Gradle Recipes for Android

Contact Info

Ken Kousen

ken.kousen@kousenit.com

@kenkousen

http://www.kousenit.com (home page)

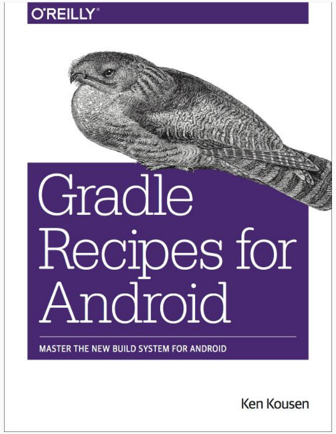
http://kousenit.wordpress.com (blog)

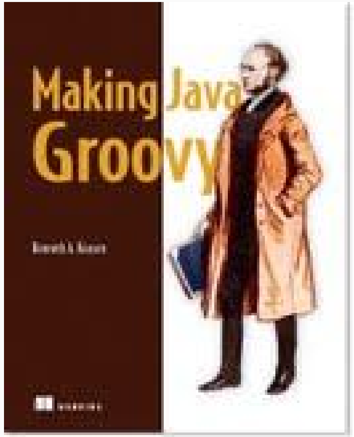
https://github.com/kousen (repositories)

Publications

O'Reilly video courses:

- Groovy Programming Fundamentals
- Practical Groovy Programming
- Mastering Groovy Programming
- Learning Android
- Practical Android
- Gradle Fundamentals
- Gradle for Android
- Spring Framework Essentials


O'REILLY
Gradle Recipes for Android
MASTER THE NEW BUILD SYSTEM FOR ANDROID
Ken Kousen



Basics

Android plugin for Gradle

- Added via `buildscript`

- Lots of customization

Basics

```
buildscript {
    repositories { jcenter() }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.0.0-beta4'
    }
}

apply plugin: 'com.android.application'
```

Properties in build.gradle

```
android {
    versionCode …
    versionName …
}
```

or even in gradle.properties

Multi-project Builds

By default, apps in AS are multi-project builds

```
build.gradle

settings.gradle
```

Can add additional libraries, other modules, and more

Changing the Gradle version

```
task wrapper(type: Wrapper) {
    gradleVersion = 2.11
}
```

Or edit distribution URL in gradle-wrapper.properties

```
distributionUrl=https\://services.gradle.org/distributions/gradle-2.11-all.zip
```

Build Types

Two default build types:

`debug`

`release`

Configuring Build Types

Use buildTypes section of build.gradle

```
buildTypes {
    release { … }
    debug { … }
}
```

Can also add custom build types that way

Configuring Build Types

Each build type defines a source set

`src/main/ …`

`src/debug/ …`

`src/release/ …`

`src/androidTest/…` (discussed below)

