

# Android APP手機程式設計實務



講師：賴貴平



# 課程內容規畫與簡介一

- 本課程時數預計**160**小時，將包含以下教學內容：

Android 相關基礎認識

Android SDK開發環境

JAVA語言基礎

APP程式設計基礎架構

UI 使用者介面設計與規劃

Android 基礎介面元件應用

系統事件處理與委派機制

Android 進階介面元件應用

使用者互動訊息與對話框

Activity類別與進階架構



# 課程內容規畫與簡介二

- 本課程時數預計**160**小時，將包含以下教學內容:

Intent類別與系統應用

服務與廣播接收器的操作

內容提供者與相關應用

硬體裝置操作實務:相機

硬體裝置操作實務:多媒體

硬體裝置操作:速度感應器

硬體裝置操作:磁場感測器

硬體裝置操作:GPS系統

Google Map 與定址實作

SQLite 資料庫與數據儲存



# 本節課程內容

- 本節課程內容將包含以下教學內容:

## Android 相關基礎認識

- ANDROID 行動作業系統發展
- ANDROID 系統重大特點、版本與市占
- ANDROID 系統架構與組成元件
- ANDROID 開發環境與工具
- ANDROID STUDIO 設定與安裝



# 本節課程內容

- 本節課程內容將包含以下教學內容:

## Android 相關基礎認識

- Java SE Development kit 設定與安裝
- 開發環境設定
- 模擬器的使用
- 程式開發初探
- Hello Android 程式



# ANDROID 行動作業系統

- **Android** 這個術語於**1863**年間，出現在美國專利文獻中，代表小型的人形玩具，一種機器人。
- **Android**（讀音：['ænˌdrɔɪd]），中文名稱為**安卓**，是一個基於**Linux**核心的開放原始碼行動作業系統，主要設計用於觸控螢幕行動裝置。由於其免費開放原始碼許可的授權方式，目前被智慧型手機和平板電腦與其他可攜式裝置上。
- **2003**年**10**月，有**Android**之父之稱的**安迪·魯賓**（**Andy Rubin**）在美國加州帕洛阿爾托建立了**Android**科技公司，並與三位主要夥伴開發製作。**2005**年**7**月**11**日，**Google**收購了**Android**科技公司，目前由**Google**成立的**Open Handset Alliance**（**OHA**，開放手機聯盟）持續發展**Android** 作業系統。



# ANDROID 行動作業系統

- 2007年11月，Google結盟共84家，包含硬體製造商、軟體開發商及電信營運商成立OHA(開放手機聯盟)，共同研發改良Android
- Google以Apache免費開放原始碼授權(Linux作業系統核心則為GPLv2)的方式，發佈了Android的原始碼，讓生產商推出搭載Android的智慧型手機，後來更逐漸拓展到平板電腦及其他領域上。
- 2010年末的統計數據顯示，Android作業系統雖僅正式推出約兩年，在市場佔有率上已經超越推出逾十年的諾基亞Symbian系統。





# ANDROID 系統重大特點

- **Android**的作業系統核心程式源自**Linux**，屬於其核心的一個分支(**Branch**)，具有典型的**Linux**排程和功能。但由於**Android**去除了**Linux**中的**X Window System**、不支持標準的**GNU**程式庫，造成**Linux**平台上的應用程式無法在**Android**平台直接執行
- **Android**作業系統 **2.2**版開始，語音功能可以用來輸入文字、或進行語音導航等功能
- **Android 3.0** ( **Honeycomb**蜂巢 ) 是第一個**Android**平板作業系統。全球第一個使用該版本作業系統的設備是摩托羅拉公司於**2011**年**2**月**24**日發布的**Motorola Xoom**平板電腦，主要特色有:專為平板電腦設計、支援**Google eBooks**、支援平板電腦大螢幕、高解析度、支援**3D**加速處理





# ANDROID 系統重大特點

- **Android作業系統 2.2版至 4.0版**能原生支援Flash，**4.4版**以後完全去除對Flash的支援(由於安全性以及執行效率問題)
- **Android作業系統**可支援多種硬體及感測器，包括:視訊/照片攝像鏡頭，多點電容/電阻觸控螢幕，**GPS**，重力加速計，陀螺儀，氣壓感測器，磁力測器(高斯計)，體感控制器，藍牙裝置，無線設備，壓力感測器，溫度計，**3D圖形加速**等
- **Android作業系統 4.0版**開始，支援畫面截圖功能，可以允許用戶直接抓取智慧型手機螢幕上的任何畫面
- **Android作業系統 4.3版本**開始支援 **OpenGL ES 3.0**，能有效率的執行各式**3D應用**



# ANDROID 系統重大特點

- **Android 4.4**引入**Android Runtime (ART)**作為新的運作環境，在安裝應用程式時，它會使用提前編譯（**AOT**）來把應用程序字節碼完全編譯為機器語言
- 在**Android 4.4**中，**ART**是一項實驗性功能，默認情況下不啟用；它成為**Android 5.0**的下一個主要版本中唯一的運作選項
- **Android**作業系統中的應用程式大部分由**Java**程式語言編寫，並轉換為**Dalvik executables**執行碼在**Dalvik**虛擬機上執行。**5.0**版（代號**Lollipop**）開始以**Android Runtime (ART)**取代**Dalvik**虛擬機



# ANDROID 行動作業系統

- 2014年6月，Google發布了Android One，是一組「硬件參考模型」，這將容許（設備製造商）以低成本輕鬆打造高品質手機。同年9月，Google宣布推出首款在印度發布的Android One手機
- 2017年3月全球網路流量和設備數量統計，Android系統均超越Microsoft Windows(視窗作業系統)，正式成為全球第一大作業系統
- 2018年3月7日，Google發布Android P的開發者預覽版本，並開放下載。2018年5月8日，Google I/O大會公布許多Android P功能。2018年8月6日，Android 9 (Pie - API 28)發行



# ANDROID 行動作業系統

- Android 10首個開發者預覽版本（測試版）在2019年3月14日發行並提供下載
  - 2019年8月23日，Google宣布命名 Android Q為Android 10（API 29）
- 正式版於 2019年9月3日發佈
  - 新增相片的動態景深格式，允許在拍照後更改景深模糊程度
  - 支援AVI視訊編解碼器、HDR10+影片格式和Opus音訊編解碼器
  - 新增浮動設定面板（Floating settings panel），允許直接從應用程式中更改系統設定
  - 為應用程式中的生物辨識技術提供更好的支援



# ANDROID 行動作業系統

- Android 11.0 ( Android S - API 30 ) 2020年2月20日發佈開發者預覽版本
- 正式版於 2020年9月發佈
  - 原生內建螢幕錄影
  - 更新語音操控
  - 更新權限管理
    - 使用者可授予應用程式一次性許可權
    - Zombie許可：當長時間沒有使用應用程式時，許可權會被重設
  - 更新通知功能通知欄新增通訊程式「對話」組別
  - 通知欄新增媒體播放的獨立區域
  - 錄影時暫停通知



# ANDROID 行動作業系統

- Android 12 (Android S)的第一個測試版於2021年5月18日發布
  - 支援AVIF圖像
  - 支援捲動長截圖
  - Android Runtime ( ART ) 模組加入可以通過Google Play升級的核心作業系統元件，現有模組功能升級
- 正式版於2021年10月4日發布 ( API 31–32 )



ANDROID

# ANDROID 行動作業系統

- Android 13 ( Tiramisu - API 33 ) ，測試版本在2022年2月10日發布，最初始用於Google Pixel系列
  - Pixel 4
- 正式版於2022年8月15日發布
  - 新增應用程式通知權限
    - APP想發送通知時會彈出確認視窗，需要使用者手動許可
  - 針對大尺寸螢幕、摺疊螢幕的手機進行最佳化處理
    - 摺疊螢幕裝置可針對不同場景、不同螢幕尺寸進行顯示設定
    - 進一步改進顯示模式，提升多螢幕設備的顯示效果
  - 改進記憶體配置較低裝置的執行效能
    - AOSP 的原始碼中新增MGLRU ( Multi-Generational Least Recently Used ) 的支援
    - 虛擬記憶體管理程序的 CPU 使用率最高可以降低 40%





# ANDROID 行動作業系統

- Android 14 ( Upside Down Cake - API 34 ) ， 測試版本Beta1 在2023年4月14日發布，可安裝於Google Pixel系列
  - Pixel 4a 或 更新機種
- 正式版於2023年10月4日發布
- 強化平板電腦和可折疊裝置功能的使用體驗
  - 針對大尺寸螢幕、折疊螢幕機種
  - 提供分離式鍵盤、新任務欄、雙面板通知欄
- 提供更高階解析度的Ultra HDR照片格式
  - 支援 Ultra HDR 攝錄，能拍攝、瀏覽色彩鮮明、陰影清晰、畫質更佳的照片或影片



# ANDROID 行動作業系統

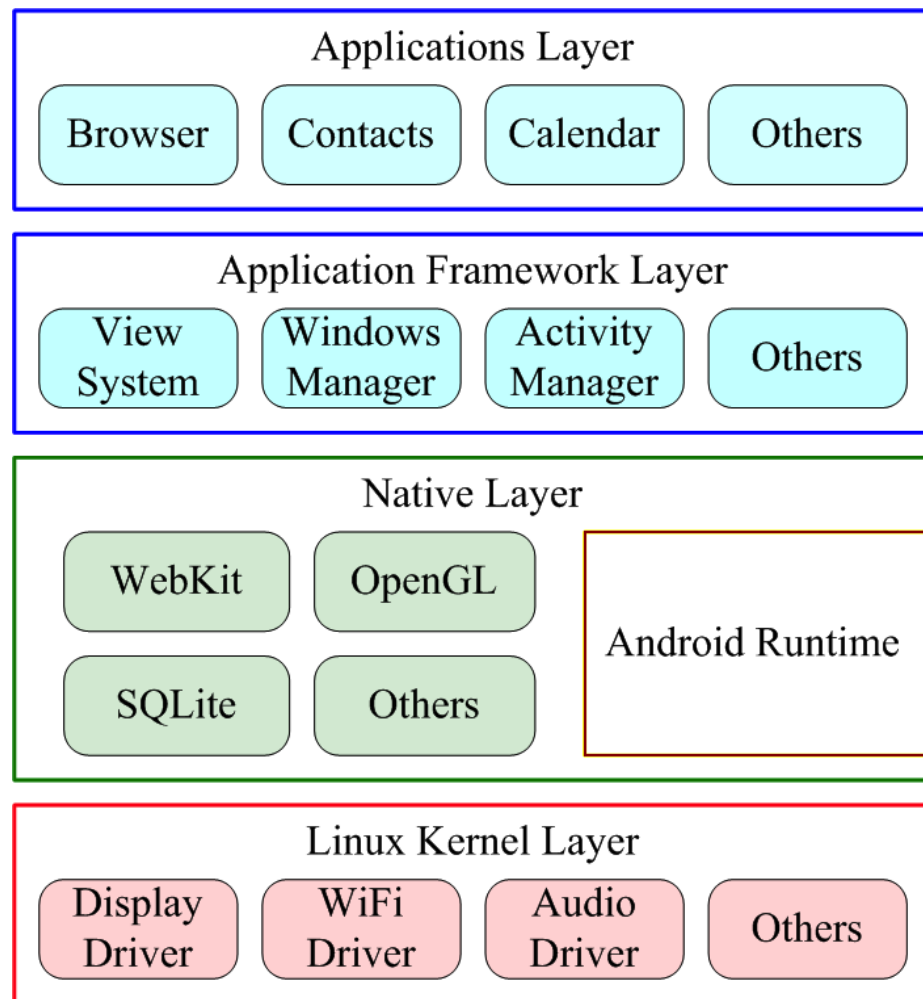
## • Android 15 ( Vanilla Ice Cream - API 35 )

