## Gradle

Treating the build as a first-class object

Michael Easter

@codetojoy













- <javac>
- <condition>
- < pathelement >
  - <condition>
    - <path><javac>
      - <condition>
      - < pathelement >
        - <condition>
          - <path>







## Basics

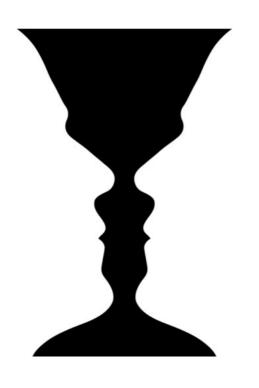
- Free, open-source, Apache 2 license
- Core in Java (1.5+), config in Groovy (embedded)
- Combine fluidity of Groovy with powerful lifecycle (DAG)
- Strong integration with Ant, some integration with Mayen

## Tasks

- Gradle unites Ant's ideas of target and task
- full API exposed (e.g. doFirst, doLast)
- dynamic properties, dependencies
- advanced features: types, predicates, and rules

# Projects

- build script configures a Project object
- full API, and dynamic properties
- consider a project as a collection of task objects
- usually has dependencies and artifacts





Context and Perception

pojo.Musician pojo.MusicianTestCase

hibernate.cfg.xml Musician.hbm.xml

common.SimpleTxnManager

# Plugins

- decoupled from core of Gradle
- each plugin uses the Project and Task API:
  - define tasks and dependencies
  - add dependency configurations
  - define a convention object

#### Java Plugin: Conventions

src/main/java
src/main/resources

src/test/java
src/test/resources

build

#### Java Plugin: Conventions

src/main/java
src/main/resources

A source set is a group of files to be compiled / executed together

build

# Java plugins: source sets

- a group of files to be compiled together
- has compile and runtime classpaths
- plugin provides 'main' and 'test' but you can define others
- any source set can be customized

#### Java Plugin: Tasks

- clean

- compileXJava

- processXResources

- test

- many, many more!

X is any sourceset. 'main' is implicit: compileJava

web.MusicianServlet

pojo.Musician pojo.MusicianTestCase

hibernate.cfg.xml Musician.hbm.xml

common.SimpleTxnManager

#### War Plugin: Conventions

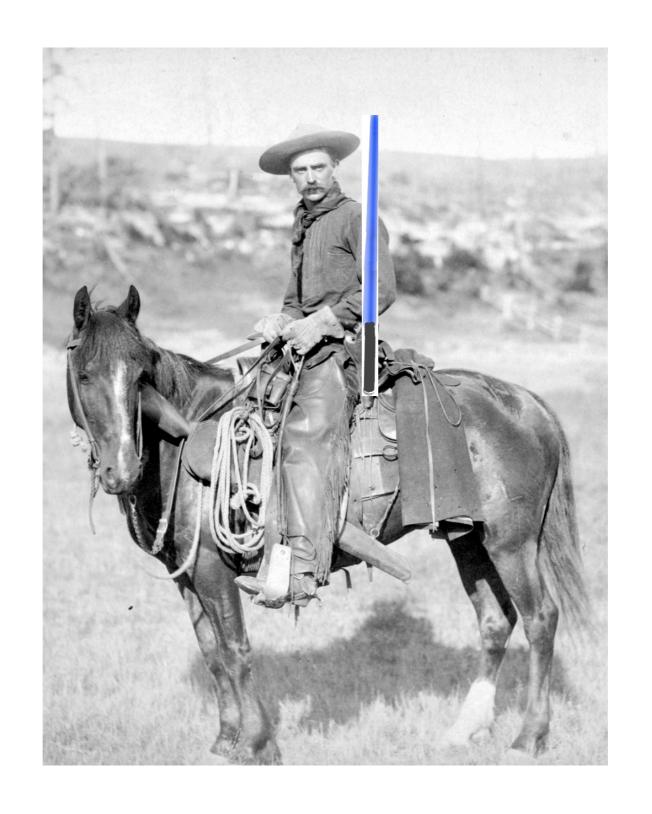
src/main/webapp

src/main/java
src/main/resources

src/test/java
src/test/resources

simply builds upon the Java plugin





## wrap up

- Fluidity and power offers first-class builds
- Built on Ivy, strong support for Ant, Maven repositories
- Strong documentation
- Momentum: Hibernate, Android SDK tools

## References

- http://www.gradle.org
- http://martinfowler.com/articles/ rake.html#DependencyBasedProgramming
- http://is.gd/bR6fj (Ken Sipe on Gradle and IntelliJ)

#### Appendix: Gradle Terms

**Task**: unit of work, similar to Ant's target

**Project**: defined and configured by the build script in a given directory

Configuration: simply a named group of files

**Repository**: normal usage, a location where artifacts are stored (e.g. Maven central)

**Dependency**: normal usage, an entity necessary for an operation (e.g. a Hibernate jar)

**Transitive Dependency**: e.g. a jar that is indirectly required, due to the needs of a direct requisite. Transitive dependencies are typically not needed at compile-time, but are required at run-time.

**Dependency Configuration**: simply a named group of files, where the files are dependencies (!)

**Artifact**: a file that is generated and provided to another client. (e.g. a war file). Generally, a 'dependency' is input to a project and, if applicable, an 'artifact' is output.

**Plugin**: a domain-oriented collection of tasks, along with task dependencies, dependency configurations, and a convention object

**Convention Object**: the mechanism where a user can easily override default values for a plugin. e.g. If a user changes a property in a convention object, it will apply to many tasks. The properties and their application to tasks are plugin specific.

**Sourceset**: part of the convention object in the Java plugin, a sourceset is a group of files to be compiled and executed together. (e.g. src/main in a typical Java project)

**Configuration Injection**: The mechanism used by Gradle to configure multiple projects, as an alternative to inheritance.

### Appendix: Options

-t	list tasks
-b	specify build file, if not build.gradle
-m	dry-run: task actions disabled
gui	launch GUI
-n	show project dependencies
-X	exclude given task from execution

#### Photo Credits

- American Cowboy, 1888, public domain: <a href="http://is.gd/bGHT3">http://is.gd/bGHT3</a>
- Raft of the Medusa, Théodore Géricault, public domain : <a href="http://is.gd/bGHWs">http://is.gd/bGHWs</a>
- Ant, Clockworks, Watch, Train, Melting Clocks, Vase in Faces: iStockphoto.com
- Light Saber based on public domain work by Søren Peo Pedersen, Creative Commons