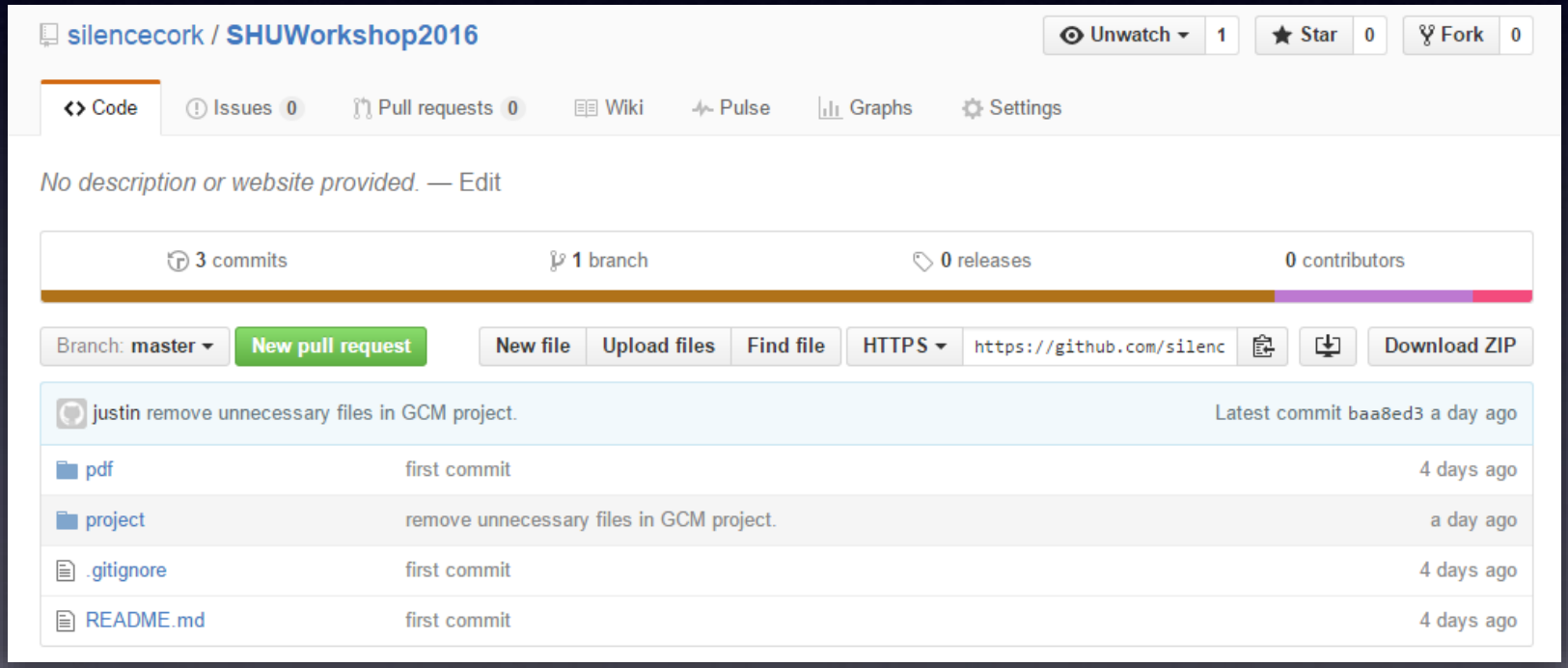
第二十一章

ANDROID STUDIO

一般下載取得教材

- 如何取得教材
  - https://github.com/silencecork/SHUWorkshop2016



使用GIT取得教材

- repository位置
  https://github.com/silencecork/SHUWorkshop2016.git
- 第一次取得教材
  - ```
    mkdir shuworkshop2016
    cd shuworkshop2016
    git clone <repository>
    ```
- 取得之後更新
  - ```
    cd shuworkshop2016
    git pull
    ```

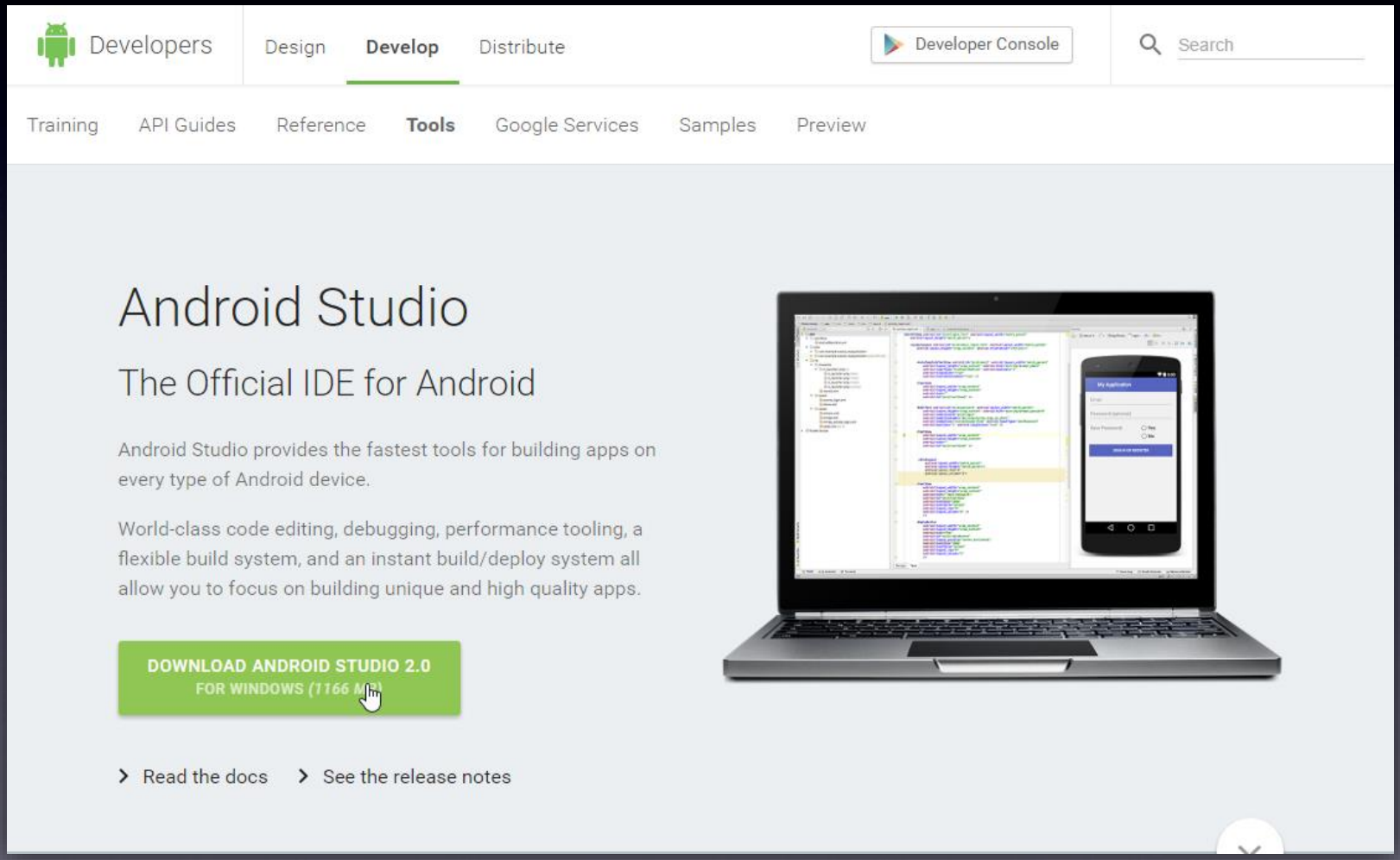
安裝ANDROID STUDIO

準備

- 安裝Java
  - http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
- 設定Java環境變數
  - http://0251077.blogspot.tw/2016/04/java-cmd.html
- 下載Android Studio
  - http://developer.android.com/sdk/index.html

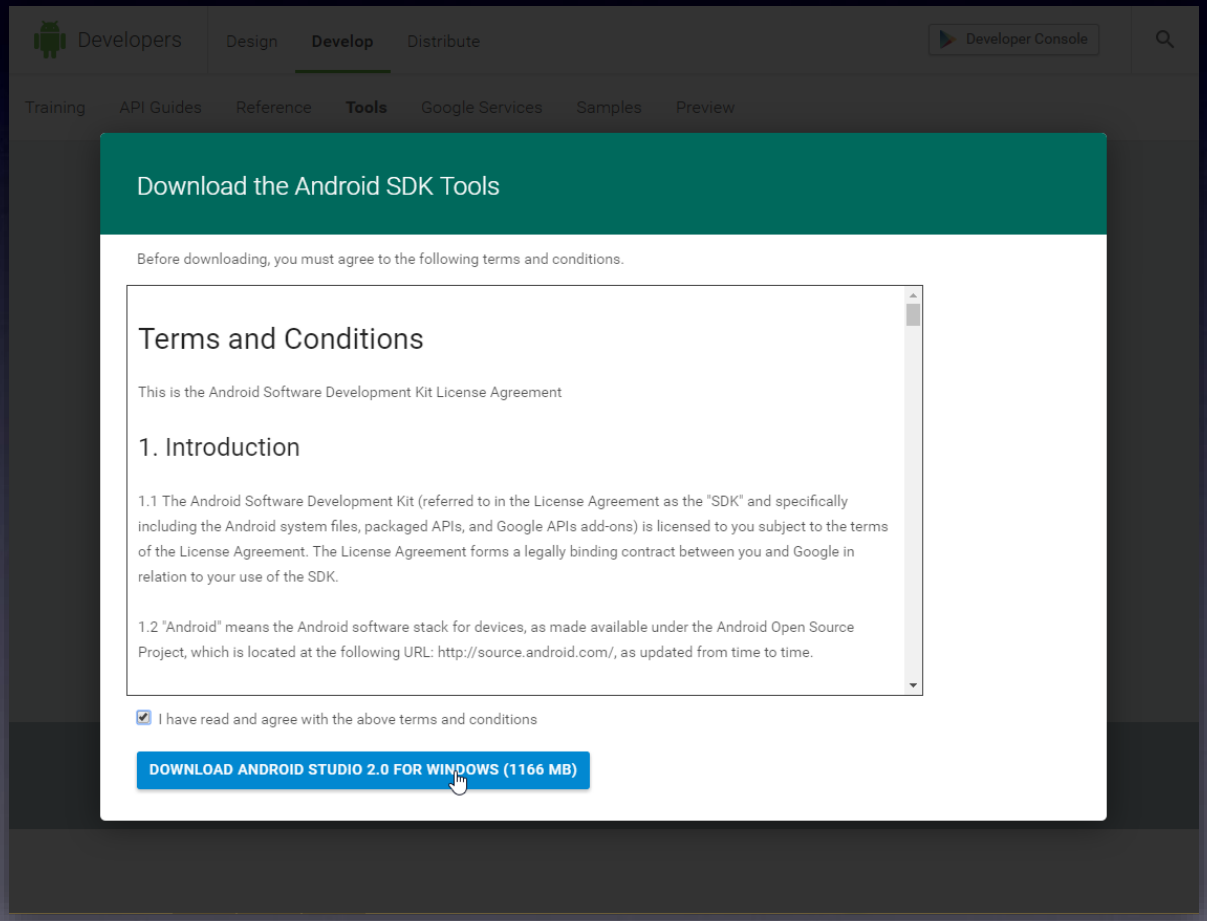
下載ANDROID STUDIO

- 連線至網站，點選Download Android Studio 2.0



下載ANDROID STUDIO

- 勾選I have read ....，接下來按下Download按鈕

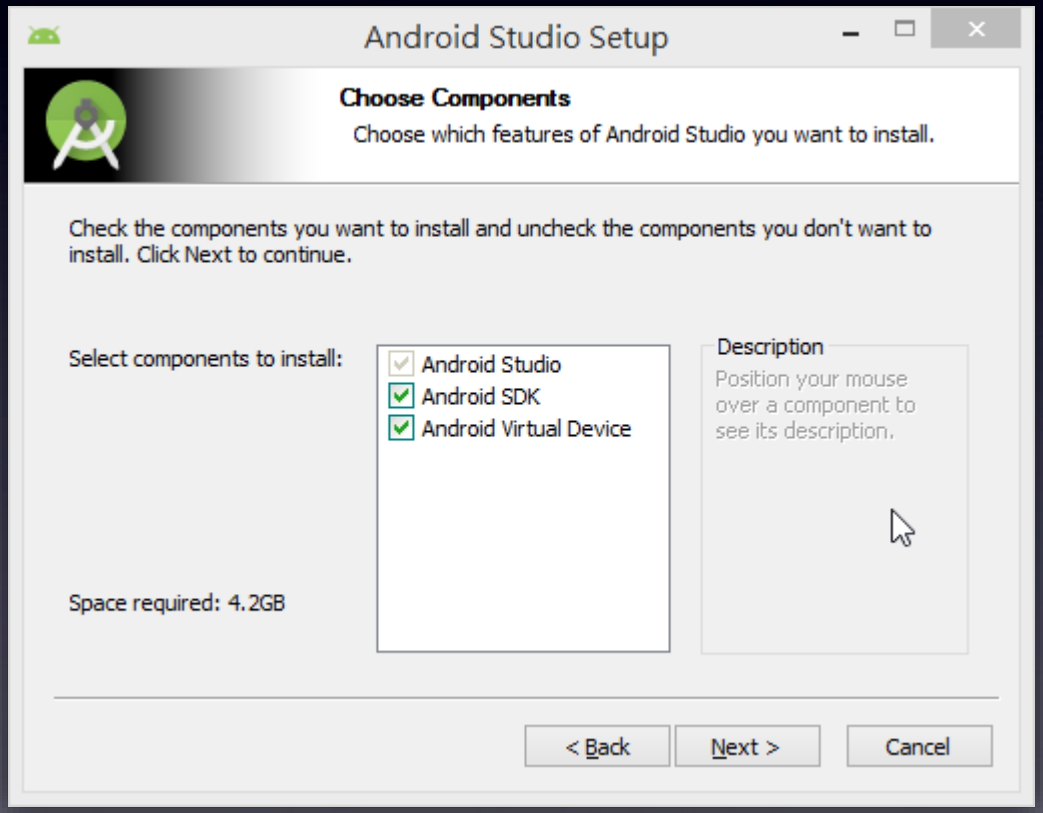


安裝ANDROID STUDIO

- 下載完畢後，點選安裝檔，按下下一步

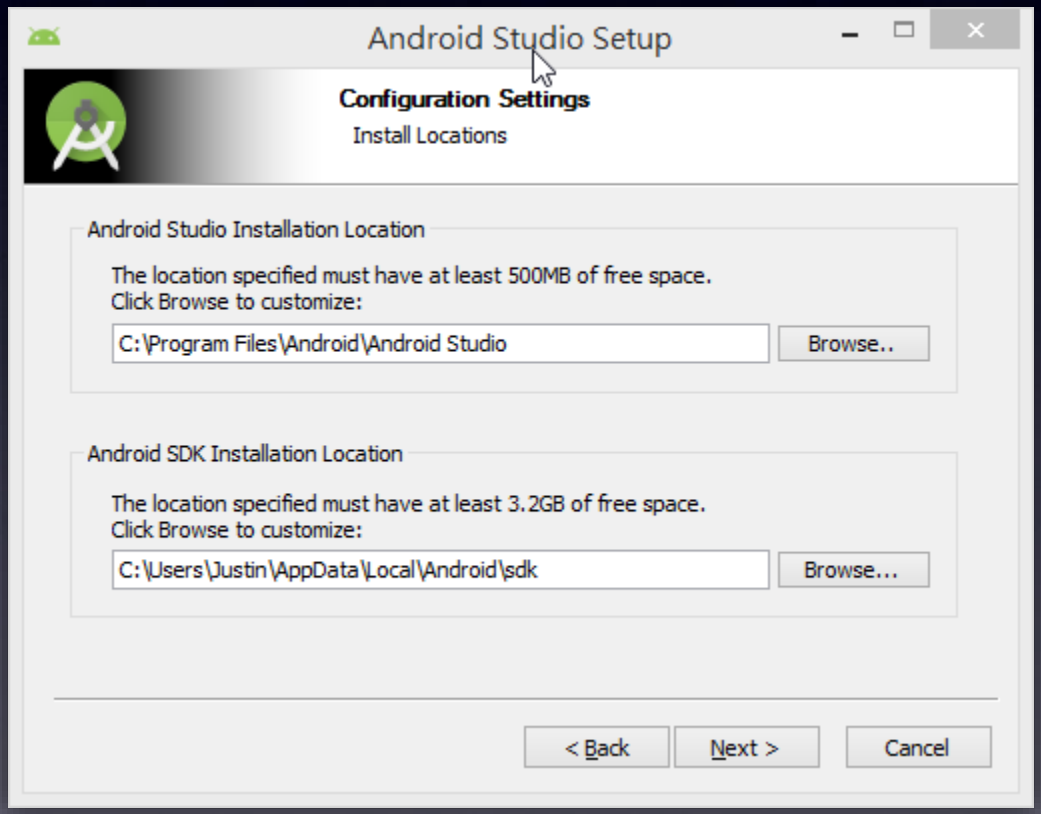
安裝ANDROID STUDIO

- 選擇要安裝項目，目前是安裝Studio、Android SDK以及Android模擬器



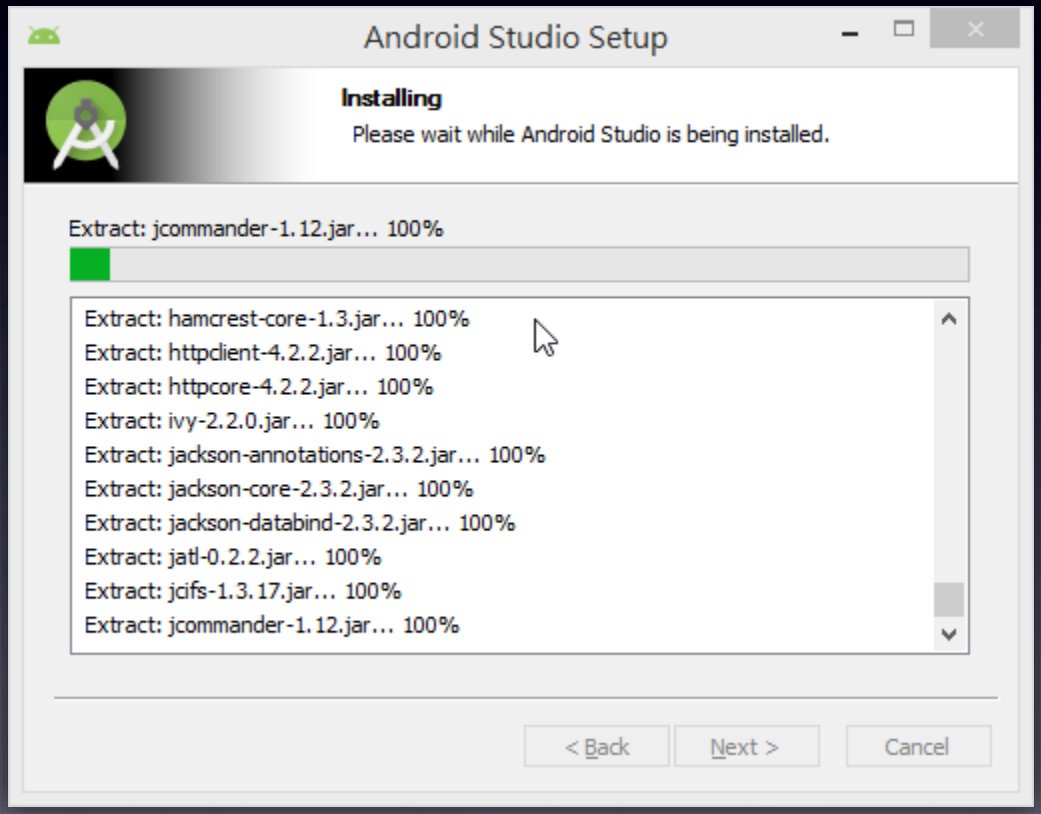
安裝ANDROID STUDIO

- 選擇安裝位置，Android Studio和SDK可以分不同位置安裝



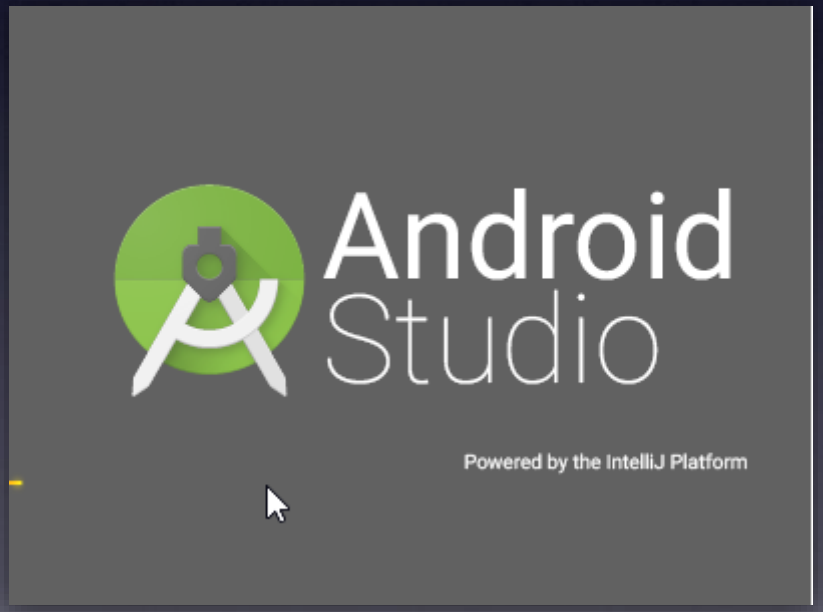
安裝ANDROID STUDIO

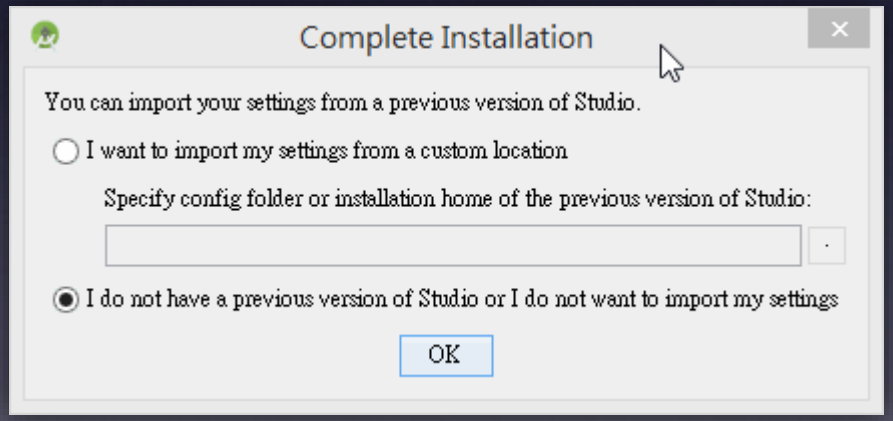
- 靜待安裝程序執行完畢



執行ANDROID STUDIO

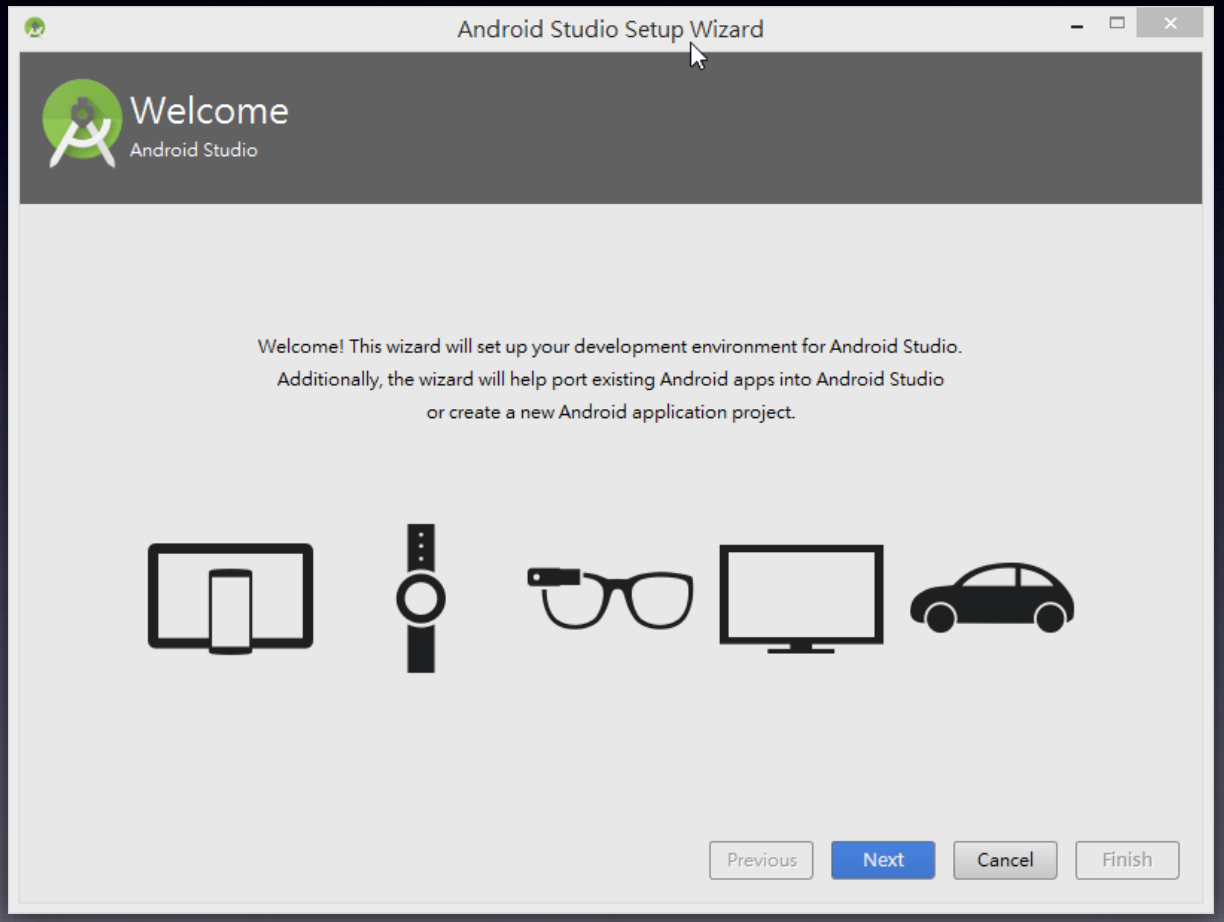
- 安裝完畢後點兩下執行Android Studio
- 第一次開啟時會出現右下圖，詢問是否有舊的設定檔，可供有安裝過舊版的人使用