- 測試版本 Beta 1.2 在2024年2月16日發布
- 原始碼於 2024年9月4日 發布
- 正式版於 2024年10月15日 發布，Google Pixel 系列手機優先更新
  - 更新 OpenJDK 17 API，改善 NIO 緩衝區、資料流程、安全性等核心功能
  - 更新 Android 動態性能框架 (ADPF) API，讓遊戲、性能負擔大的應用程式得以更直接的控制 Android 設備的電源和散熱系統
  - 更新衛星功能支援
  - 改進硬體的 NFC 操作，使用者體驗更流暢
  - 虛擬 MIDI 2.0 裝置的應用擴充 UMP 支援，允許合成器應用程式能作為虛擬 MIDI 2.0 裝置操作控制，一致性 USB MIDI 2.0 裝置操作方式



# ANDROID 系統架構

- Android 作業系統架構大致上分為四層



應用程式層

應用程式框架層

原生函式庫層(不含GNU C函式庫)

Linux 作業系統核心層



# ANDROID 程式組成元件

- **Activity（活動）**
  - Activity 類似於一個使用者介面元件，通常對應於一個畫面
- **BroadcastReceiver（廣播接收器）**
  - 回應廣播的意向類別（**Intents Class**）
- **Service（服務）**
  - 在背景運行的無畫面工作任務
  - 音樂播放器
- **ContentProvider（內容提供者）**
  - 允許應用程式共用資料內容



# ANDROID 程式組成元件

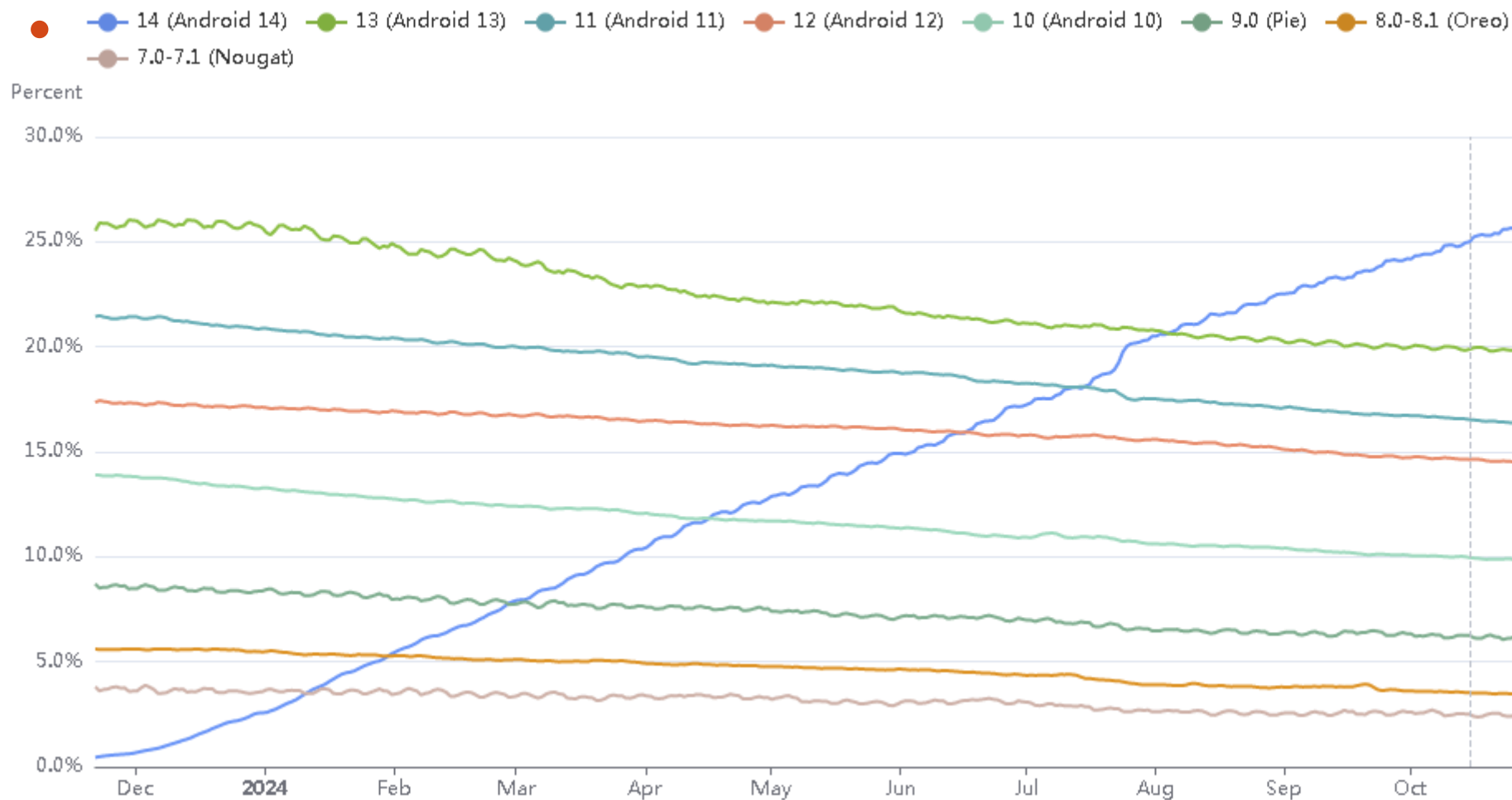
- 統計至2024 年10 月，Android作業系統各版本的用戶比例

版本號+版本名稱	<u>API</u>	用戶分佈
Android 4.4 KitKat	19	0.4 %
Android 5.0 Lollipop - 5.1 Lollipop	21 - 22	0.7 %
Android 6 Marshmallow	23	0.7 %
Android 7.0 Nougat - 7.1 Nougat	24 - 25	2.6 %
Android 8.0 Oreo - 8.1 Oreo	26 - 27	3.5 %
Android 9 Pie	28	6.3 %
Android 10 Q	29	10 %
Android 11 R	30	16.4 %
Android 12 S - S_V2	31-32	12.4 %
Android 13 Tiramisu	33	19.3 %
Android 14 Upside Down Cake	34	25.8 %
Android 15T	35	-



# ANDROID 程式組成元件

● 統計至2024年10月，Android作業系統各版本的用戶比例





# ANDROID 開發環境與工具

- **Android Studio**是一個Android平臺開發程式的整合式開發環境 (IDE)。最初在2013年5月16日在Google 開發者大會上發佈，可供開發者免費使用。 <https://androidstudio.googleblog.com/>
  - 穩定版本: 2024.2.1 ( Patch 2 - 2024年10月 )
  - 預覽版本: 2024.2.2 ( Canary 9 - 2024年10月)
- 系統需求如下

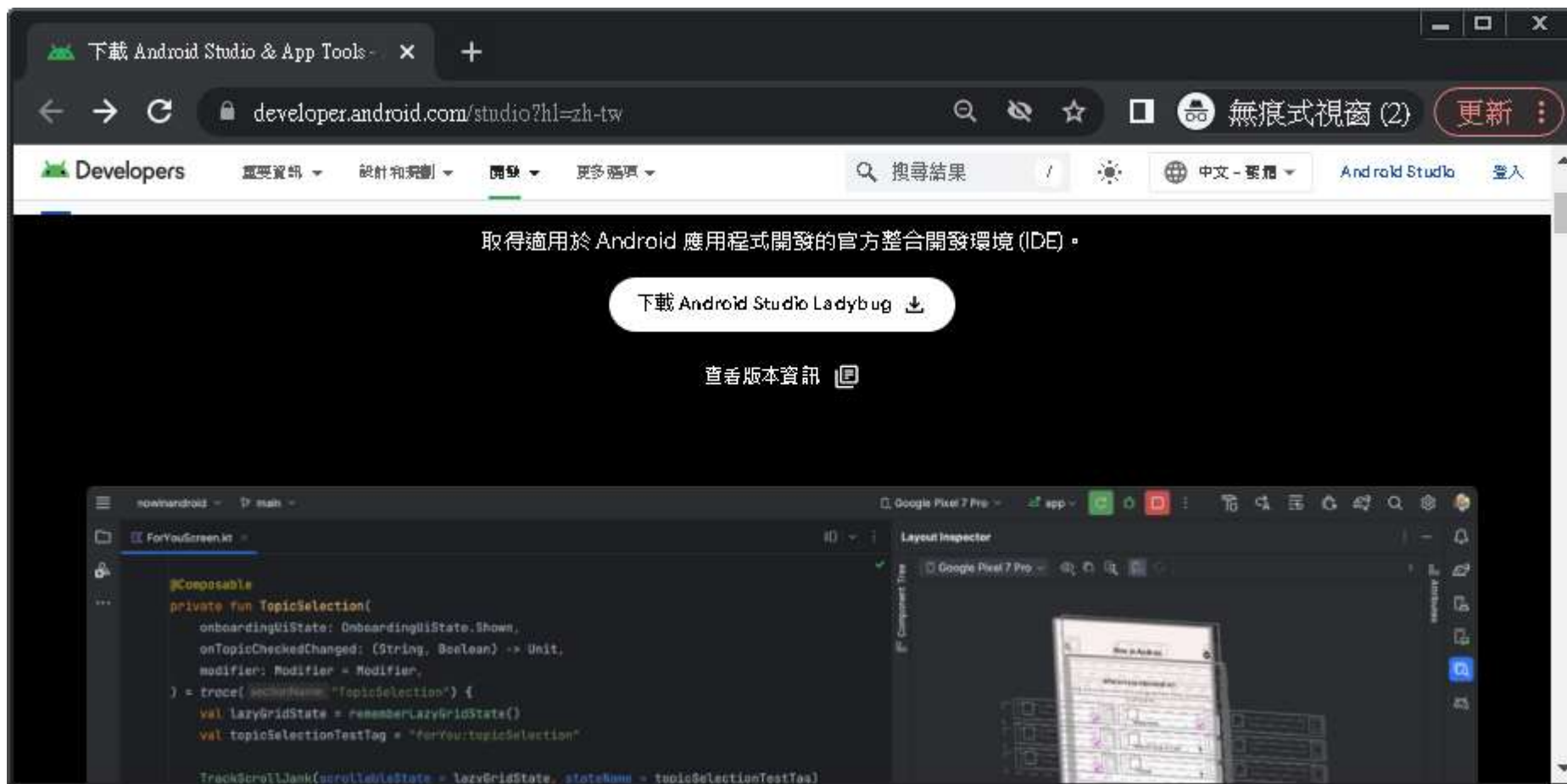
	Windows	macOS	Linux
作業系統版本	Microsoft Windows 11/10/8	OS X 10.14或更高版本	GNOME或KDE桌面
記憶體	最低4GB，推薦8GB記憶體；1GB用於Android Emulator		
磁碟空間	最低2GB，推薦4GB磁碟空間；500MB磁碟空間，至少1.5GB用於AndroidSDK		
JDK版本	Java Development Kit (JDK) 17 或以下版本		
螢幕解析度	最低1280×800螢幕解析度		





