Gradle Release Notes

### Version 4.10.1

The Gradle team is pleased to announce Gradle 4.10. This is a big release.

First and foremost, this release of Gradle features an improved [incremental Java compiler, now enabled by default](#gjdgxs). This will result in significantly reduced Java compilation time in subsequent builds when outputs are not up-to-date or resolved from the build cache.

Chances are caches in those .gradle/ directories have accumulated a few (or a few dozen) gigabytes over time. If so, you'll be relieved to know that Gradle will now [periodically clean up unused /caches](#30j0zll) under GRADLE\_USER\_HOME and project root directories.

A moment you have anticipated is nearly here, as the [Kotlin DSL reaches version 1.0 RC6](https://github.com/gradle/kotlin-dsl/releases/tag/v1.0-RC6). Configuration avoidance, buildSrc refactoring propagation to the IDE, and lots of DSL polish make this the release to try. Gradle Kotlin DSL 1.0 will ship with the next version of Gradle, 5.0. Read [this blog post](https://blog.gradle.org/gradle-kotlin-dsl-release-candidate) for guidance on trying the Kotlin DSL and submitting feedback.

You can now use SNAPSHOT plugin versions with the plugins {} and pluginManagement {} blocks. This is especially good news for Kotlin DSL users, who will get code assistance and auto-completion for these SNAPSHOT plugins. Special thanks to [Sébastien Deleuze](https://github.com/sdeleuze) for contributing.

Last but not least, [included builds can now be nested](#1fob9te). This makes some common workflows more convenient, such as working on multiple source repositories at the same time to implement a cross-cutting feature.

We hope you will build happiness with Gradle 4.10, and we look forward to your feedback [via Twitter](https://twitter.com/gradle) or [on GitHub](https://github.com/gradle/gradle).

## Table Of Contents

* [Upgrade Instructions](#3znysh7)
* [New and noteworthy](#2et92p0)
  + [Incremental Java compilation by default](#gjdgxs)
  + [Periodic cache cleanup](#30j0zll)
  + [Kotlin DSL 1.0 RC](#tyjcwt)
  + Use SNAPSHOT plugin versions with the plugins {} block
  + [Nested included builds](#1fob9te)
  + [Authorization for Maven repositories with custom HTTP headers](#3dy6vkm)
  + [Environment variables kept up-to-date on Java 9+](#1t3h5sf)
  + [Incremental build uses less memory](#4d34og8)
  + [Build Scan Plugin default version updated to 1.16](#2s8eyo1)
* [Fixed issues](#17dp8vu)
* [Deprecations](#3rdcrjn)
  + [Creating instances of JavaPluginConvention](#26in1rg)
  + [Creating instances of ApplicationPluginConvention](#lnxbz9)
  + [Creating instances of WarPluginConvention](#35nkun2)
  + [Creating instances of EarPluginConvention](#1ksv4uv)
  + [Creating instances of BasePluginConvention](#44sinio)
  + [Creating instances of ProjectReportsPluginConvention](#2jxsxqh)
  + [Adding tasks via TaskContainer.add() and TaskContainer.addAll()](#z337ya)
* [Potential breaking changes](#3j2qqm3)
  + [Changes to the Gradle Kotlin DSL](#1y810tw)
  + [Restricting cross-configuration and lifecycle hooks from lazy configuration APIs](#4i7ojhp)
  + [Cross Account AWS S3 Artifact Publishing](#2xcytpi)
  + [Changes to the Java Gradle Plugin plugin](#1ci93xb)
* [External contributions](#3whwml4)
* [Known issues](#2bn6wsx)

## Upgrade Instructions

Switch your build to use Gradle 4.10 quickly by updating your wrapper properties:

./gradlew wrapper --gradle-version=4.10

Standalone downloads are available at [gradle.org/releases](https://gradle.org/releases/).

## New and noteworthy

Here are the new features introduced in this Gradle release.