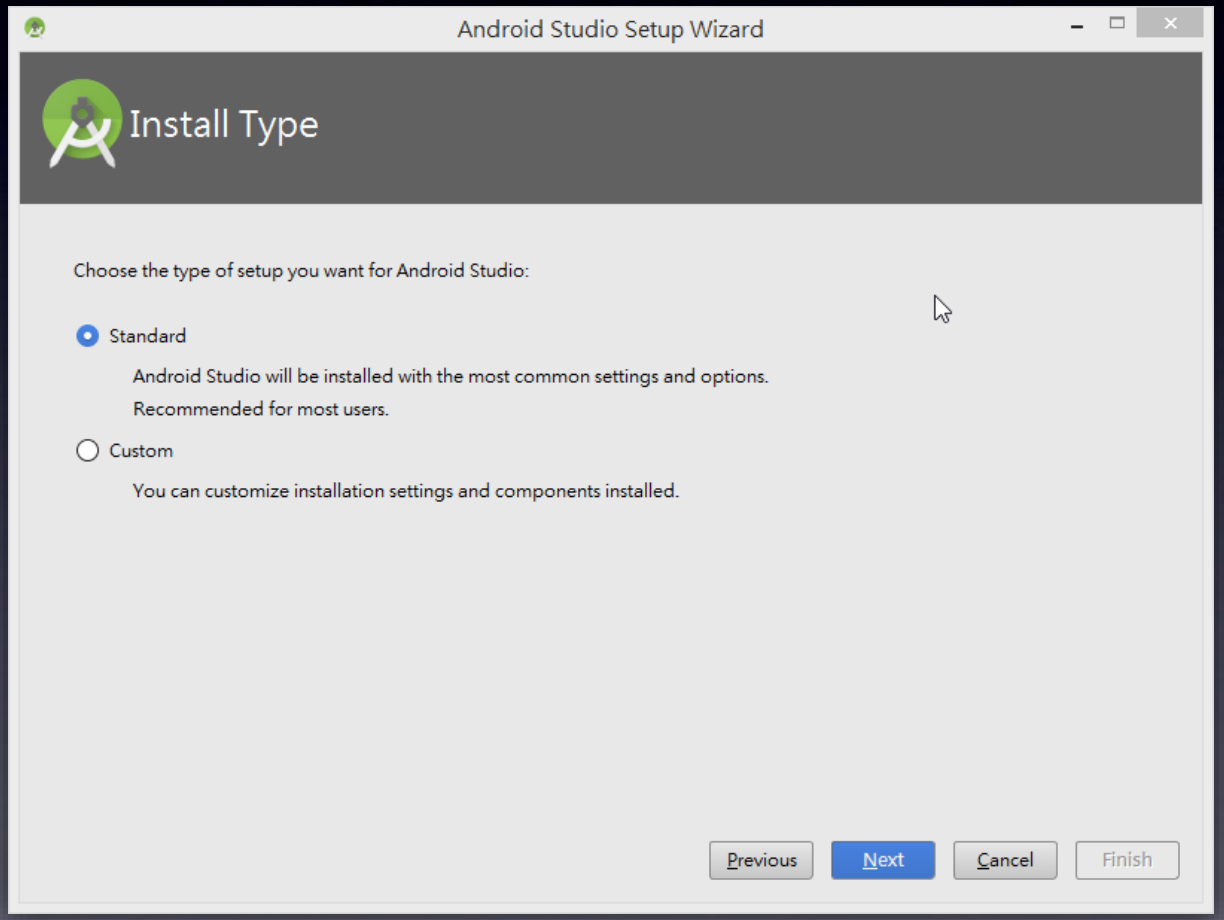
執行ANDROID STUDIO

- 開啟後會有下方的說明頁，點下Next



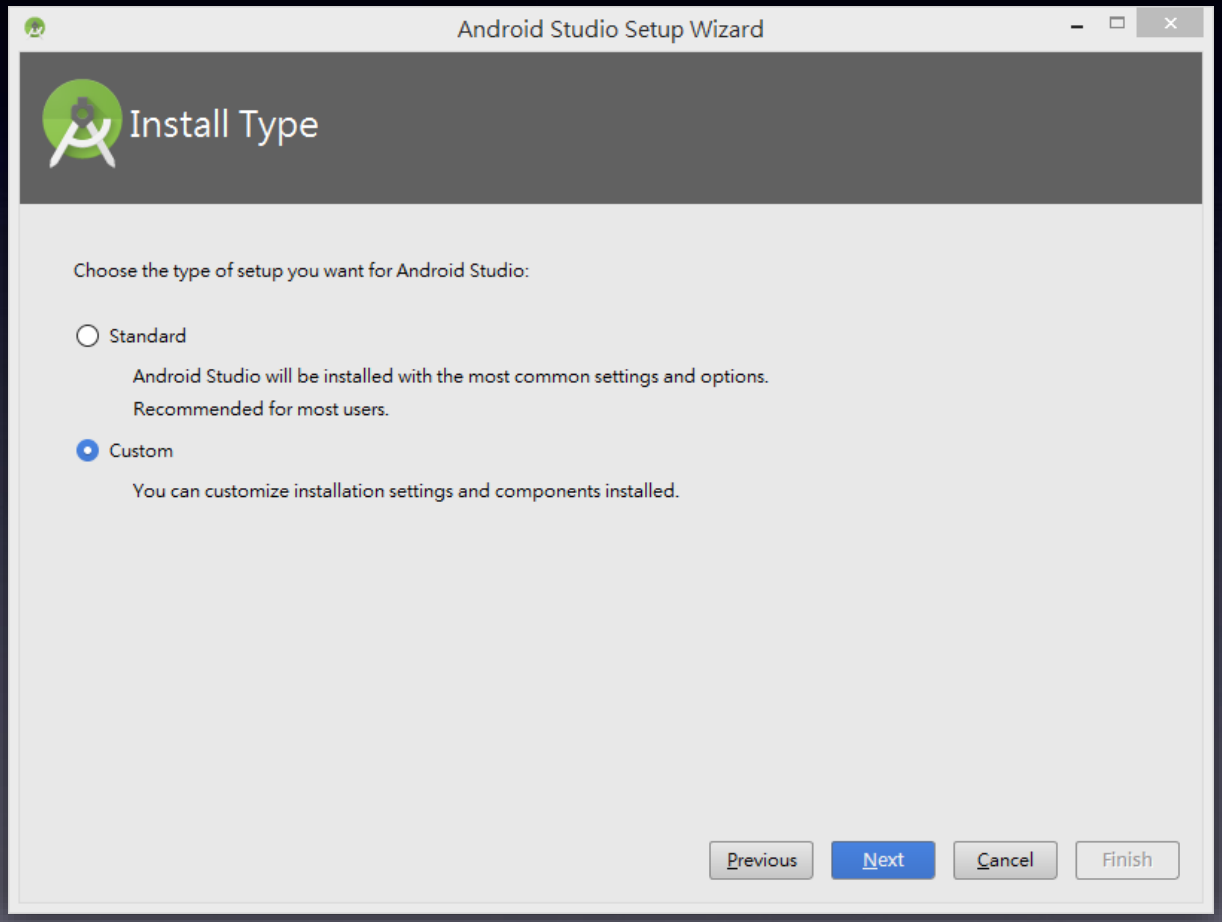
執行ANDROID STUDIO

- 一般安裝都選擇Standard即可



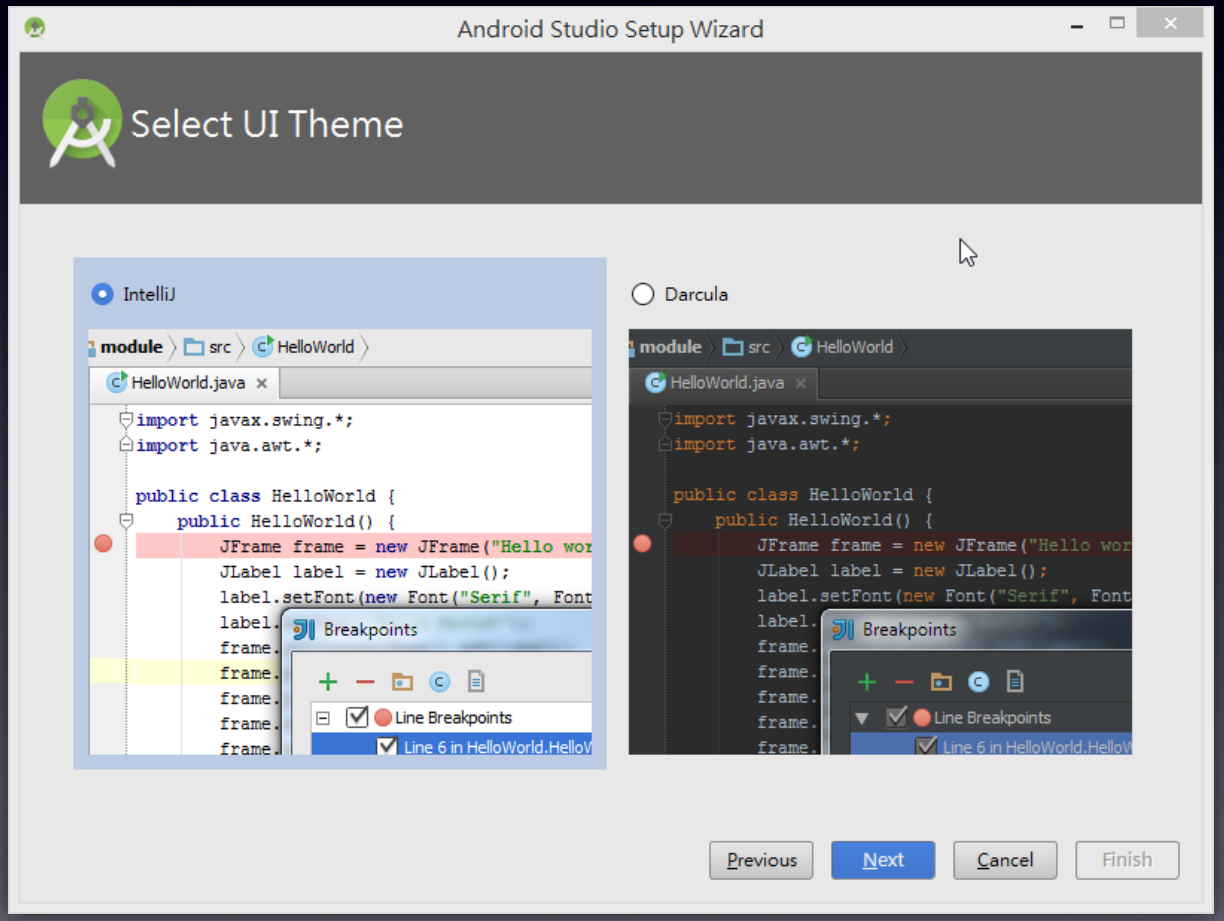
執行ANDROID STUDIO

- 此處使用Custom來了解一下執行的過程



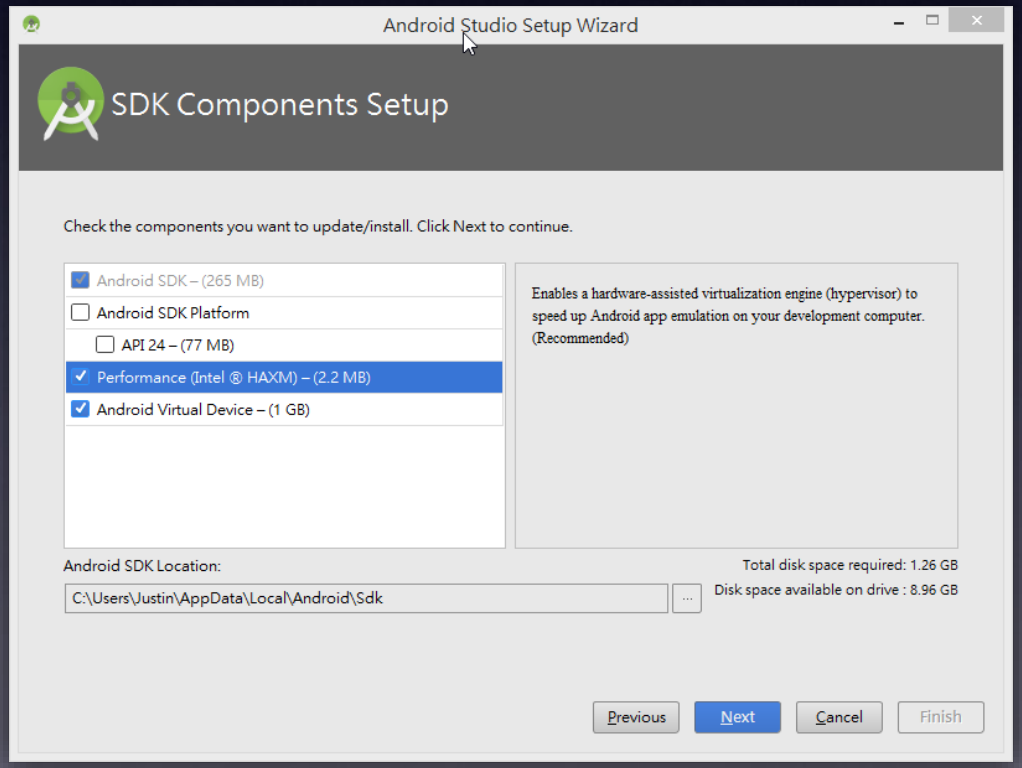
執行ANDROID STUDIO

- 首先讓使用者選擇佈景主題



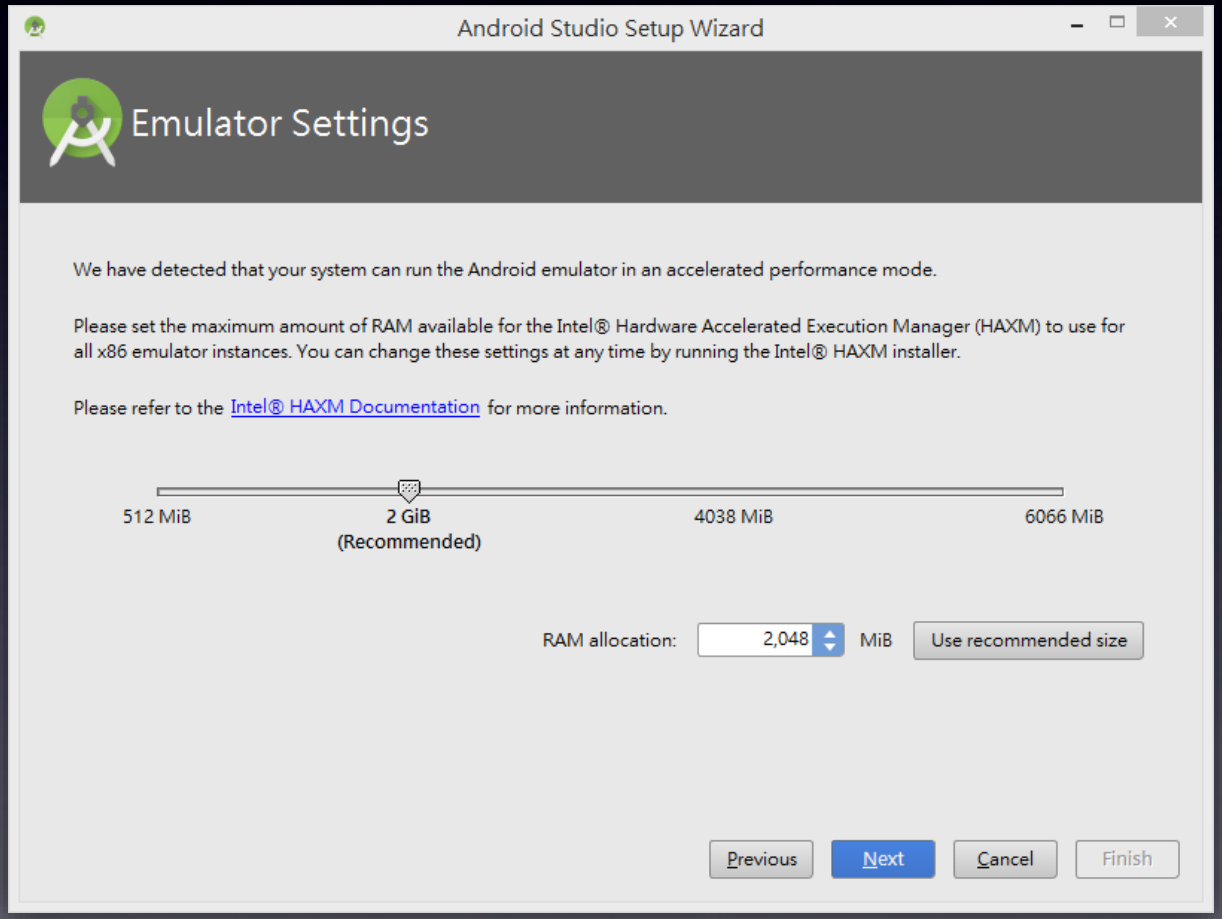
執行ANDROID STUDIO

- 選擇要額外安裝的相關項目
  - SDK Platform 表示目前的SDK中沒有包含的新平台
  - Performance針對模擬器加速的套件(只與Intel相容)
  - Android Virtual Device建立一個模擬器