# ANDROID 開發環境與工具

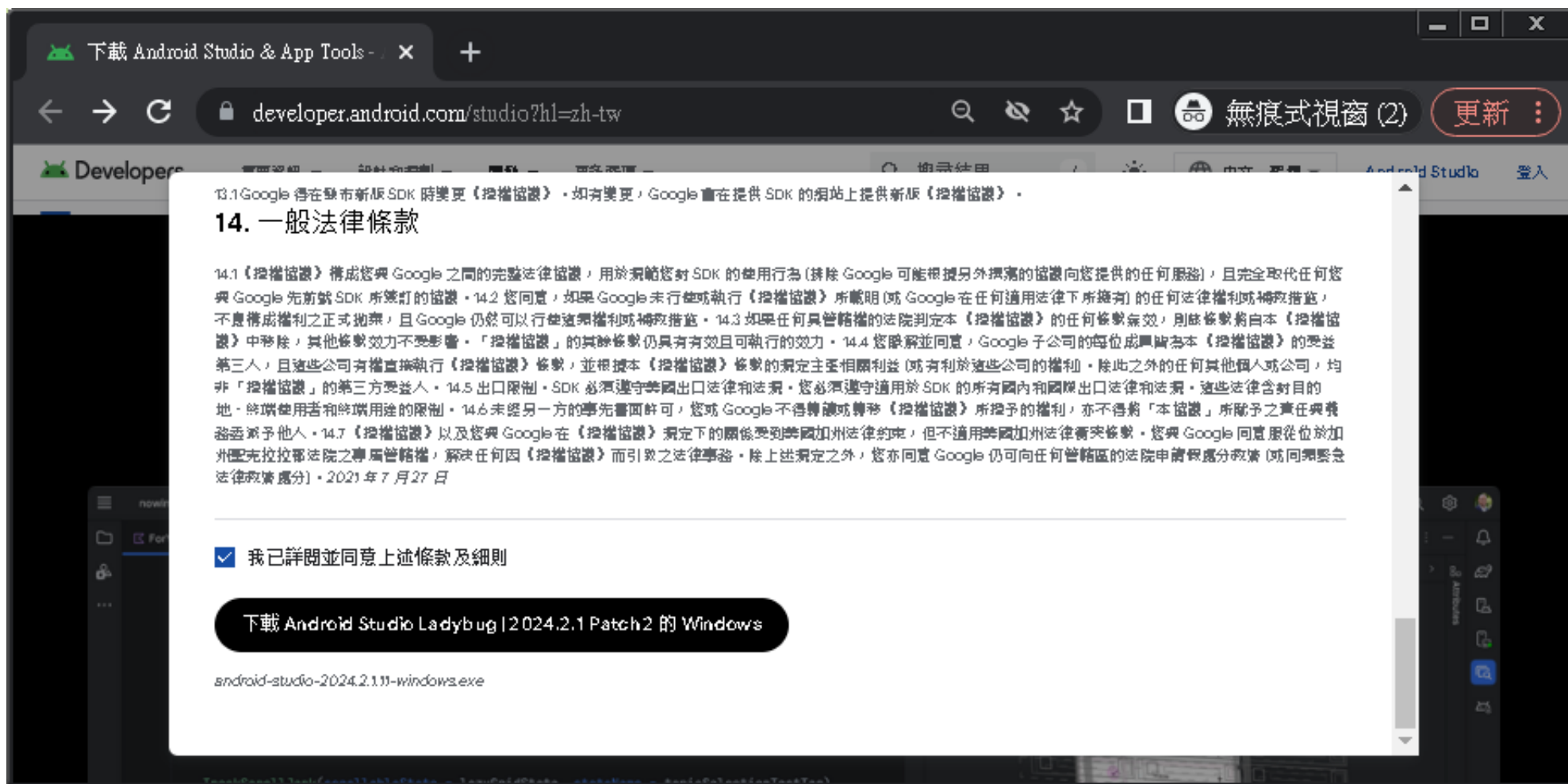
- 下載與安裝 Android Studio
- 網址: [developer.android.com/studio/](https://developer.android.com/studio/)





