

(This wiki page applies to people migrating code written before December 2017. It is unlikely to still be relevant.)

Introduction

Prior to [#13492](#) (merged on 2017-12-13), Flutter tooling created projects whose Android part used Gradle 3.3 and the Android Studio Gradle plugin version 2.3.3.

After [#13492](#), Flutter tooling creates projects based on Gradle 4.1 and the Android Studio Gradle plugin version 3.0.1.

This page describes how you can manually upgrade your Flutter projects from the old mode to the new one. This may be needed to consume new-mode Flutter plugins.

Where in doubt, please consult <https://developer.android.com/studio/build/gradle-plugin-3-0-0-migration.html>.

Upgrading a Flutter app project

1. In `android/gradle/wrapper/gradle-wrapper.properties`, replace `3.3` with `4.1` in the last line.

2. In `android/build.gradle`, replace

```
repositories {  
    jcenter()  
    maven {  
        url "https://maven.google.com"  
    }  
}
```

by

```
repositories {  
    google()  
    jcenter()  
}
```

twice.

Also replace

```
dependencies {  
    classpath 'com.android.tools.build:gradle:2.3.3'  
}
```

by

```
dependencies {  
    classpath 'com.android.tools.build:gradle:3.0.1'  
}
```

Finally, replace

```

subprojects {
    project.buildDir = "${rootProject.buildDir}/${project.name}"
    project.evaluationDependsOn(':app')
}

```

by

```

subprojects {
    project.buildDir = "${rootProject.buildDir}/${project.name}"
}
subprojects {
    project.evaluationDependsOn(':app')
}

```

This change prevents a "Gradle build failed to produce an Android package" error message when your app depends on plugins whose name comes before "app" in lexicographic order.

3. In `android/app/build.gradle`, replace version `25` by `27` and `25.0.3` by `27.0.3` (three places total).

If you are using a version of Flutter from before [#13558](#), add the following to the `android, buildTypes` section:

```

profile {
    matchingFallbacks = ['debug', 'release']
}

```

Replace configurations named `compile` by `api` (or `implementation`) and `provided` by `compileOnly` in the `dependencies` section. You may also want to update dependencies with newer versions as applicable. In the default project templates generated by Flutter tooling we have replaced

```

dependencies {
    androidTestCompile 'com.android.support:support-annotations:25.4.0'
    androidTestCompile 'com.android.support.test:runner:0.5'
    androidTestCompile 'com.android.support.test:rules:0.5'
}

```

by

```

dependencies {
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.1'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'
}

```

loosely following the lead of Android Studio.

Upgrading a Flutter plugin project

1. Upgrade the `example/` Flutter app project as indicated above.
2. Remove the Gradle wrapper files and folder (`gradlew` , `gradlew.bat` , `gradle/`) from the top-level `android` folder, if present. The Gradle wrapper is used by the `example` app only, and it should already contain these files.
3. In `android/build.gradle` , replace

```
repositories {  
    jcenter()  
    maven {  
        url "https://maven.google.com"  
    }  
}
```

by

```
repositories {  
    google()  
    jcenter()  
}
```

twice.

Also replace

```
dependencies {  
    classpath 'com.android.tools.build:gradle:2.3.3'  
}
```

by

```
dependencies {  
    classpath 'com.android.tools.build:gradle:3.0.1'  
}
```

Replace version `25` by `27` and `25.0.3` by `27.0.3` .

Lastly, replace configurations named `compile` by `api` (or `implementation`) and `provided` by `compileOnly` in any custom `dependencies` section.