MAKING THE MOST OF YOUR GRADLE BUILD

ANDRES ALMIRAY

@AALMIRAY

ANDRESALMIRAY.COM

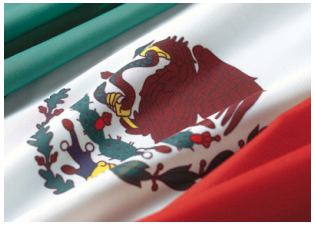
MAKING THE MOST OF YOUR GRADLE BUILD

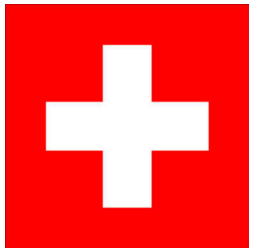
ANDRES ALMIRAY

@AALMIRAY

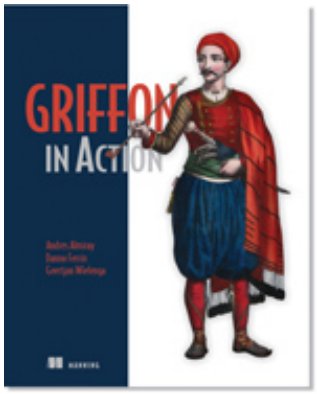
ANDRESALMIRAY.COM


Kino





canoo



JAVA™
CHAMPIONS

Groovy

1

HOW TO GET GRADLE INSTALLED ON YOUR SYSTEM


```
$ curl -s get.sdkman.io | bash
$ sdk install gradle
```

2

INITIALIZING A GRADLE PROJECT

GRADLE INIT

THIS COMMAND WILL GENERATE A BASIC STRUCTURE AND A MINIMUM SET OF FILES TO GET STARTED.

YOU CAN INVOKE THIS COMMAND ON A MAVEN PROJECT TO TURN IT INTO A GRADLE PROJECT. MUST MIGRATE PLUGINS MANUALLY.

$ sdk install lazybones
$ lazybones create gradle-quickstart sample

3

FIX GRADLE VERSION TO A PARTICULAR RELEASE

GRADLE WRAPPER

INITIALIZE THE WRAPPER WITH A GIVEN VERSION.

CHANGE FROM –bin.zip TO –all.zip TO HAVE ACCESS TO GRADLE API SOURCES.

4

INVOKING GRADLE ANYWHERE ON YOUR PROJECT

GDUB

https://github.com/dougborg/gdub

A SCRIPT THAT CAN INVOKE A GRADLE BUILD ANYWHERE INSIDE THE PROJECT STRUCTURE.

WILL ATTEMPT RESOLVING WRAPPER FIRST, THEN LOCAL GRADLE.

FAILS IF NEITHER IS FOUND.

5

MULTI-PROJECT SETUP

CHANGE BUILD FILE NAMES

CHANGE BULD FILE NAME FROM build.gradle TO

${project.name}.gradle

USE GROOVY EXPRESSIONS TO FACTORIZE PROJECT INCLUSIONS IN settings.gradle

SETTINGS.GRADLE (1/2)

```
['common', 'full', 'light'].each { name ->
    File subdir = new File(rootDir, name)
    subdir.eachDir { dir ->
        File buildFile = new File(dir, "${dir.name}.gradle")
        if (buildFile.exists()) {
            include "${name}/${dir.name}"
        }
    }
}
```

