JEP-14: Tip & Tail

Feedback from Gradle

Gradle

Oleg Nenashev
BT DevRel & Education Team

Disclaimers

- This slidedeck does not represent the official feedback/opinion of Gradle Inc. as a company/vendor
- The feedback is collected from Gradle Build Tool engineers and contributors, then compiled

> The information contained in this presentation and the roadmap (and the "Roadmap") is intended to outline our general product direction, which is subject to change at any time. The content provided in this Roadmap is provided for informational purposes only and is not a commitment, promise or legal obligation to deliver any material, code, or functionality and should not be relied upon in making purchasing or other decisions. The development, release and timing of any features or functionality described in this Roadmap remains at the sole discretion of Gradle, Inc. Product capabilities, timeframes and features are subject to change and should not be viewed as commitments.

Disclaimers

Gradle BUILD TOOL

// f.k.a. and a.k.a. Gradle

DEVELOCITY

// f.k.a. Gradle Enterprise

Developer Productivity Engineering (DPE)

DEVELOCITY

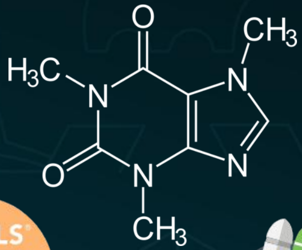
For many FOSS projects: we sponsor

Develocity licenses / SaaS hosting

Help with optimization, O11y, platform/dependencies

Maven, Bazel and sbt users welcome!

gradle/develocity-oss-projects



gradle/develocity-oss-projects

About Gradle Build Tool (fka/aka Gradle)

- One of the most popular build tools in the JVM ecosystem
- We build with Java, we run with Java
- Support for projects beyond JVM - Android, Swift, C/C++, etc.
- Member of OpenJDK Quality outreach

https://gradle.org/

Java - Our Tips

- Toolchain and Runtime isolation
- Toolchain is a priority
- We adopt new Java quickly
- We normally start with Java support discovery before the GA