# ANDROID 開發環境與工具

- 同意Android Studio相關授權





# ANDROID 開發環境與工具

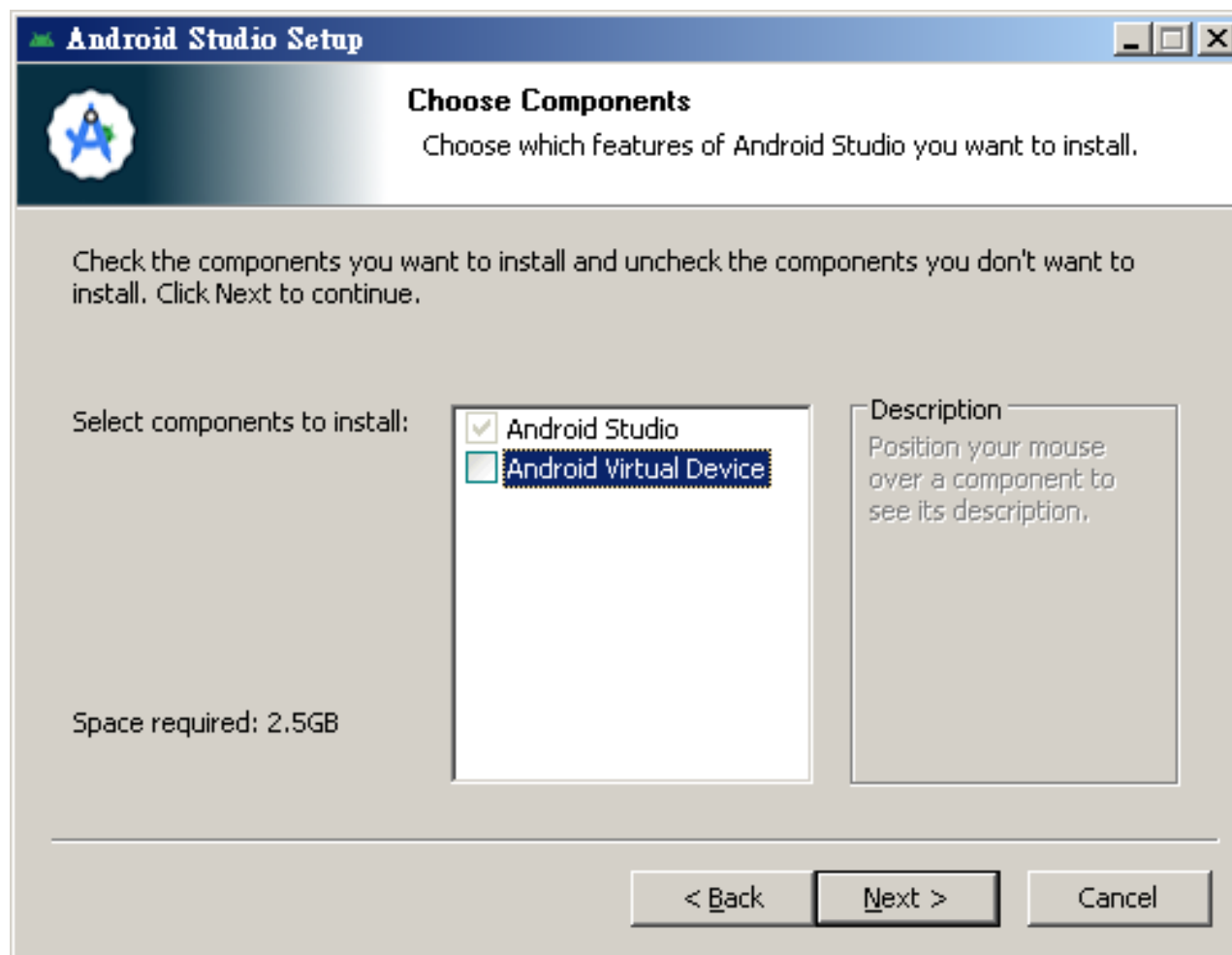
- 啟動 Android Studio 安裝程式





# ANDROID 開發環境與工具

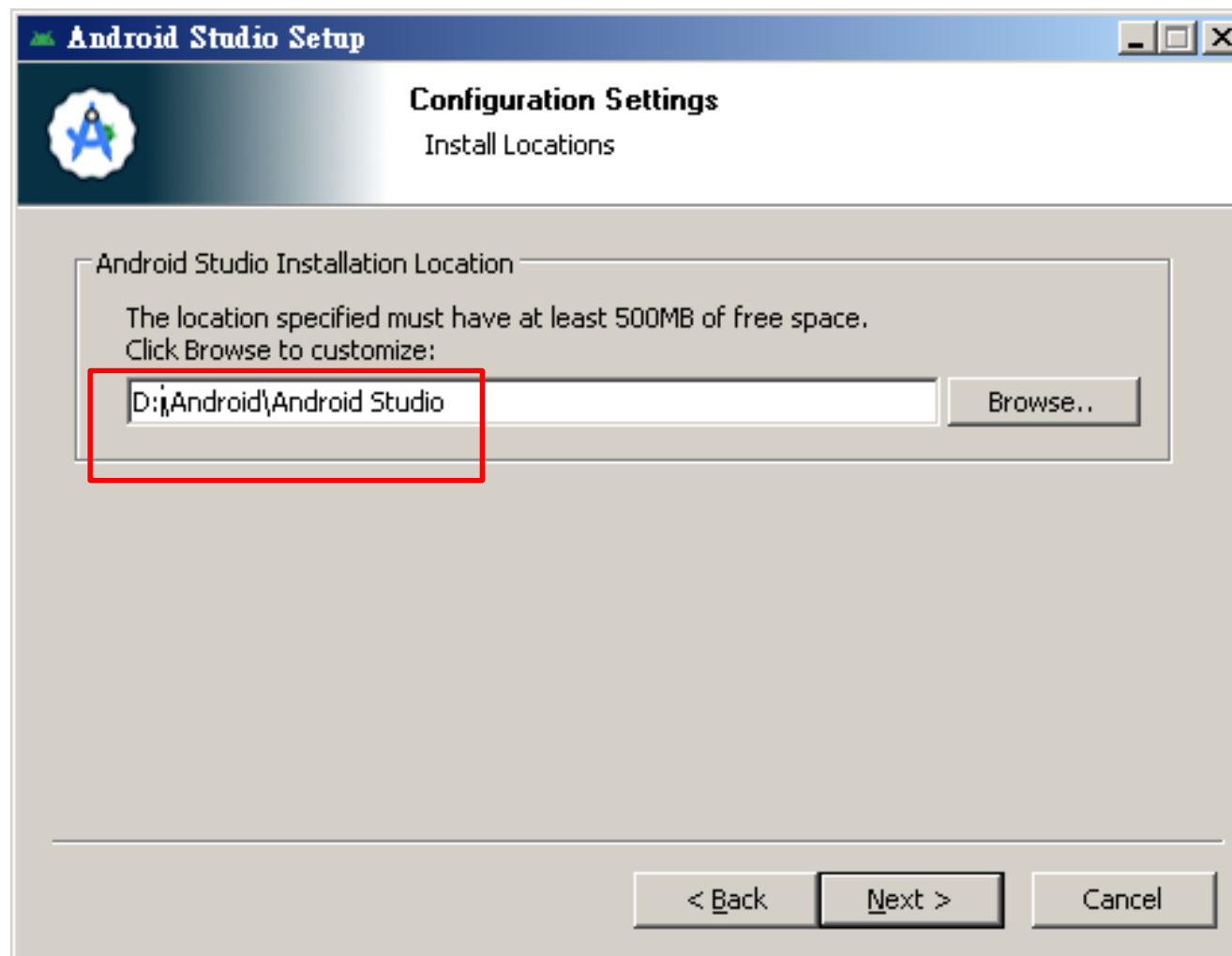
- 安裝 Android Studio，選擇所需元件





# ANDROID 開發環境與工具

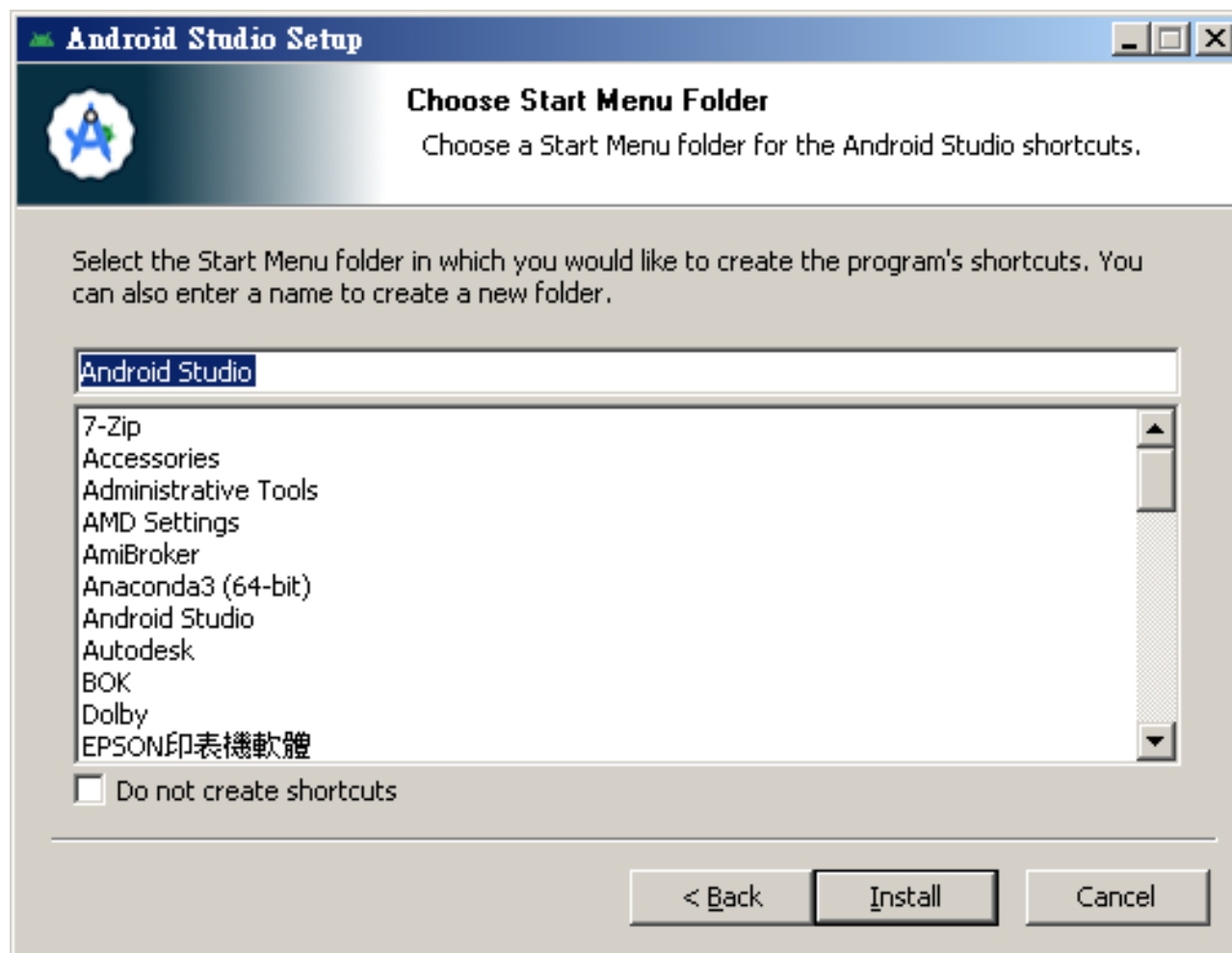
- 安裝 Android Studio，設定安裝環境目錄





# ANDROID 開發環境與工具

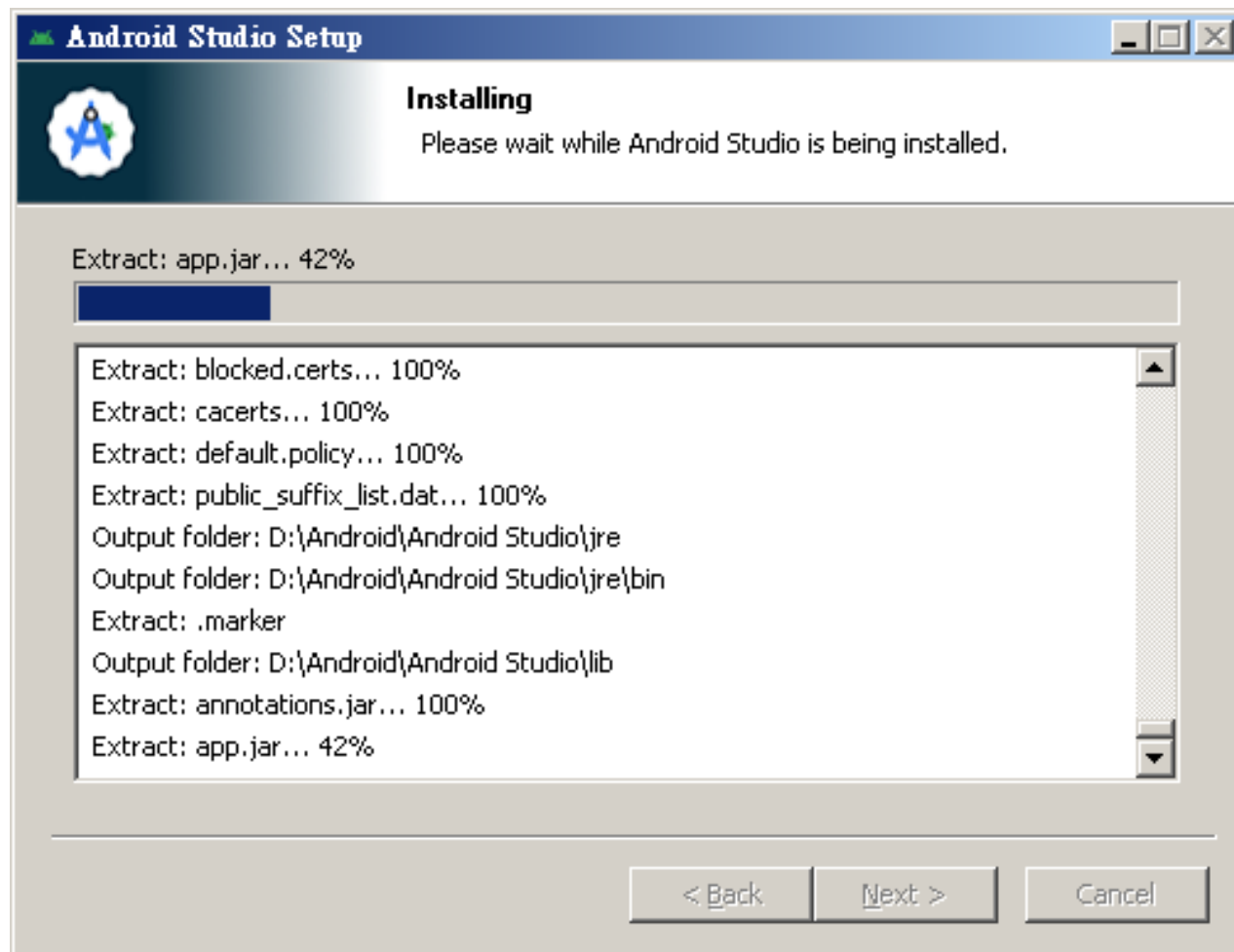
- 安裝 Android Studio，選擇是否安裝捷徑





# ANDROID 開發環境與工具

- 按下 Next，開始安裝

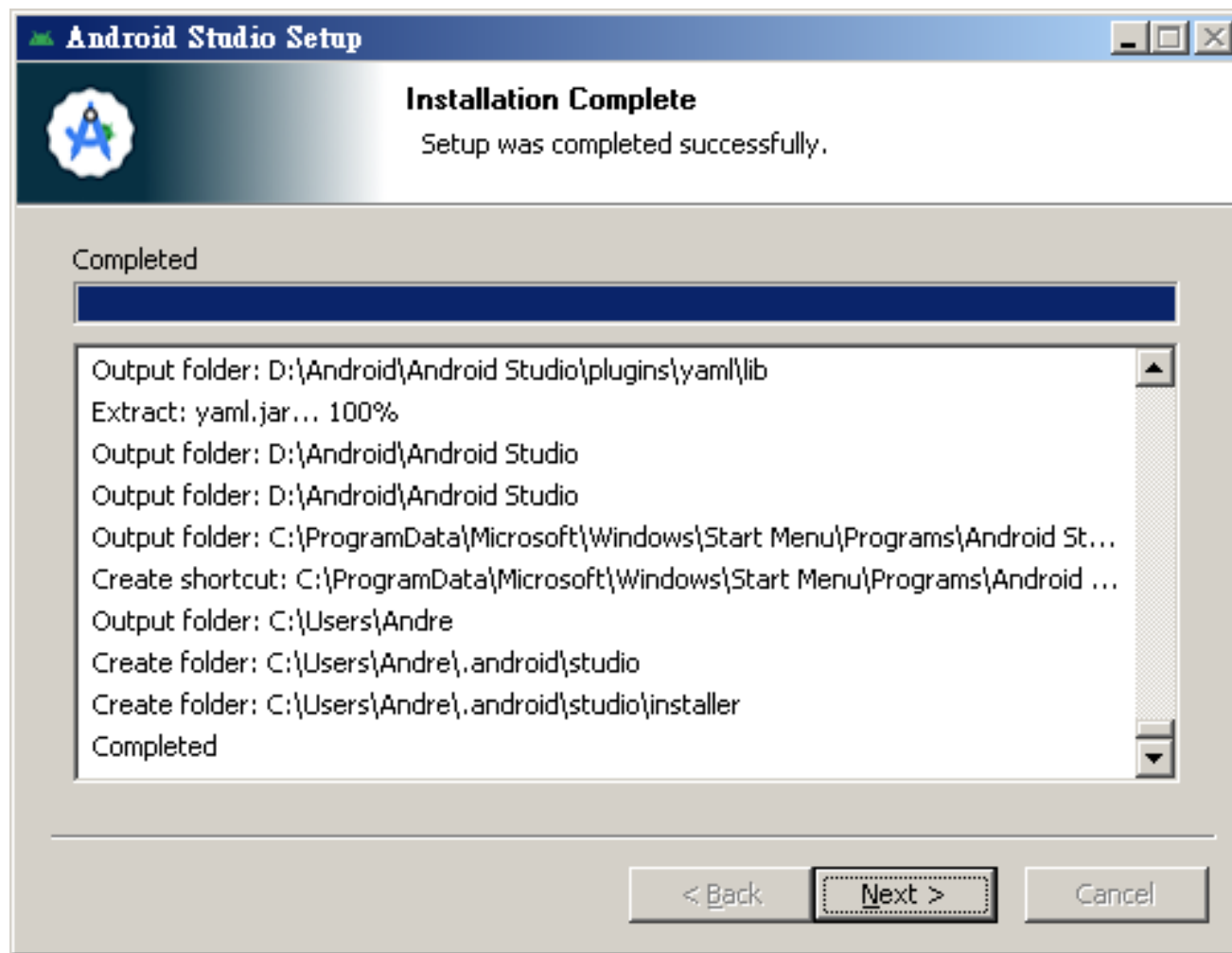






# ANDROID 開發環境與工具

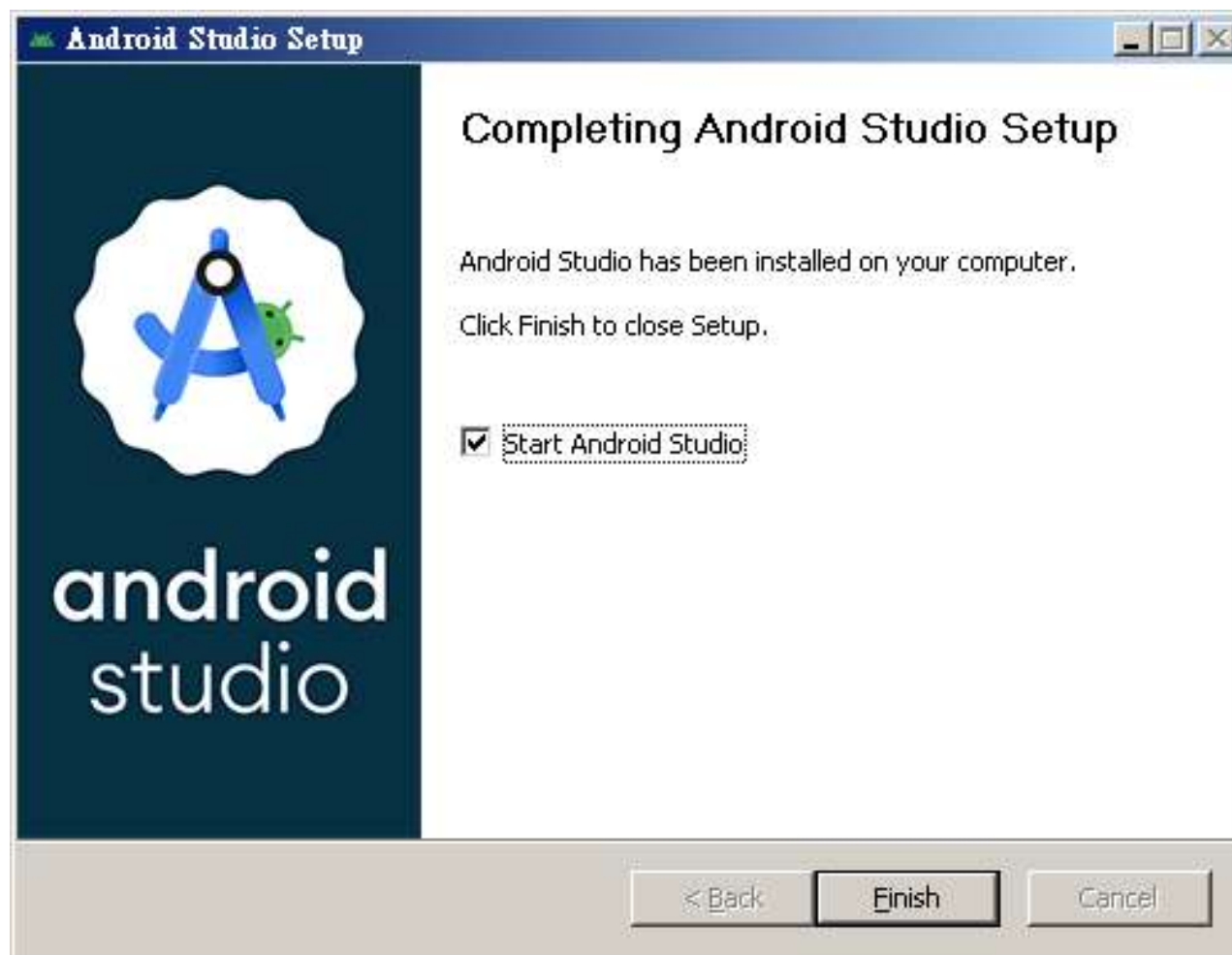
- 按下 Next，完成安裝





# ANDROID 開發環境與工具

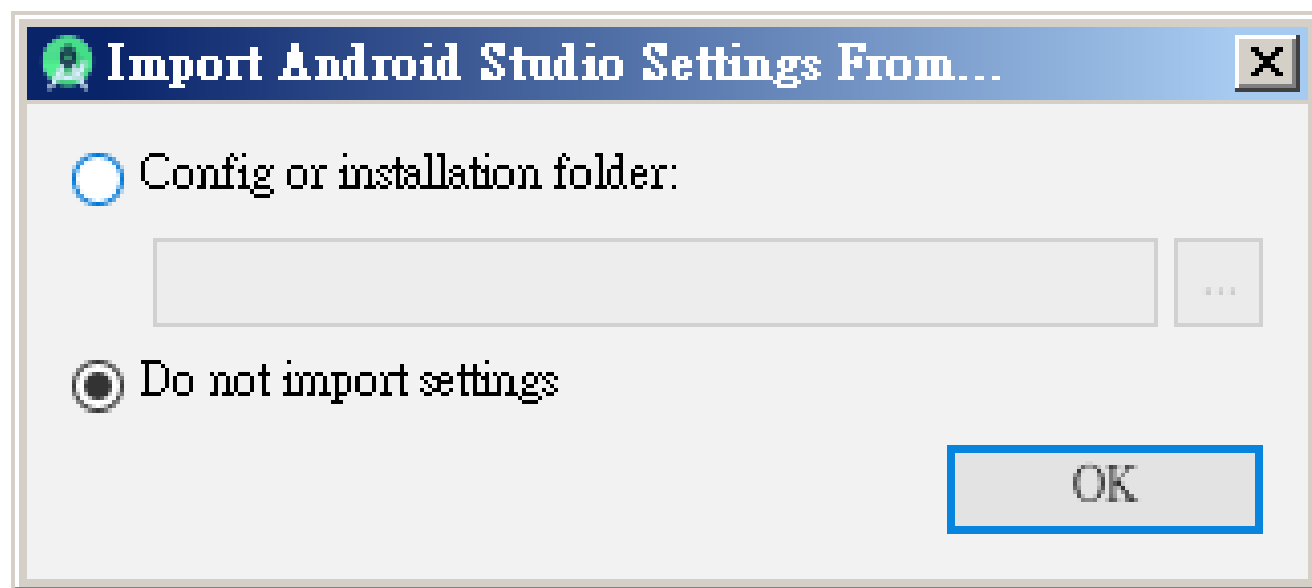
- 完成安裝 Android Studio





# ANDROID 開發環境與工具

- 首次啟動Android Studio與相關設定





# ANDROID 開發環境與工具

- 首次啟動Android Studio與相關設定



## Help improve Android Studio

Allow Google to collect usage data for Android Studio and its related tools, such as how you use features and resource usage along with software identifiers such as package name and class names and plugin configuration. This data helps improve Android Studio and is collected in accordance with [Google's Privacy Policy](#). Anonymous and aggregated usage data may be shared with Google's partners to improve Android Studio.

You can always change this behavior in Settings | Appearance & Behavior | System Settings | Data Sharing.

