Guava Release 17.0: Release Notes

- Release 17.0 was released on April 22, 2014.
- Release 17.0-rc2 was released on April 10, 2014.
- Release 17.0-rc1 was released on April 8, 2014.

(See [[ReleaseHistory]].)

Full API Documentation

Using Guava in your project

	Guava	Guava (GWT)	Guava (JDK5 Backport)
Maven Identifier	com.google.guava:guava:17.0	com.google.guava:guava-gwt:17.0	com.google.guava:guava-jdk5:17.0
Jar	guava-17.0.jar	guava-gwt-17.0.jar	guava-jdk5-17.0.jar
Javadoc	guava-17.0-javadoc.jar	guava-gwt-17.0-javadoc.jar	guava-jdk5-17.0-javadoc.jar
Sources	guava-17.0-sources.jar	guava-gwt-17.0-sources.jar	guava-jdk5-17.0-sources.jar

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

Issues resolved

11 issues are resolved in this release.

API Changes

Full JDiff Report of changes since release 16.0.

To build a combined report of the API changes between release 17.0 and any older release, check out our docs tree and run `jdiff/jdiff.sh` with the previous release number as argument (example: `jdiff.sh 5.0`).

Significant API additions and changes

common.base

`Verify` and `VerifyException`

`Converter.from(Function<A, B>, Function<B, A>)`

common.cache

`CacheLoader.asyncReloading(CacheLoader<K, V>, Executor)`

common.io

`ByteStreams.newDataInput(ByteArrayInputStream)`

`ByteStreams.newDataOutput(ByteArrayOutputStream)`

`Closeables.closeQuietly(InputStream)`

`Closeables.closeQuietly(Reader)`

common.net

`HostAndPort.fromHost(String)`

common.util.concurrent

`Futures.inCompletionOrder(Iterable<ListenableFuture<T>>)`

`MoreExecutors.shutdownAndAwaitTermination(ExecutorService, long, TimeUnit)`

`Service` (and subclasses) - deprecated methods removed.

A note on `BloomFilter`

Release 17 fixes an issue (1119) with the performance of very large `BloomFilter`s. For most users, this fix should be completely transparent. `BloomFilter` objects serialized with a previous version of Guava will be deserializable by and work fine in Guava 17. However, `BloomFilter`s created by Guava 17 will *not* be deserializable by previous versions of Guava. This should still only affect you if both of the following are true:

- You are serializing `BloomFilter`s and sending them from one server or process to another.
- You can't upgrade all your servers to Guava 17 at the same time.

In this case, a server that's been upgraded to 17 could send a `BloomFilter` to a server that hasn't, which will then fail to deserialize it.

For this release only, we're providing the ability to use a system property to work around this issue. If the system property `com.google.common.hash.BloomFilter.useMitz32` is set to `true` (ignoring case), Guava will create `BloomFilter`s that are compatible with previous versions of Guava rather than using the new strategy. So while you're rolling out Guava 17, you can set this system property to keep everything working. Once it's fully rolled out, you can remove the system property to start using the new strategy. Guava 18.0 will no longer recognize the system property and will always use the new strategy for newly created `BloomFilter`s.