https://docs.gradle.org/current/userguide/compatibility.html

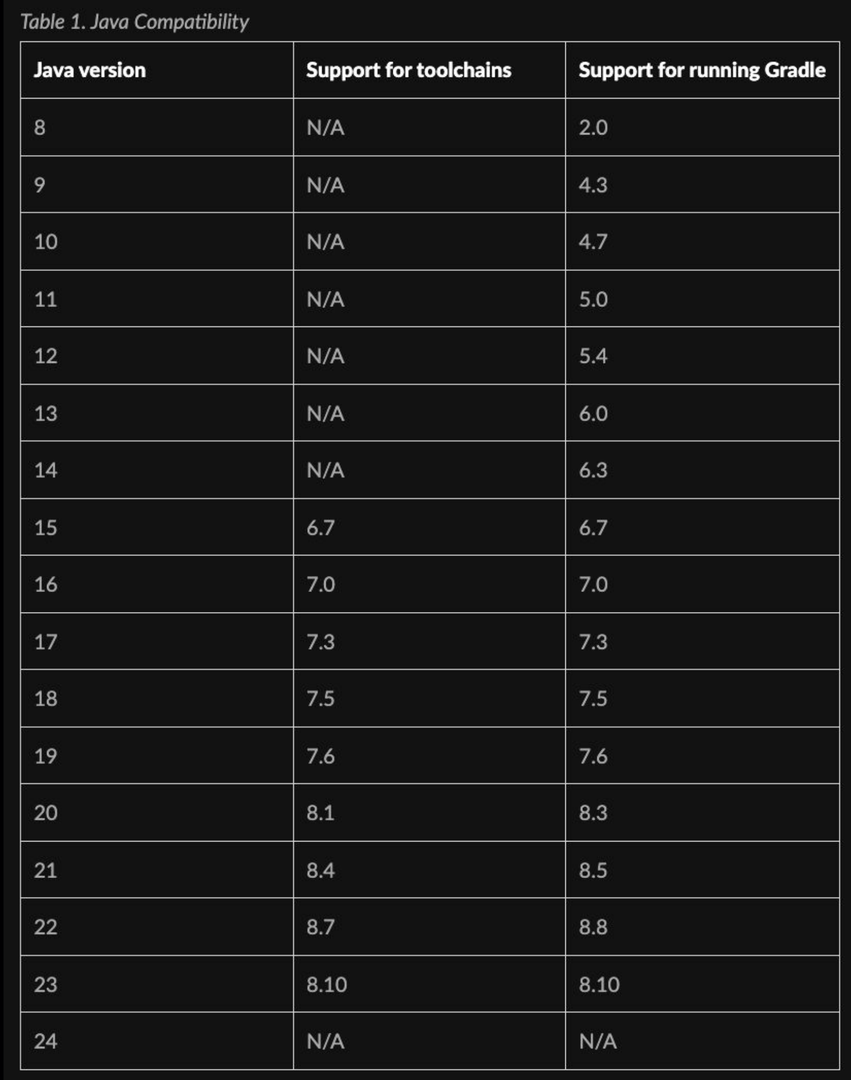


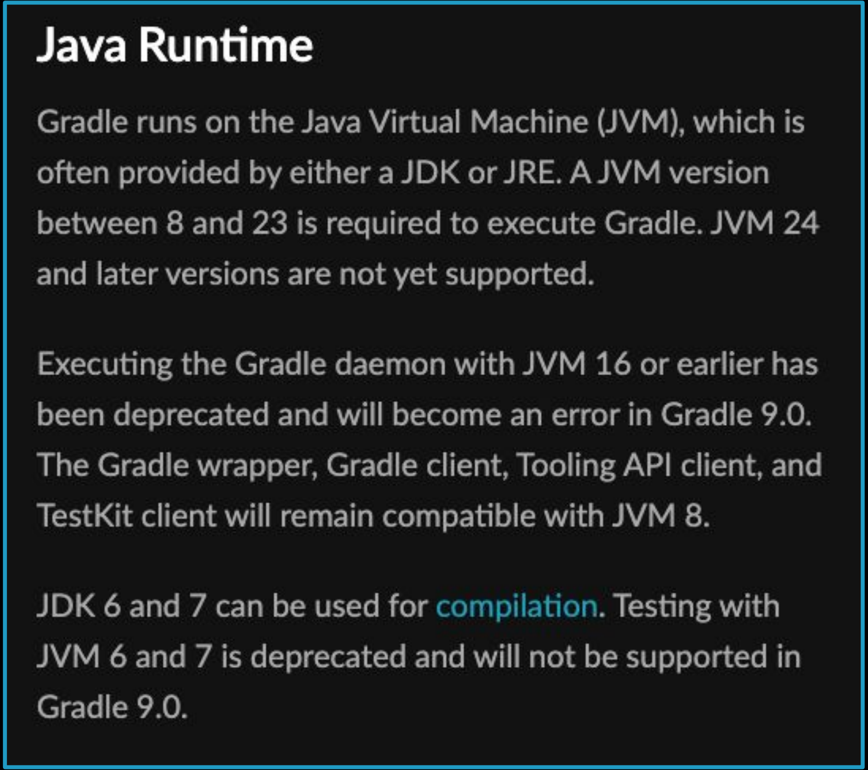
Table 1. Java Compatibility

Java version	Support for toolchains	Support for running Gradle
8	N/A	2.0
9	N/A	4.3
10	N/A	4.7
11	N/A	5.0
12	N/A	5.4
13	N/A	6.0
14	N/A	6.3
15	6.7	6.7
16	7.0	7.0
17	7.3	7.3
18	7.5	7.5
19	7.6	7.6
20	8.1	8.3
21	8.4	8.5
22	8.7	8.8
23	8.10	8.10
24	N/A	N/A

Java - Our Tails

- We still support Java 1.8+
- Gradle 9.0 plans to raise min required runtime version to 17
- Java 21 as minimum is TBD

https://docs.gradle.org/current/userguide/compatibility.html



Java Runtime

Gradle runs on the Java Virtual Machine (JVM), which is often provided by either a JDK or JRE. A JVM version between 8 and 23 is required to execute Gradle. JVM 24 and later versions are not yet supported.

Executing the Gradle daemon with JVM 16 or earlier has been deprecated and will become an error in Gradle 9.0. The Gradle wrapper, Gradle client, Tooling API client, and TestKit client will remain compatible with JVM 8.

