

Guava Release 13.0: Release Notes

Release 13.0 was released on August 3, 2012.

Release 13.0.1 was released on August 28, 2012.

(See [ReleaseHistory](#).)

[Full API Documentation](#)

Using Guava in your project

This release will be identified in the Maven Central repository as `com.google.guava:guava:13.0.1` and `com.google.guava:guava-gwt:13.0.1`

See [UseGuavaInYourBuild](#) for help integrating Guava into your build environment.

If you don't use managed dependencies, you can also just manually download JARs of the classes, sources and documentation from:

- `guava-13.0.1.jar`
- `guava-gwt-13.0.1.jar` (for GWT users)
- `guava-13.0.1-javadoc.jar` (Javadoc)
- `guava-13.0.1-sources.jar` (Source)

Note: Release 13 removes transitivity of the JSR-305 artifact (commonly used for `@Nullable`). Please provide your own direct dependency on this if you declare things in your own code `@Nullable`. More detail below.

Issues resolved

26 issues are resolved in this release.

Changes between 13.0 and 13.0.1

Fixed `guava-gwt` artifact.

API Changes

Full JDiff Report of changes since release 12.0

To build a combined report of the API changes between release 13.0 and any older release, check out our docs tree and run `jdifff/jdifff.sh` with the previous release number as argument (example: `jdifff.sh 5.0`).

Significant API additions

Added `RateLimiter`.

Added `Striped`.

Added `FluentIterable.toSortedImmutableList` and `transformAndConcat`.
Added `ContiguousSet.create(Range, DiscreteDomain)`.
Added `Maps.synchronizedNavigableMap`.
Added `Sets.synchronizedNavigableSet`.
Added `Ordering.allEqual`
Added `Funnels.asOutputStream`, `integerFunnel` and `longFunnel`.
Added `DoubleMath.fuzzyCompare` and `fuzzyEquals`.
Added a few more constants to `MediaType`.
Added `UnsignedBytes.parseUnsignedByte`, `toString` and `MAX_VALUE`.
Added `UnsignedInts.decode`.
Added `UnsignedLongs.decode`.
Added `CycleDetectingLockFactory`.
Added `Service.Listener` and `addListener`.
Added `Futures.dereference`.
Added `ImmutableTypeToInstanceMap` and `MutableTypeToInstanceMap`

Significant API changes

`Iterators.skip` was deprecated in favor of `Iterators.advance`
The static methods on `Equivalences` were deprecated in favor of instance methods on `Equivalence`.
Removed `Files.getDigest` and `ByteStreams.getDigest` in favor of `Files.hash` and `ByteStreams.hash`.

Promoted from @Beta

`SortedMap` overloads for `Maps.filter{Keys,Values,Entries}`
`SortedSet` overloads for `Sets.filter`
`Ascii.MIN` and `Ascii.MAX`
`Equivalence`, `EquivalenceWrapper`
`Joiner.appendTo` overloads for `Iterator`
`Optional` (though `or(Optional)`, `or(Supplier)`, `transform(Function)`, and `presentInstances(Iterable<Optional>)` are still `@Beta`)
`Strings.common{Prefix, Suffix}`

`Lists.newCopyOnWriteArrayList`

`Sets.newCopyOnWriteArraySet`

`Multimaps.filter{Keys, Values, Entries}`

`Multisets.{contains, remove, retain}Occurrences`

`Tables` (though `newCustomTable`, `transformValues`, and `unmodifiableRowSortedTable` are still `@Beta`)

Non-API changes

Made `findbugs` a provided dependency to avert dep conflicts when using `findbugs` 2.0. The side-effect of this change is that projects which relied upon Guava to grant access to the JSR-305 annotations “for free” will break unless they provide their own direct dependency on that jar (or an equivalent). Projects should always have been directly depending on JSR-305 (per maven best-practice), but this change makes that should into a must.

Caliper benchmarks for much of Guava have been open-sourced.

`LinkedHashMultimap` has been entirely rewritten, reducing its memory consumption by ~50%. (Previously, it was one of the worst potential memory hogs in the library.)

`ImmutableSortedMultiset` has been rewritten, reducing its memory consumption by ~40%.

`ImmutableSortedMap` has been rewritten, reducing its memory usage by ~50%.

Fixed a significant performance regression in `LocalCache` (12.0 was affected, but 12.0.1 was not)