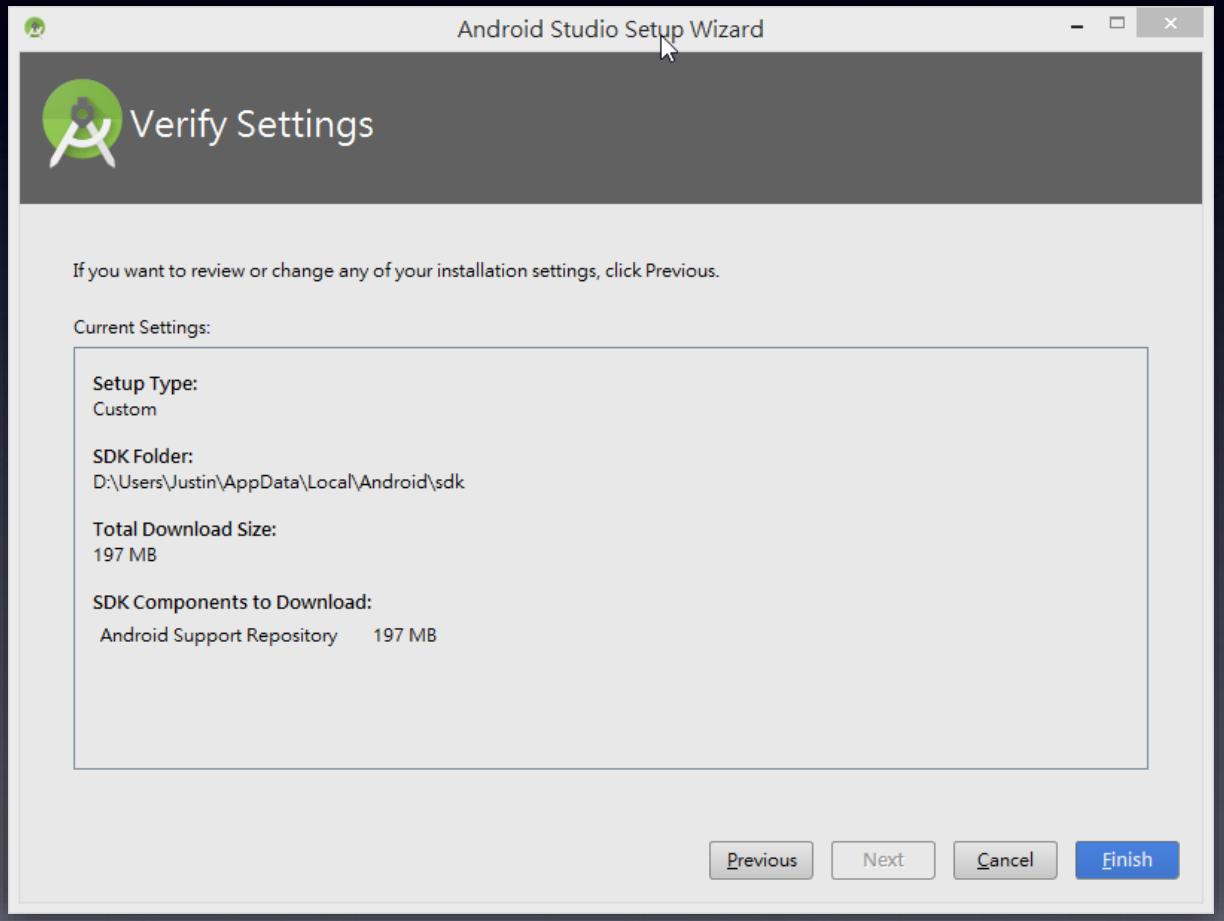
執行ANDROID STUDIO

- 調整執行模擬器時，預設系統分配多少記憶體



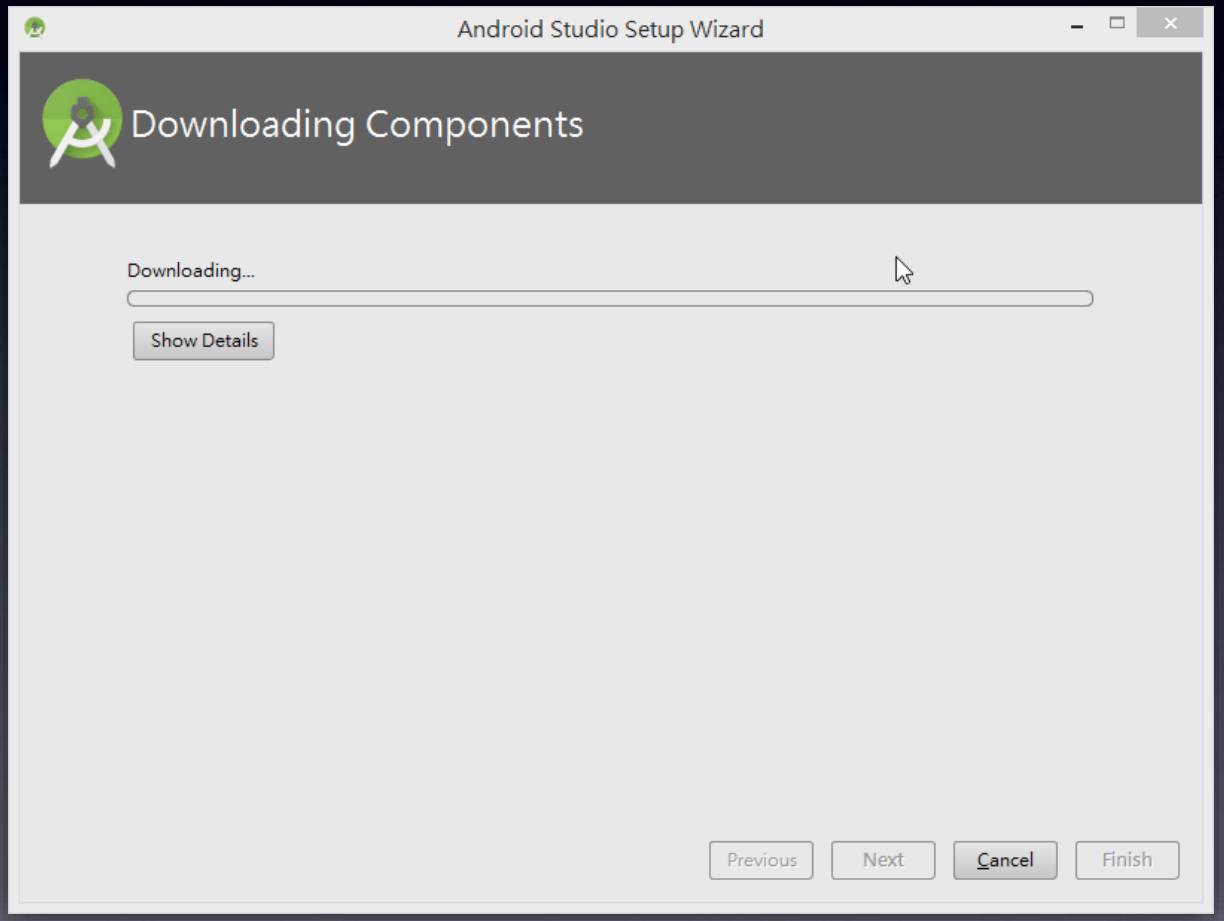
執行ANDROID STUDIO

- 全部選擇完畢，按下Finish



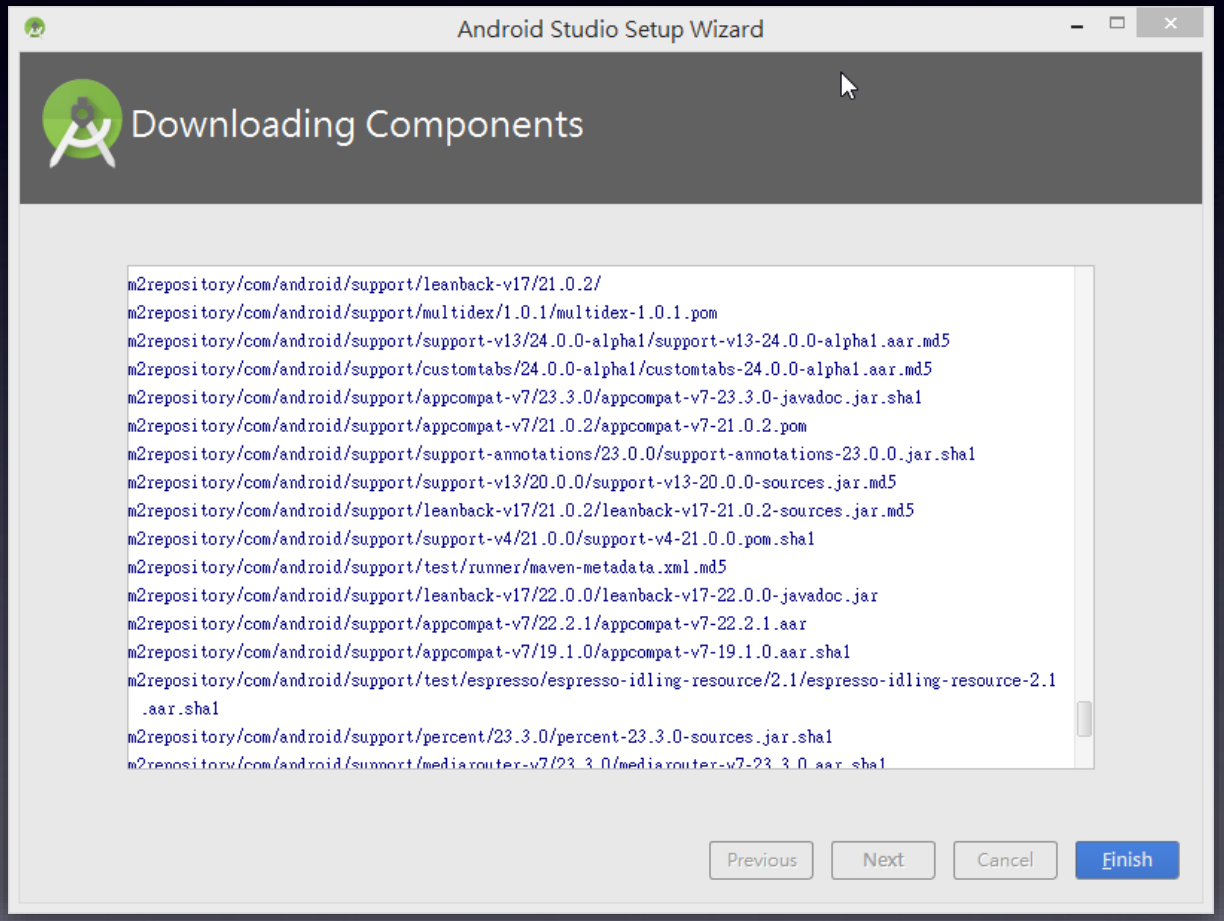
執行ANDROID STUDIO

- 等待下載結束



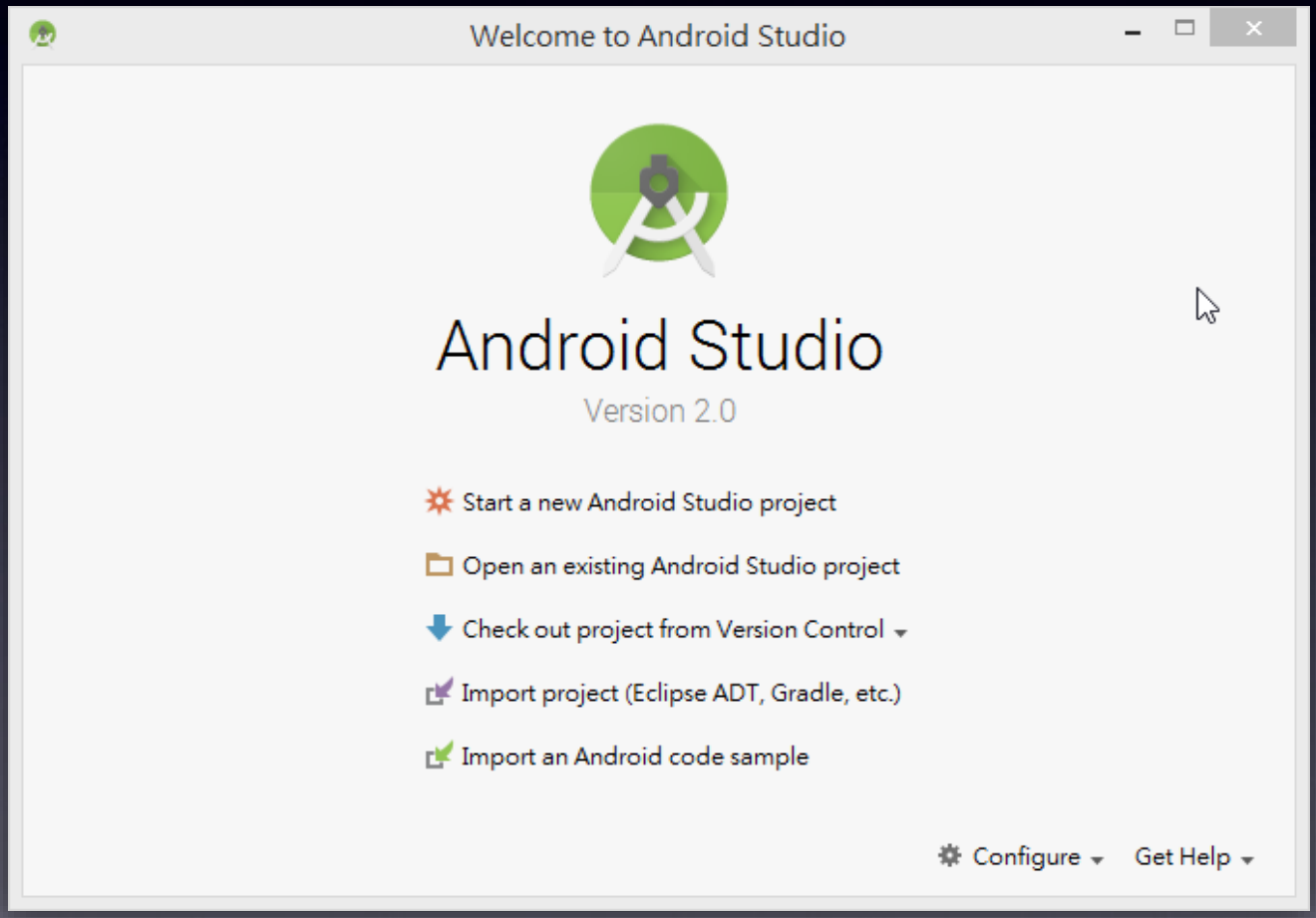
執行ANDROID STUDIO

- 下載完成！點下Finish



執行ANDROID STUDIO

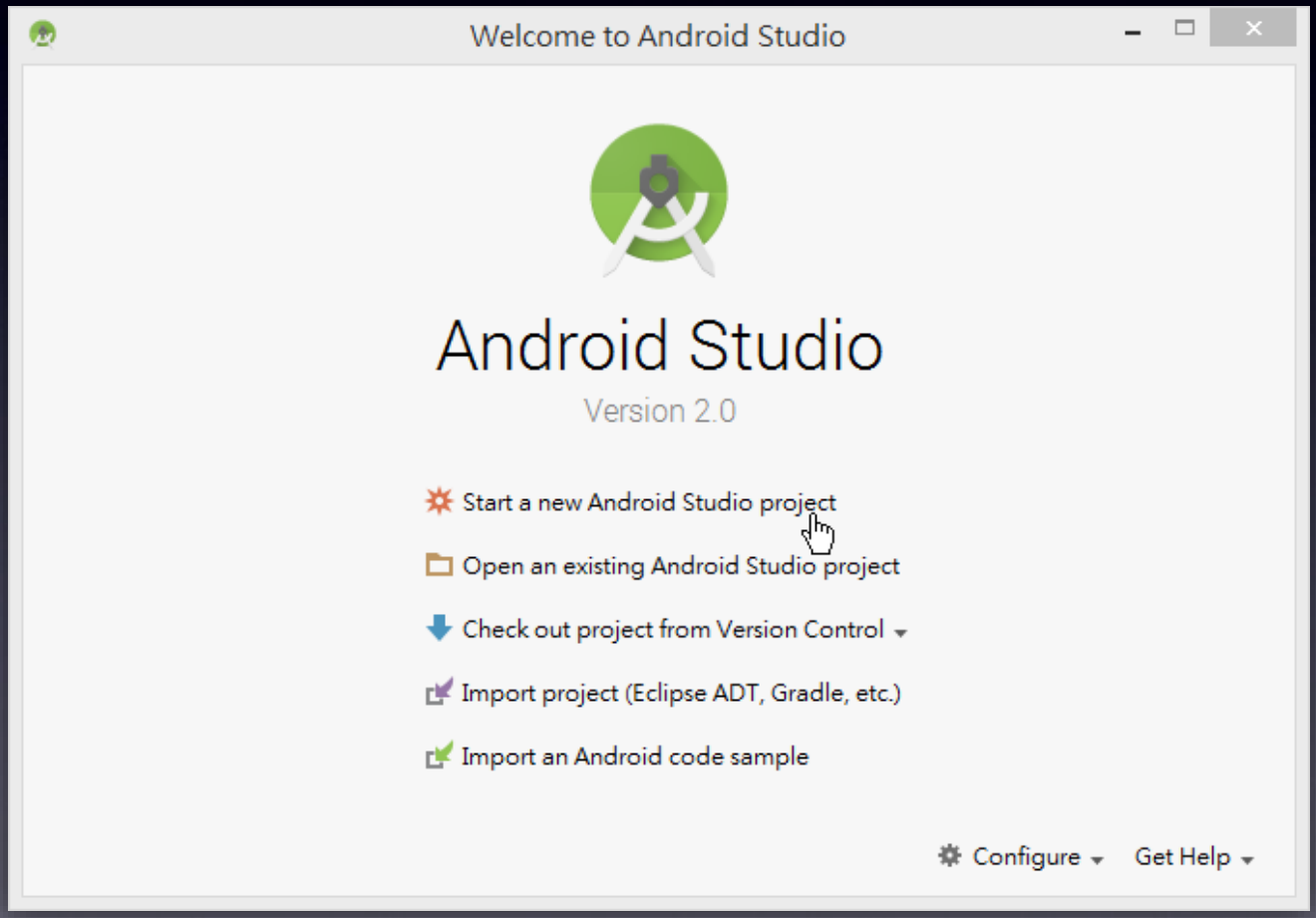
- 完成執行的步驟了



開新專案

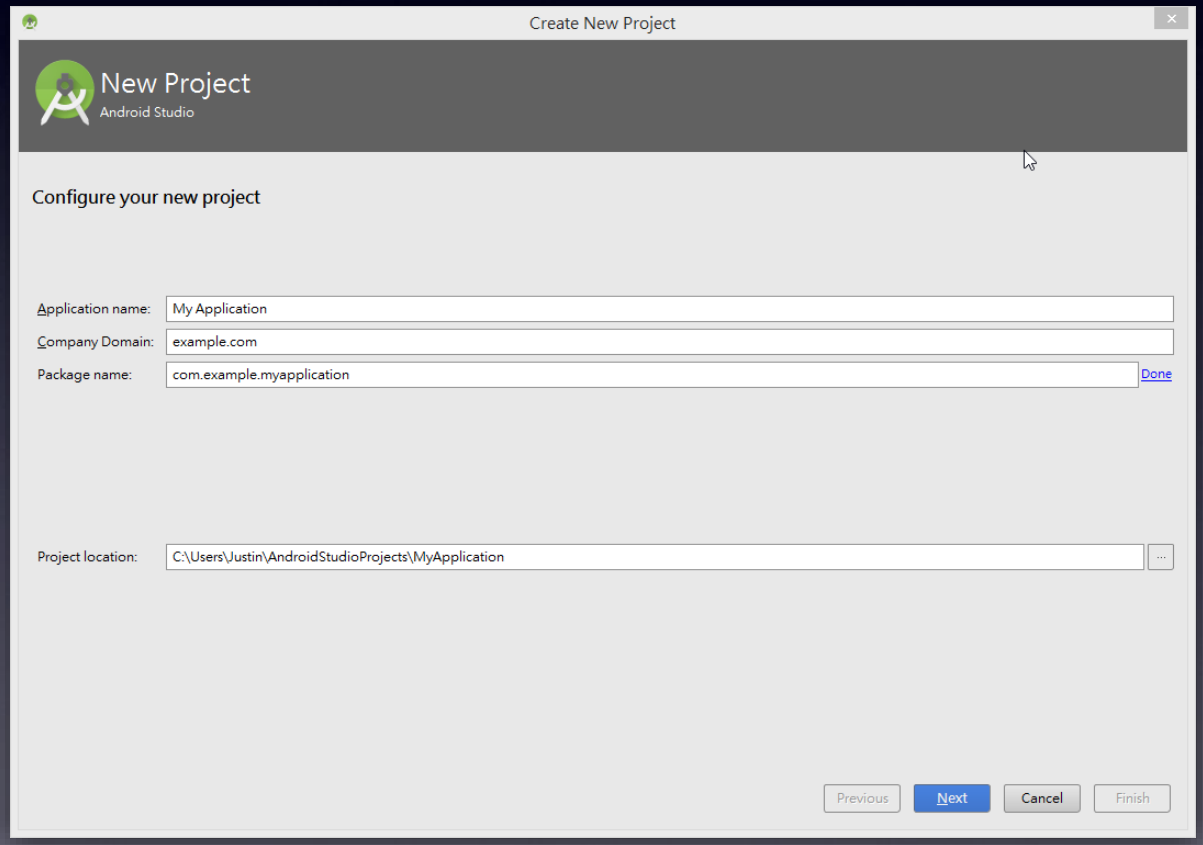
開新專案

- 選擇Start a new Android Studio project



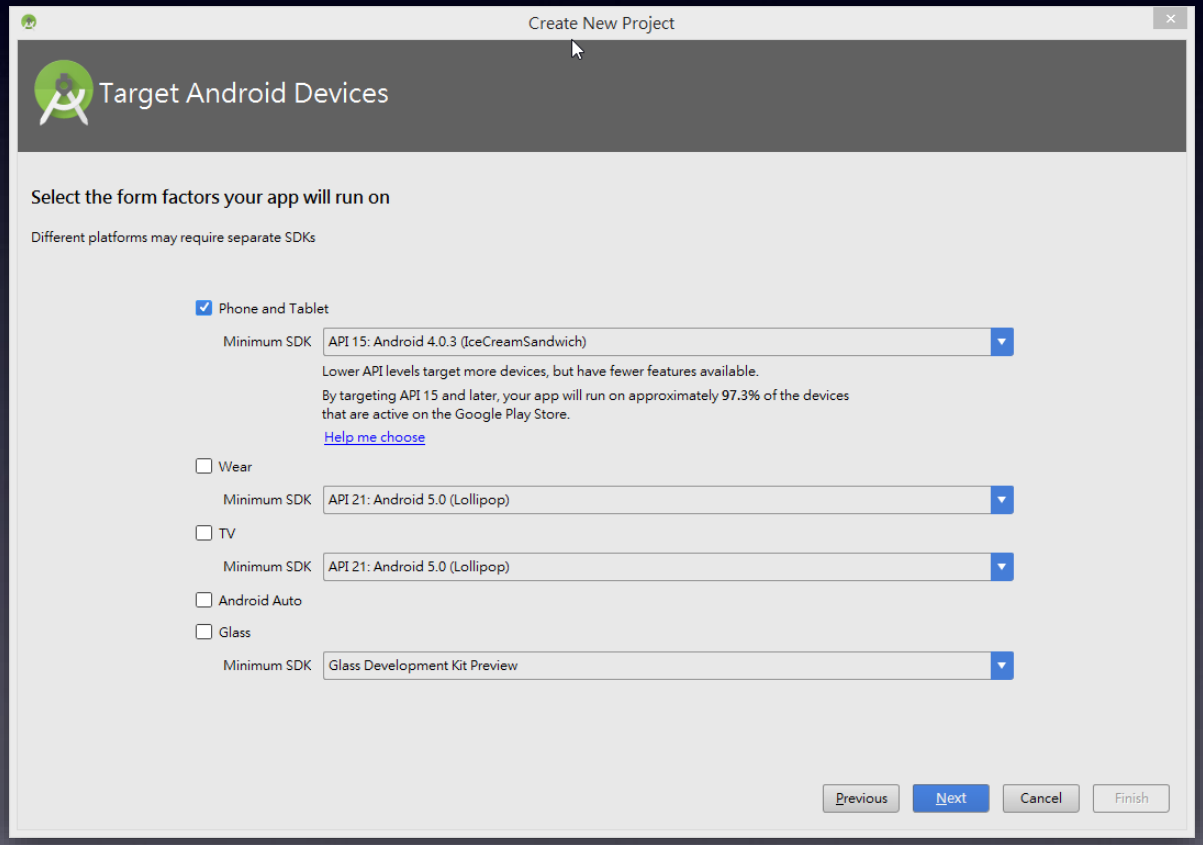
開新專案

- Application name 輸入app名稱
- Company Domain 公司網址，用來產生package name



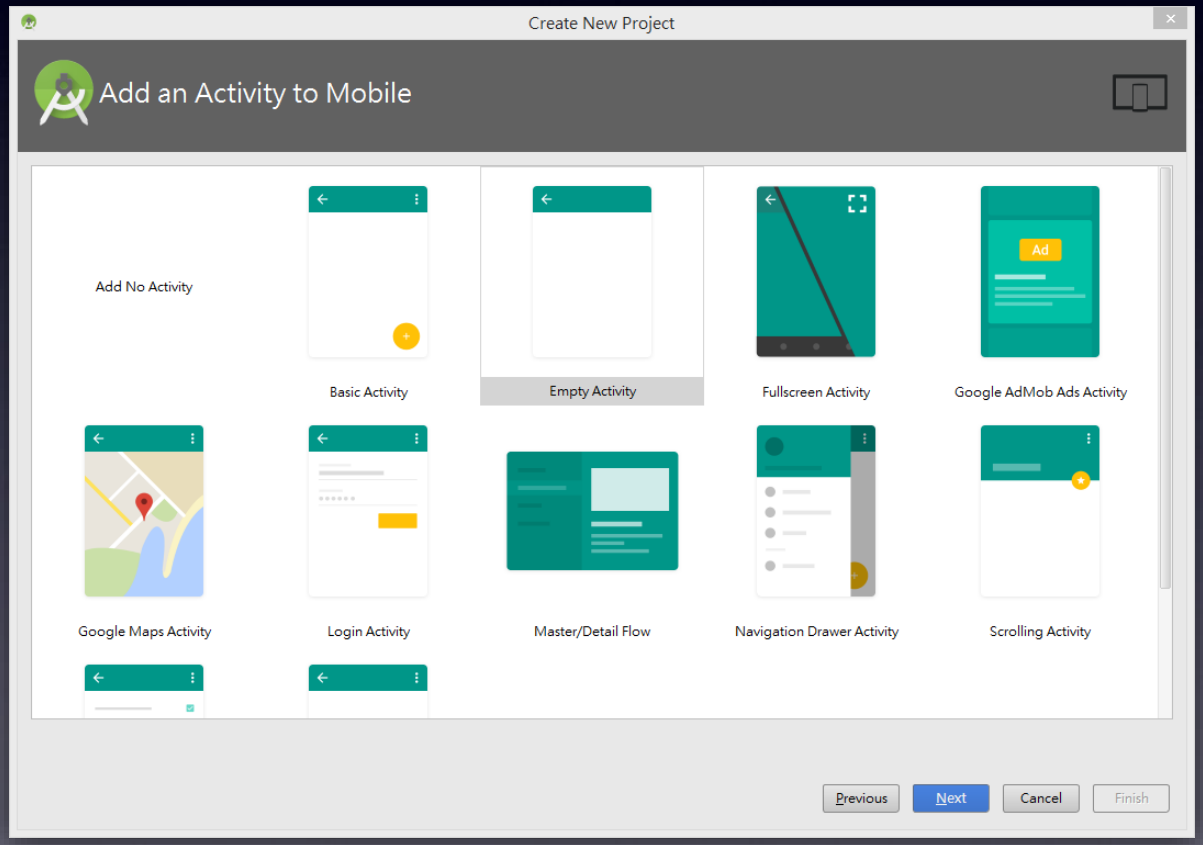
開新專案

- Minimum SDK決定了app能運行最低的Android版本
  - 可點選Help me choose來看看選擇可涵蓋市場的比率



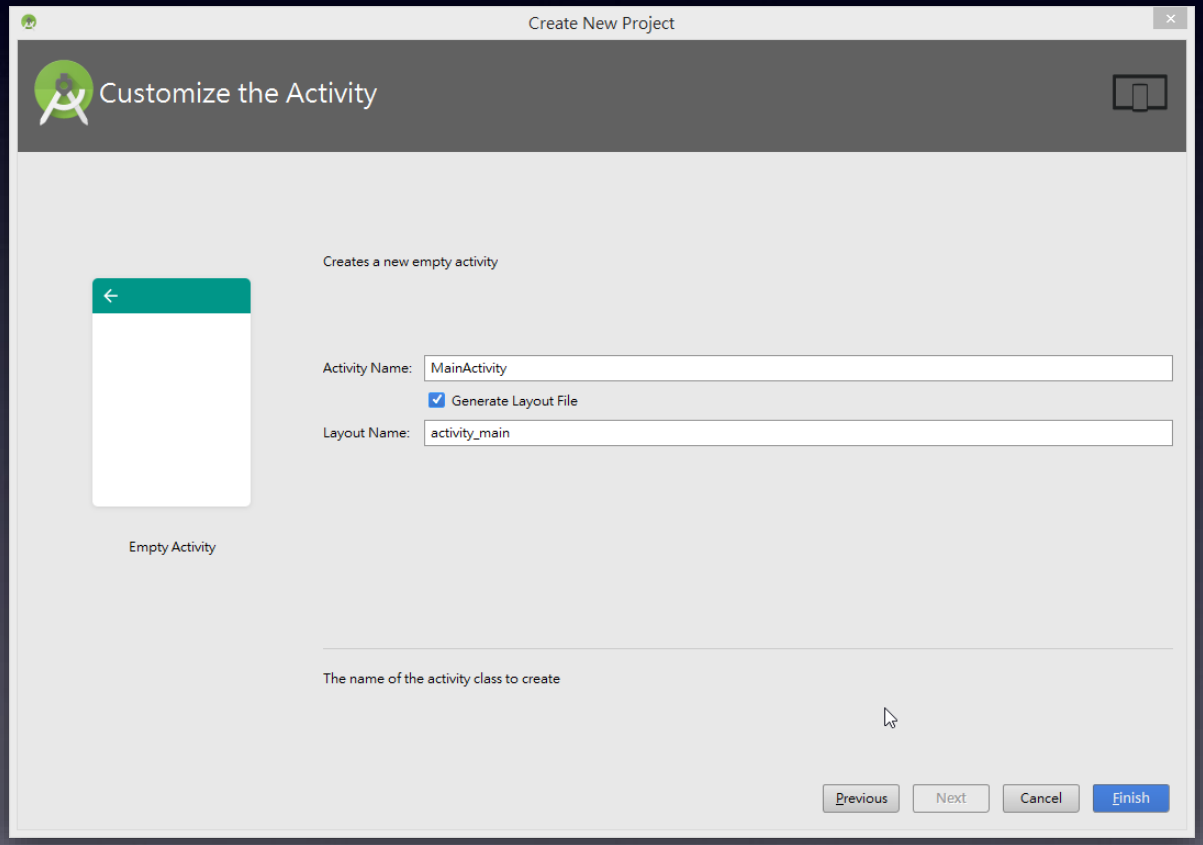
開新專案

- 選擇App第一個Activity類型，本範例選擇Empty Activity



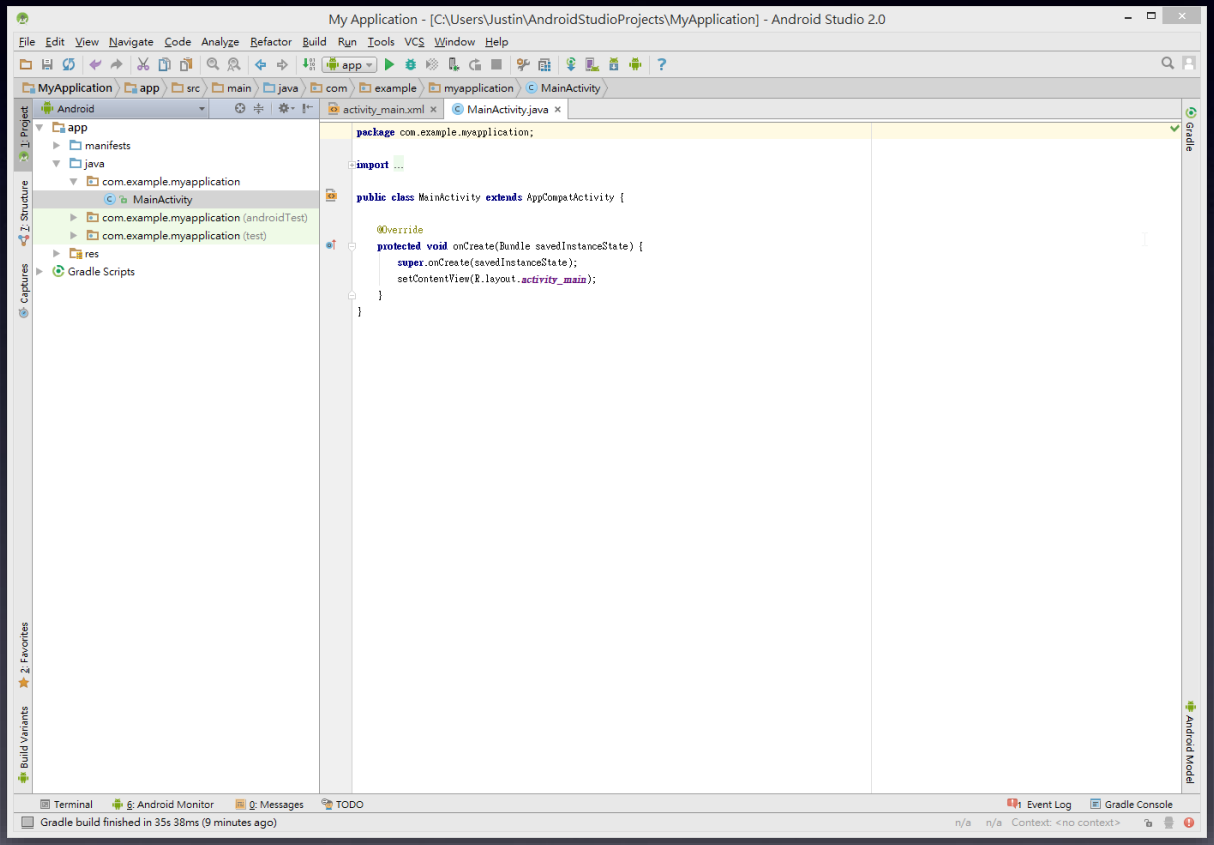
開新專案

- Activity Name 頁面Activity class的名稱
