The new Android Build System!

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

= 

Gradle +
Android Build System

= 

Gradle +

most advanced build system
(a.k.a. best)
Android Build System

= 

Gradle + Android Gradle Plugin + IDE integration

most advanced build system (a.k.a. best)
Android Build System

= 

Gradle + Android Gradle Plugin + IDE integration

most advanced build system (a.k.a. best)

Magnificent piece of engineering by the Android Team
Show case for a state of the art build system
Why a new system?
Demo

Basic Android
• Product Flavors
• Build Types
• Build Variants
• SourceSets
• Library Projects
• Testing
Product Flavors

- A way to generate several versions of the same app
- Source code + resources overlay
  - Also custom dependencies
- Customize app parameters:
  - package name
  - min/targetSdkVersion
  - versionCode/Name
  - signing info
BuildType

- Controls how an app is built
- Source code + resources overlay
  - Also custom dependencies
- Customize app parameters:
  - debuggable flag
  - native debug compilation flag
  - package suffix
  - debug signing
  - proguard options
- 2 default types: debug & release
- orthogonal to product flavors
Build Variant

- Build is always a flavor + a type

<table>
<thead>
<tr>
<th></th>
<th>Debug</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free</td>
<td>Free-Debug</td>
<td>Free-Release</td>
</tr>
<tr>
<td>Paid</td>
<td>Paid-Debug</td>
<td>Paid-Release</td>
</tr>
</tbody>
</table>

Rules to manage overlay
build type > flavor > main
Demo

Build Variants
### Build Variant, cont'd

- **Product Flavor groups**
  - add extra dimensions of flavors

<table>
<thead>
<tr>
<th></th>
<th>Debug</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free</td>
<td>x86</td>
<td>Free-x86-Debug</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Free-x86-Release</td>
</tr>
<tr>
<td>ARM</td>
<td>Free-Arm-Debug</td>
<td>Free-Arm-Release</td>
</tr>
<tr>
<td>Paid</td>
<td>x86</td>
<td>Paid-x86-Debug</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Paid-x86-Release</td>
</tr>
<tr>
<td>ARM</td>
<td>Paid-Arm-Debug</td>
<td>Paid-Arm-Release</td>
</tr>
</tbody>
</table>
SourceSets

```
src/
  main/
    AndroidManifest.xml
  java/
  resources/
  assets/
  ...
  debug/
  ...
  release/
  ...
  paid/
  ...
  free/
  ...
```
Library Projects

• Binary bundle
• Uploadable to Maven, Ivy, ...
  ○ Declare their own dependencies
• Contain:
  ○ Compiled Code
  ○ Resources
  ○ + more
    ▪ custom proguard rules?
    ▪ custom lint checks?
    ▪ annotation processor?
    ▪ IDE plugin?
Testing

- Embedded in the project
- test sourceset is default
- test<flavor> sourceset allows flavor specific tests
- APK for library project embed the library to test
Dynamic Build-By-Convention
Performance
Demo

 Incremental Build
apply plugin: 'java'

version = '1.0'

sourceCompatibility = '1.6'

repositories {
    mavenCentral()
}

dependencies {
    compile 'commons-collections:commons-collections:3.2.1'
    testCompile 'junit:junit:4.7'
}
Gradle Daemon
Relax IDE, ain’t need to be a build tool anymore.
Close to canonical build

Relax IDE, ain’t need to be a build tool anymore.
Close to canonical build

Optimized Performance

Relax IDE, ain’t need to be a build tool anymore.
Relax IDE, ain’t need to be a build tool anymore.

Close to canonical build

Optimized Performance

More automation
Close to canonical build

Optimized Performance

Relax IDE, ain’t need to be a build tool anymore.

More automation

More feedback
Tool Integration
=
First Class Gradle citizen
Provisioning

Tool Integration

= First Class Gradle citizen
Tool Integration

= First Class Gradle citizen
Tool Integration

= First Class Gradle citizen

Provisioning

Tailored Models

Back & Forwards compatibility
Deep IDE customization
Declarative Elements

Deep IDE customization
Declarative Elements

Deep IDE customization

Programmatic Access
eclipse.classpath.file {
    whenMerged { classpath ->
        classpath.entries.findAll { entry ->
            entry.kind == 'lib'
        }*.exported = false
    }
}
Android Studio based on IntelliJ is available!
Version 0.4

- Proguard Support
- NDK Support
- Lint Support

Version 0.5

- Emma
- AVD Management

Version 0.6

- T.b.d.
Demo

Gradle Wrapper
apply plugin: 'java'
apply from: '../helper-plugins/test-generator.gradle'

version = '1.0'

repositories {
    mavenCentral()
}

dependencies {
    compile group: 'commons-collections', name: 'commons-collect'
testCompile group: 'junit', name: 'junit', version: '4.7'
}
Build Environment

Project specific gradle.properties

```
org.gradle.daemon=true
org.gradle.java.home=/Library/Java/JavaVirtualMachines/1.7.0.jdk/Contents/Home
org.gradle.jvmargs=-Xmx512M
```
Gradle Summit 2013
June 13-14
Santa Clara, CA, USA

http://gradlesummit.com
http://tools.android.com/tech-docs/new-build-system

http://www.gradle.org

http://m.gradleware.com
http://gradleware.com/subscribe
Register for 'The new Android Build System' webinar on April 16th
http://gradle.org/webinars

Android Online Training
http://gradleware.com/trainings

http://gradleware.com/services
Questions?

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