



Gradle Isn't Just a Bad Word



A FAILURE OCCURRED EXECUTING ...
AAPT2 UNEXPECTED ERROR DURING
LINK,

GRAAADLE!

Goals



- reduce fear of Gradle
- help you understand Stack Overflow snippets
- new tools to help debug builds

[github.com/phatblat/
CompositeProject](https://github.com/phatblat/CompositeProject)

Agenda



- Project Structure
- Invoking Gradle from the CLI
- Creating Tasks
- Plugins
- Dependencies
- Composite Projects

Files

Gradle Project Structure

- Wrapper
- Ignore
- File Layout
- Project Hierarchy

A photograph of a man with long hair and a beard, wearing dark sunglasses and a red zip-up hoodie. He is holding a green rectangular card with a gold emblem in his right hand and pointing his left index finger towards the camera. A gold chain with a large medallion hangs around his neck. The background is dark.

Rapper?

Wrapper

```
└── gradle
    └── wrapper
        └── gradle-wrapper.jar
        └── gradle-wrapper.properties
    └── gradlew
    └── gradlew.bat
```

Ignore

→ .gradle /

→ build /

Gradle Files

- build.gradle
- settings.gradle
- gradle.properties

Kotlin Script

- build.gradle.kts
- settings.gradle.kts

build.gradle

- gradle DSL
- plugin "extension" DSLs

settings.gradle

- rootProject.name
- multi-project configuration (include)
- composite configuration (includeBuild)

gradle.properties

- Comment lines with #
- Leading/trailing whitespace ignored key = value
- Values read as String
- Properties inherited by subprojects
 - AGP 4.2+ ignores these files in subprojects ¹

¹ See the AGP 4.2 [release notes](#)

local.properties

→ sdk.dir

→ ndk.dir

Project Hierarchy

```
Root project 'CompositeProject'  
+--- Project ':app'  
\--- Project ':module'
```

View using the "projects" task.

**Invoking
Gradle from
the cli**

Invoking the Wrapper

- `./gradlew` `taskName`
- `gradlew.bat` `taskName`
- `gw` `taskName`

Wrapper Function (Bash & Z Shell)

```
function gw {  
    if [ -e ./gradlew ]; then  
        ./gradlew $argv  
        return  
    fi  
  
    echo "There is no Gradle wrapper in the current dir."  
    gradle $argv  
}
```

Wrapper Function (Fish Shell)

```
function gw
  if test -e ./gradlew
    ./gradlew $argv
  return
end

echo "There is no Gradle wrapper in the current dir."
gradle $argv
end
```

Input

- build.gradle
- gradle.properties
- ~/.gradle/gradle.properties
- -Pproperty=value

Output

- console
- log levels
- stacktrace
- build scans

--console

→ plain

Log Levels

- error: -q, --quiet
- warn: -W, --warn
- lifecycle (default)
- -i, --info
- -d, --debug

Stack Trace

→ -S, --stacktrace

→ -S, --full-stacktrace

Build --scan

--scan

Tasks

Listing Tasks

→ gw tasks

→ gw tasks --all

```
Zzz tasks
-----
clonePlugin - Clones the ShellExec plugin project into a sibling directory.
cloneRetrofit - Clones the Retrofit project into a sibling directory.
fortune - An example script task.
hello - An example task class.
```