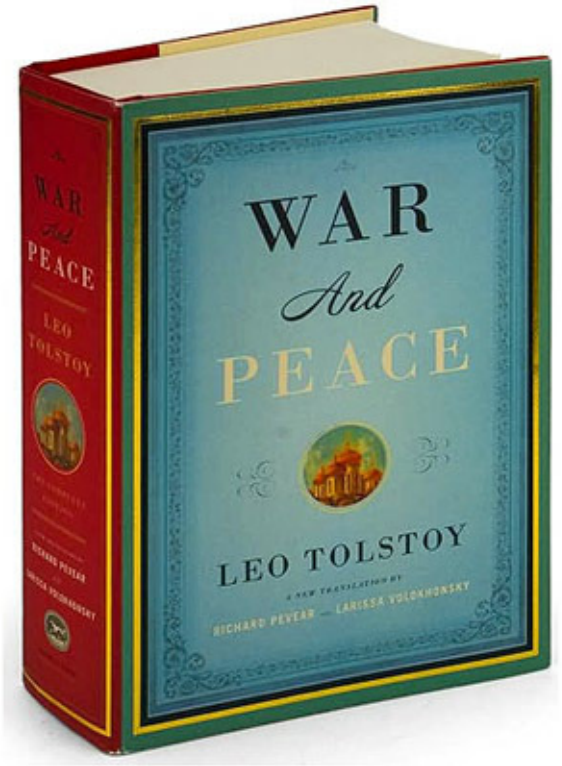
SETTINGS.GRADLE (2/2)

```
rootProject.name = 'my-project'
rootProject.children.each { project ->
    int slash = project.name.indexOf('/')
    String fileBaseName = project.name[(slash + 1)..-1]
    String projectDirName = project.name
    project.name = fileBaseName
    project.projectDir = new File(settingsDir, projectDirName)
    project.buildFileName = "${fileBaseName}.gradle"
    assert project.projectDir.isDirectory()
    assert project.buildFile.isFile()
}
```

6

BUILD FILE ETIQUETTE

WHAT BUILD.GRADLE IS NOT



KEEP IT DRY

EMBRACE THE POWER OF CONVENTIONS.

BUILD FILE SHOULD DEFINE HOW THE PROJECT DEVIATES FROM THE PRE-ESTABLISHED CONVENTIONS.

RELY ON SECONDARY SCRIPTS, SEGGREGATED BY RESPONSIBILITIES.

TASK DEFINITIONS

PUT THEM IN SEPARATE FILES.

1. **INSIDE SECONDARY SCRIPTS**
    1. REGULAR TASK DEFINITIONS
    2. PLUGIN DEFINITIONS
2. **AS PART OF buildSrc SOURCES**
3. **AS A PLUGIN PROJECT**

APPLYING PLUGINS

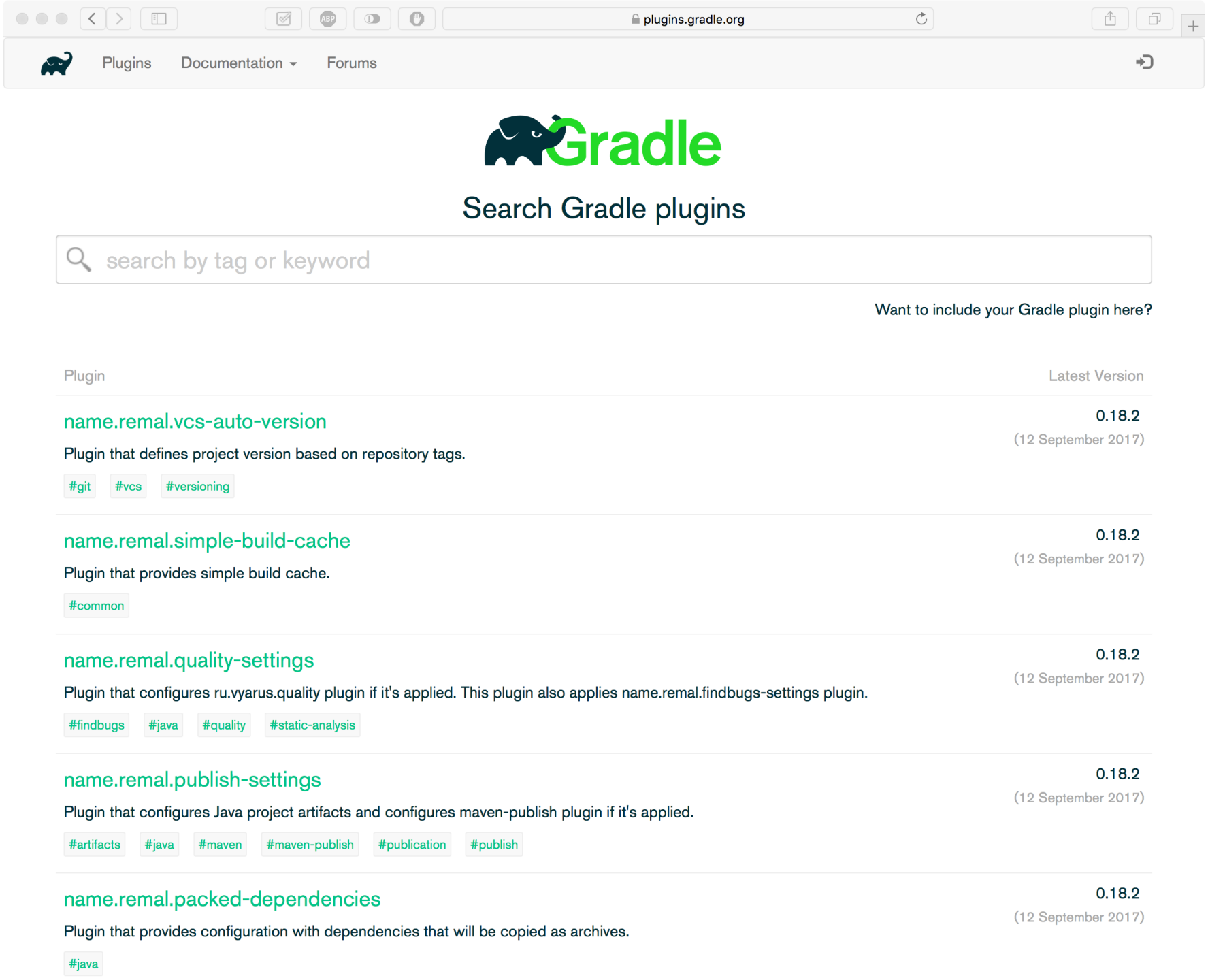
USE THE PLUGINS BLOCK DSL IF IN A SINGLE PROJECT.

