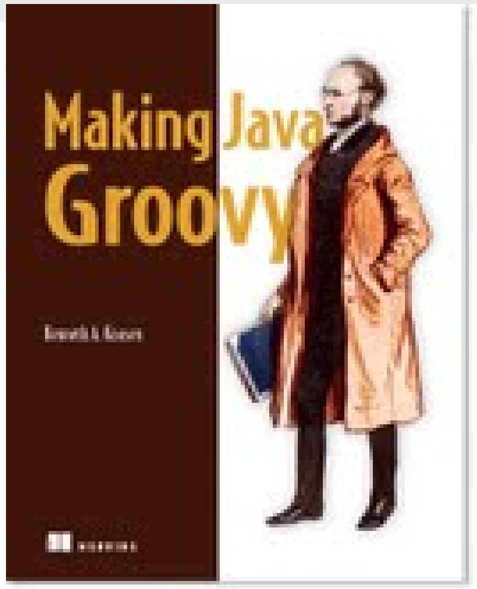
Gradle Fundamentals

Who Am I?

Ken Kousen
President, Kousen IT, Inc.
http://www.kousenit.com
ken.kousen@kousenit.com
@kenkousen

Making Java Groovy
http://manning.com/kousen



Build Process

Goals:

- Automated
- Managed dependencies
- Easy to customize

Existing Tools

Ant / Ivy
- Define low-level tasks in XML
- Ivy for dependencies

Maven
- Define high-level tasks in XML
- "Opinionated" architecture

Gradle

Build tool written in Groovy (and Java)

Build file uses Groovy

Constructs DAG

  Directed Acyclic Graph of tasks

Plugin-based architecture

Gradle

DSL for builds

- Domain Specific Language

Full class structure:

- Project, Task, ...
- Access from build as needed

"Java is a DSL for generating stack traces"
"JavaScript is a DSL for detecting browser bugs"
"Maven is a DSL for downloading the Internet"

Installing Gradle

Gradle home page: http://gradle.org
    (Gradle wrapper discussed later)

JDK 1.5+ required
Groovy NOT required
    Included in install

Unzip, set GRADLE_HOME environment var
Add bin to path and ready to run

Testing Installation

```
gradle -v
```

Gives versions of:

- Gradle
- Groovy
- Ant
- Ivy
- JVM
- OS

Documentation

Distribution includes:

- User guide
- DSL guide
- JavaDocs
- GroovyDocs

Also, *"Building and Testing with Gradle"*

- Tim Berglund and Matthew McCullough
- Available free online (registration required)

Trivial Build

Use Maven project structure

- `src/main/java`
- `src/test/java`
- Easy to change later

Use java plugin

Slightly more complex

Add testing

```
repositories {
   mavenCentral()
}

dependencies {
   testCompile 'junit:junit:4.9'
}
```

More additions

Add IDE support and Groovy

```
apply plugin:'groovy'
apply plugin:'eclipse'
apply plugin:'idea'

repositories {
    mavenCentral()
}

dependencies {
    groovy 'org.codehaus.groovy:groovy-all:1.8.6'
}
```

Stages

Based on plugins

Java stages:

```
:compileJava
:processResources
:classes
:jar
:assemble
:compileTestJava
:processTestResources
:testClasses
:test
:check
:build
```

Groovy plugin

Added stages

```
:compileJava
:compileGroovy
...
:compileTestJava
:compileTestGroovy
```

Alternative Layout

For standard Eclipse projects

- Use `groovyc` for both Java and Groovy

```
sourceSets {
   main {
      java { srcDirs = [] }
      groovy { srcDir 'src' }
   }
   // same for test
}
```

Other Eclipse details

Generate Eclipse project files

Sets classpath to gradle cached jars

```
gradle cleanEclipse eclipse
```

Web projects:

```
apply plugin:'eclipse-wtp'
```

Similar approach for IntelliJ IDEA

Defining Tasks

```
task hello {
   doLast {
      println 'Hello, World!'
   }
}

task hello << {
   println 'Hello, World!'
}
```

Add dependencies

```
task intro(dependsOn: 'hello') << {
   println 'Hello from intro'
}
```

Tasks

Action methods: `doFirst, doLast` (or <<)

Properties:

```
dependsOn, logger, name
inputs, outputs
group, project, enabled
```

Can assign arbitrary properties as well

Inputs and Outputs

Gradle decides when to skip tasks

Declare `inputs` and `outputs`
- Tasks only run when inputs change

Multiproject Builds

Use hierarchical structure by default

Wrapper

Allows clients to use gradle before install

```
task wrapper(type:Wrapper) {
  gradleVersion = ...
}
```

Generates `gradlew, gradlew.bat`

Plugins

Comes with many standard plugins

- java, groovy, scala, antlr
- war, ear, jetty, maven, osgi
- eclipse, eclipse-wtp, idea

Third-party plugins on wiki

Examples

Samples directory with distribution

Spock-Example
spockframework

Spring-Framework

Conclusions

Gradle builds written in Groovy

- Uses "builder" syntax with DSL
- Full Groovy / Java libraries available

Constructs and executes Directed Acyclic Graph

Rich set of plugins

Easy to customize

Session Evaluations

Please complete your session evals