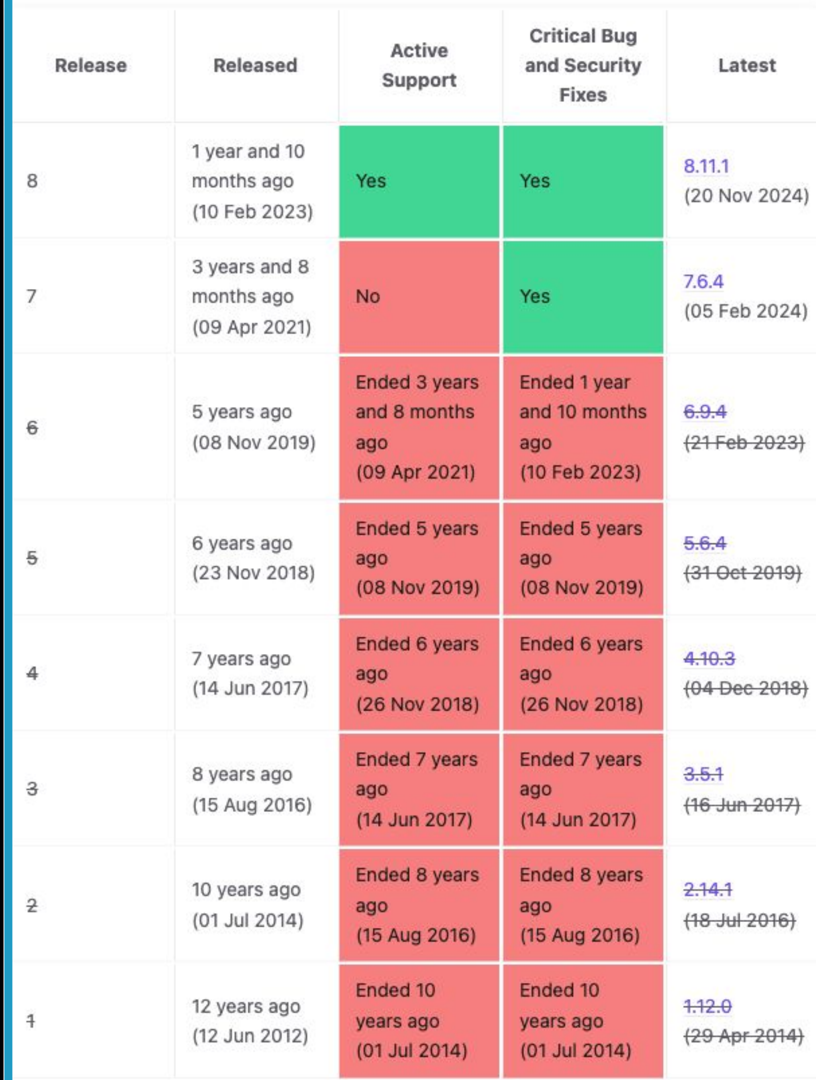
JDK 6 and 7 can be used for compilation. Testing with JVM 6 and 7 is deprecated and will not be supported in Gradle 9.0.