PLUGINS BLOCK DOES NOT WORK IN SECONDARY SCRIPTS.

USE THE OLD-SCHOOL SYNTAX IF IN A MULTI-PROJECT AND NOT ALL SUBPROJECTS REQUIRE THE SAME PLUGINS.

7

DON’T REINVENT THE WHEEL, APPLY PLUGINS INSTEAD


plugins.gradle.org
Plugins
Documentation
Forums
Gradle
Search Gradle plugins
search by tag or keyword
Want to include your Gradle plugin here?
Plugin
Latest Version
name.remal.vcs-auto-version
0.18.2
(12 September 2017)
Plugin that defines project version based on repository tags.
#git
#vcs
#versioning
name.remal.simple-build-cache
0.18.2
(12 September 2017)
Plugin that provides simple build cache.
#common
name.remal.quality-settings
0.18.2
(12 September 2017)
Plugin that configures ru.vyarus.quality plugin if it's applied. This plugin also applies name.remal.findbugs-settings plugin.
#findbugs
#java
#quality
#static-analysis
name.remal.publish-settings
0.18.2
(12 September 2017)
Plugin that configures Java project artifacts and configures maven-publish plugin if it's applied.
#artifacts
#java
#maven
#maven-publish
#publication
#publish
name.remal.packed-dependencies
0.18.2
(12 September 2017)
Plugin that provides configuration with dependencies that will be copied as archives.
#java

USEFUL PLUGINS

Versions

```
id 'com.github.ben-manes.versions' version '0.15.0’
```

License

```
id 'com.github.hierynomus.license' version '0.11.0’
```

Versioning

```
id 'net.nemerosa.versioning' version '2.6.1'
```

FILTERING VERSIONS

```
apply plugin: 'com.github.ben-manes.versions'
dependencyUpdates.resolutionStrategy = {
    componentSelection { rules ->
        rules.all { selection ->
            boolean rejected = ['alpha', 'beta', 'rc’].any { qualifier ->
                selection.candidate.version ==~ /(?i).*[.-]${qualifier}[.\d-]*/
            }
            if (rejected) {
                selection.reject('Release candidate')
            }
        }
    }
}
```