Configuring Build Types

Resources in build type source sets

replace their counterparts in main

Java classes *conflict*, however

Define class in each, or just in main

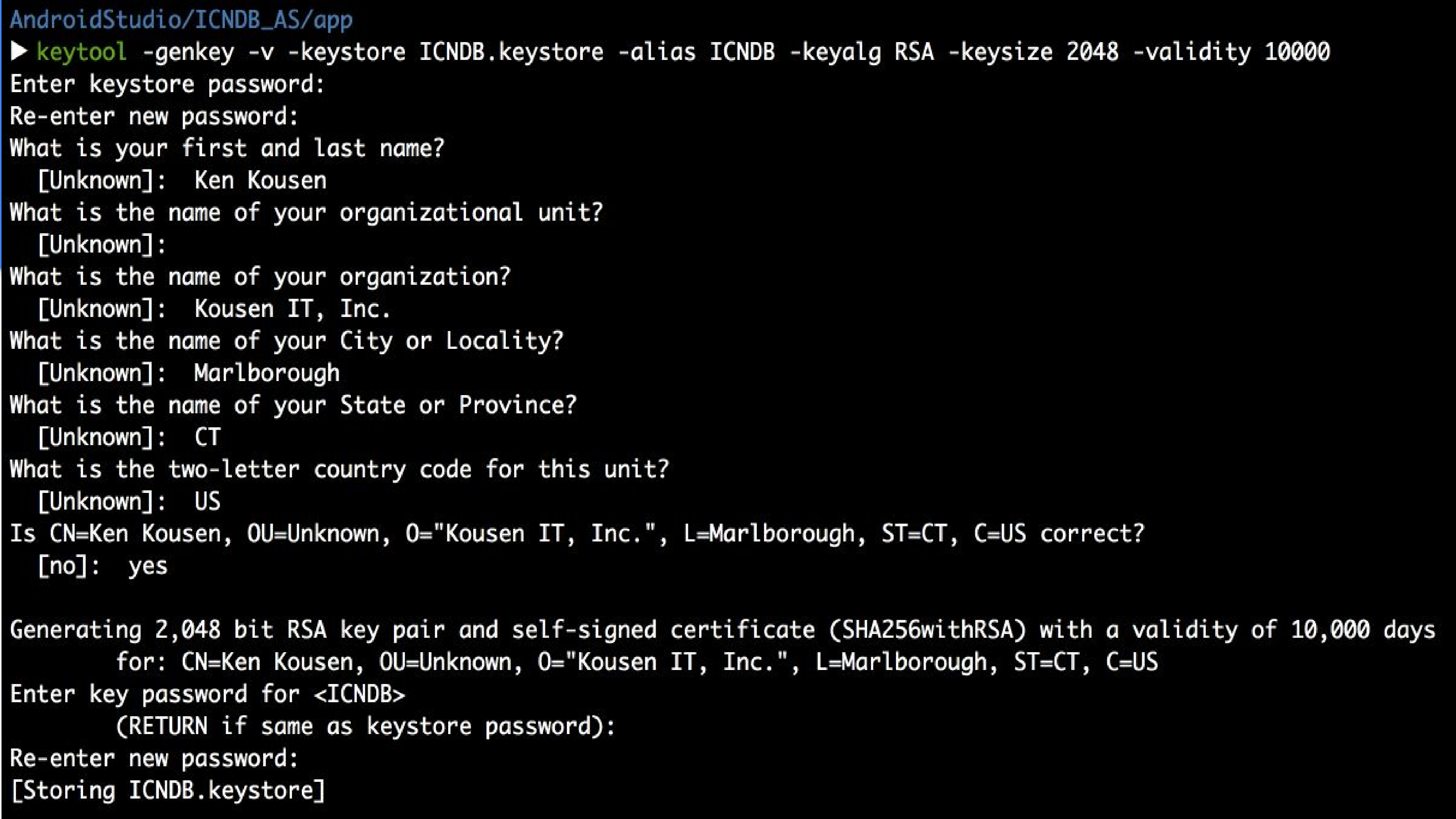
Generating a Release

Can't assemble a release until

you can sign it

Signing Your App

Use Java's **`keytool`** to generate cert



```
AndroidStudio/ICNDB_AS/app
▶ keytool -genkey -v -keystore ICNDB.keystore -alias ICNDB -keyalg RSA -keysize 2048 -validity 10000
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  Ken Kousen
What is the name of your organizational unit?
  [Unknown]:
What is the name of your organization?
  [Unknown]:  Kousen IT, Inc.
What is the name of your City or Locality?
  [Unknown]:  Marlborough
What is the name of your State or Province?
  [Unknown]:  CT
What is the two-letter country code for this unit?
  [Unknown]:  US
Is CN=Ken Kousen, OU=Unknown, O="Kousen IT, Inc.", L=Marlborough, ST=CT, C=US correct?
  [no]:  yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10,000 days
        for: CN=Ken Kousen, OU=Unknown, O="Kousen IT, Inc.", L=Marlborough, ST=CT, C=US
Enter key password for <ICNDB>
        (RETURN if same as keystore password):
Re-enter new password:
[Storing ICNDB.keystore]
```

Signing Your App

```
signingConfigs {
    release {
        storeFile file('ICNDB.keystore')
        keyAlias 'ICNDB'
        storePassword 'gradle_rules'
        keyPassword 'carlos_ray_aka_chuck'
    }
}
```

Signing Your App

Passwords don't have to be in build file

- Can use system properties

- Can prompt user

- Can use gradle.properties

See docs for suggestions

Signing Your App

Add `signingConfig` to build type config

```
buildTypes {
    release {
        // ...
        signingConfig signingConfigs.release
    }
}
```

Signing Your App

The **`signingReport`** task shows details

Signing Your App

Invoke **`assembleRelease`** task

Resulting apk in `build/outputs/apk` folder

Flavors and Variants

buildTypes

- debug, release

flavors → different versions of same app

- arrogant, friendly, obsequious

Flavors and Variants

Each flavor generates an apk

Variants combine buildTypes and flavors

- debug+arrogant, debug+friendly, debug+obsequious
- release+arrogant, release+friendly, release+obsequious

Multiple Flavors

```
flavorDimensions 'attitude', 'client'

productFlavors {
    arrogant { dimension 'attitude' }
    …
    stark { dimension 'client' }
    wayne { dimension 'client' }
}
```

Custom Tasks

```
task copyApks(type: Copy, dependsOn: assembleDebug) {
    from("$buildDir/outputs/apk") {
        exclude '**/*unsigned.apk', '**/*unaligned.apk'
    }
    into '../apks'
}
```

Custom Tasks

```
task printVariantNames {
   doLast {
       android.applicationVariants.all { variant ->
           println variant.name
       }
   }
}
```

Testing

- Unit testing
- Functional testing
  - Robotium
  - Espresso

Testing

With Gradle:

Tests run on all connected devices simultaneously

Testing

Use `androidTest` source set

```
src/androidTest/java
```

Testing

Use `androidTest` source set

```
    src/androidTest/java
```

`androidTestCompile` dependencies

Testing

Use `androidTest` source set

```
    src/androidTest/java
```

Run **`connectedCheck`** task

References

Android new build system

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

Developer's Guide section on Gradle

- https://developer.android.com/sdk/installing/studio-build.html

Android Developers on G+

- https://plus.google.com/+AndroidDevelopers/posts

References

Gradle Recipes for Android

- http://shop.oreilly.com/product/0636920032656.do
- Ask me for a coupon code :)