Enter the Gradle

Hans Dockter
CEO, Gradleware
Founder Gradle
hans.dockter@gradleware.com

What is Gradle?

- A general purpose build system
- It comes with a Groovy DSL and a Java core.
- Provides build-in support for **Java**, **Groovy**, **Scala**, **Web**, **OSGi** projects.
- Gradle provides exciting solutions for many of the big pain points you often have with current builds.
  - Maintainability
  - Performance
  - Usability

What is Gradle?

Gradle is declarative

You specify the **WHAT**

Gradle figures out the **HOW**

Extensible

Build Language

instead of a

Build Framework

Extending Gradle

- Deep Configuration API
- Deep Execution API
- Rich API
- Extendable Domain Objects
- Custom Tasks
- Custom Plugins

Custom Declarative Elements

```
usePlugin ‘editions’

productEditions {
    enterprise core, plugins, powerAddons
    public core, plugins, openApi
}
```

```
> gradle enterpriseEditionZip
> gradle publicEditionTar
```

Gradle

is

declarative

without

being rigid

The Build Language

Source Sets

Custom Tasks

Archives

Dependencies

Artifacts

Projects

Configurations

Plugins

XML and the What

It’s the design, stupid!

Please

no

messy

build scripts

Organizing Build Logic

- Separate Imperative from Declarative
- Custom Tasks/Plugins
- BuildSrc
- Jar

Build Script Libraries

```
buildscript {
    repositories {
        mavenCentral()
    }
    dependencies {
        classpath 'commons-math:commons-math:1.1'
    }
}

task math << {
    org.apache.commons.math.fraction.Fraction lhs = new
        org.apache.commons.math.fraction.Fraction(1, 3);
    // do something
}
```

It’s the performance, stupid!

Should **clean** be **required** for a **reliable build**?

(Hint: We have the 21st century)

Speed Improvements

- Incremental Build
- Parallel Testing
- Soon: Parallel Builds, Distributed testing/builds
- Rich Model

Gradle is a Build Integration Tool

Build Integration Features

- Ant Tasks
- Deep import of Ant builds
- Retrieve/Deploy to Maven/Ivy repositories
- Autogeneration of pom.xml/ivy.xml
- Future: Deep import of Maven builds

Build Migration

- Mission Critical!
- Very expensive if the build system can’t adapt to the existing project layout:
  - Freeze
  - Project automation not working for a while
  - Different branches (unreliable, hard to compare, ...)
- Gradle’s suppleness enables baby steps.
  - Gradle can adapt to any project layout.
  - No separate branches
  - Comparable --> You can write tests

Dependency Management

- Transitive Dependencies
- Repository less dependencies are 1st class citizens.
- Excludes per configuration or dependency
- Very flexible repository handling
- Based on Apache Ivy
- Powerful API
- Much More ...

Deep Integration with Ant Builds

```
<project>
   <target name="hello" depends="intro">
      <echo>Hello, from Ant</echo>
   </target>
</project>
```

```
ant.importBuild 'build.xml'

hello.doFirst { println 'Here comes Ant' }
task intro << { println 'Hello, from Gradle'}
```

```
> gradle hello
Hello, from Gradle...Here comes Ant...
[ant:echo] Hello, from Ant
```

Build Eco System

artifactory

- Supports Maven/Ant/Gradle with pom or ivy.
- Supports any repository layout
- Very advanced features
- Gradle Artifactory plugin
- Integrated in Gradle Hudson plugin.
- Gradle Inc. has business partnership with JFrog.

Process is very important.

But it is all about

YOUR

process

Their is

no one-size-fits-all

project structure

for the

enterprise.

The physical structure of your projects should be determined by **your requirements**.

(What, if not?)

Be as

rigid

as

YOU

want

FRAME-
WORKITIS

Fighting with a Framework

Me in 2007
There was no way I could win this fight.
We need a new build system.

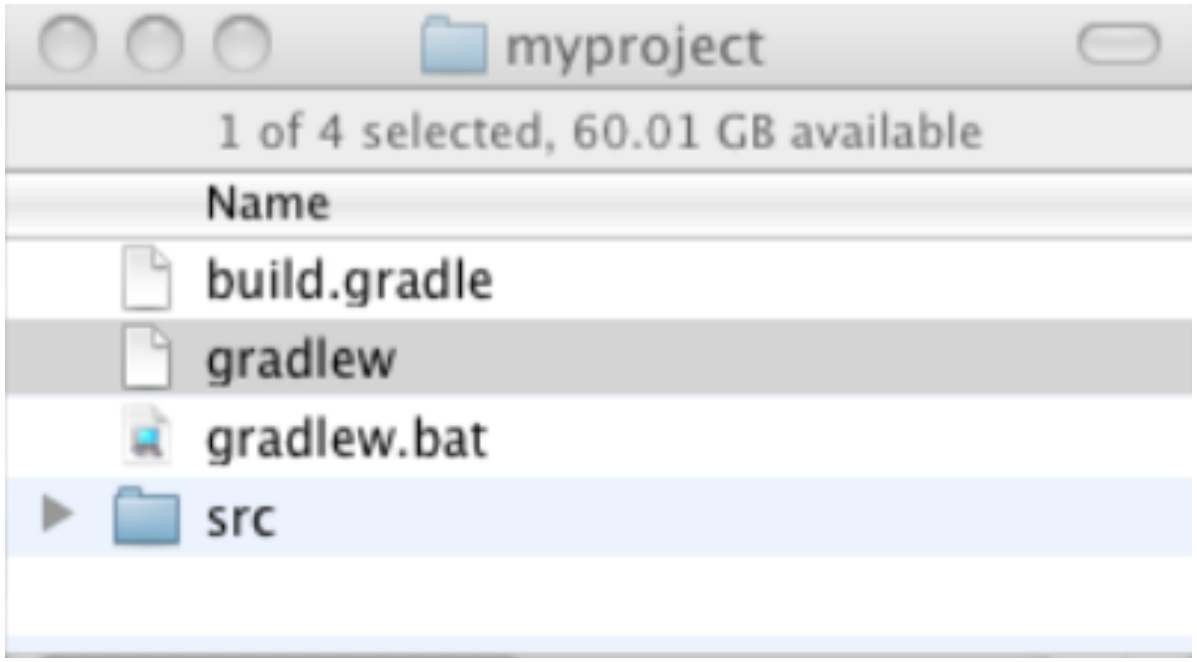
Extensible

Build Language

instead of a

Build Framework

Gradle Wrapper



```
>./gradlew assemble
```

Project Background

- Very active community (mailing-list, patches, issues)
- Apache v2 license.
- Excellent user’s guide (200+ pages) + many samples
- Frequent releases, multiple commits per day
- Quality is king:
  - 2800 unit tests, Many hundreds of integration test
  - Healthy codebase
  - low defect rate

News & Roadmap

- Ken Sipe, Peter Niederwieser and Szczepan Faber entered the Gradle!
- Sonar Support in gradle-1.0-milestone-2
- Very cool new DSL reference
- Excellent Eclipse support work in progress
- 4-6 weekly milestones towards gradle-1.0

Q&A