### Incremental Java compilation by default

This release fixes all known issues of the incremental compiler. It now

* deletes empty package directories when the last class file is removed
* recompiles all classes when module-info files change
* recompiles all classes in a package when that package's package-info changes

Its memory usage has also been reduced. For the gradle/gradle build, heap usage dropped from 350MB to just 10MB.

We are now confident that the incremental compiler is ready to be used in every build, so it is now the new default setting.

### Periodic cache cleanup

Caching has always been one of the strong suits of Gradle. Over time, more and more persistent caches have been added to improve performance and support new features, requiring more and more disk space on build servers and developer workstations. Gradle now addresses one of the most highly voted issues on GitHub and introduces the cleanup strategies for the caches in the [Gradle user home directory](http://docs.google.com/userguide/directory_layout.html#dir:gradle_user_home:cache_cleanup) and the [project root directory](http://docs.google.com/userguide/directory_layout.html#dir:project_root:cache_cleanup).

### Kotlin DSL 1.0 RC

[Kotlin DSL version 1.0 RC](https://github.com/gradle/kotlin-dsl/releases/tag/v1.0-RC6) is now available. Major updates from v0.18 include:

* API Documentation in IDE and reference forms
* Script compilation build cache
* IDE integration improvements
* Support for configuration avoidance
* buildSrc refactoring propagation to the IDE

This version includes the last set of backward compatibility-breaking DSL changes until Gradle 6.0.

Please give it a go and file issues in the [gradle/kotlin-dsl](https://github.com/gradle/kotlin-dsl) project. *If you are interested in using the Kotlin DSL, please check out the* [*Gradle guides*](https://gradle.org/guides/)*, especially the* [*Groovy DSL to Kotlin DSL migration guide*](https://guides.gradle.org/migrating-build-logic-from-groovy-to-kotlin/)*.*

### Use SNAPSHOT plugin versions with the plugins {} block

Starting with this release, it is now possible to use SNAPSHOT plugin versions in the plugins {} and pluginManagement {} blocks. For example:

plugins {  
    id 'org.springframework.boot' version '2.0.0.BUILD-SNAPSHOT'  
}

### Nested included builds

Composite builds is a feature that allows a Gradle build to 'include' another build and conveniently use its outputs locally rather than via a binary repository. This makes some common workflows more convenient, such as working on multiple source repositories at the same time to implement a cross-cutting feature. In previous releases, it was not possible for a Gradle build to include another build that also includes other builds, which limits the usefulness of this feature for these workflows. In this Gradle release, a build can now include another build that also includes other builds. In other words, composite builds can now be nested.

There are a number of limitations to be aware of. These will be improved in later Gradle releases:

* A buildSrc build cannot include other builds, such as a shared plugin build.
* The root project of each build must have a unique name.

### Authorization for Maven repositories with custom HTTP headers

Now it is possible to define a custom HTTP header to authorize access to a Maven repository. This enables Gradle to access private Gitlab and TFS repositories used as Maven repositories or any OAuth2 protected Maven repositories.

### Environment variables kept up-to-date on Java 9+

Early previews of Java 9 failed when Gradle tried to update the environment variables of the daemon to match the client that requested the build. The final Java 9 version added ways to work around this, but Gradle was not updated accordingly, meaning that environment variables always had the same value as when the daemon was started. This has now been fixed and the environment variables of the daemon will match the variables of the client again. Changes to the PATH will be visible to Exec tasks and calling System.getenv will yield the expected result.

However, we strongly recommend that build and plugin authors use Gradle properties instead of System.getenv for a more idiomatic end user experience.

### Incremental build uses less memory

Memory usage for up-to-date checking has been improved. For the gradle/gradle build, heap usage dropped by 60 MB to 450 MB, that is a 12% reduction.

### Build Scan Plugin default version updated to 1.16

The built-in build scan plugin version has been updated to 1.16. When used with Gradle Enterprise 2018.4 or [scans.gradle.com](https://scans.gradle.com/), this provides deeper configuration time profiling, dependency repository insights, deprecated Gradle functionality usage analysis, and more.

