

Nebula Lint plugin

<https://dev.to/gradle-community/unlocking-configuration-cache-with-gsoc-contributor-374l>

PR (partial changes towards a fix): <https://github.com/nebula-plugins/gradle-lint-plugin/pull/433>

Play Framework (Gradle own)

Repository: <https://github.com/Nouran-11/playframework>

PR: <https://github.com/gradle/playframework/pull/207>

Liquibase Gradle Plugin

While testing the Liquibase Gradle plugin, I encountered configuration cache (CC) issues. Specifically, the `:update` task fails CC compatibility because it accesses `project` during task execution. For example:

Even using `liquibase.activities` directly doesn't help, because it still implicitly accesses `project.liquibase`, which violates configuration cache rules. Gradle expects all configuration to be finalized and immutable before execution starts. Referencing anything from `project` (like extensions, configurations, or logging) during execution causes CC to break.

Upon reviewing the plugin's source code ([lines 59–60](#)), this access pattern occurs within the task action itself. This is repeated in multiple areas, such as `runLiquibase(activity)` ([line 73](#)) and usage of `project.configurations`, `systemProperties`, `project.liquibase.jvmArgs`, and `project.logger` ([line 86](#)). // `project.logger` doesn't break cc

The `LiquibaseExtension` used via `project.liquibase` is added to `project.extensions` and is initialized during the configuration phase. So there's no need to access it at execution time — doing so is unnecessary and problematic for configuration cache support.

Issue Overview

In `LiquibaseTask`, the following lines are one of the problematic for **configuration cache** compatibility because they directly access `project`:

```
def activities = project.liquibase.activities
def runList = project.liquibase.runList
```

Even removing `project` (i.e., using just `liquibase.activities`) doesn't solve the problem, as it still implicitly accesses `project.liquibase`, breaking CC due to project state being queried at execution time.

- **Why does this break CC?**

Accessing `project.liquibase` (an extension of type `LiquibaseExtension`) during execution time breaks configuration cache compatibility. Configuration cache expects all configuration-related state to be collected and made immutable during the configuration phase.

- **How are `activities` and `runList` initialized?**

They are properties on the `LiquibaseExtension`, which is added to `project.extensions`. These properties are configured during the **configuration phase**, so they are already available before task execution.

- **So why access them during execution?**

There's no actual need. These values can and should be captured during configuration time using **providers**, avoiding project access at execution.

- **Recommended fix:**

Use **Provider**-based inputs to defer the value resolution but still resolve them during the configuration. This approach:

- Captures the necessary state early
- Avoids referencing `project` during task execution
- Makes the task compatible with the configuration cache

This is similar to what was done in other plugins (e.g., Nebula Lint), where values needed at execution were projected early via providers.

- **Project instance scope:**

It appears that the `project` instance is not needed at execution time for any other logic, so the dependency can be removed safely.

Js2d-gradle

Error msg when trying to run : ./gradlew
generateJsonSchema2DataClassConfigExample

* What went wrong:

Configuration cache problems found in this build.

2 problems were found storing the configuration cache, 1 of which seems unique.

- Task `:generateJsonSchema2DataClassConfigExample` of type
`org.jsonschema2dataclass.js2p.Js2pGenerationTask`: value 'fixed(class
org.gradle.api.internal.file.DefaultFileCollectionFactory\$ResolvingFileCollection, file
collection)' is not assignable to 'org.gradle.api.NamedDomainObjectProvider'

Test repo link: <https://github.com/Nouran-11/js2d-test>

My changes:

<https://github.com/jsonschema2dataclass/js2d-gradle/compare/main...Nouran-11:js2d-gradle:fix-cc>

Note: This plugin was solved at their code but not released yet , that what made the conflict

Android plugin version :

First issue I got :

* What went wrong:

Configuration cache state could not be cached: field `\_\_outputFolder\_\_` of task

`:plugin-gradle:processors:jsonschema2pojo:processorVersion` of type

`org.jsonschema2dataclass.internal.plugin.lib.ProcessorVersionGeneratorTask`: error writing value of type
'org.gradle.api.internal.file.DefaultFilePropertyFactory\$DefaultDirectoryVar'

> Extension 'processorVersion' property 'outputFolder' is declared as an output property of extension
'processorVersion' (type ProcessorVersionPExtension) but does not have a task associated with it.

After solving it , I got :

1 problem was found storing the configuration cache.

- Task `:plugin-gradle:processors:jsonschema2pojo:processorVersion` of type

`org.jsonschema2dataclass.internal.plugin.lib.ProcessorVersionGeneratorTask`: invocation of
'Task.project' at execution time is unsupported.

Root Cause

In the current version of the plugin, a task input receives a resolved file collection at configuration time, which violates Gradle's configuration cache requirements. Specifically, the task attempts to assign a value that's not lazily resolved or wrapped in a `Provider` or `NamedDomainObjectProvider`.

Strategy Used to Fix It:

Capture the configuration using a local variable typed as `FileCollection` instead of assigning a resolved value directly to the task input.