Gradle

newsletter.gradle.org/2024/10#gradle-9-roadmap-updates

Gradle Build Tool - Release Model

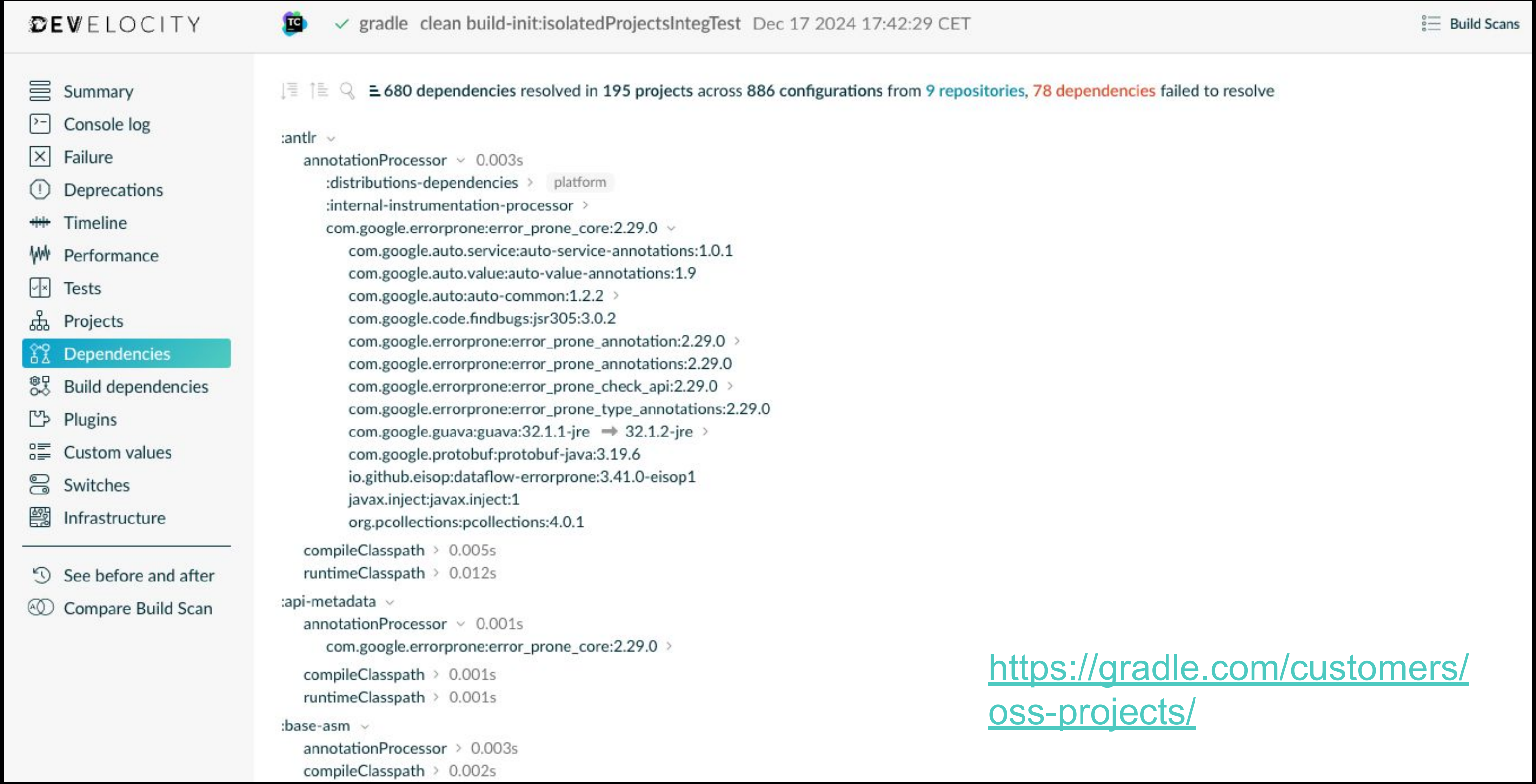
- Gradle itself is Tip&Tails
- Single Tip release
- Single tail: porting of critical bug fixes and security fixes only to N-1.x
- docs.gradle.org/current/userguide/feature_lifecycle.html#eol_support



Release	Released	Active Support	Critical Bug and Security Fixes	Latest
8	1 year and 10 months ago (10 Feb 2023)	Yes	Yes	8.11.1 (20 Nov 2024)
7	3 years and 8 months ago (09 Apr 2021)	No	Yes	7.6.4 (05 Feb 2024)
~~6~~	5 years ago (08 Nov 2019)	Ended 3 years and 8 months ago (09 Apr 2021)	Ended 1 year and 10 months ago (10 Feb 2023)	~~6.9.4~~ ~~(21 Feb 2023)~~
~~5~~	6 years ago (23 Nov 2018)	Ended 5 years ago (08 Nov 2019)	Ended 5 years ago (08 Nov 2019)	~~5.6.4~~ ~~(31 Oct 2019)~~
~~4~~	7 years ago (14 Jun 2017)	Ended 6 years ago (26 Nov 2018)	Ended 6 years ago (26 Nov 2018)	~~4.10.3~~ ~~(04 Dec 2018)~~
~~3~~	8 years ago (15 Aug 2016)	Ended 7 years ago (14 Jun 2017)	Ended 7 years ago (14 Jun 2017)	~~3.5.1~~ ~~(16 Jun 2017)~~
~~2~~	10 years ago (01 Jul 2014)	Ended 8 years ago (15 Aug 2016)	Ended 8 years ago (15 Aug 2016)	~~2.14.1~~ ~~(18 Jul 2016)~~
~~1~~	12 years ago (12 Jun 2012)	Ended 10 years ago (01 Jul 2014)	Ended 10 years ago (01 Jul 2014)	~~1.12.0~~ ~~(29 Apr 2014)~~