For more information on how to use build scans, see <https://scans.gradle.com/>. For more information on new features in Gradle Enterprise 2018.4, see the [Gradle Enterprise 2018.4 Release Notes](https://gradle.com/enterprise/releases/2018.4/).

## Fixed issues

## Deprecations

Features that have become superseded or irrelevant due to the natural evolution of Gradle become *deprecated*, and scheduled to be removed in the next major Gradle version (Gradle 5.0). See the User guide section on the “[Feature Lifecycle](http://docs.google.com/userguide/feature_lifecycle.html)” for more information.

The following are the newly deprecated items in this Gradle release. If you have concerns about a deprecation, please raise it via the [Gradle Forums](https://discuss.gradle.org).

### Creating instances of JavaPluginConvention

Instances of this class are intended to be created only by the java-base plugin and should not be created directly. Creating instances using the constructor of JavaPluginConvention will become an error in Gradle 5.0. The class itself is not deprecated and it is still be possible to use the instances created by the java-base plugin.

### Creating instances of ApplicationPluginConvention

Instances of this class are intended to be created only by the application plugin and should not be created directly. Creating instances using the constructor of ApplicationPluginConvention will become an error in Gradle 5.0. The class itself is not deprecated and it is still be possible to use the instances created by the application plugin.

### Creating instances of WarPluginConvention

Instances of this class are intended to be created only by the war plugin and should not be created directly. Creating instances using the constructor of WarPluginConvention will become an error in Gradle 5.0. The class itself is not deprecated and it is still be possible to use the instances created by the war plugin.

### Creating instances of EarPluginConvention

Instances of this class are intended to be created only by the ear plugin and should not be created directly. Creating instances using the constructor of EarPluginConvention will become an error in Gradle 5.0. The class itself is not deprecated and it is still be possible to use the instances created by the ear plugin.

### Creating instances of BasePluginConvention

Instances of this class are intended to be created only by the base plugin and should not be created directly. Creating instances using the constructor of BasePluginConvention will become an error in Gradle 5.0. The class itself is not deprecated and it is still be possible to use the instances created by the base plugin.

### Creating instances of ProjectReportsPluginConvention

Instances of this class are intended to be created only by the project-reports plugin and should not be created directly. Creating instances using the constructor of ProjectReportsPluginConvention will become an error in Gradle 5.0. The class itself is not deprecated and it is still be possible to use the instances created by the project-reports plugin.

### Adding tasks via TaskContainer.add() and TaskContainer.addAll()

These methods have been deprecated and the create() or register() methods should be used instead.

## Potential breaking changes

### Changes to the Gradle Kotlin DSL

The Kotlin DSL enables experimental Kotlin compiler features in order to expose an uniform org.gradle.api.Action<T> based API to both Groovy DSL and Kotlin DSL scripts.

The DSL types and behavior of containers elements delegated properties (e.g. val jar by tasks) and containers scopes (e.g. tasks { }) changed.

The source set container can now be accessed using project.sourceSets, or just sourceSets. Previously it was located at project.java.sourceSets, or just java.sourceSets.

All these changes could cause script compilation errors.

See the [Gradle Kotlin DSL release notes](https://github.com/gradle/kotlin-dsl/releases/tag/v1.0-RC6) for more information and how to fix builds broken by the changes described above.

### Restricting cross-configuration and lifecycle hooks from lazy configuration APIs

In Gradle 4.9, we [introduced a new API](https://blog.gradle.org/preview-avoiding-task-configuration-time) for creating and configuring tasks.

The following hooks are disallowed when called from these new APIs:

* project.afterEvaluate(Action) and project.afterEvaluate(Closure)
* project.beforeEvaluate(Action) and project.beforeEvaluate(Closure)

If you attempt to call any of these methods an exception will be thrown. Gradle restricts these APIs because mixing these APIs with lazy configuration can cause hard to diagnose build failures and complexity.