Creating Tasks

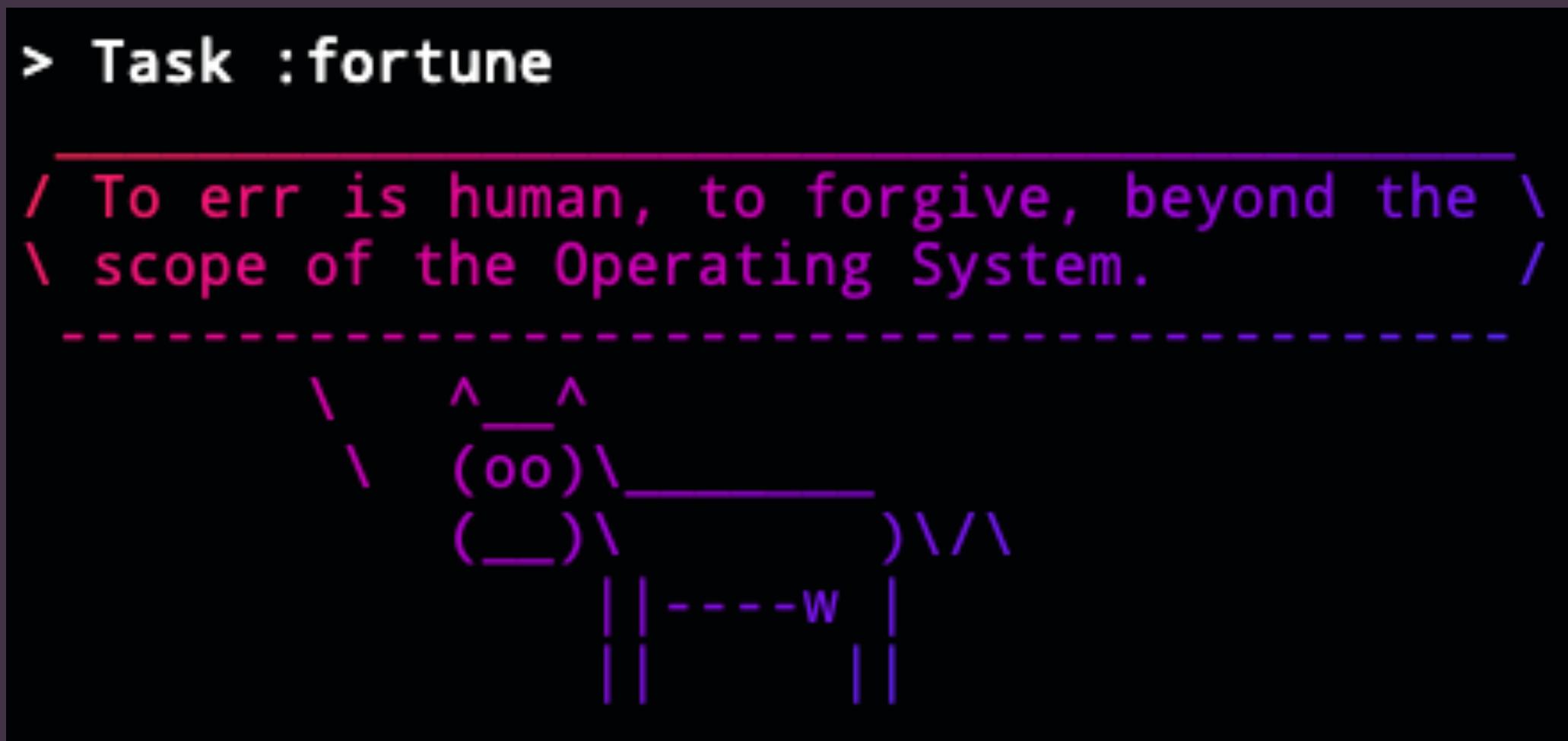
```
tasks.register('hello') {  
    doLast {  
        println 'hello world'  
    }  
}
```

Example Tasks

- buildTypes
- productFlavors
- flavorDimensions
- buildVariants
- variantFilter

ShellExec

```
tasks.register('fortune', ShellExec) {
    command "fortune | cowsay | lolcat --force"
}
```



dependsOn

→ Chains tasks

 `assembleDebug.dependsOn(fortune)`

```
tasks.whenTaskAdded { task ->
    if (task.name == 'assembleDebug') {
        task.dependsOn(fortune)
    }
}
```

Dry Run

→ -m, --dry-run

```
↪ gw --dry-run assembleDebug
:fortune SKIPPED
:app:preBuild SKIPPED
...
:app:packageFullDebug SKIPPED
:app:assembleFullDebug SKIPPED
:app:assembleDebug SKIPPED
..
:module:compileDebugSources SKIPPED
:module:assembleDebug SKIPPED
```

Plugins

Script Plugins

apply from: "android-info.gradle"

→ File path

→ URL

Binary Plugins

- Buildscript Block
- Plugins Block
- Maven Coordinates vs Plugin ID

Buildscript Block

```
buildscript {  
    repositories {  
        google()  
        mavenCentral()  
        jcenter()  
    }  
    dependencies {  
        classpath 'at.phatbl:shellexec:+'  
    }  
}  
  
apply plugin: "at.phatbl.shellexec"
```

Plugins Block

```
plugins {  
    id "at.phatbl.shellexec" version "1.5.2"  
}
```

- version only needed in root project
- auto-applies plugin to the current project
- opt out with: apply false

Maven Coordinates

- group ID
- artifact ID
- version
- classifier (optional)
- extension (optional)

configurationName 'group:name:version:classifier@extension'

configurationName group: group, name: name, version: version, classifier: classifier, ext: extension

Maven Group ID

- typically reverse-domain syntax of author
- phatbl.at <-> at.phatbl
- org.jetbrains.kotlin
- com.android.tools.build