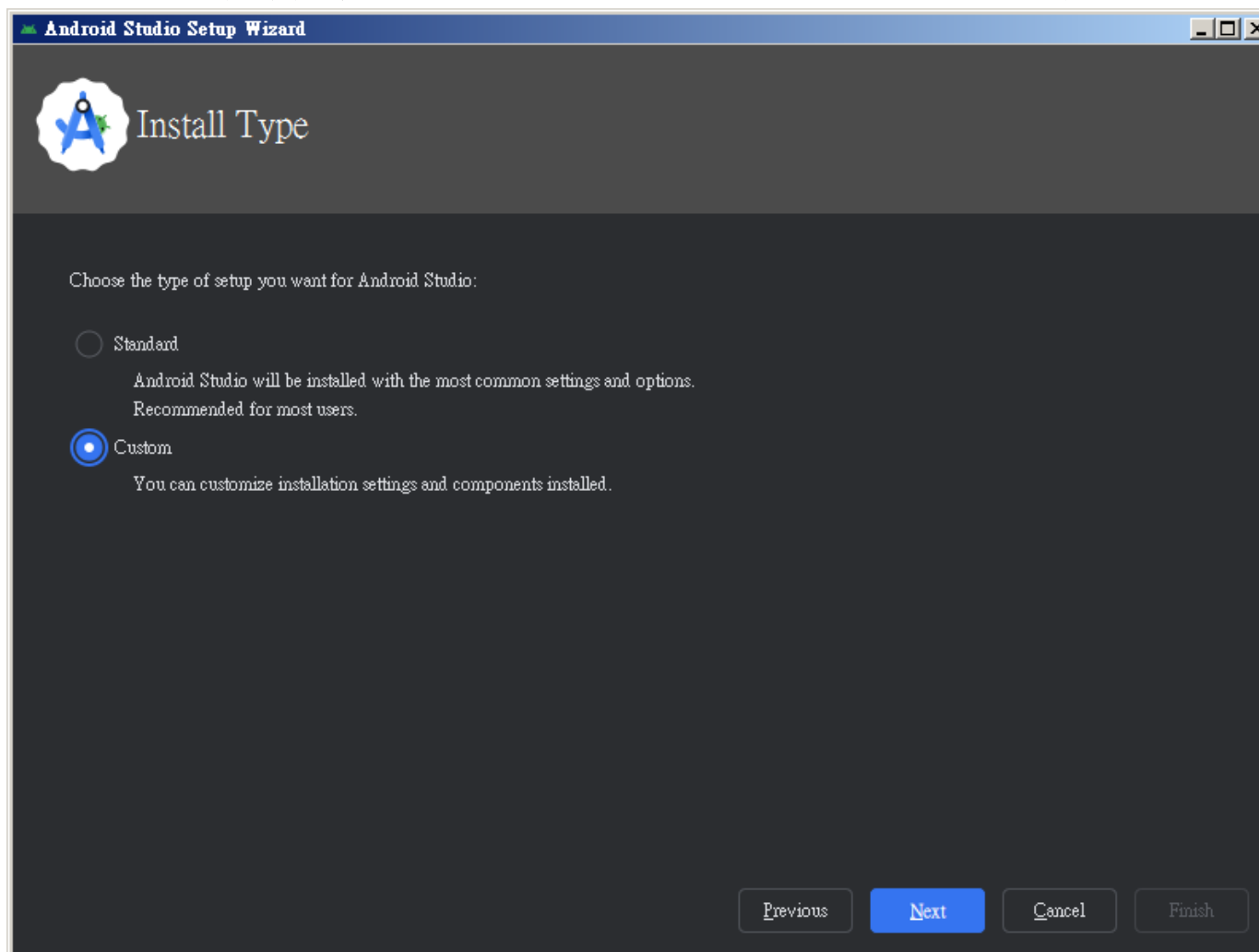
Don't send

Send usage statistics to Google



# ANDROID 開發環境與工具

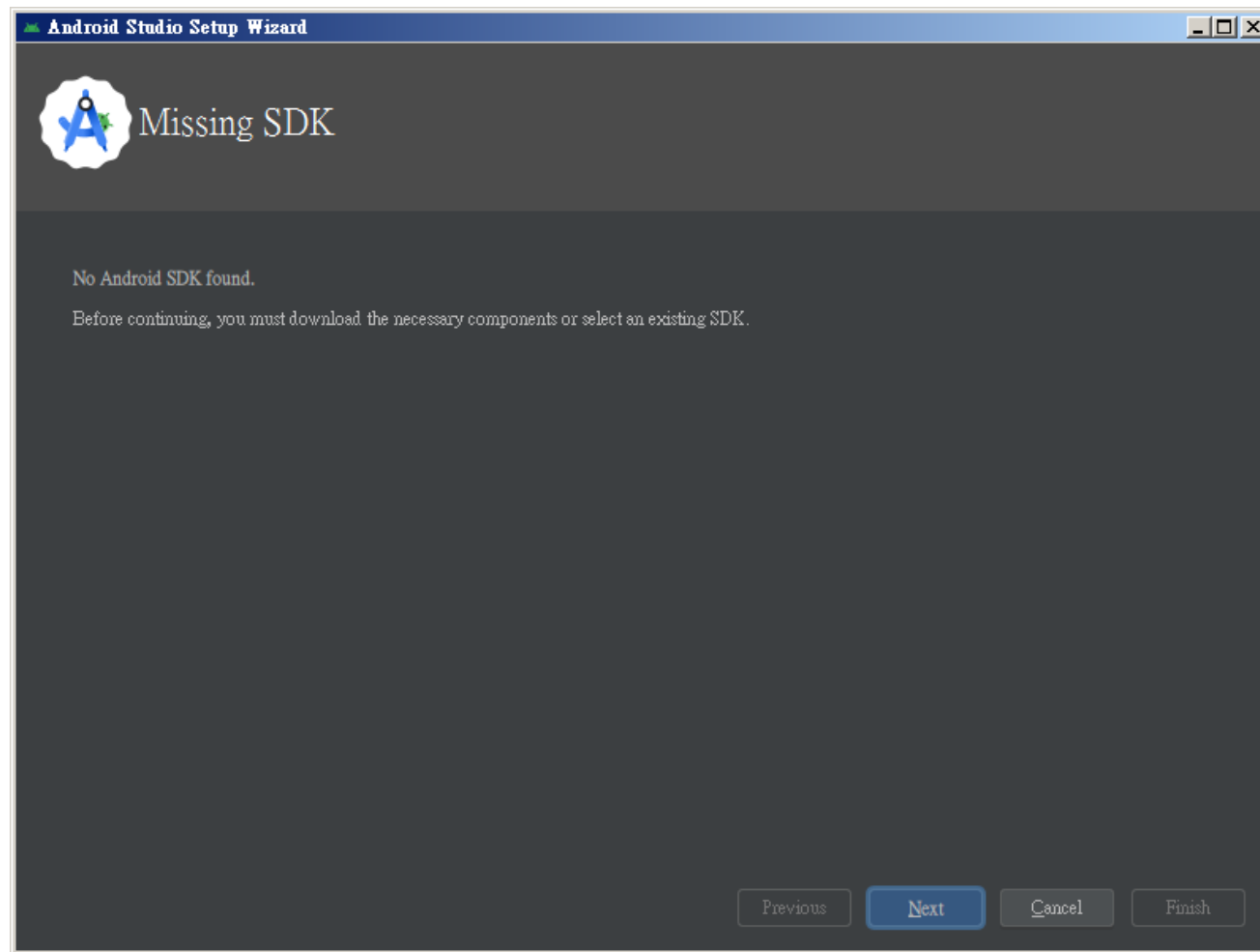
- 首次啟動Android Studio與相關設定





# ANDROID 開發環境與工具

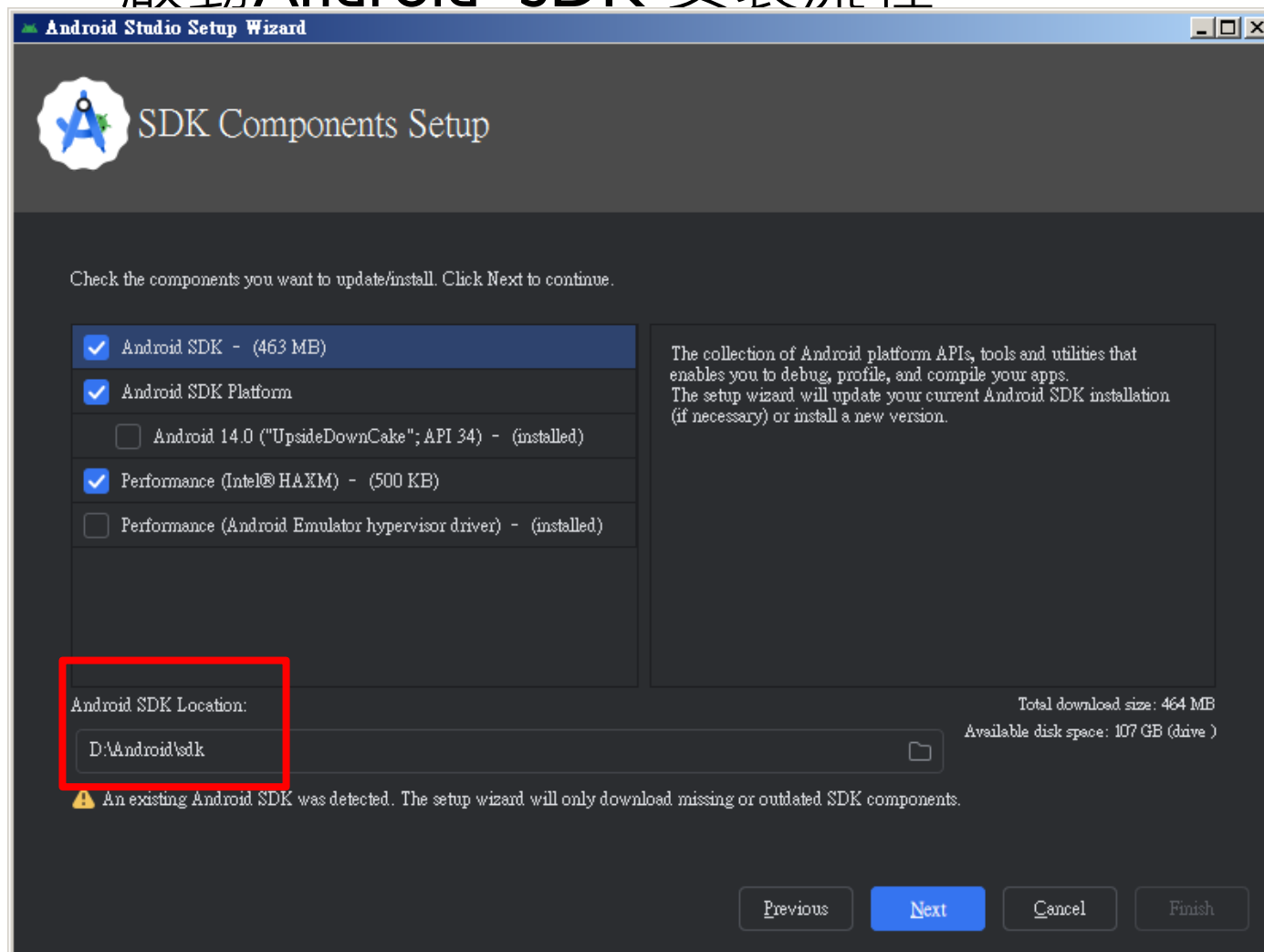
- 下載並安裝 Android SDK ，按下 Next