- Layout Name 自動產生的layout檔名稱



開新專案

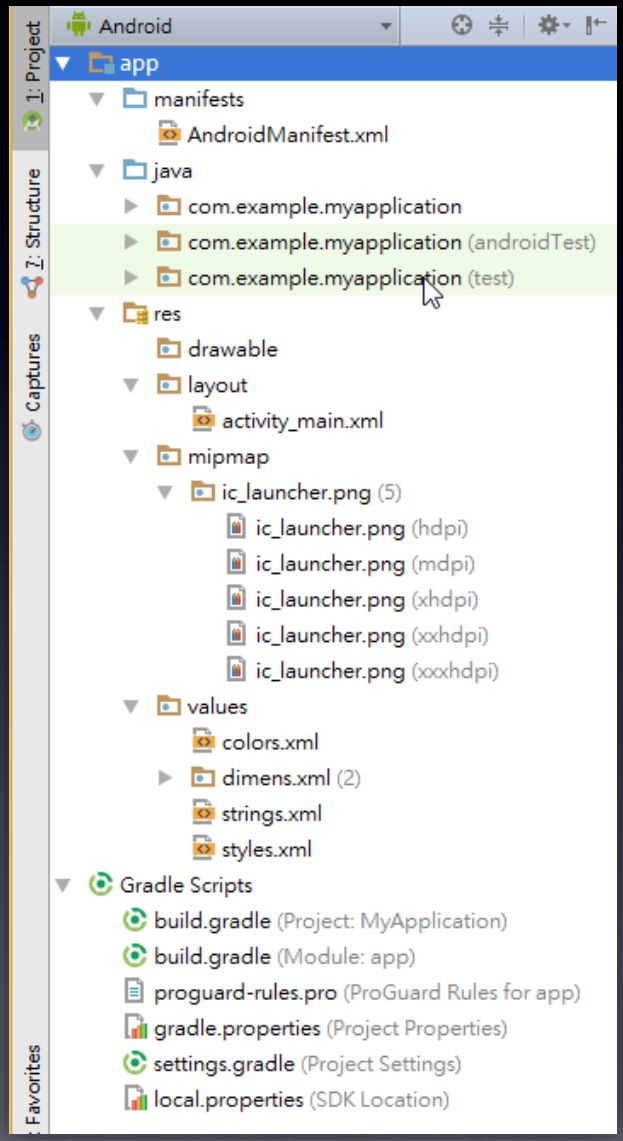
- 新專案建立完成！



專案結構說明

專案結構說明

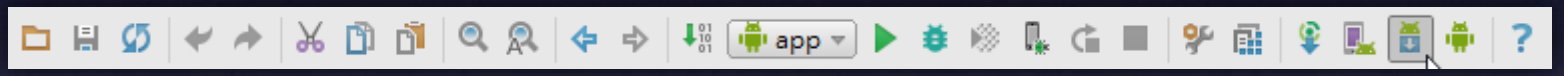
- app (Module)
  - manifest 應用程式設定檔
  - java 原始碼
  - res 資源檔
- Gradle Scripts
  - 編譯用Gradle Scripts以及專案設定檔



SDK MANAGER

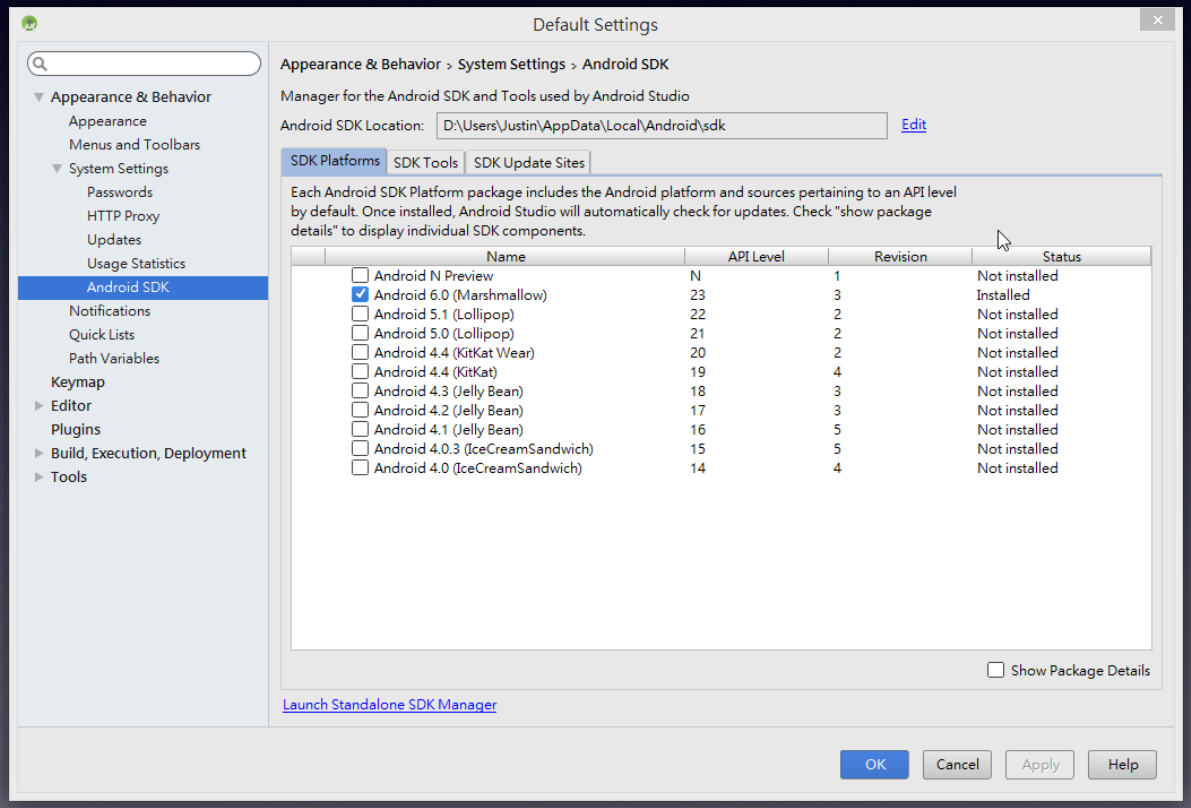
SDK MANAGER

- 點選如下圖滑鼠指標所指到的圖示



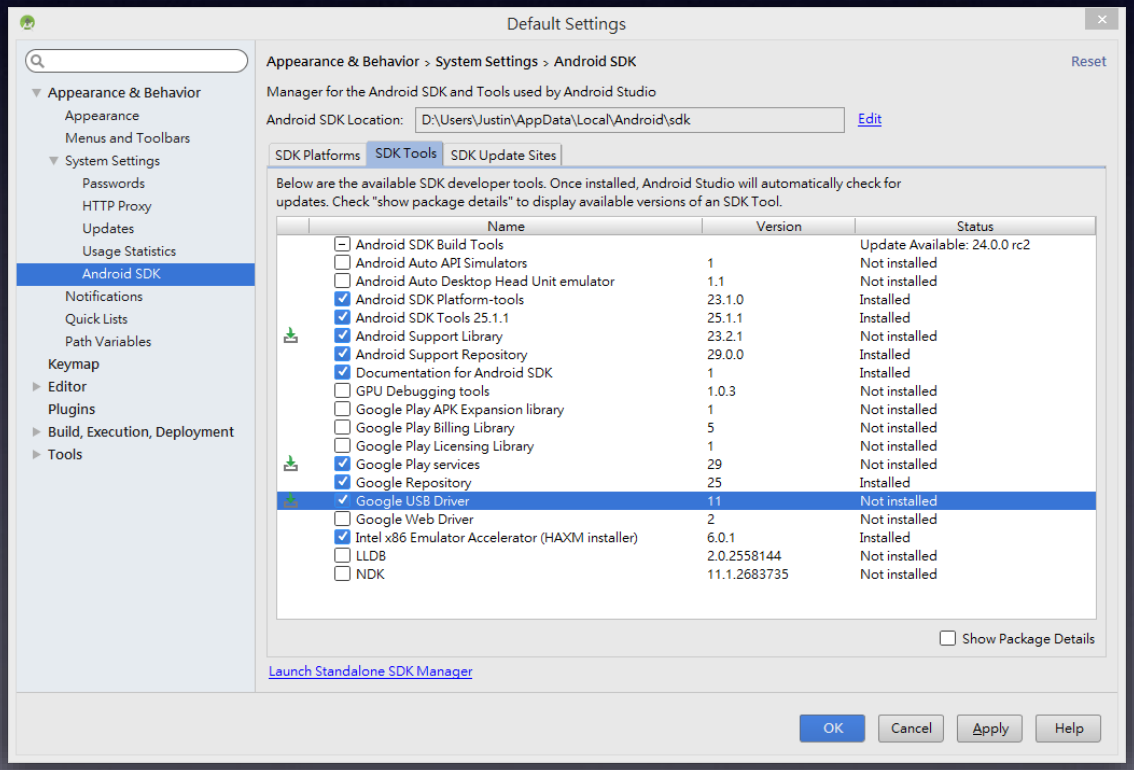
SDK MANAGER

- SDK Platforms 目前不同版本的SDK下載
- SDK Tools 專案使用的函式庫、編譯工具



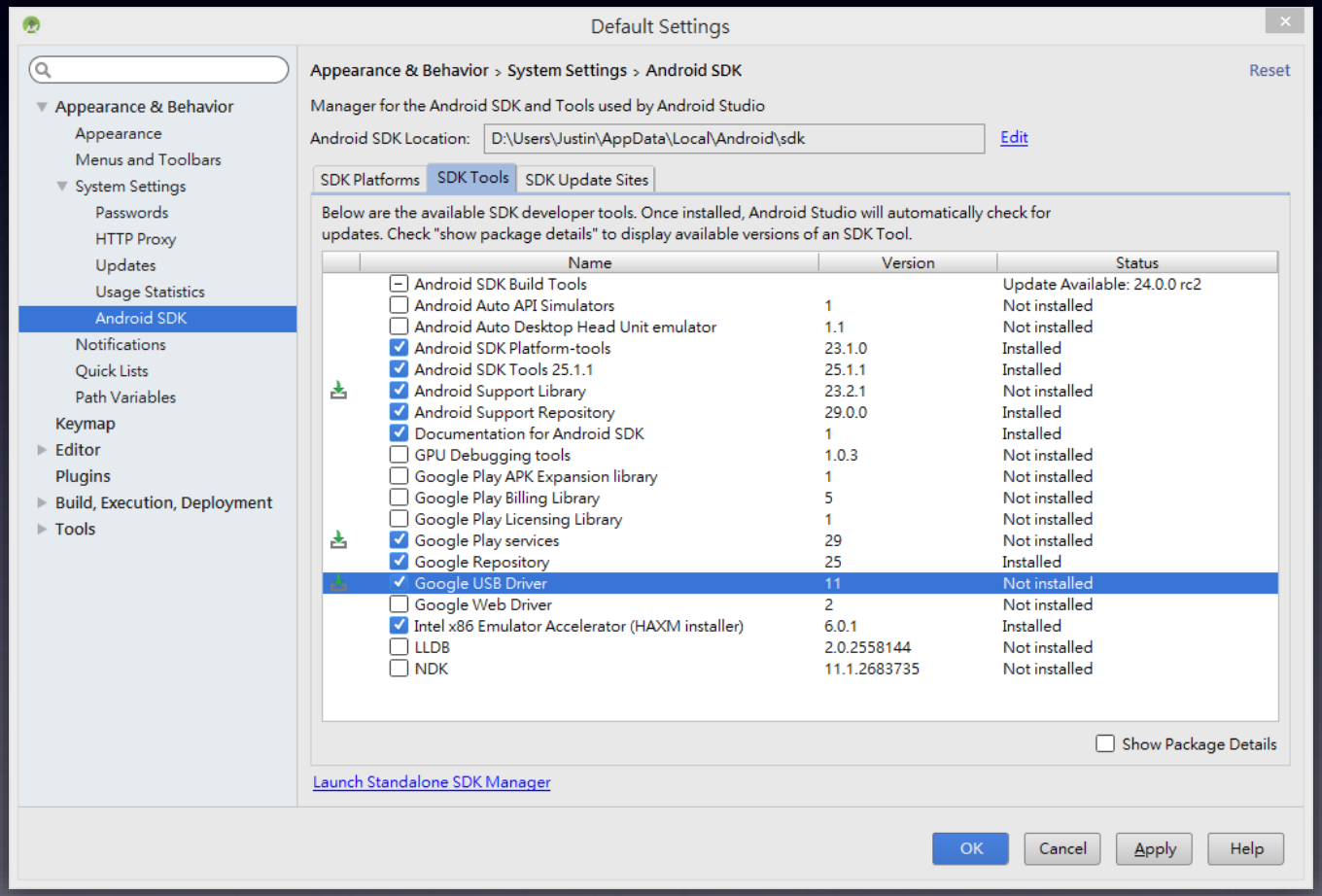
SDK MANAGER

- SDK Tools要確定下載了Android Support Repository、Google Repository
- 若平台允許還可安裝Intel x86 Emulator Accelerator



SDK MANAGER

- 要更多的細節操作，可以點選下方 Launch Standalone SDK Manager



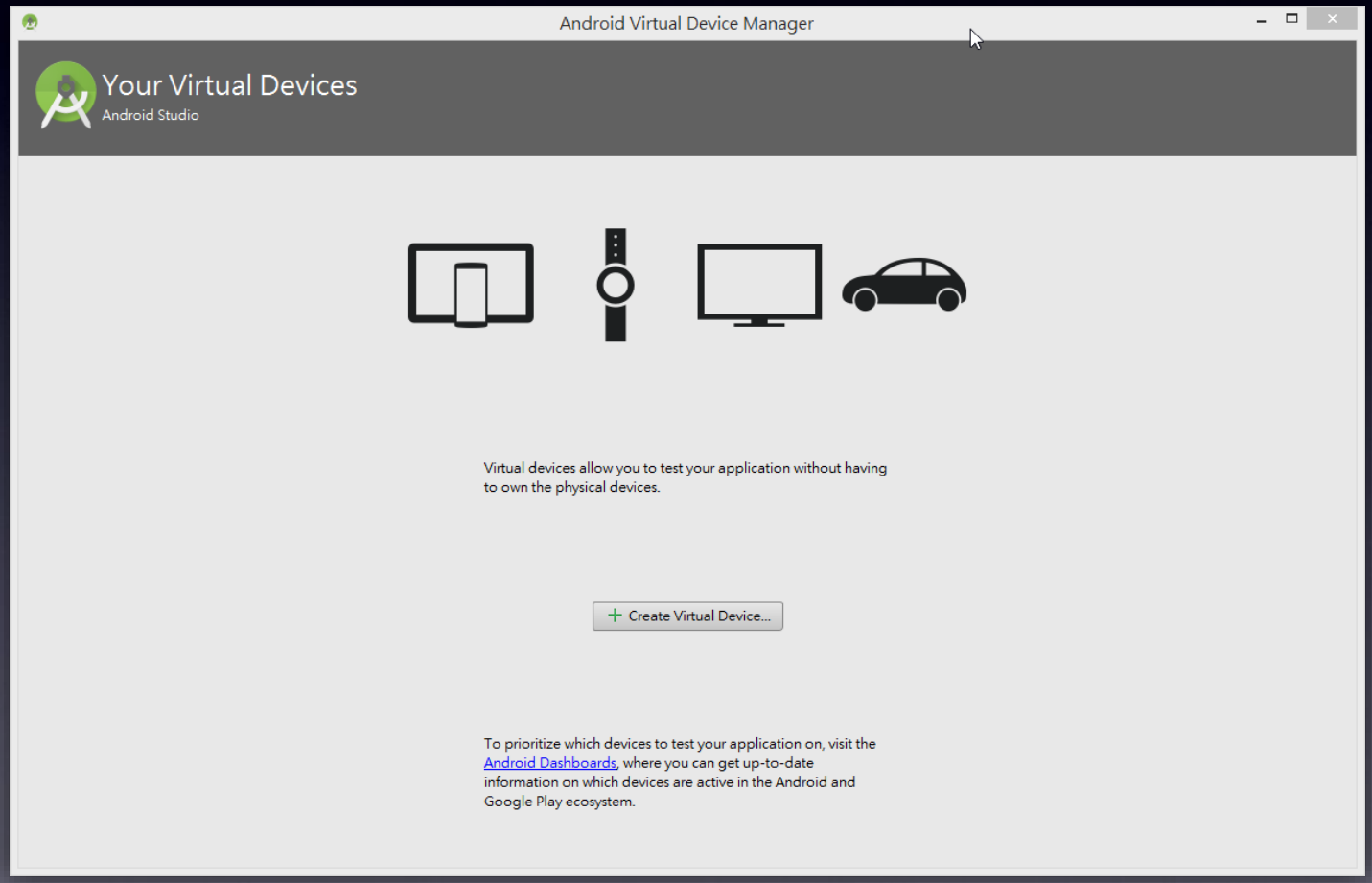
建立模擬器

建立模擬器

- 在Android Studio最上方快捷工具列
  點選如下圖中滑鼠所點選的圖示

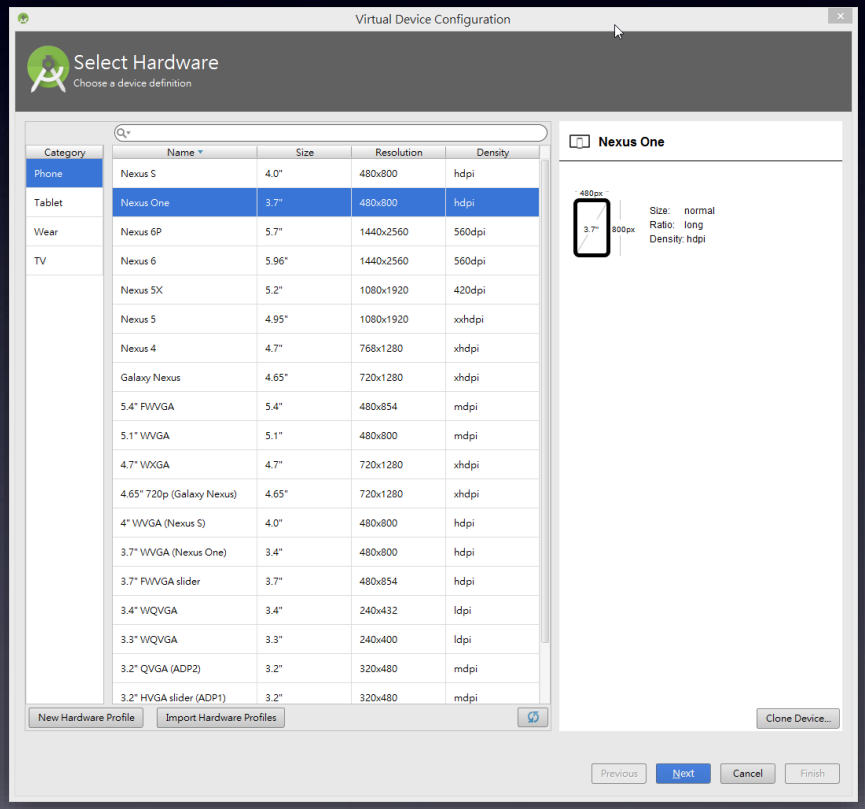
建立模擬器

- 此處說明能建立的模擬器類型，選擇Create Virtual Device...



