

JDK 21u

Welcome to OpenJDK 21 Updates!

OpenJDK 21 updates are part of the [JDK updates project](#) of the OpenJDK. [Rob McKenna](#) serves as the Project Lead. The list of Reviewers, Committers, and Authors can be found in the [jdk updates](#) section of the OpenJDK Census.

Maintainers

- [Severin Gehwolf](#) (Lead Maintainer)
- [Goetz Lindenmaier](#)
- [Andrew Haley](#)
- [Andrew John Hughes](#)
- [Christoph Langer](#)

General Information

OpenJDK 21 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 [jdk21u-dev](#) Git 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 [jdk21u](#) 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 might want to continue reading [here](#).

Contributing

New fixes should first be submitted to the development repository for the current version of OpenJDK, [jdk/jdk](#). Most changes submitted to the OpenJDK 21 project will be *backports* from this repository. Exceptions are made if an issue only applies to 21.

Everybody is encouraged to submit fixes for OpenJDK 21 by creating a pull request to the [jdk21u-dev](#) Git repository. Established community members will help new developers without commit access in getting their patch reviewed. For details on the process involved, continue reading these [backport instructions](#).

Should you not be willing or not be able to drive a fix into JDK 21 updates, you can still suggest changes by dropping a mail to the [jdk-updates-dev mailing list](#). But by only doing that, you are at the grace of the community to pick up your suggestion.

Fix Approvals

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

In addition to this list, we will consider

- Build system improvements, particularly for stability and reproducibility
- Changes necessary to make 21u work on new operating systems and hardware

However, if there is an enhancement that provides substantial benefit to OpenJDK 21u users and it is of low risk, we will consider it. Please ask on the [jdk-updates mailing list](#) mailing list if you are in any doubt about whether your proposed backport might qualify.

If the backport requires more than just cosmetic changes (file location changes, copyright header updates) to apply to the 21u repository, it should first be submitted for review.

Push approval for a fix is requested by setting the [jdk21u-fix-request](#) label on the original [JBS](#) bug. The maintainers will either approve by setting [jdk21u-fix-yes](#) or reject by setting [jdk21u-fix-no](#). Outstanding approvals can be monitored here. If and only if the fix gets approved, it may be pushed to the [jdk21u-dev](#) repository. Approved fixes show up in this [JBS filter](#) (login required).

During the later stages of a release cycle, the release enters *rampdown*. The master [jdk21u](#) repository contains the latest version of that release, while the [jdk21u-dev](#) repository is 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 [jdk21u-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. Minor bug fixes and new features should go in the next release being developed in [jdk21u-dev](#). The maintainers may approve with [jdk21u-critical-yes](#), defer to [jdk21u-dev](#) or reject altogether. Outstanding approvals for critical fixes can be monitored here. If, and only if, the fix gets approved with [jdk21u-critical-yes](#), it may be pushed to the [jdk21u](#) repository. Approved critical fixes show up in this [JBS filter](#) (login required).

At the end of the month prior to the release month, the [jdk21u](#) 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 [jdk21u](#) repository and source bundles made available.

Repositories

[jdk21u-dev](#): Always open repository for development of upcoming JDK 21 Update releases. Pushes after [jdk21u-fix-yes](#) approval. Check [here](#) for clearance.

[jdk21u](#): Repository used for Rampdown of JDK 21 Update releases. Pushes only during 4-weeks rampdown period after [jdk21u-critical-yes](#) approval. Check here for clearance.

Timelines

The release date follows the schedule of [Oracle Critical Patch Updates](#) as this is the date embargoed security fixes can be published.

OpenJDK 21.0.3

- Tuesday, April 16 2024 GA; OpenJDK 21.0.3 released (tag: jdk-21.0.3-ga)

OpenJDK 21.0.4

- Wednesday, February 28 2024: jdk21u-dev repo open (tag: 21.0.4+0)
- Tuesday, April 30 2024: First merge from jdk21u-dev to jdk21u (tag: 21.0.4+1)
- Tuesday, May 28 2024: Rampdown; last merge from jdk21u-dev to jdk21u
- Tuesday, June 25 2024: Last tag before code freeze
- Tuesday, July 16 2024 GA; OpenJDK 21.0.4 released (tag: jdk-21.0.4-ga)

OpenJDK 21.0.5

- Wednesday, May 29 2024: jdk21u-dev repo open (tag: 21.0.5+0)
- Tuesday, July 30 2024: First merge from jdk21u-dev to jdk21u (tag: 21.0.5+1)
- Tuesday, August 27 2024: Rampdown; last merge from jdk21u-dev to jdk21u
- Tuesday, September 24 2024: Last tag before code freeze
- Tuesday, October 15 2024 GA; OpenJDK 21.0.5 released (tag: jdk-21.0.5-ga)

Latest Release

Latest GA release: 21.0.3

- 21.0.3+9 (GA), April 16th 2024, [\[Release\]](#) [\[Tag\]](#) [\[Binaries\]](#)

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

Downloads

Latest Generally Available (GA) binary releases of the jdk21u stream of the OpenJDK jdk-updates project are available at: <https://adoptium.net/temurin/releases/?version=21>

Latest Early Access (EA) binary releases of the jdk21u stream of the OpenJDK jdk-updates project are available at: <https://adoptium.net/temurin/nightly/?version=21>

JBS Filters

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

Standard Fix Requests

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

Critical Fix Requests

[\[Unapproved requests\]](#)

Filters for Release 21.0.3

[21.0.3/all: OpenJDK but not OracleJDK](#): Changes in OpenJDK 21.0.3 that are not in OracleJDK 21.0.3.

[21.0.3/new: OracleJDK but not OpenJDK](#): Changes new in OracleJDK 21.0.3 but not in OpenJDK 21.0.3

Filters for Release 21.0.4

[21.0.4/all: OpenJDK but not OracleJDK](#): Changes in OpenJDK 21.0.4 that are not in OracleJDK 21.0.4.

[21.0.4/all: OracleJDK but not OpenJDK](#): Changes backported to OracleJDK 21.0.4-21.0.3 but not in OpenJDK 21.0.4.

[21.0.4/new: OracleJDK but not OpenJDK](#): Changes new in OracleJDK 21.0.4 but not in OpenJDK 21.0.4

Filters for Release 21.0.5

[21.0.5/all: OpenJDK but not OracleJDK](#): Changes in OpenJDK 21.0.5 that are not in OracleJDK 21.0.5.

[21.0.5/all: OracleJDK but not OpenJDK](#): Changes backported to OracleJDK 21.0.3-21.0.5 but not in OpenJDK 21.0.5.

[21.0.5/new: OracleJDK but not OpenJDK](#): Changes new in OracleJDK 21.0.5 but not in OpenJDK 21.0.5

Common jdk21u JBS labels

The above filters exclude issues that are marked with the following labels:

- [jdk21u-na](#) Change does not apply to open jdk21u. In general, this is a change done by Oracle for their build or test infrastructure, for their own delivery or in their tests not open sourced. Any other change might not be deemed appropriate for OpenJDK 21u and get labelled that way.
- [jdk21u-defer-next](#) Change is generally acceptable, but should go into the next JDK 21 update release. I.e. 21.0.4 is currently collecting fixes in jdk21u-dev, but the proposed change should get into 21.0.5 (the next release) instead.