8

KEEP ALL VERSIONS IN ONE PLACE

IN GRADLE.PROPERTIES (1)

```
awaitilityVersion  = 3.0.0
bootstrapfxVersion = 0.2.1
guiceVersion       = 4.1.0
hamcrestVersion    = 1.3
ikonliVersion      = 1.9.0
jacksonVersion     = 2.8.7
jdeferredVersion   = 1.2.5
jukitoVersion      = 1.5
junitVersion       = 4.12
lombokVersion      = 1.16.16
mbassadorVersion   = 1.3.0
miglayoutVersion   = 5.0
mockitoVersion     = 2.8.9
reactfxVersion     = 2.0-M5
retrofitVersion    = 2.2.0
slf4jVersion       = 1.7.25
testfxVersion      = 4.0.6-alpha
wiremockVersion    = 2.5.1
```

IN BUILD.GRADLE (2)

```
dependencies {
    compile "net.engio:mbassador:$mbassadorVersion"
    compile "org.reactfx:reactfx:$reactfxVersion"
    compile "org.slf4j:slf4j-api:$slf4jVersion"
    compile "org.jdeferred:jdeferred-core:$jdeferredVersion"
    compile "com.squareup.retrofit2:retrofit:$retrofitVersion"
    compile "com.squareup.retrofit2:converter-jackson:$retrofitVersion"
    compile "com.fasterxml.jackson.core:jackson-core:$jacksonVersion"
    compile "com.fasterxml.jackson.core:jackson-annotations:$jacksonVersion"
    compile "com.fasterxml.jackson.core:jackson-databind:$jacksonVersion"
    /* and more ... */
}
```

9

DEPENDENCY VERSIONS CONVERGENCE

FORCE VERSIONS AND CHECK

```
configurations.all {
    resolutionStrategy.force "jdepend:jdepend:$jdependVersion",
        "com.google.guava:guava:$guavaVersion",
        "junit:junit:$junitVersion",
        "cglib:cglib-nodep:$cglibVersion",
        "org.asciidoctor:asciidoctorj:$asciidoctorjVersion",
        "org.codehau.groovy:groovy-all:$groovyVersion",
        "org.slf4j:slf4j-api:$slf4jVersion",
        "org.slf4j:slf4j-simple:$slf4jVersion",
        "org.easytesting:fest-util:$festUtilVersion"

    resolutionStrategy.failOnVersionConflict()
}
```

10

SUPPLY MORE INFORMATION TO DEVELOPERS

MANIFEST ENTRIES

ENRICH JAR MANIFESTS WITH ADDITIONAL ENTRIES SUCH AS

'Built-By': System.properties['user.name'],

'Created-By': "${System.properties['java.version']} (${System.properties['java.vendor']} ${System.properties['java.vm.version']})".toString(),

'Build-Date': buildDate,

'Build-Time': buildTime,

'Build-Revision': versioning.info.commit,

'Specification-Title': project.name,

'Specification-Version': project.version,

'Specification-Vendor': project.vendor,

'Implementation-Title': project.name,

'Implementation-Version': project.version,

'Implementation-Vendor': project.vendor

11

INCREMENTAL BUILDS

INCREMENTAL BUILD

ACTIVATED BY ADDING –t TO THE COMMAND LINE.

EXECUTES A GIVEN COMMAND WHENEVER ITS RESOURCES (INPUTS) CHANGE.

AVOID INVOKING THE CLEAN TASK AS MUCH AS YOU CAN.