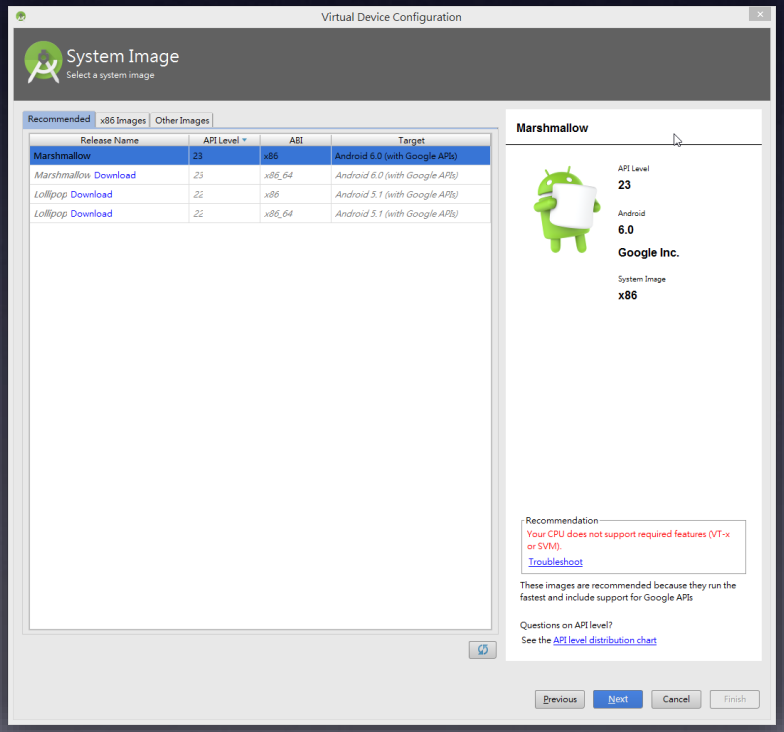
建立模擬器

- 選擇模擬器要模擬的硬體規格
  - 類型是Phone, Tablet, Wear, TV
  - 實際硬體規格和螢幕大小



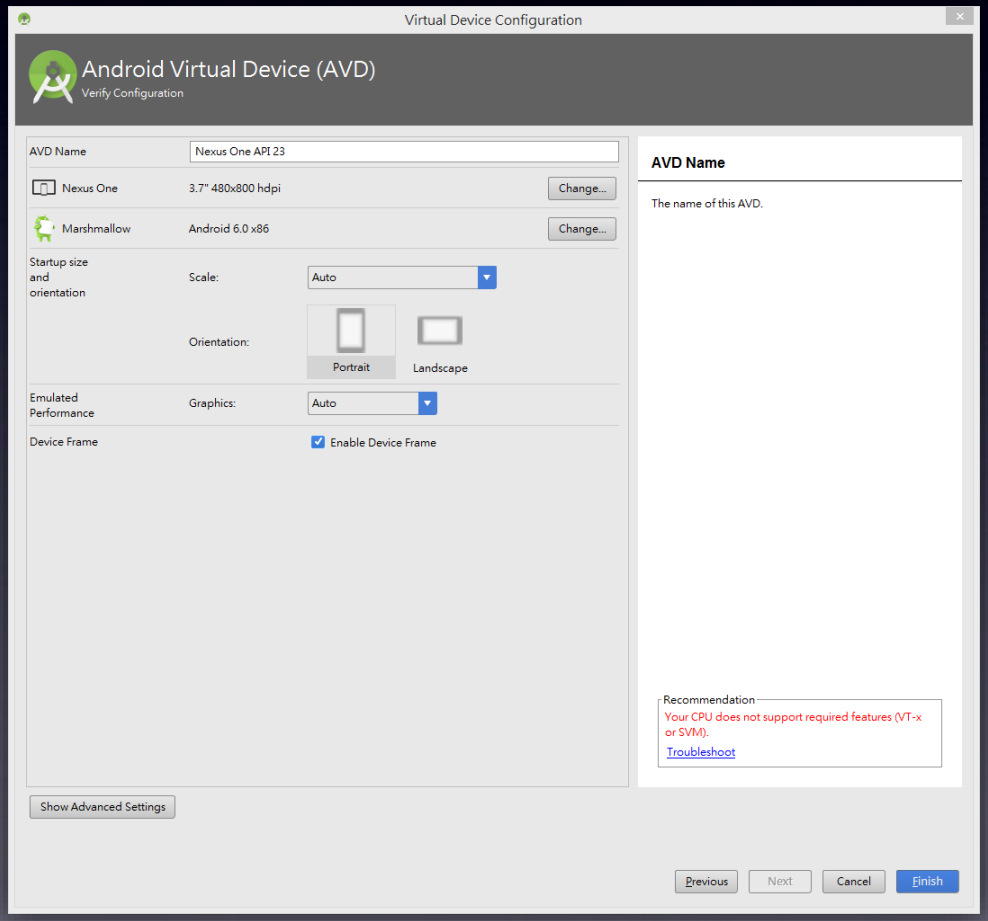
建立模擬器

- 選擇下載模擬器的映像檔(Image)
- 若你電腦的CPU非Intel的，請選擇Other Image
  - 可以看圖右下角的Recommendation
- 建議下載要有Google Play包含的映像檔



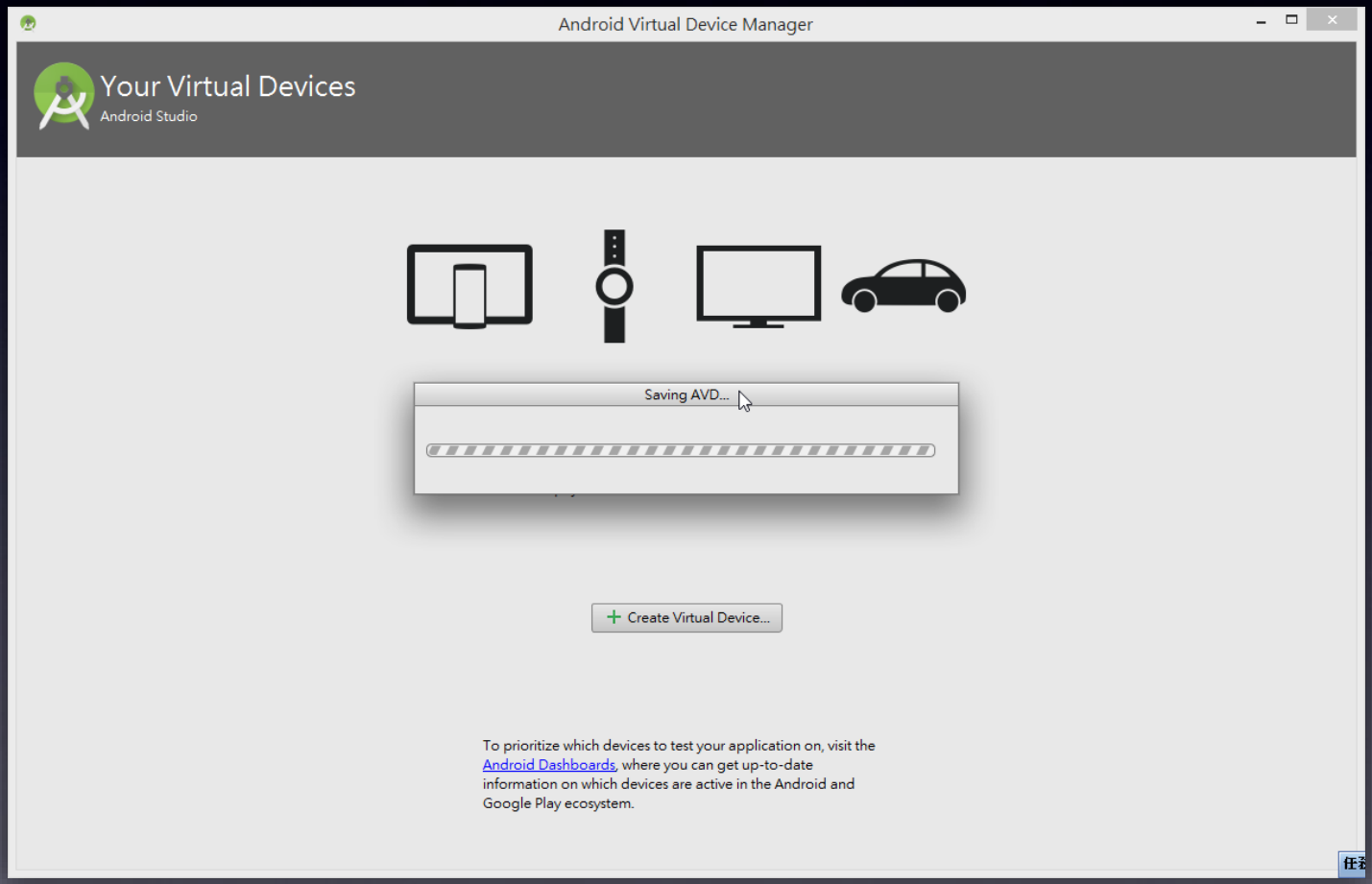
建立模擬器

- AVD Name 給模擬器取個名字



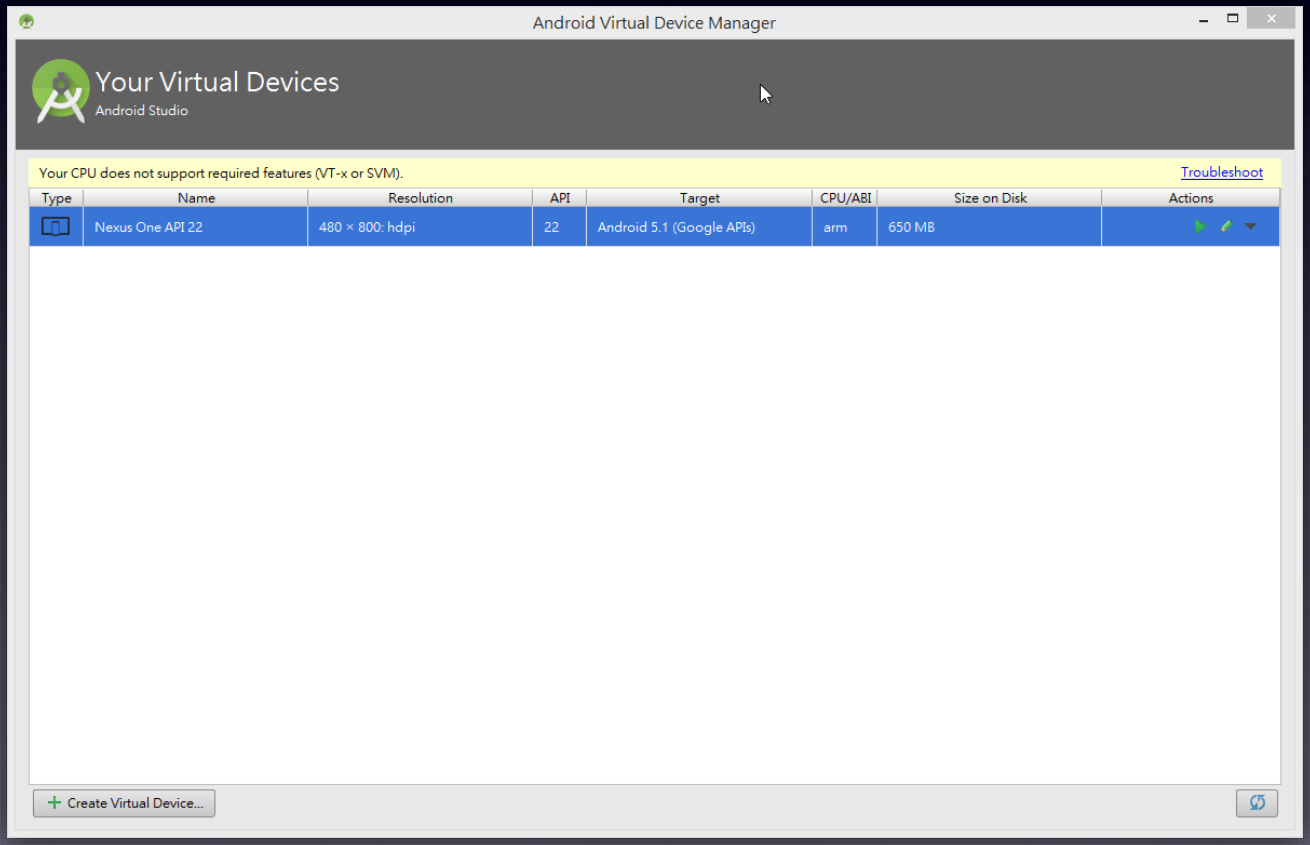
建立模擬器

- 等待建立完成



建立模擬器

- 建立完成後會在AVD Manager的列表中出現
- 點選最右方的執行按鈕開始啟動

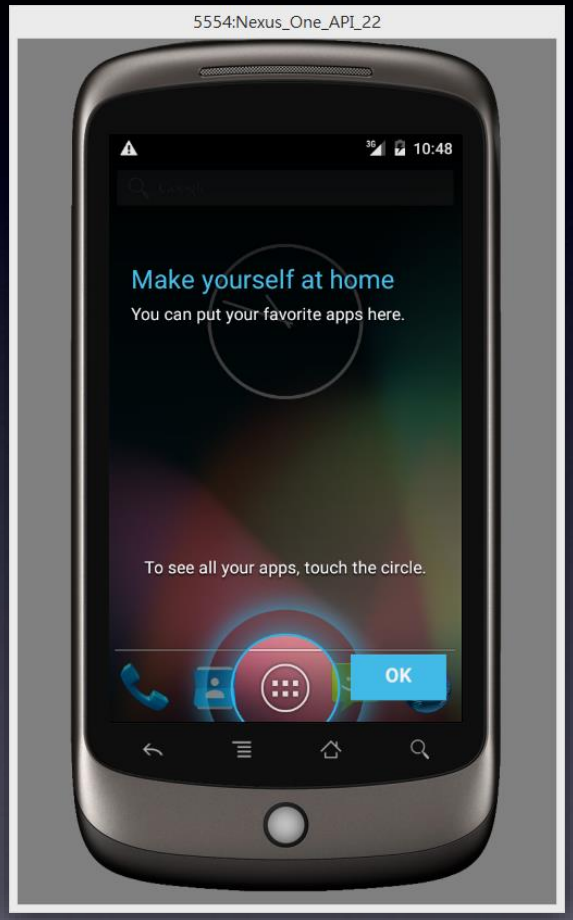


建立模擬器

- 若是Intel Accelerator安裝成功的電腦，且使用x86的Image建立的模擬器開機速度將會非常快
- 但一般arm的Image建立的模擬器，開機速度將會慢超過10倍

建立模擬器

- 開機成功！


5554:Nexus_One_API_22
10:48
Make yourself at home
You can put your favorite apps here.
To see all your apps, touch the circle.
OK

模擬器其他選擇

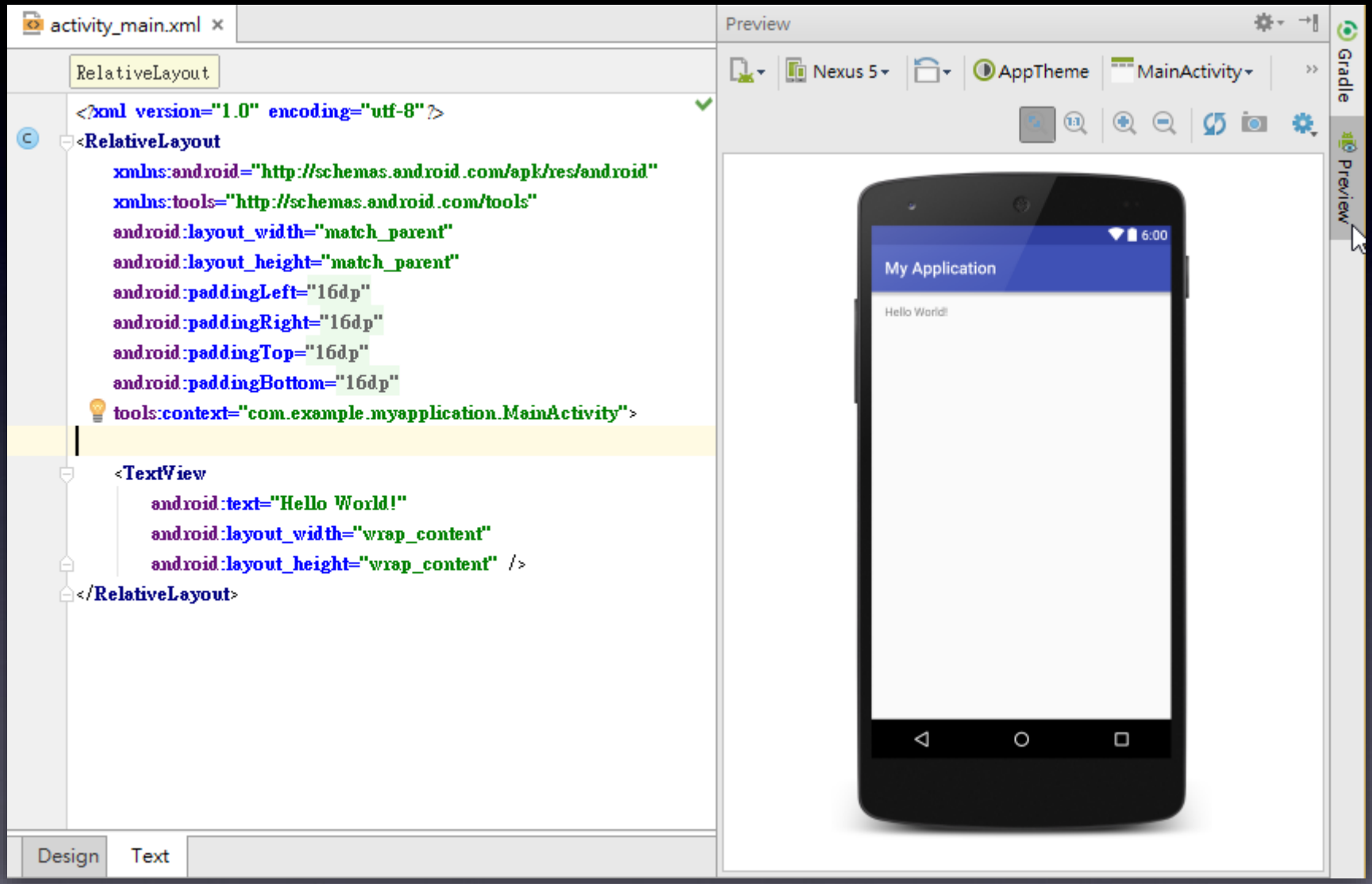
- Genymotion
  - https://www.genymotion.com/

LAYOUT EDITOR

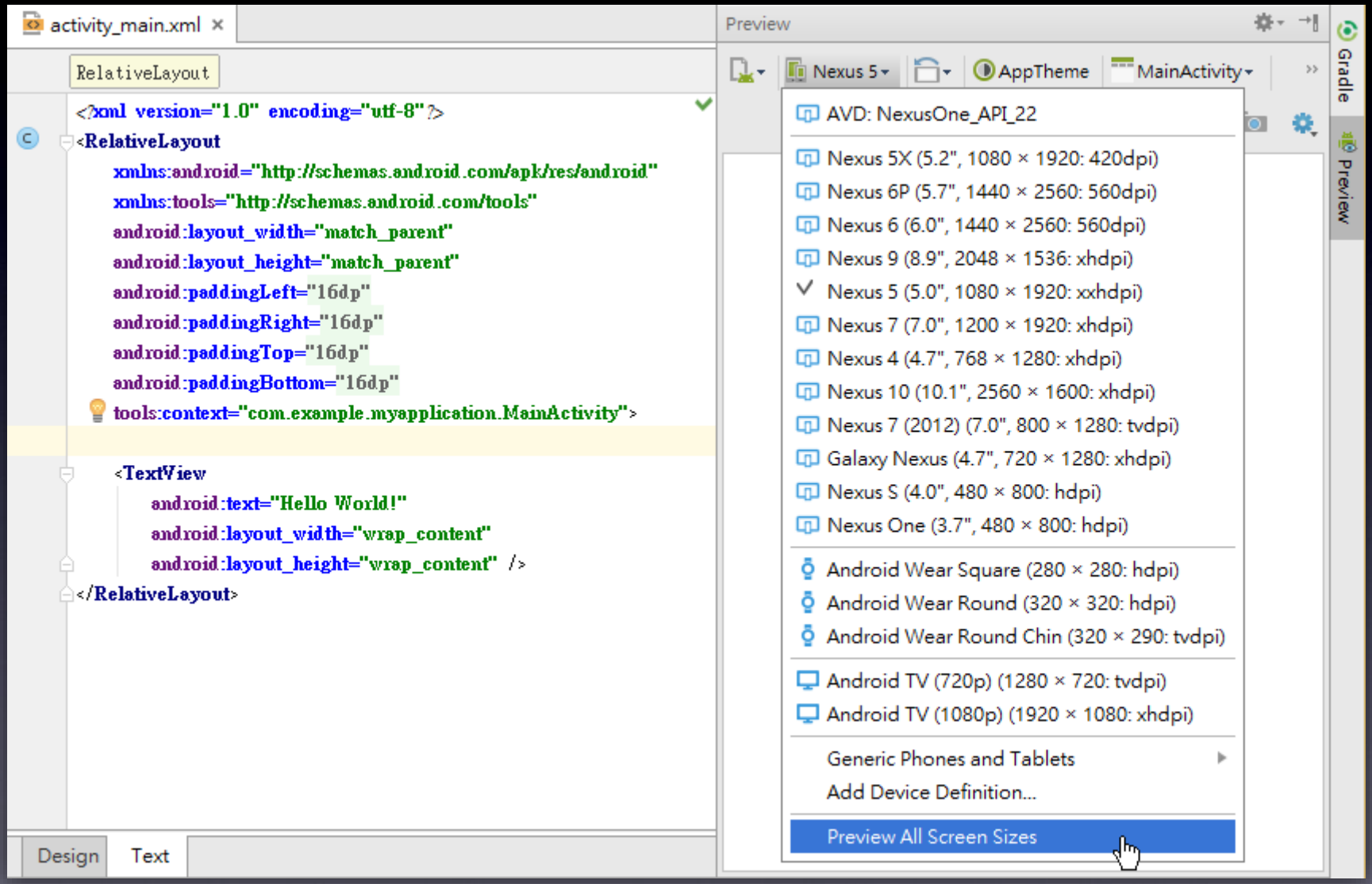
LAYOUT EDITOR

- 開啟專案介面設定檔案 app/res/layout/activity_main.xml

FAST WYWYG


activity_main.xml
RelativeLayout
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingLeft="16dp"
android:paddingRight="16dp"
android:paddingTop="16dp"
android:paddingBottom="16dp"
tools:context="com.example.myapplication.MainActivity">
<TextView
android:text="Hello World!"
android:layout_width="wrap_content"
android:layout_height="wrap_content" />
</RelativeLayout>
Design
Text
Preview
Nexus 5
AppTheme
MainActivity
Gradle
Preview
6:00
My Application
Hello World!

