

Joining The Winning Team

An Android Studio/Gradle presentation

Isn't Android Studio in Beta?

Android Studio is soooo buggy! I'm not going to leave the world's most stable IDE (Eclipse) to go to one that's only an "early access preview"!



Source: <http://developer.android.com/sdk/installing/studio.html>

Snap out of it!



Android Studio Fun Facts!

- Announced at Google I/O 2013
- IntelliJ by JetBrains + Android tools
- Created to integrate closely with a Gradle build system

Android Studio vs. Eclipse

- | | |
|--|---|
| 1. Automatic Code Completion | 1. Additional Typing Practice |
| 2. XML layout rendering in text input mode | 2. Nothing to distract you from your XML layout code |
| 3. Project -> Module | 3. Workspace -> Project |
| 4. Resource preview / automatically collapsed code | 4. Plain view of the code so you don't have to infer anything |
| 5. Automatically saves | 5. Manual saving |
| 6. Quicker searching mechanism of entire project | 6. Gives you more time for sword fights |
| 7. Fast and friendly support | 7. Plug-ins |
| 8. Free | 8. Free |

Gradle Fun Facts!

- Uses Groovy scripting language
- Dependency management with Maven
- Allows for different build types and build flavors
- Lint integrated with Android plug-in

Migrating

It's easy to migrate from Eclipse/Ant to Android Studio/Gradle.

Don't believe us?

Before we get started...

Requirements

- **Update Android SDK Build Tools to 19.1.0+**
(We suggest you also install Extras/Google Repository and Extras/Android Support Repository if you haven't.)
- **Download latest Android Studio** (0.6.1 currently)

Importing an Eclipse Project

Four Easy Steps:

1. Open AS and select Import Project...
2. Select project folder to import
3. Change path to new import folder - will not overwrite original folder!
4. Click Finish - leaving all checkboxes on Import dialog selected

Demo

Adding a little flavor...

Gradle supports product flavors

A product flavor defines a customized version of the application built by the project.

For example:

```
android {  
    productFlavors {  
        generic {  
            packageName "com.generic.app"  
            versionCode 33  
            versionName "1.0"  
        }  
  
        phunware {  
            packageName "com.phunware.app"  
            versionCode 33  
            versionName "2.0"  
        }  
    }  
}
```

Customizing the Flavor

A flavor can redefine almost anything about the build.

The following code will add another folder to the resource sourceSet for each of our new flavors:

```
android {
    sourceSets {
        generic {
            res.srcDirs = ['src/main/custom/generic/res']
        }
        phunware {
            res.srcDirs = ['src/main/custom/phunware/res']
        }
    }
}
```

This will merge the main resource sourceSet with the new sourceSet, overriding anything that may conflict.

Demo