Sonar-scanner-gradle

1 problem was found storing the configuration cache.

- Plugin 'com.jfrog.artifactory': registration of listener on 'Gradle.addListener' is unsupported

The solution: only update jfrog plugin version to the latest "6.0.0" as we did to PF

The plugin already compatible , the plugin's build was not compatible so that why it was breaking while using composite build

Flyway issue

14 problems were found storing the configuration cache, 2 of which seem unique.

- Task `:flywayClean` of type `org.flywaydb.gradle.task.FlywayCleanTask`: invocation of 'Task.project' at execution time is unsupported.
- Task `:flywayMigrate` of type `org.flywaydb.gradle.task.FlywayMigrateTask`: invocation of 'Task.project' at execution time is unsupported.

Unfortunately flyway doesn't work with gradle

:<https://github.com/flyway/flyway/issues/3550>

Forbidden apis plugin

<https://github.com/policeman-tools/forbidden-apis/issues/269>

The issue was using Task.project at execution time

From a maintainer there : I **don't want to touch it or fix any bugs**. Gradle Plugin of forbiddenapis is no longer understandable and therefore not maintained anymore. Standard stuff works but caching is not in my priority list. Sorry!

Compose multiplatform localize plugin

<https://github.com/hyperether/compose-multiplatform-localize/issues/10>

Main issue : GenerateTranslationTask

Error: Task.project invocation at execution time

* What went wrong:

Configuration cache problems found in this build.

2 problems were found storing the configuration cache, 1 of which seems unique.

- Task `:composeApp:generateTranslateFile` of type
`com.hyperether.localization.GenerateTranslationsTask`: invocation of 'Task.project' at execution time is unsupported.

See

https://docs.gradle.org/8.13/userguide/configuration_cache.html#config_cache:requirements:use_project_during_execution

See the complete report at

file:///Users/nouranatef/AndroidStudioProjects/localizeetest/build/reports/configuration-cache/6xsq9hlemt75nqmi5xn3auo4c/cx65zw03h03xp4rl6o80qt6pd/configuration-cache-report.html

> Invocation of 'Task.project' by task ':composeApp:generateTranslateFile' at execution time is unsupported.

To test this specific task: `./gradlew generateTranslateFile --configuration-cache`

My test project repo , note you need to install android studio and xcode to use it , it is a compose multiplatform project: Statues : Done

Rewrite gradle plugin

Initial line :

<https://github.com/Nouran-11/rewrite-gradle-plugin/blob/4c6b62c8535d9fc6161b0ae5b141843e0e1784c5/plugin/src/main/java/org/openrewrite/gradle/RewriteDryRunTask.java#L52>

Line 2 :

<https://github.com/Nouran-11/rewrite-gradle-plugin/blob/de5dd219f8b87314f6039200fd8992da3416918b/plugin/src/main/java/org/openrewrite/gradle/AbstractRewriteTask.java#L90>

Part 3 :

<https://github.com/Nouran-11/rewrite-gradle-plugin/blob/de5dd219f8b87314f6039200fd8992da3416918b/plugin/src/main/java/org/openrewrite/gradle/DelegatingProjectParser.java#L44-L83>

Part 4 :

<https://github.com/Nouran-11/rewrite-gradle-plugin/blob/de5dd219f8b87314f6039200fd8992da3416918b/plugin/src/main/java/org/openrewrite/gradle/DelegatingProjectParser.java#L177-L200>

Java

```
org.gradle.api.InvalidUserCodeException: Invocation of 'Task.project' by
    task ':rewriteDryRun' at execution time is unsupported.
    at org.gradle.api.DefaultTask.getProject(DefaultTask.java:59)
    at
    org.openrewrite.gradle.AbstractRewriteTask.getProjectParser(AbstractRewriteTask.java:90)
    at
    org.openrewrite.gradle.RewriteDryRunTask_Decorated.getProjectParser(Unknown Source)
    at
    org.openrewrite.gradle.AbstractRewriteTask.getActiveRecipes(AbstractRewriteTask.java:98)
```

```
at
  org.openrewrite.gradle.RewriteDryRunTask_Decorated.getActiveRecipes(Unknown
  Source)
at
  org.gradle.internal.properties.bean.DefaultPropertyWalker$CachedPropertyV
  alue.lambda$new$0(DefaultPropertyWalker.java:107)
at
  com.google.common.base.Suppliers$NonSerializableMemoizingSupplier.get(Sup
  pliers.java:181)
at java.base/java.lang.Thread.run(Thread.java:833)
```

Unlisted Issues:

<https://github.com/GTNewHorizons/RetroFuturaGradle/issues/86>

<https://github.com/architecture/architecture-plugin/issues/55>

<https://github.com/LFDT-web3j/web3j-build-tools/issues/14>

<https://github.com/jfrog/gradle-dep-tree/issues/25>

<https://github.com/digma-ai/digma-intellij-plugin/issues/2729>

<https://github.com/AppGalleryConnect/agc-android-demos/issues/8>