ALL SIZE PREVIEW

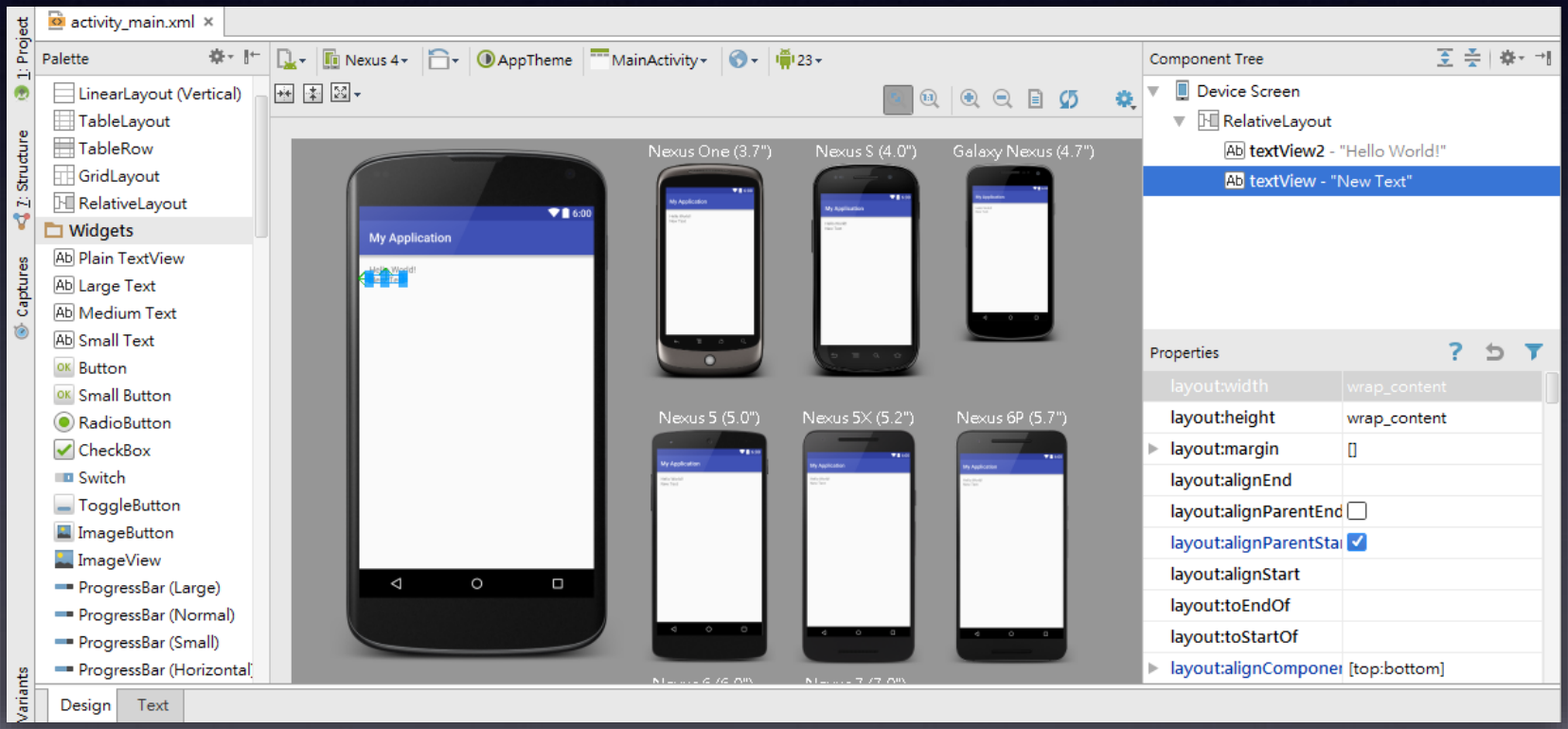


ALL SIZE PREVIEW

- Android Studio在Layout的預覽上非常全面快速

DESIGN模式

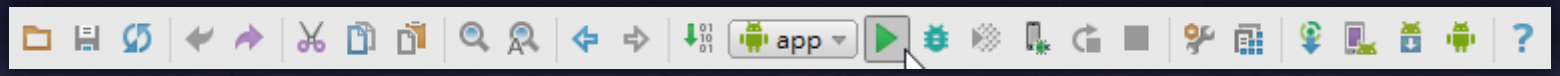
- 比Eclipse的Layout編輯模式更快速和方便



執行專案

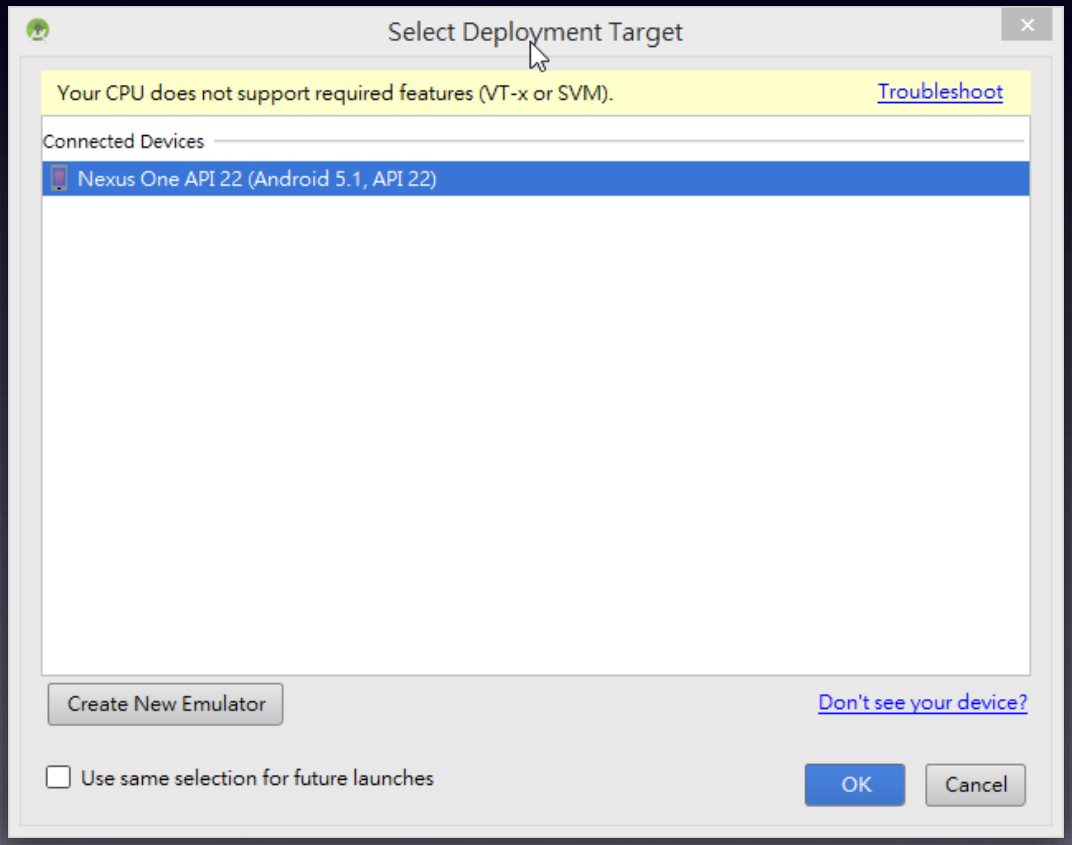
執行專案

- 在Android Studio最上方快捷工具列
  點選如下圖中滑鼠所點選的圖示



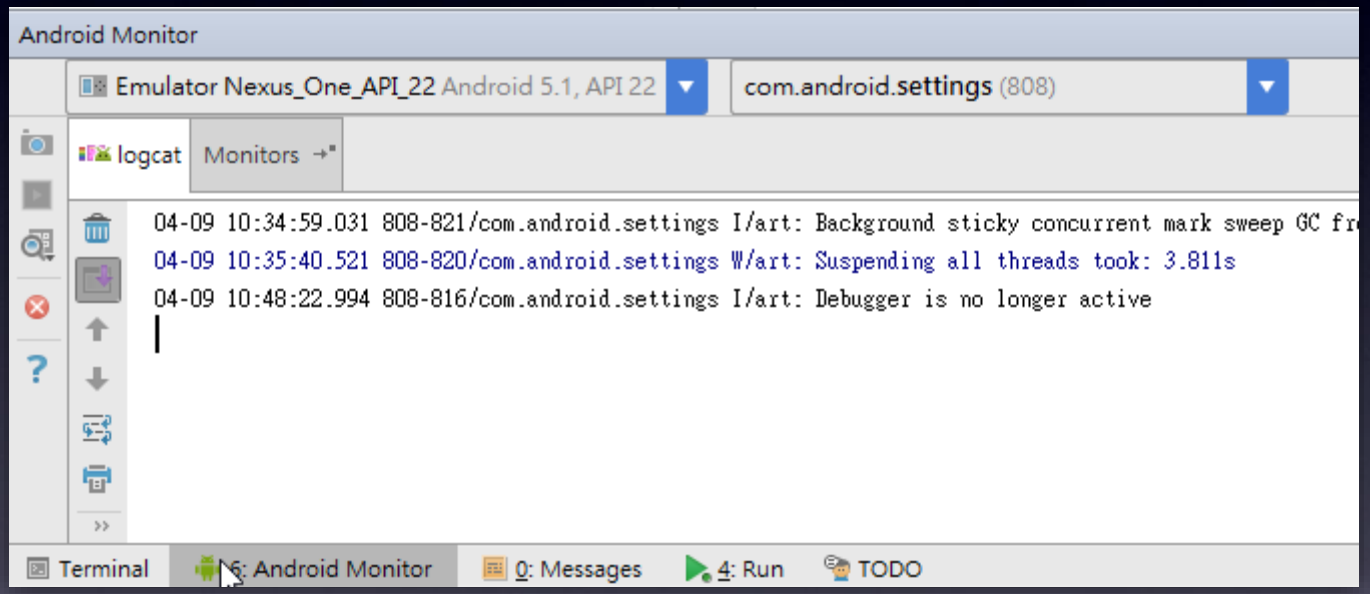
執行專案

- 顯示出部屬視窗，供選擇要部屬的機器
- 在Android Studio改為先決定要部屬的機器



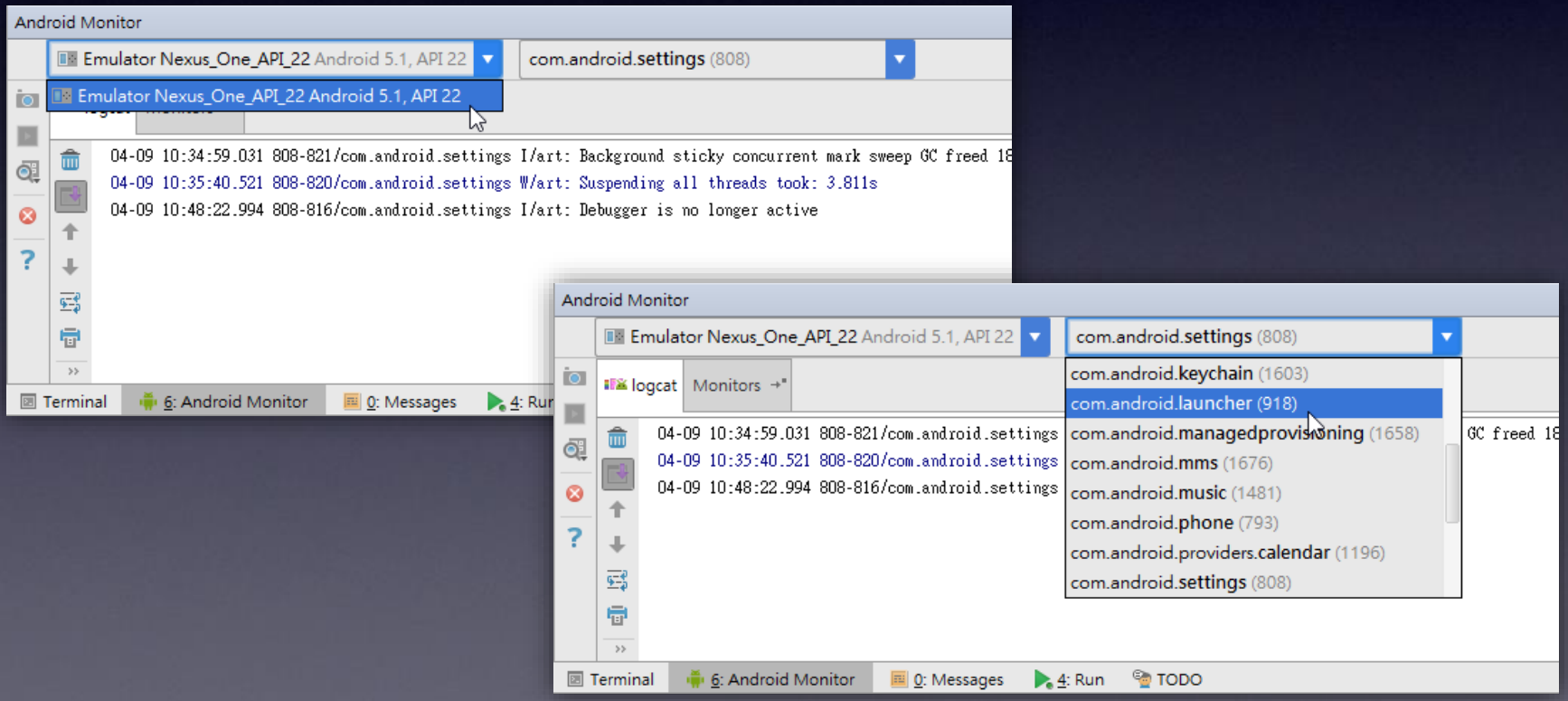
ANDROID MONITOR

- Android Studio整合Android Monitor (以前DDMS)的部分功能至預設第一層介面中



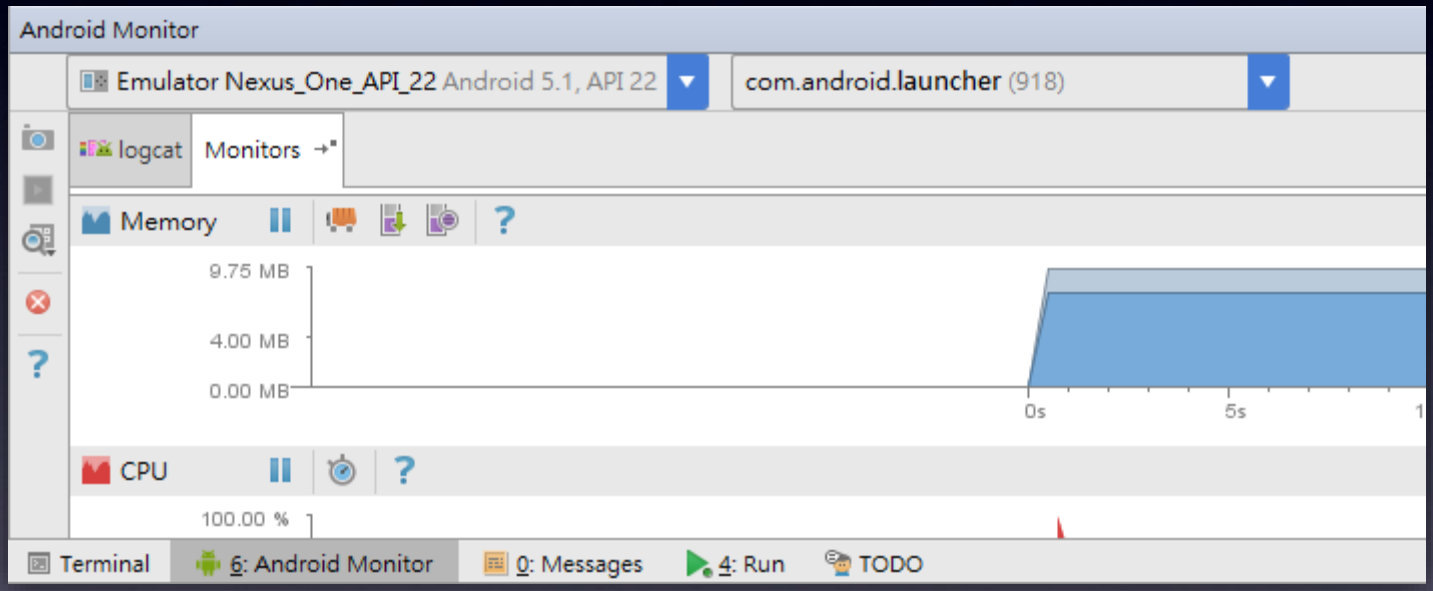
ANDROID MONITOR

- 下拉式選單可以選擇目前觀看哪台機器的Log
- 並可選擇該台機器的的Process去進行觀察



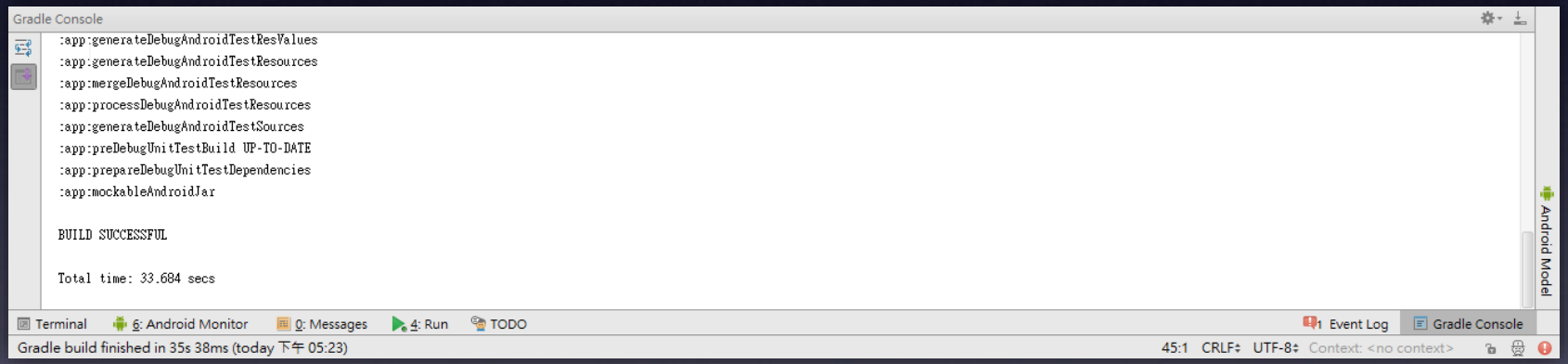
ANDROID MONITOR

- 新版整合視覺化即時的監控器，可以監控Process所花費的記憶體、CPU、網路流量等資訊



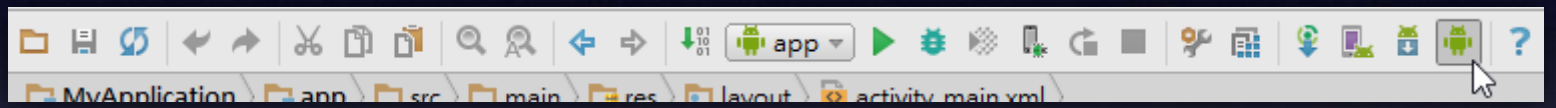
GRADLE CONSOLE

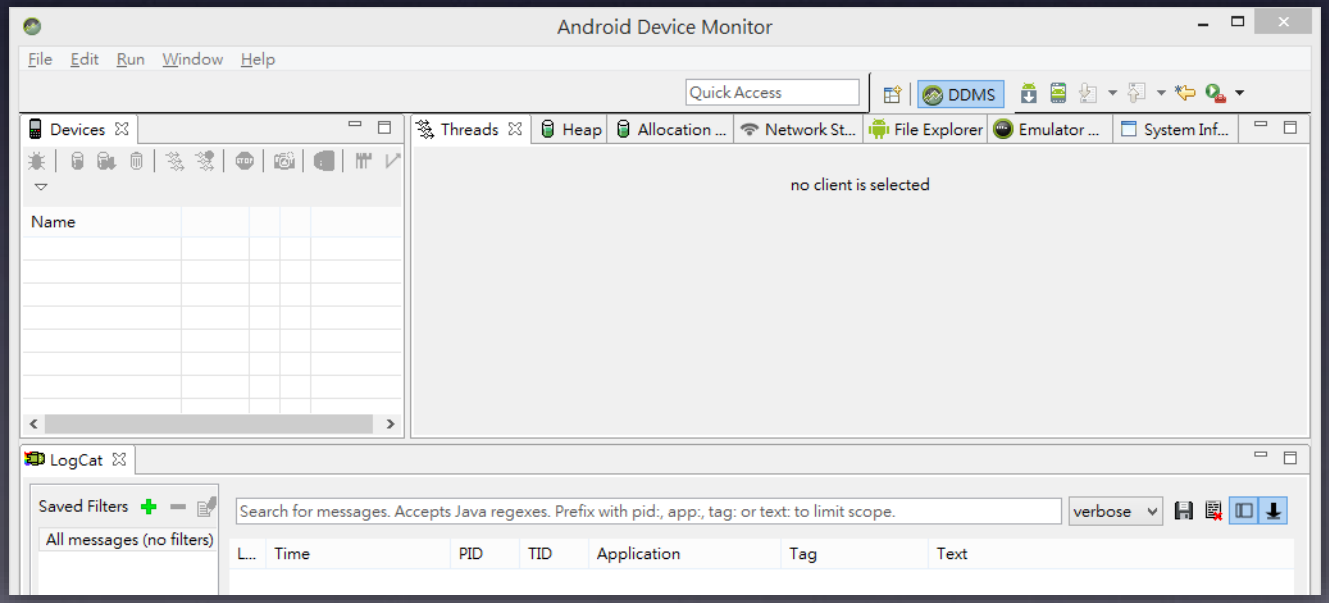
- Android Studio的最右下方可以看到Gradle Console
- 可以看到編譯App的過程，Gradle系統正在執行的Task，有錯誤出現也可更容易掌握