Maven Artifact ID

- library name
- typically words separated by dash
- e.g. protobuf-gradle-plugin

Version

- Semantic verion number
- Rolling -SNAPSHOTS
- Optional pre-release qualifier (e.g. 1.0-alpha-3)

Specifying versions

- Support for version ranges ²
- Dynamic versions (+)

² Maven Dependency [Version Requirement Spec](#)

Dynamic Versions

→ + - any version

→ 1.+ - any v1 minor version

→ 1.2.+ - any v1.2 patch version

→ Version cached for 24h

→ Force check with --refresh-dependencies

Dynamic Version Caveats

- Discouraged by Google
- Build non-determinism if dependency updates are being released
- Slower dependency resolution
 - Every repo is checked for dependency versions

Plugin ID

- Used when applying plugins
- Used in newer Gradle plugins block

Maven Coordinates vs Plugin ID

- at.phatbl:shellexec:1.0.0
- at.phatbl.shellexec
- View on plugins.gradle.org for most plugins

Dependencies

Configuration

- compile (don't use)
- implementation (use this)

implementation ...

testImplementation ...

Configuration

- api - for modules & libraries
- when types defined in a dependency are exposed by your public API
- see app -> module.GitHub -> retrofit in example project

List Plugins

```
↪ gw buildEnvironment
```

```
classpath
+--- com.android.application:com.android.application.gradle.plugin:4.2.2
|   \--- com.android.tools.build:gradle:4.2.2
|   ...
\--- at.phatbl.shellexec:at.phatbl.shellexec.gradle.plugin:1.5.2
    \--- gradle.plugin.at.phatbl:shellexec:1.5.2
        +--- org.jetbrains.kotlin:kotlin-stdlib:1.3.61 -> 1.5.21 (*)
        +--- org.jetbrains.kotlin:kotlin-stdlib-jdk8:1.3.61 -> 1.4.31 (*)
        \--- org.jetbrains.kotlin:kotlin-reflect:1.3.61 -> 1.5.21 (*)
```

Gradle Dependencies Task

```
↪ gw :app:dependencies --configuration implementation
```

```
implementation - Implementation only dependencies for 'main' sources. (n)
+--- project module (n)
+--- org.jetbrains.kotlin:kotlin-stdlib:1.5.21 (n)
+--- androidx.core:core-ktx:1.6.0 (n)
+--- androidx.appcompat:appcompat:1.3.0 (n)
+--- com.google.android.material:material:1.4.0 (n)
\--- androidx.constraintlayout:constraintlayout:2.0.4 (n)
```