12

AGGREGATE CODE COVERAGE REPORTS

JACOCO OR BUST

STEP 1: DEFINE A LIST OF PROJECTS TO INCLUDE

```
ext {
    projectsWithCoverage = []
    baseJaCocoDir = "${buildDir}/reports/jacoco/test/"
    jacocoMergeExecFile = "${baseJaCocoDir }jacocoTestReport.exec"
    jacocoMergeReportHTMLFile = "${baseJaCocoDir }/html/"
    jacocoMergeReportXMLFile = "${baseJaCocoDir}/jacocoTestReport.xml"
}
```

JACOCO OR BUST

STEP 2: ADD PROJECT TO COVERAGE LIST

```
jacocoTestReport {
    group = 'Reporting'
    description = 'Generate Jacoco coverage reports after running tests.'
    additionalSourceDirs = project.files(sourceSets.main.allSource.srcDirs)
    sourceDirectories = project.files(sourceSets.main.allSource.srcDirs)
    classDirectories = project.files(sourceSets.main.output)
    reports {
        xml.enabled = true
        csv.enabled = false
        html.enabled = true
    }
}
projectsWithCoverage << project
```

JACOCO OR BUST

STEP 3: DEFINE AGGREGATE TASKS IN ROOT PROJECT

```
evaluationDependsOnChildren() // VERY IMPORTANT!!


task jacocoRootMerge(type: org.gradle.testing.jacoco.tasks.JacocoMerge) {
    dependsOn = projectsWithCoverage.test +
projectsWithCoverage.jacocoTestReport
    executionData = projectsWithCoverage.jacocoTestReport.executionData
    destinationFile = file(jacocoMergeExecFile)
}
```

JACOCO OR BUST

STEP 3: DEFINE AGGREGATE TASKS IN ROOT PROJECT

```
task jacocoRootMergeReport(dependsOn: jacocoRootMerge, type: JacocoReport) {
    executionData projectsWithCoverage.jacocoTestReport.executionData
    sourceDirectories = projectsWithCoverage.sourceSets.main.allSource.srcDirs
    classDirectories = projectsWithCoverage.sourceSets.main.output

    reports {
        html.enabled = true
        xml.enabled = true
        html.destination = file(jacocoMergeReportHTMLFile)
        xml.destination = file(jacocoMergeReportXMLFile)
    }
}
```

13

MIND THE TARGET JAVA VERSION

JDK8 VS JDK7

JOINT COMPILING GROOVY CODE MAY GIVE YOU HEADACHES IF THE SOURCE/TARGET COMPATIBILITY IS NOT SET RIGHT

```
tasks.withType(JavaCompile) { t ->
    t.sourceCompatibility = project.sourceCompatibility
    t.targetCompatibility = project.targetCompatibility
}

tasks.withType(GroovyCompile) { t ->
    t.sourceCompatibility = project.sourceCompatibility
    t.targetCompatibility = project.targetCompatibility
}
```

JDK8 JAVADOC TROUBLES

JAVADOC IN JDK8 IS VERY PEDANTIC

```
if (JavaVersion.current().isJava8Compatible()) {
    tasks.withType(Javadoc) {
        options.addStringOption('Xdoclint:none', '-quiet')
    }
}
```

COMPILE WARNINGS

ACTIVATE COMPILER WARNINGS AT WILL

```
tasks.withType(AbstractCompile) {
    if (rootProject.hasProperty('lint') && rootProject.lint.toBoolean()) {
        options.compilerArgs = [
            '-Xlint:all', '-Xlint:deprecation', '-Xlint:unchecked'
        ]
    }
}
```

```
$ gw –Plint=true compile
```

14

GET READY FOR JDK9

JDEPS TO THE RESCUE

VERIFIES PRODUCTION CLASSES AND DEPENDENCIES

```
plugins {
    id 'org.kordamp.jdeps' version '0.2.0'
}
```

```
$ gw jdeps
```

15

PLUGINS! PLUGINS! PLUGINS!

license

versions

stats

bintray

shadow

izpack

java2html

git

coveralls

asciidoctor

jbake

markdown

gretty

nexus

apt

ossindex

versioning

pitest

semantic-release

clirr

compass

flyway

jmh

macappbundle

osspackage

jnlp

spotless

dependency-management

sshoogr


KEEP
CALM
AND
OPEN
SOURCE

HTTP://ANDRESALMIRAY.COM/NEWSLETTER

HTTP://ANDRESALMIRAY.COM/EDITORIAL

Willkommen an der BaselOne 2017

http://baselone.ch

THANK YOU!

ANDRES ALMIRAY

@AALMIRAY

ANDRESALMIRAY.COM