# ANDROID 開發環境與工具

## ● 啟動Android SDK 安裝流程

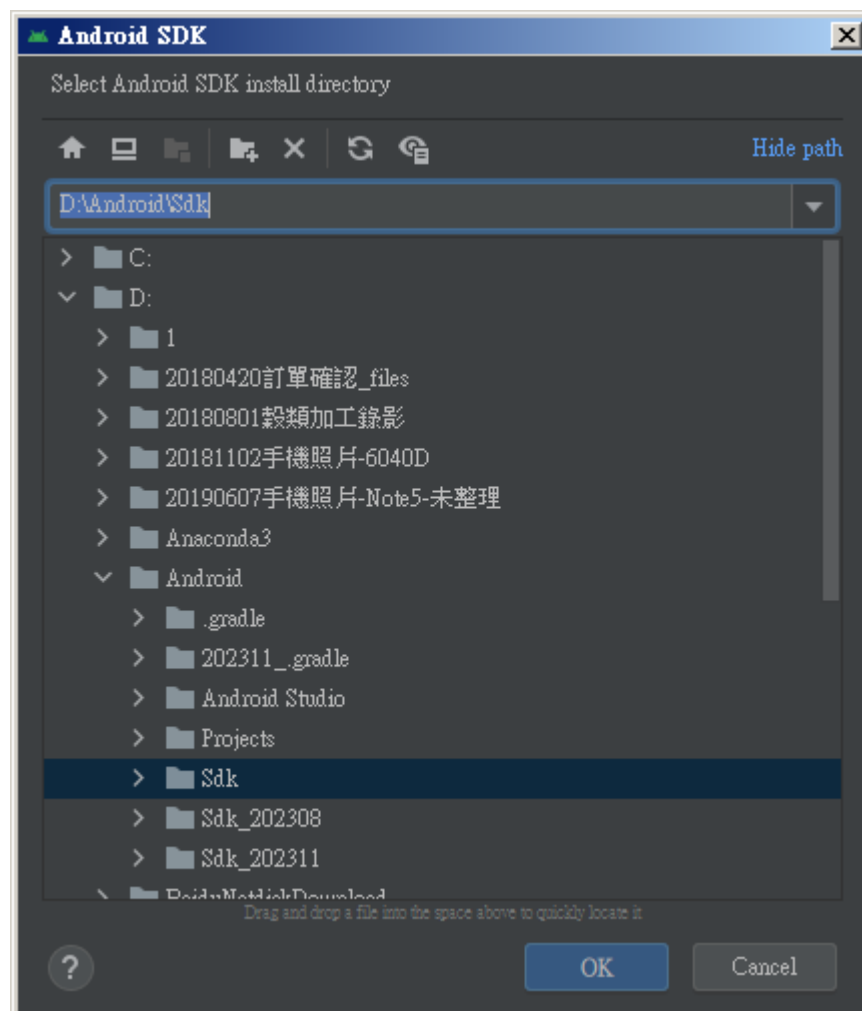






# ANDROID 開發環境與工具

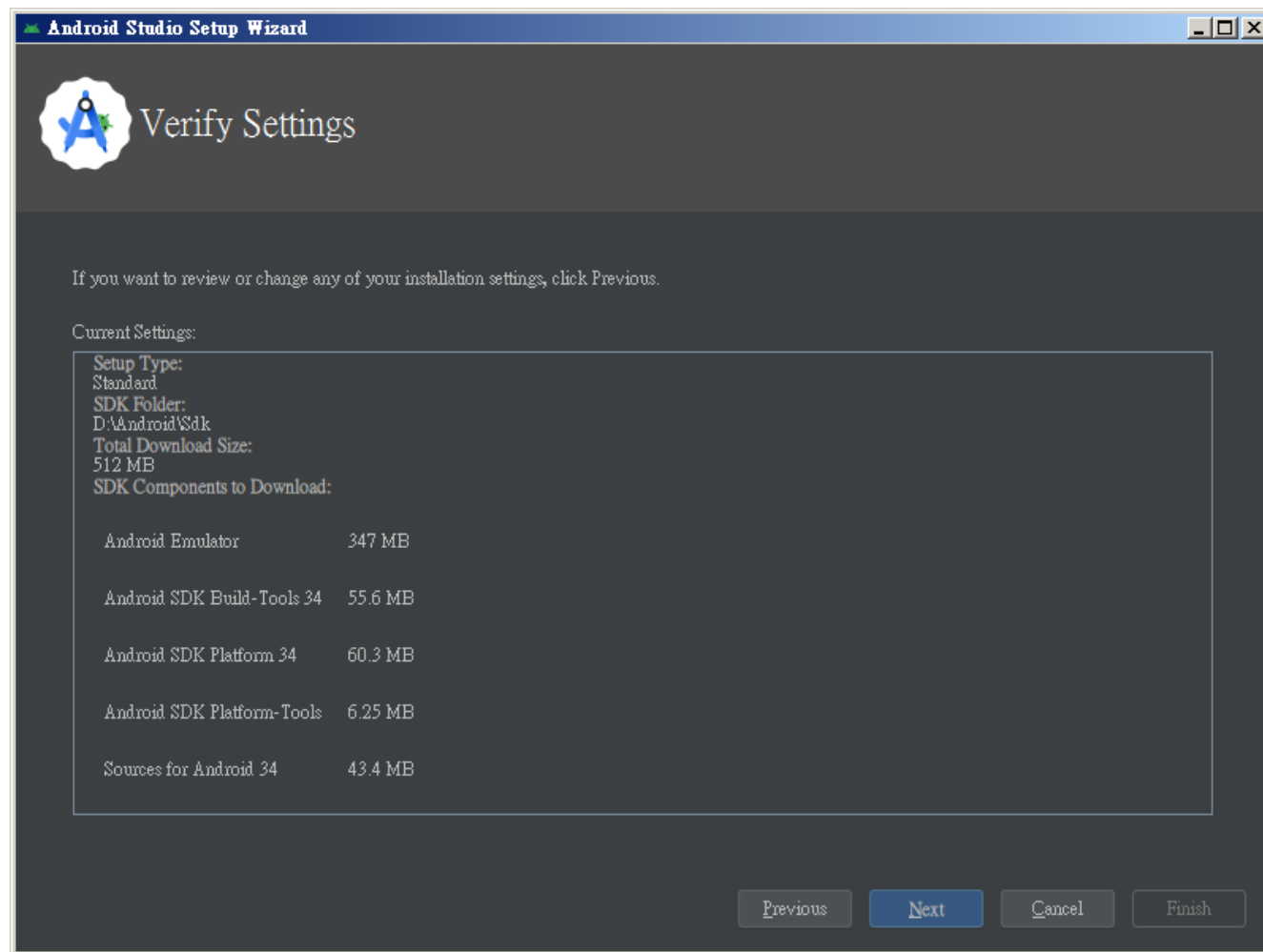
- 變更 SDK 存放位置，整合 Android SDK 安裝目錄





# ANDROID 開發環境與工具

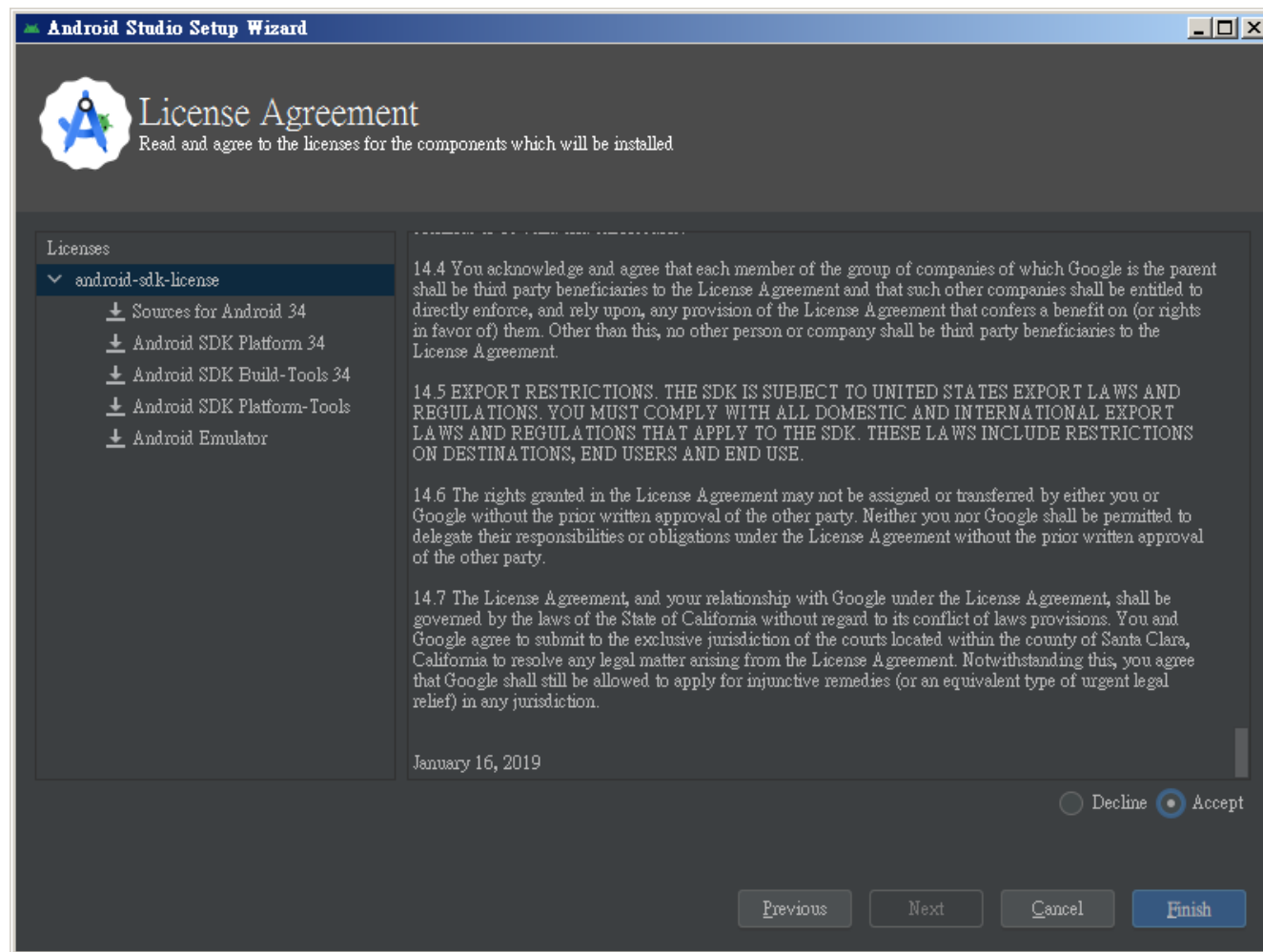
- 按下 Next，開始 Android SDK 初次安裝