### Cross Account AWS S3 Artifact Publishing

The S3 [repository transport protocol](http://docs.google.com/userguide/repository_types.html#sub:supported_transport_protocols) allows Gradle to publish artifacts to AWS S3 buckets. Starting with this release, every artifact uploaded to an S3 bucket will be equipped with bucket-owner-full-control canned ACL. Make sure the used AWS credentials can do s3:PutObjectAcl and s3:PutObjectVersionAcl to ensure successful artifacts uploads.

{  
    "Version":"2012-10-17",  
    "Statement":[  
        // ...  
        {  
            "Effect":"Allow",  
            "Action":[  
                "s3:PutObject", // necessary for uploading objects  
                "s3:PutObjectAcl", // required starting with this release  
                "s3:PutObjectVersionAcl" // if S3 bucket versioning is enabled  
            ],  
            "Resource":"arn:aws:s3:::myCompanyBucket/\*"  
        }  
    ]  
}

See the User guide section on “[Repository Types](http://docs.google.com/userguide/repository_types.html#sub:s3_cross_account)” for more information.

### Changes to the Java Gradle Plugin plugin

* PluginUnderTestMetadata and GeneratePluginDescriptors were updated to use the Provider API.
* All setters were removed and can be replaced with calls to the new Property set(...) method.

## External contributions

We would like to thank the following community members for making contributions to this release of Gradle.

We love getting contributions from the Gradle community. For information on contributing, please see [gradle.org/contribute](https://gradle.org/contribute).

* [Sébastien Deleuze](https://github.com/sdeleuze) - Add support for SNAPSHOT plugin versions in the plugins {} block ([gradle/gradle#5762](https://github.com/gradle/gradle/issues/5762))
* [Paul Wellner Bou](https://github.com/paulwellnerbou) - Authorization for Maven repositories with custom HTTP headers ([gradle/gradle#5571](https://github.com/gradle/gradle/issues/5571))
* [Salvian Reynaldi](https://github.com/salvianreynaldi) - Give S3 bucket owner full control over the published Maven artifacts ([gradle/gradle#5329](https://github.com/gradle/gradle/issues/5329))
* [Mike Kobit](https://github.com/mkobit) - Add ability to use RegularFile and Directory as publishable artifacts ([gradle/gradle#5109](https://github.com/gradle/gradle/issues/5109))
* [Thomas Broyer](https://github.com/tbroyer) - Convert java-gradle-plugin to use lazy configuration API ([gradle/gradle#6115](https://github.com/gradle/gradle/issues/6115))
* [Björn Kautler](https://github.com/Vampire) - Update Spock version in docs and build init ([gradle/gradle#5627](https://github.com/gradle/gradle/issues/5627))
* [Ben McCann](https://github.com/benmccann) - Decouple Play and Twirl versions ([gradle/gradle#2062](https://github.com/gradle/gradle/issues/2062))
* [Kyle Moore](https://github.com/DPUkyle) - Use latest Gosu plugin 0.3.10 ([gradle/gradle#5855](https://github.com/gradle/gradle/issues/5855))
* [Jean-Baptiste Nizet](https://github.com/jnizet) — Avoid double deprecation message when using the publishing plugin ([gradle/gradle#6653](https://github.com/gradle/gradle/issues/6653))
* [Mata Saru](https://github.com/matasaru) - Add missing verb into docs ([gradle/gradle#5694](https://github.com/gradle/gradle/issues/5694))
* [Mészáros Máté Róbert](https://github.com/mrmeszaros) - Fix typo in userguide java plugin configuration image ([gradle/gradle#6011](https://github.com/gradle/gradle/issues/6011))
* [Kenzie Togami](https://github.com/kenzierocks) - Docs are unclear on how JavaExec parses --args ([gradle/gradle#6056](https://github.com/gradle/gradle/issues/6056))
* [Sebastian Schuberth](https://github.com/sschuberth) - Fix an application plugin example to work for Windows ([gradle/gradle#5994](https://github.com/gradle/gradle/issues/5994))

## Known issues

Known issues are problems that were discovered post release that are directly related to changes made in this release.