PR is correctly recognised by Skara as a backport and bugs updated appropriately.
9. Wait for testing to complete and a successful review from an [OpenJDK 8u reviewer](#).
10. Request for approval of the bug to go into OpenJDK 8u. You can do this either by using the Skara `/approval` command for which you don't need to have access to JBS, or go to the bug in JIRA and add the 'jdk8u-fix-request' label. In either case, be sure to include proper reasons as to why you think that fix needs to get into OpenJDK 8u. Do this either by writing a comment preceded with "Fix Request [8u]" or using the `/approval <comment>` approach. Explain the motivation for the fix, and either explain that it applies cleanly to 8u with patch shuffling, or refer to the GitHub review process.
11. Wait for an 8u maintainer to add jdk8u-fix-yes to the bug.
12. The change can now be pushed to the [jdk8u-dev repository](#) using the Skara `/integrate` command. If you are not yet an [OpenJDK 8u Committer](#), you may need to wait for someone who is to issue the `/sponsor` command.
13. Pat yourself on the back, having successfully got a patch into OpenJDK 8u.

Backport bugs will be automatically created on push. 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 such bugs will not be resolved 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 yesterday at 1:50 AM [view change](#)

Archive updated Jul 30, 2024 [view change](#)

Space contributors

- [Andrew Hughes](#) (1 day ago)
- [Severin Gehwolf](#) (278 days ago)
- [Christoph Langer](#) (1573 days ago)
- [Aleksey Shipilev](#) (1993 days ago)
- [Andrew Haley](#) (1993 days ago)
- ...



Severin Gehwolf

Main updated Feb 06, 2024 [view change](#)



Andrew Hughes

Detailed Process Description updated Mar 05, 2022 [view change](#)



Severin Gehwolf

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