Ref: https://endoflife.date/gradle

Our Dependencies



DEVELOCITY   ✓ gradle clean build-init:isolatedProjectsIntegTest Dec 17 2024 17:42:29 CET   Build Scans

- Summary
- Console log
- Failure
- Deprecations
- Timeline
- Performance
- Tests
- Projects
- Dependencies
- Build dependencies
- Plugins
- Custom values
- Switches
- Infrastructure

- See before and after
- Compare Build Scan

680 dependencies resolved in 195 projects across 886 configurations from 9 repositories, 78 dependencies failed to resolve

```
:antlr
    annotationProcessor  0.003s
        :distributions-dependencies  platform
        :internal-instrumentation-processor
        com.google.errorprone:error_prone_core:2.29.0
            com.google.auto.service:auto-service-annotations:1.0.1
            com.google.auto.value:auto-value-annotations:1.9
            com.google.auto:auto-common:1.2.2
            com.google.code.findbugs:jsr305:3.0.2
            com.google.errorprone:error_prone_annotation:2.29.0
            com.google.errorprone:error_prone_annotations:2.29.0
            com.google.errorprone:error_prone_check_api:2.29.0
            com.google.errorprone:error_prone_type_annotations:2.29.0
            com.google.guava:guava:32.1.1-jre ➡ 32.1.2-jre
            com.google.protobuf:protobuf-java:3.19.6
            io.github.eisop:dataflow-errorprone:3.41.0-eisop1
            javax.inject:javax.inject:1
            org.pcollections:pcollections:4.0.1
    compileClasspath  0.005s
    runtimeClasspath  0.012s
:api-metadata
    annotationProcessor  0.001s
        com.google.errorprone:error_prone_core:2.29.0
    compileClasspath  0.001s
    runtimeClasspath  0.001s
:base-asm
    annotationProcessor  0.003s
    compileClasspath  0.002s
```

https://gradle.com/customers/oss-projects/

Gradle

Our Upstream Dependencies are not Tip & Tail (and that's fine?)

- The approach is not widely adopted by our dependency tree
- Updating to Tip libraries in Gradle is a big investment. We usually use the Tail
- Build-facing dependencies are a problem for Gradle users (e.g. Kotlin DSL, Groovy)
- Java-dependent libraries (e.g. ASM)

Gradle

Our Downstream Dependencies

- Upgrading Gradle is complicated. We are working on it ©
- Partnerships with key vendors, e.g. Android, JetBrains, GitHub, etc.
- Projects like Gradle Build Server for better isolation

Gradle

Gradle Plugins

- We have no strict policy on Gradle plugin versioning schemas
- Semver is most popular
- Gradle plugins usually follow the Tip, but no strong policy
- We may recommend the Tip&Tail approach in the new developer guide

Developer Guide:
https://docs.gradle.org/current/userguide/implementing_gradle_plugins_binary.html

Gradle

The End?

Gradle

TL;DR: Feedback to JCP (Unofficial)

- We see a big Java evolution speed-up in Java 11 and beyond
- Supporting faster evolution of Java is great!
- Tip & Tail is a reasonable approach
- Dependencies are a bigger problem
- More ecosystem alignment might be needed

References - Gradle

- Java compatibility matrix: https://docs.gradle.org/current/userguide/compatibility.html
- Public roadmap: roadmap.gradle.org
- Newsletter with key updates: newsletter.gradle.org

Gradle

References - Java JEP-14

- https://openjdk.org/jeps/14 and references
- 101 Explainer by Nicolai Parlog - https://www.youtube.com/watch?v=ozUE4YN_WhI
- Tip & Tail for Library Maintainers - Brian Goetz and Georges Saab at Devoxx Belgium 2024: https://devoxx.be/talk/tip-and-tail-for-library-maintainers/

Gradle

Open Questions?

Gradle Community Slack, #roadmap