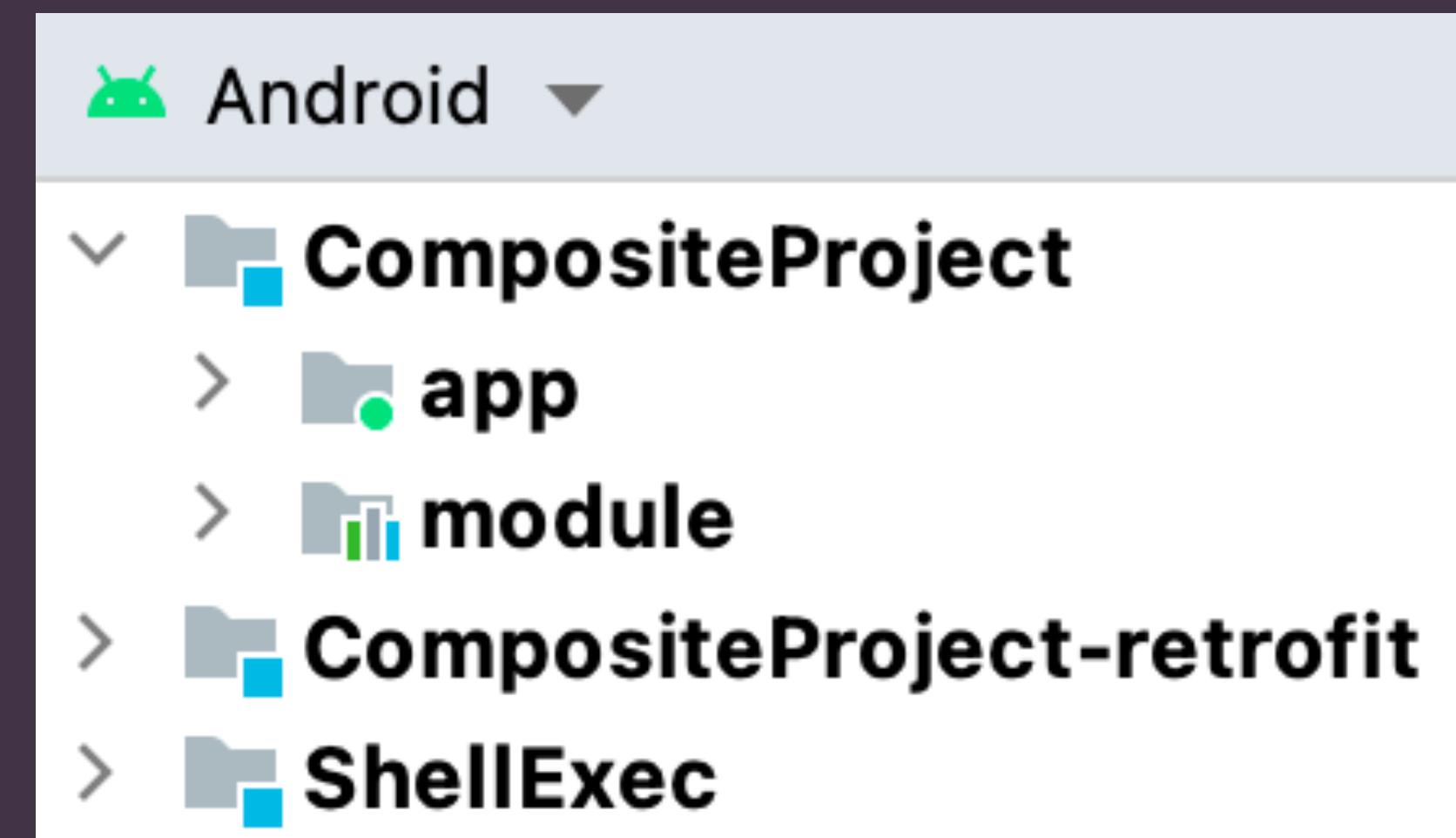
Android Dependencies Task

```
↪ gw androidDependencies
```

```
releaseUnitTestRuntimeClasspath - Dependencies for runtime/packaging
+--- :module (variant: release)
+--- junit:junit:4.13.2@jar
+--- com.squareup.retrofit2:converter-gson:2.9.0@jar
+--- com.squareup.retrofit2:adapter-java8:2.9.0@jar
+--- com.squareup.retrofit2:retrofit:2.9.0@jar
+--- androidx.core:core-ktx:1.6.0@aar
+--- com.google.code.gson:gson:2.8.5@jar
\--- com.squareup.okio:okio:1.17.2@jar
```

Composite Project

gw --include-build ../retrofit ...



includeBuild

```
File retrofitProject = file("../retrofit")

includeBuild(retrofitProject) {
    dependencySubstitution {
        substitute module('com.squareup.retrofit2:retrofit') with project(':retrofit')
    }
}
```

Sample Project

```
↪ gw clonePlugin cloneRetrofit
```

```
> Task :clonePlugin  
Cloning into '../ShellExec'...
```

```
> Task :cloneRetrofit  
Cloning into '../retrofit'...
```

Sample Project

↪ gw projects

Including .../ShellExec project in composite build.
Including .../retrofit project in composite build.

> Configure project :ShellExec

...

> Configure project :retrofit:retrofit

...

Caveats

- AGP version
- Android build variants
- Gradle plugin using same library

AGP Version

```
> Failed to apply plugin 'com.android.internal.application'.
    > Using multiple versions of the Android Gradle plugin in the same build is not allowed.
      - Project `..../CompositeProject/app` is using version `4.2.2`
      - Project `..../retrofit/retrofit/android-test` is using version `4.2.1`
```

Gradle Properties to the Rescue! 💪

gw -P`agp_version=4.2.1` projects

> Configure project :retrofit:retrofit

Variants

Comment out:

→ flavorDimensions

→ productFlavors

Gradle plugin using same library

→ `includeBuild file("../retrofit")`

```
> Could not resolve com.squareup.retrofit2:retrofit:2.7.1.
```

Required by:

```
  project :retrofit > com.vanniktech:gradle-maven-publish-plugin:0.14.0
```

```
  project :retrofit > com.vanniktech:gradle-maven-publish-plugin:0.14.0 > com.squareup.retrofit2:converter-moshi:2.7.1
```

→ Removed the gradle-maven-publish-plugin from retrofit

Problem

Profanity



→ frak

→ p'tak

→ frell

→ cruk

→ shazbot

→ kriff

→ gorram