ANDROID DEVICE MONITOR

- 等同於舊有的DDMS
- 在Android Studio最上方快捷工具列
  點選如下圖中滑鼠所點選的圖示



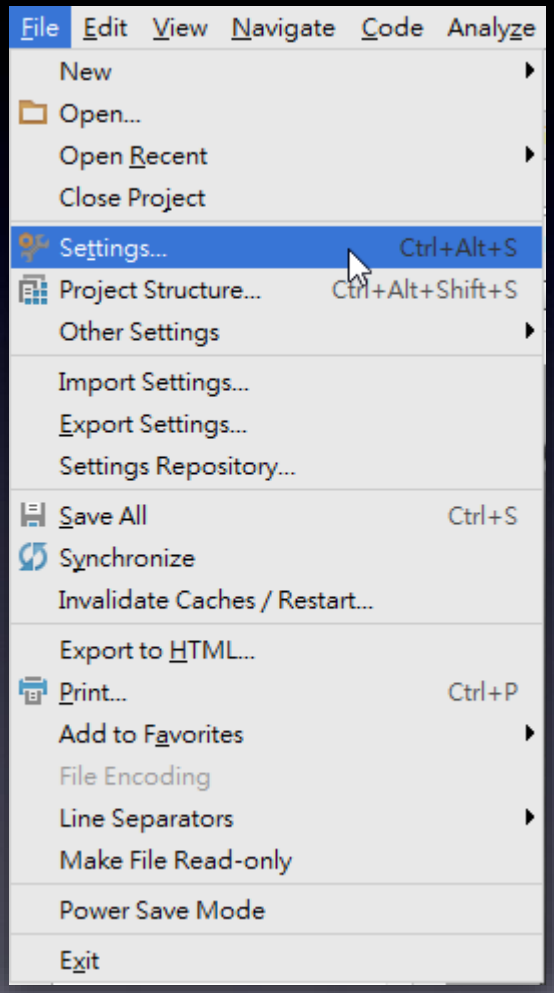


INSTANT RUN

- App執行後，就自動進入Instant Run模式
  - Activity不用重新啟動，畫面立刻呈現改變
  - https://www.youtube.com/watch?v=AefJXhCxrK8

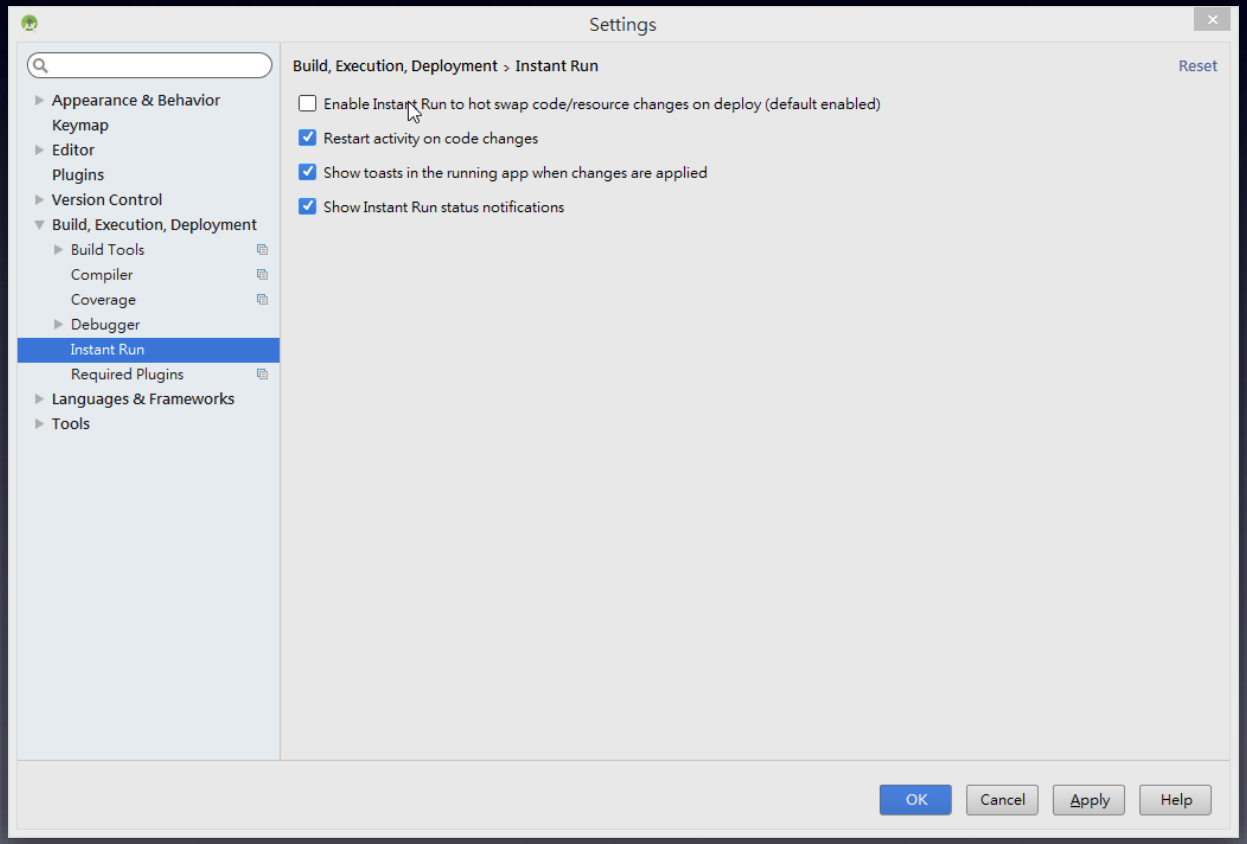
取消INSTANT RUN

- 選擇File -> Settings



取消INSTANT RUN

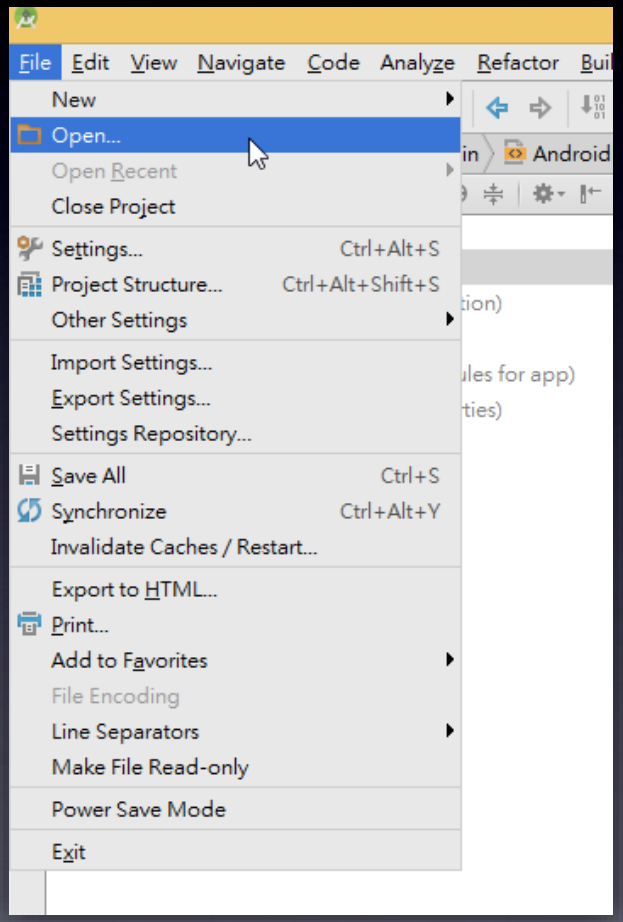
- 選擇Build, Execution, Deployment -> Instant Run
- 取消勾選 Enable Instant Run to hot .....
- 按下OK



開啟ANDROID STUDIO專案

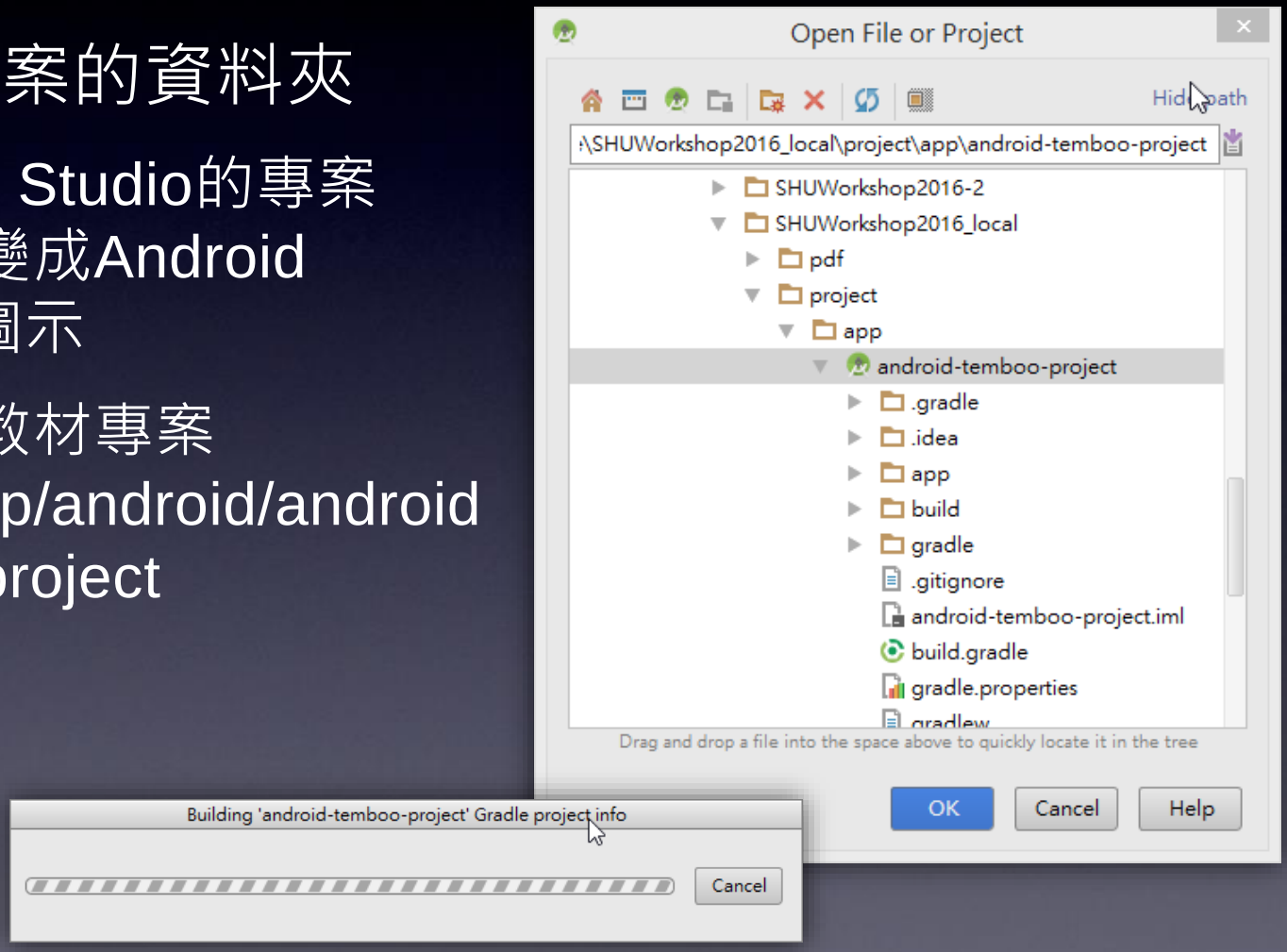
開啟ANDROID STUDIO專案

- 選擇右上方工具列
- File -> Open



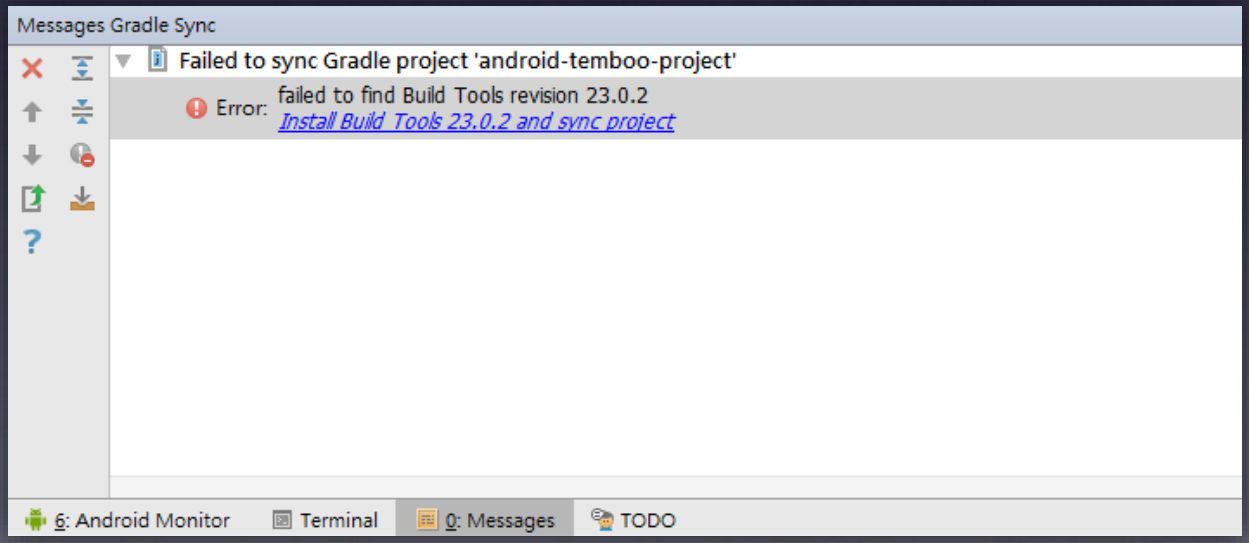
開啟ANDROID STUDIO專案

- 出現資料夾選取視窗
- 選擇含有專案的資料夾
  - 是Android Studio的專案資料夾會變成Android Studio的圖示
  - 試著開啟教材專案 project/app/android/android-temboo-project



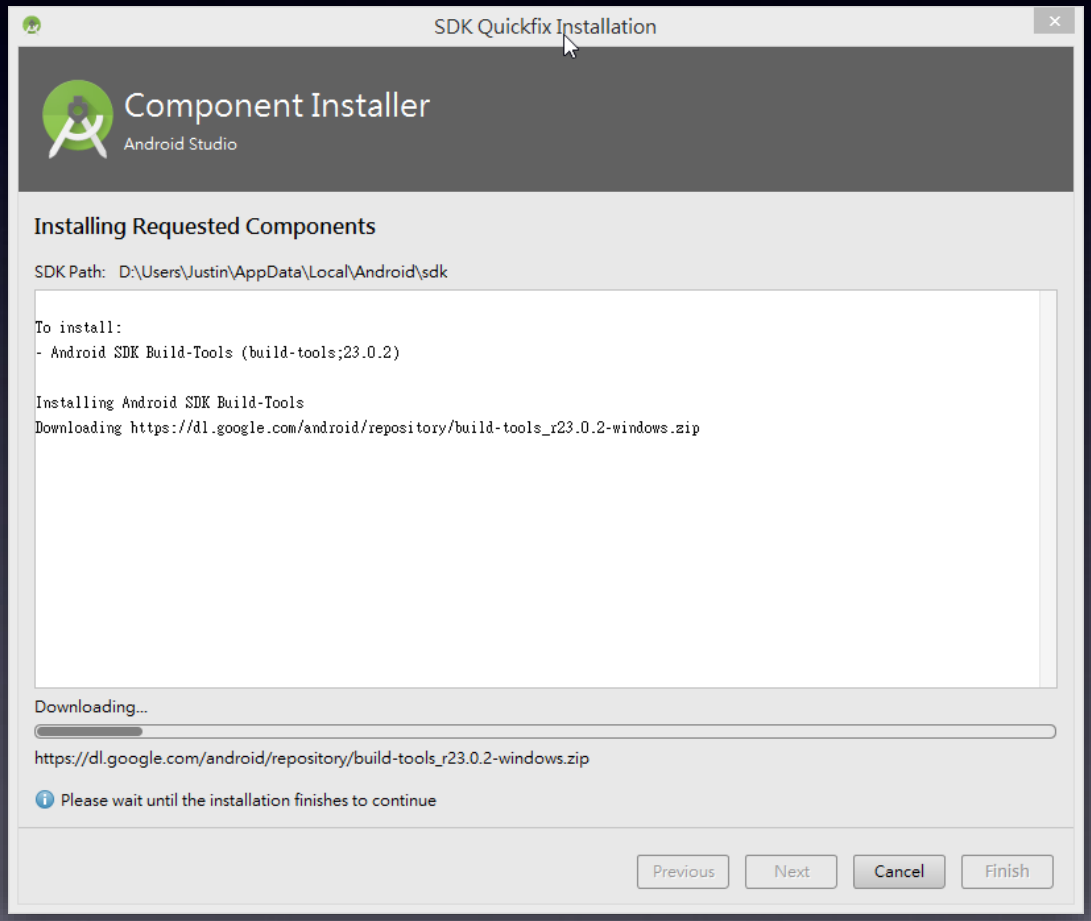
疑難排解

- 開啟別人專案時常會遇到如下圖的狀況
- 這表示你目前的電腦中有缺少編譯工具或是相依性套件導致不能編譯
- Android Studio會提示你如何解決，只要點下Error的連結即可



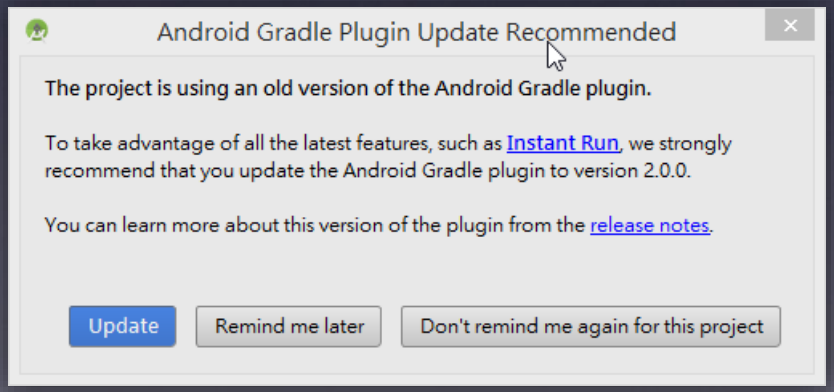
疑難排解

- Android Studio將會自動下載缺少的工具或套件



疑難排解

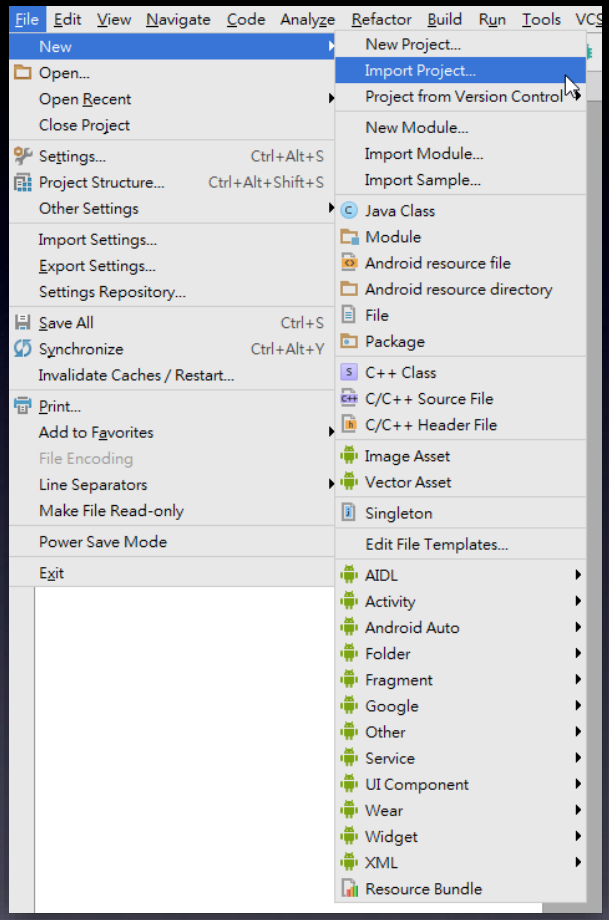
- 2016年4月初Android Studio推出了2.0版
  很多之前建立的Android Studio專案打開後會出現以下警示
- 因為Android Studio 2.0使用Gradle 2.0 (之前是1.8)
  這是Android Studio提示要升級編譯系統了
- 點下Update即可 (不升級就無法使用Instant Run)



開啟ADT專案

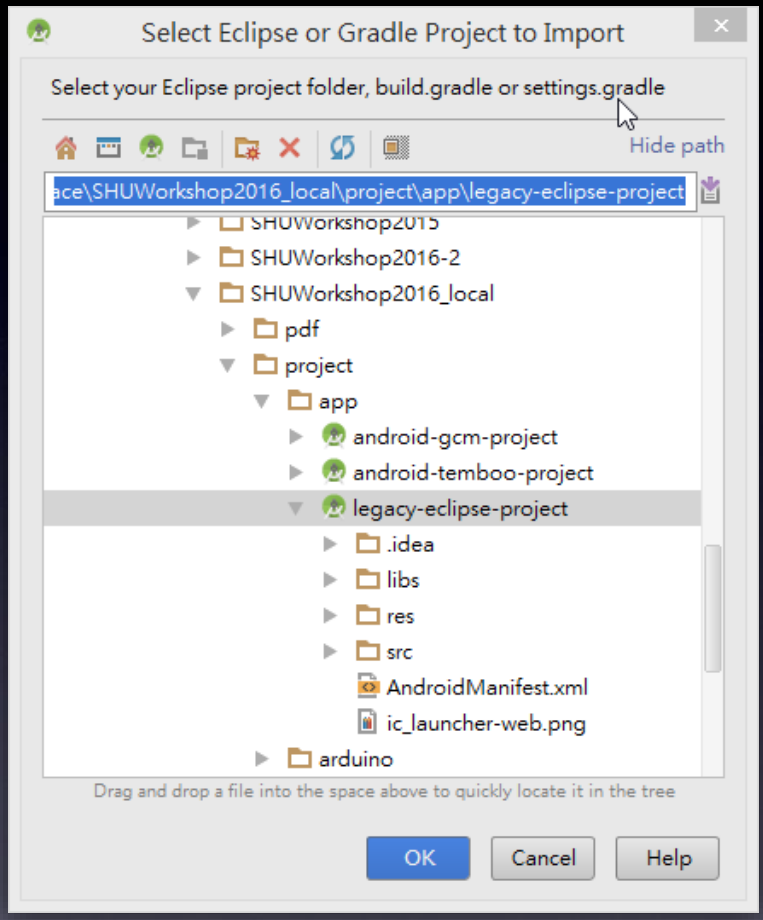
開啟ADT專案

- ADT是在Eclipse開發的Android專案
- 在Android Studio 1.5開始支援
- 到上方的工具選單
- File -> New -> Import Project...



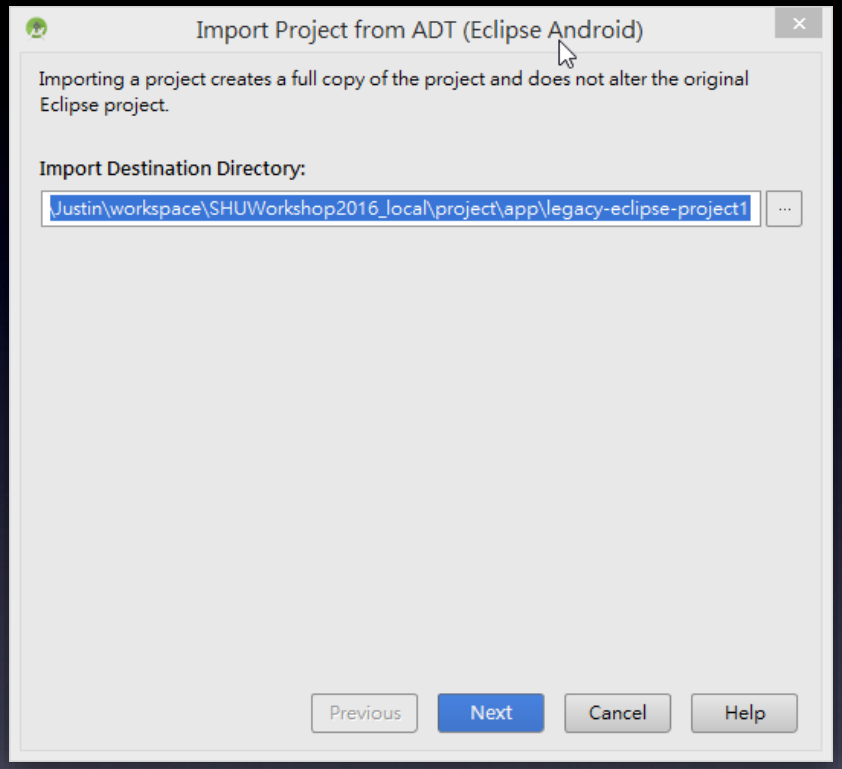
開啟ADT專案

- 出現資料夾選取視窗
- 選擇含有專案的資料夾
  - 是ADT的專案
    資料夾也會變成Android Studio的圖示
  - 試著開啟教材專案
    project/app/android/legacy-eclipse-project



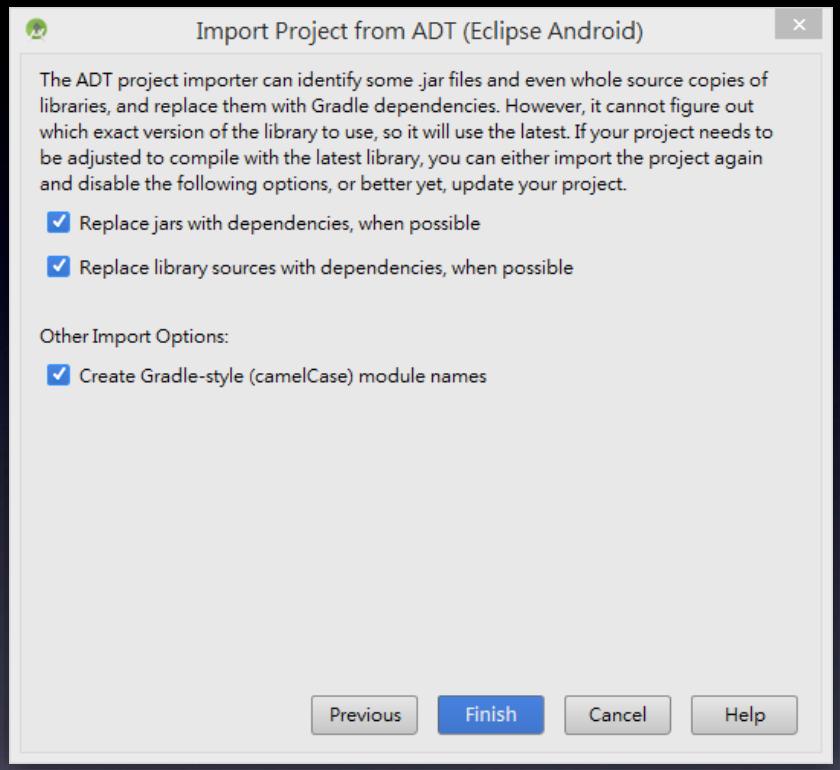
開啟ADT專案

- Android Studio採取專案重建的方式
  將ADT專案轉換為Android Studio專案
- 此處選擇轉換專案要存放的位置



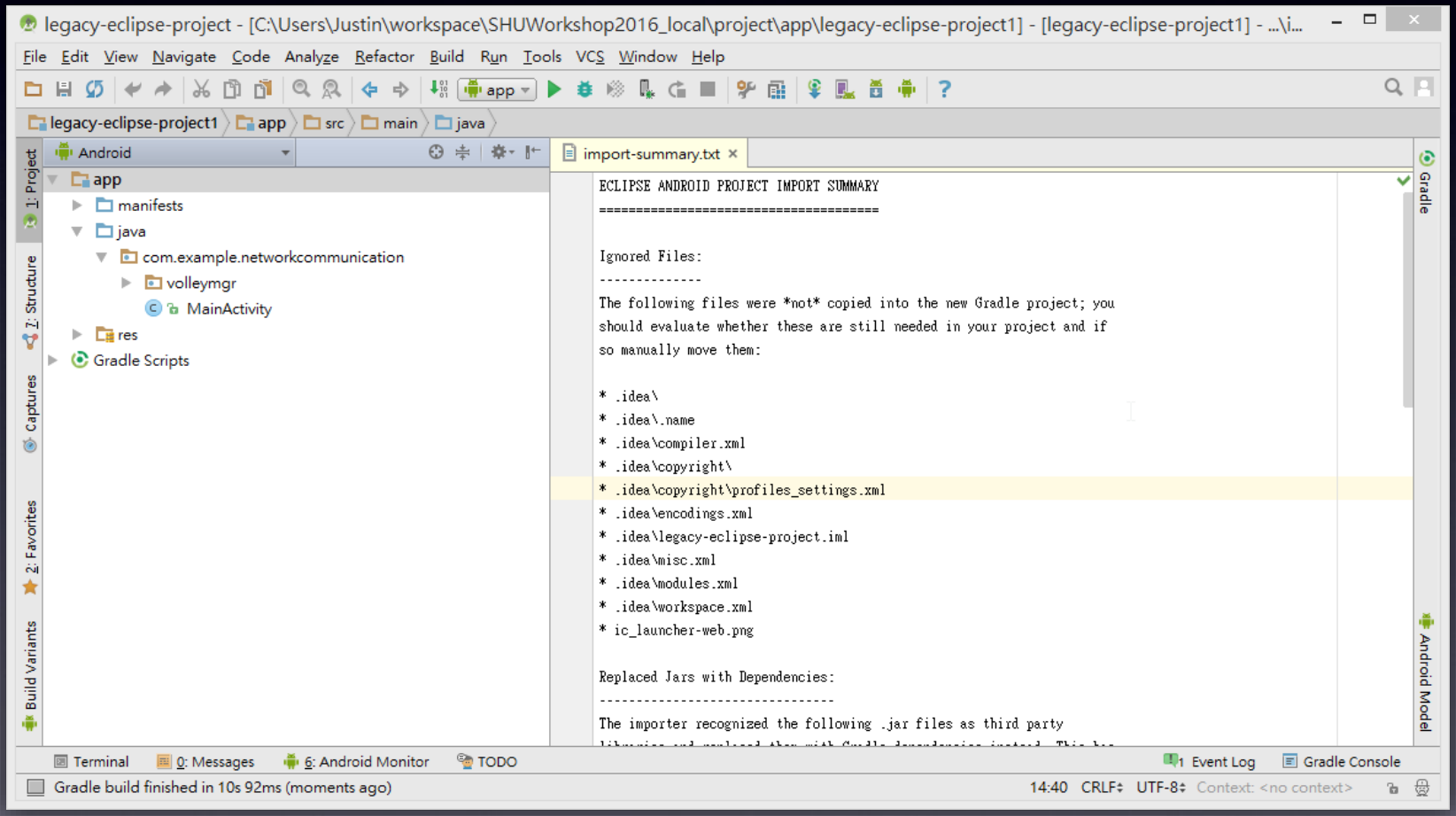
開啟ADT專案

- 轉換時會建議將ADT專案用的jar檔轉換為Gradle的dependencies
- 轉換ADT專案用到的函式庫類型專案變為Gradle的dependencies
  - 在Gradle稱為AAR
- 可將專案轉換為駝峰式命名法 (camelCase)



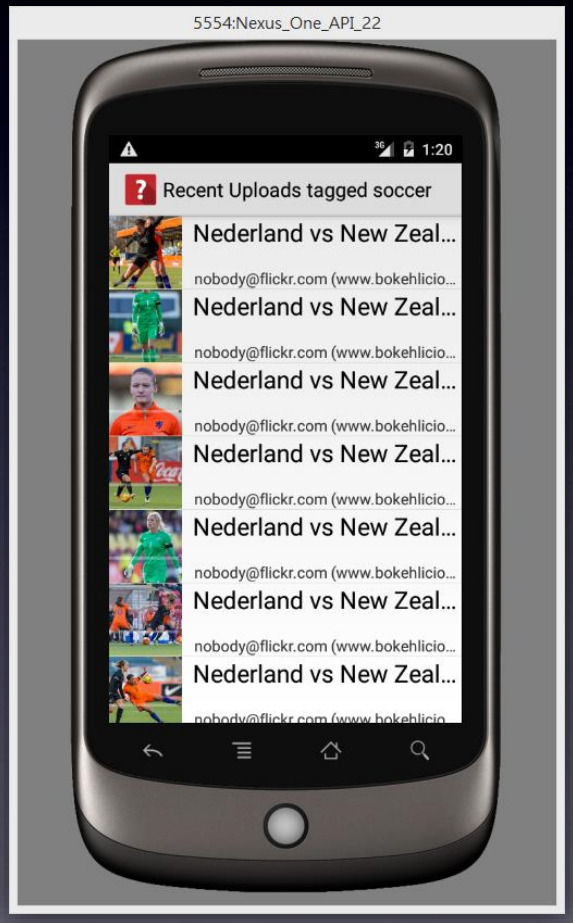
開啟ADT專案

- 匯入成功後會產生import-summary.txt說明專案變動



開啟ADT專案

- 試著執行轉換後的Android Studio專案



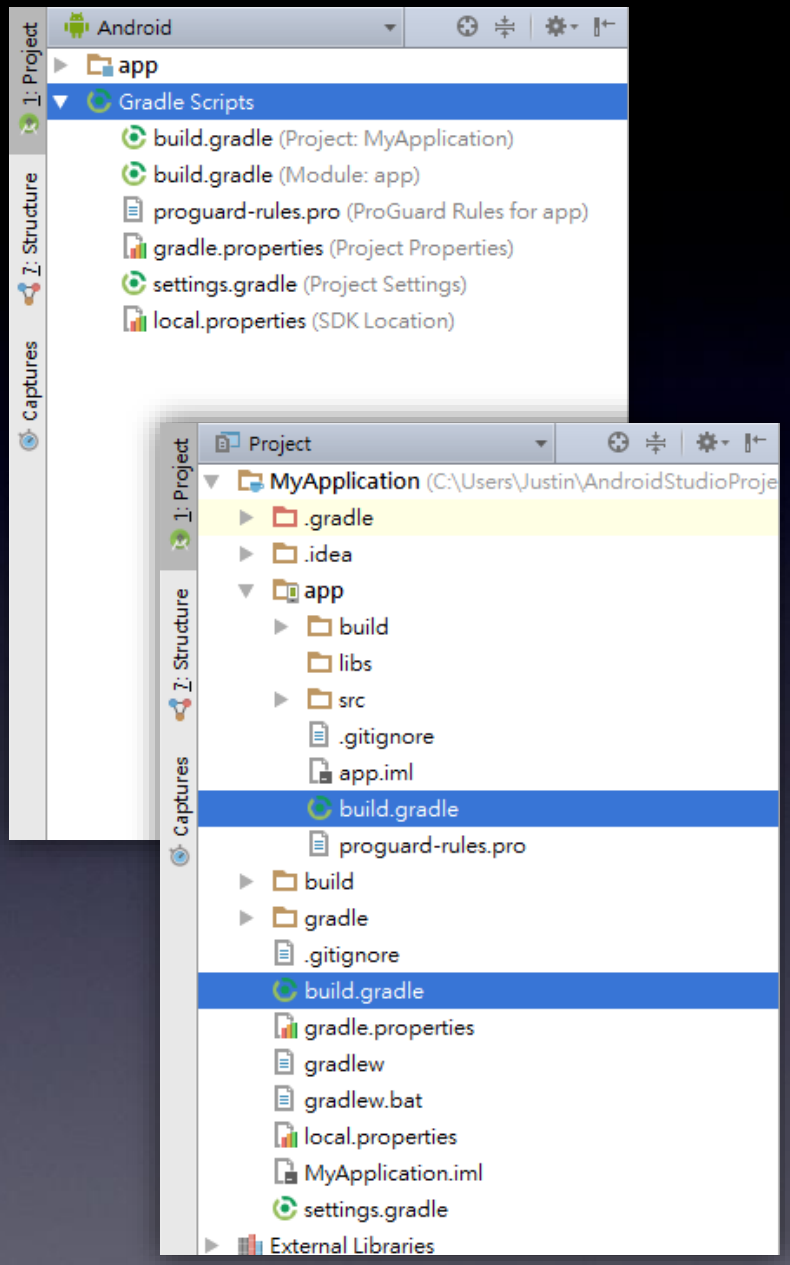
GRADLE説明

GRADLE說明

- http://www.codedata.com.tw/java/understanding-gradle-1-ant/
- 編譯系統，負責決定專案編譯的流程，相依檔如何匯入，中繼檔產生位置，編譯不同階段時要使用的工具等等
- 常使用的編譯系統
  - ANT
  - Maven
  - Gradle
- Gradle有相容於Maven

GRADLE說明

- 在Android Studio中，Gradle的編譯設定檔稱為build.gradle
- build.gradle有分為專案等級和模組等級的



BUILD.GRADLE

套入預設應用程式編譯的流程

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

android {
    compileSdkVersion 23
    buildToolsVersion "23.0.3"

    defaultConfig {
        applicationId "com.example.myapplication"
        minSdkVersion 15
        targetSdkVersion 22
        versionCode 1
        versionName "1.0"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.3.0'
}
```

BUILD.GRADLE

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

android {
    compileSdkVersion 23
    buildToolsVersion "23.0.3"

    defaultConfig {
        applicationId "com.example.myapplication"
        minSdkVersion 15
        targetSdkVersion 22
        versionCode 1
        versionName "1.0"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.3.0'
}
```

定義App的相關設定，輸出方式

BUILD.GRADLE

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

android {
    compileSdkVersion 23
    buildToolsVersion "23.0.3"

    defaultConfig {
        applicationId "com.example.myapplication"
        minSdkVersion 15
        targetSdkVersion 22
        versionCode 1
        versionName "1.0"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.3.0'
}
```

編譯App的Android SDK版本

BUILD.GRADLE

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

android {
    compileSdkVersion 23
    buildToolsVersion "23.0.3"

    defaultConfig {
        applicationId "com.example.myapplication"
        minSdkVersion 15
        targetSdkVersion 22
        versionCode 1
        versionName "1.0"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.3.0'
}
```

編譯App時使用的工具版本

BUILD.GRADLE

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

android {
    compileSdkVersion 23
    buildToolsVersion "23.0.3"

    defaultConfig {
        applicationId "com.example.myapplication"
        minSdkVersion 15
        targetSdkVersion 22
        versionCode 1
        versionName "1.0"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.3.0'
}
```

應用程式的版本、相容性、
package定義處

BUILD.GRADLE

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

android {
    compileSdkVersion 23
    buildToolsVersion "23.0.3"

    defaultConfig {
        applicationId "com.example.myapplication"
        minSdkVersion 15
        targetSdkVersion 22
        versionCode 1
        versionName "1.0"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.3.0'
}
```

輸出版本時要做的步驟
分為release和debug

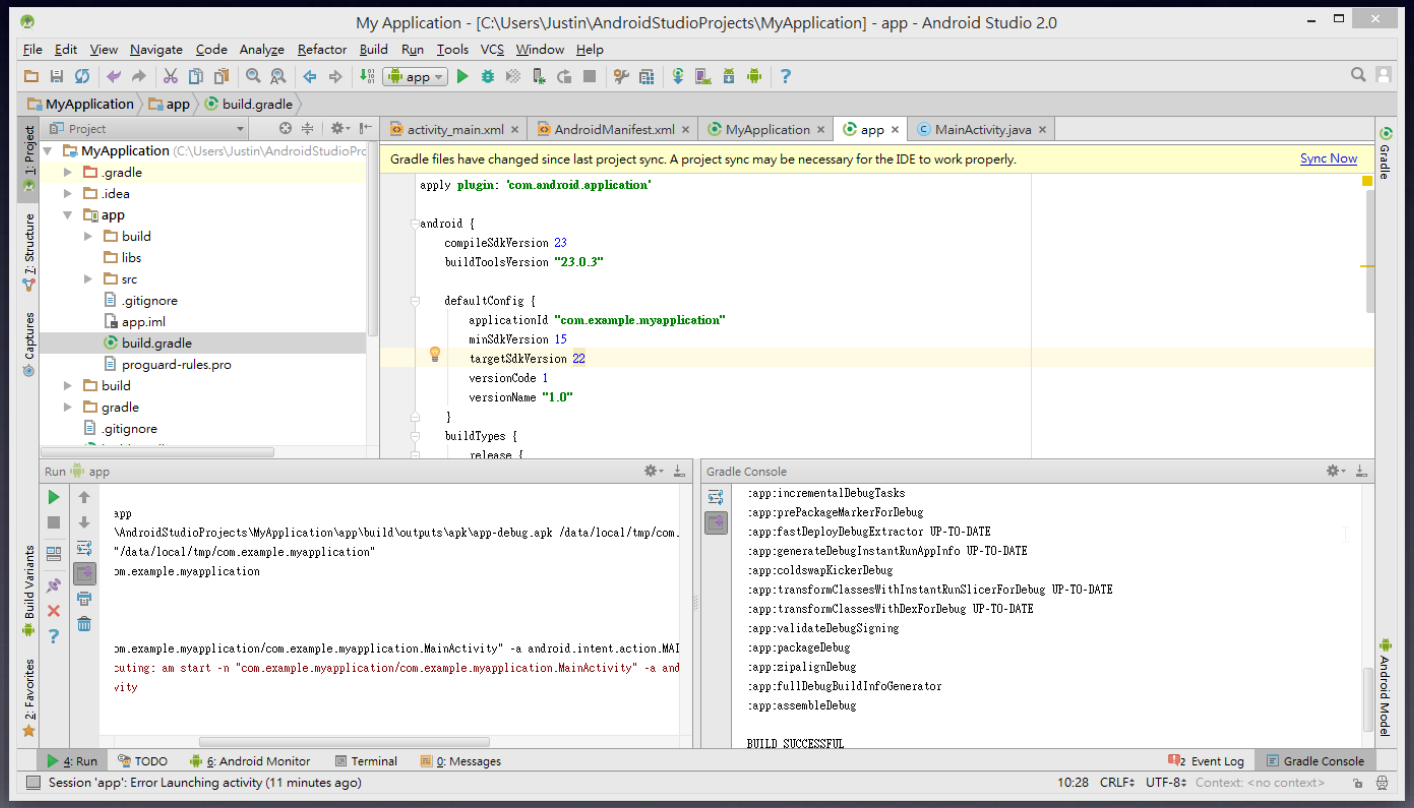
BUILD.GRADLE

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

android {
    compileSdkVersion 23
    buildToolsVersion "23.0.3"

    defaultConfig {
        applicationId "com.example.myapplication"
        minSdkVersion 15
        targetSdkVersion 22
        versionCode 1
        versionName "1.0"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.3.0'
}
```

設定專案要使用到的
相依性函式庫

GRADLE說明

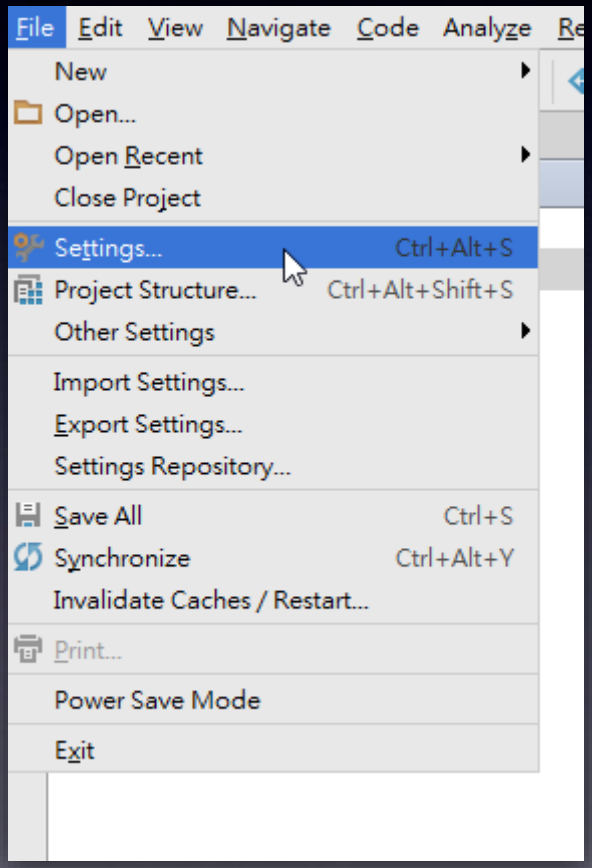
- 在Android Studio中有更動到build.gradle，都可點選上方Sync Now。Android Studio自動去下載相依套件、將設定套入至編譯流程中



快速鍵

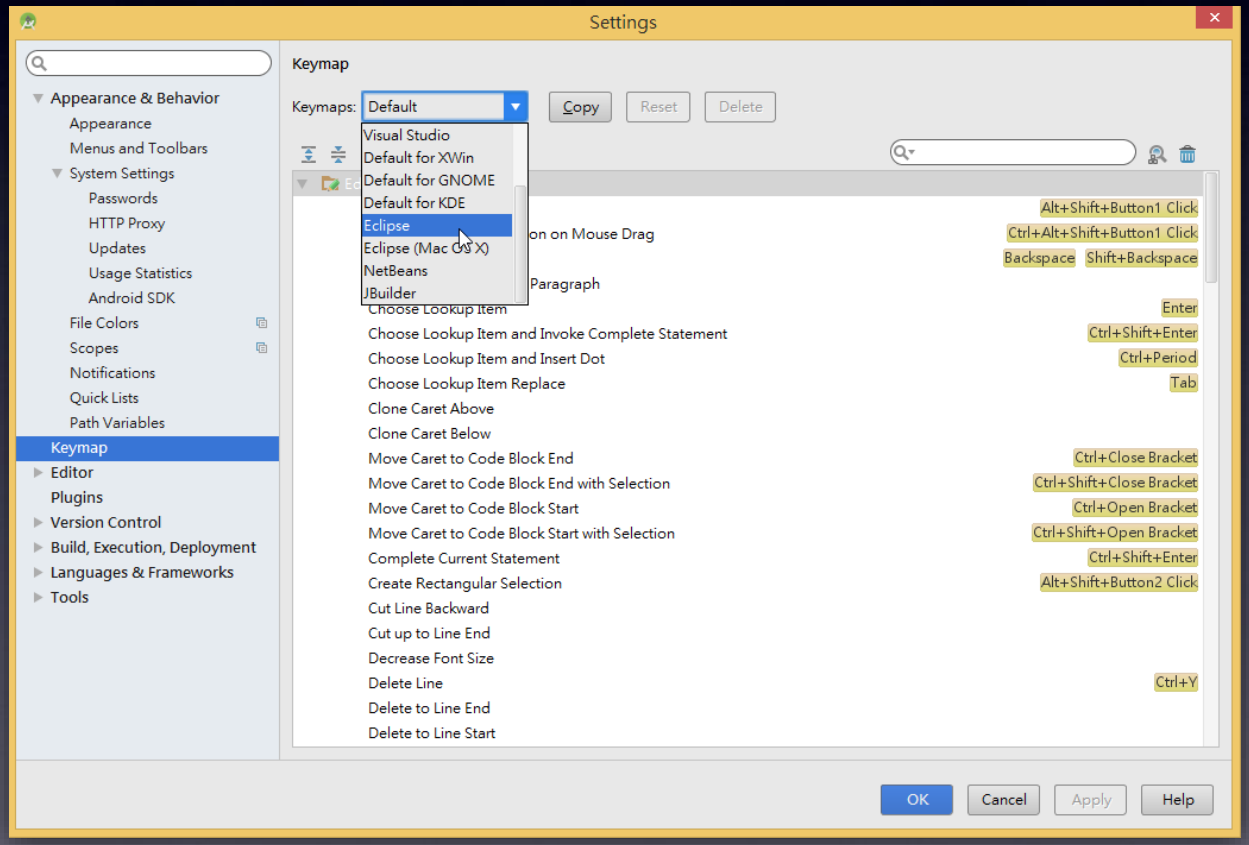
修改KEYMAP

- 選擇File -> Settings



修改KEYMAP

- 選擇Keymap
- 下拉式選單選擇Eclipse



快速鍵

- CTRL + D 刪除一整行
- ALT + / 協助修復、完成
- CTRL + F 尋找檔案
- F3 快速跳轉至宣告處