# ANDROID 開發環境與工具

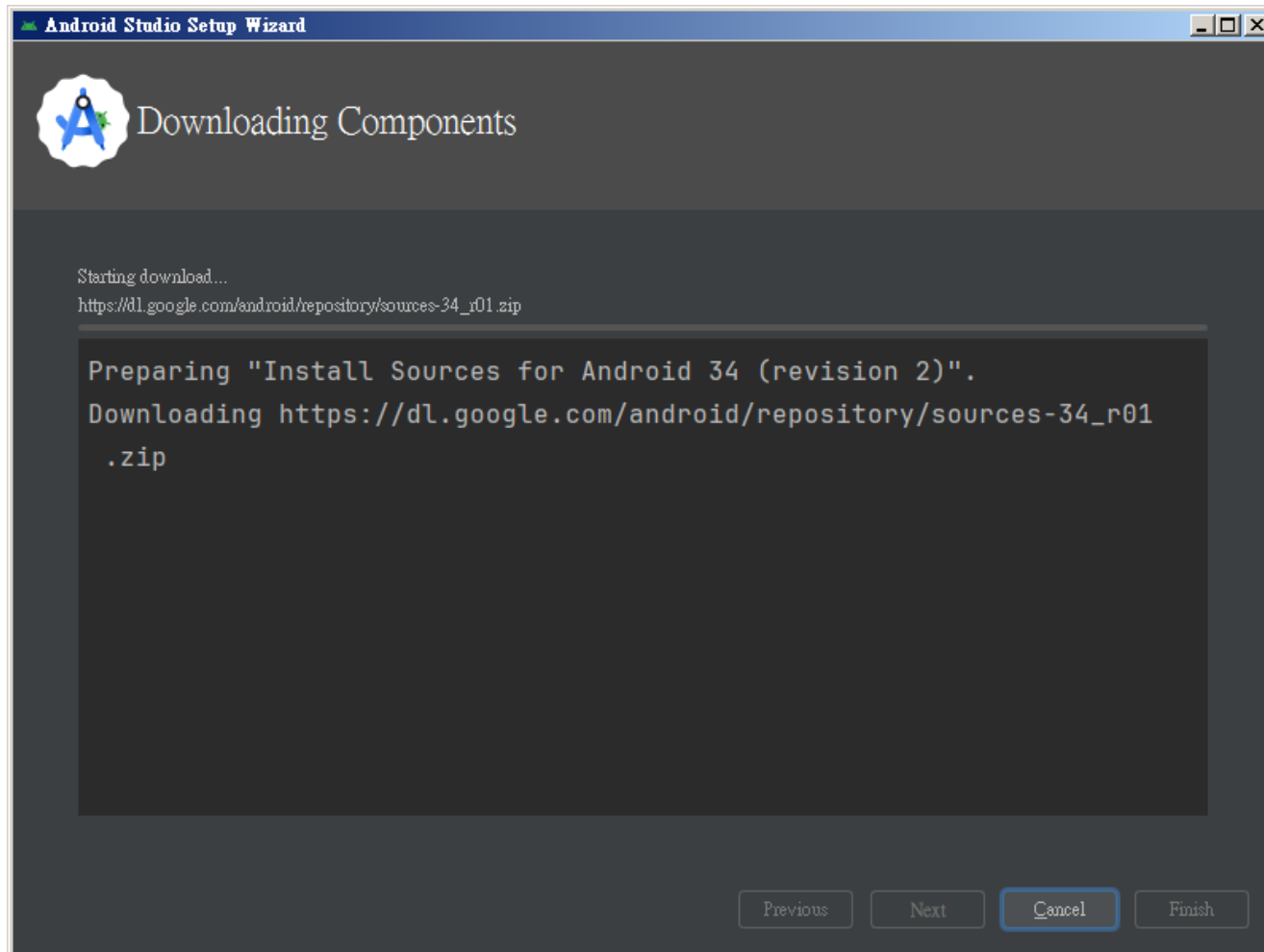
- 授權同意畫面





# ANDROID 開發環境與工具

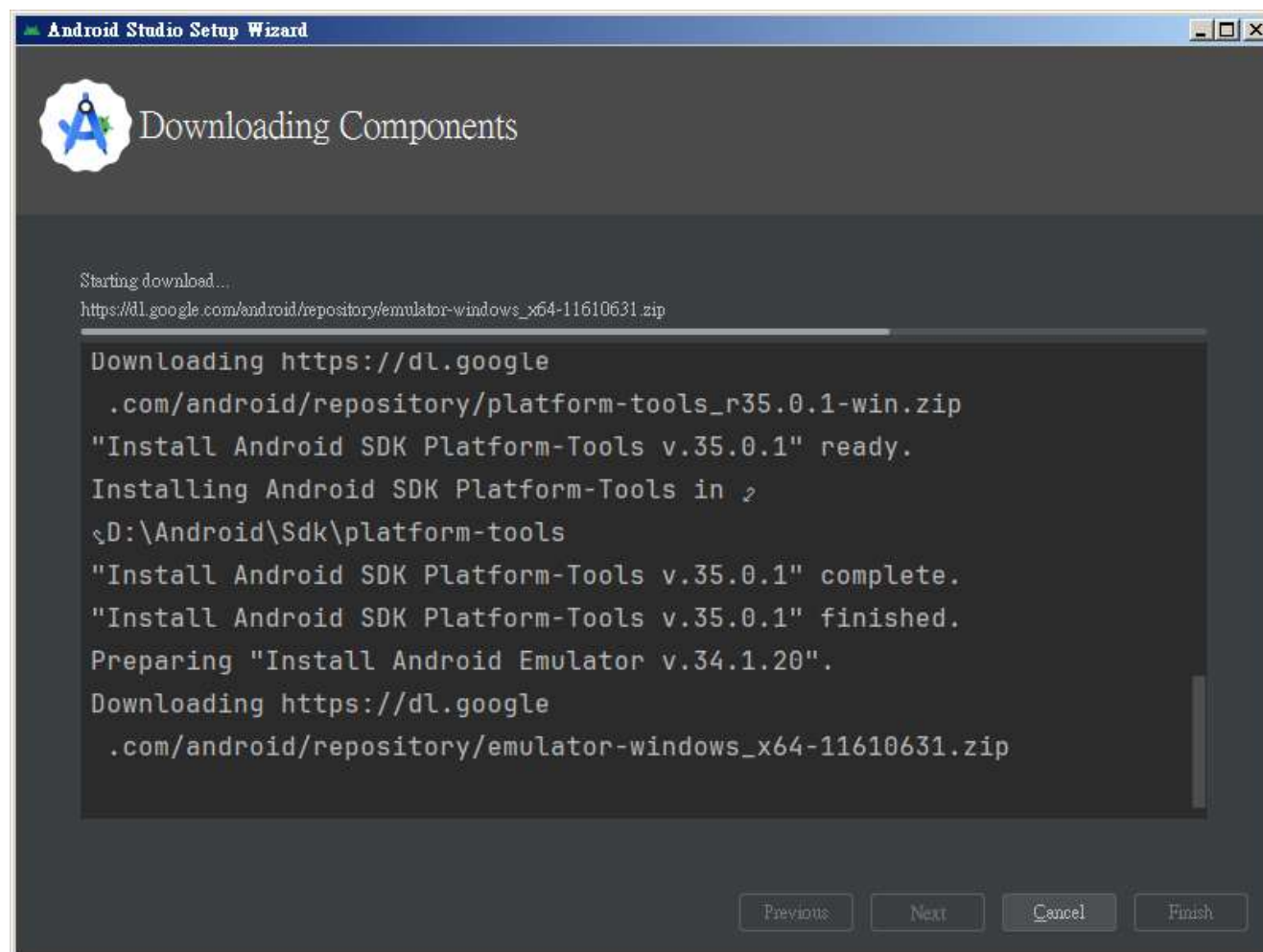
- 開始下載套件檔案，依據電腦及網路速度花費時間不同





# ANDROID 開發環境與工具

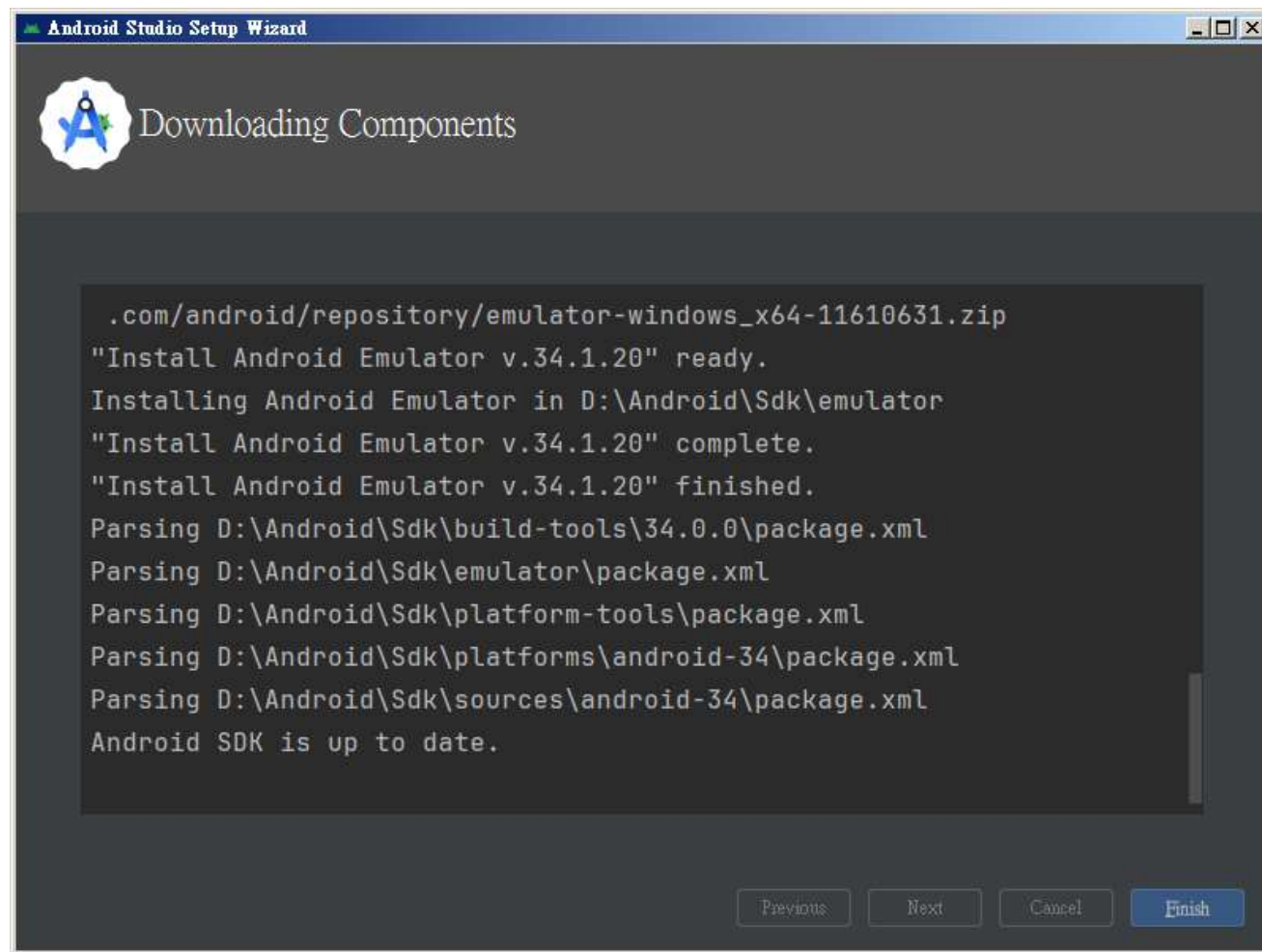
- 安裝中...





# ANDROID 開發環境與工具

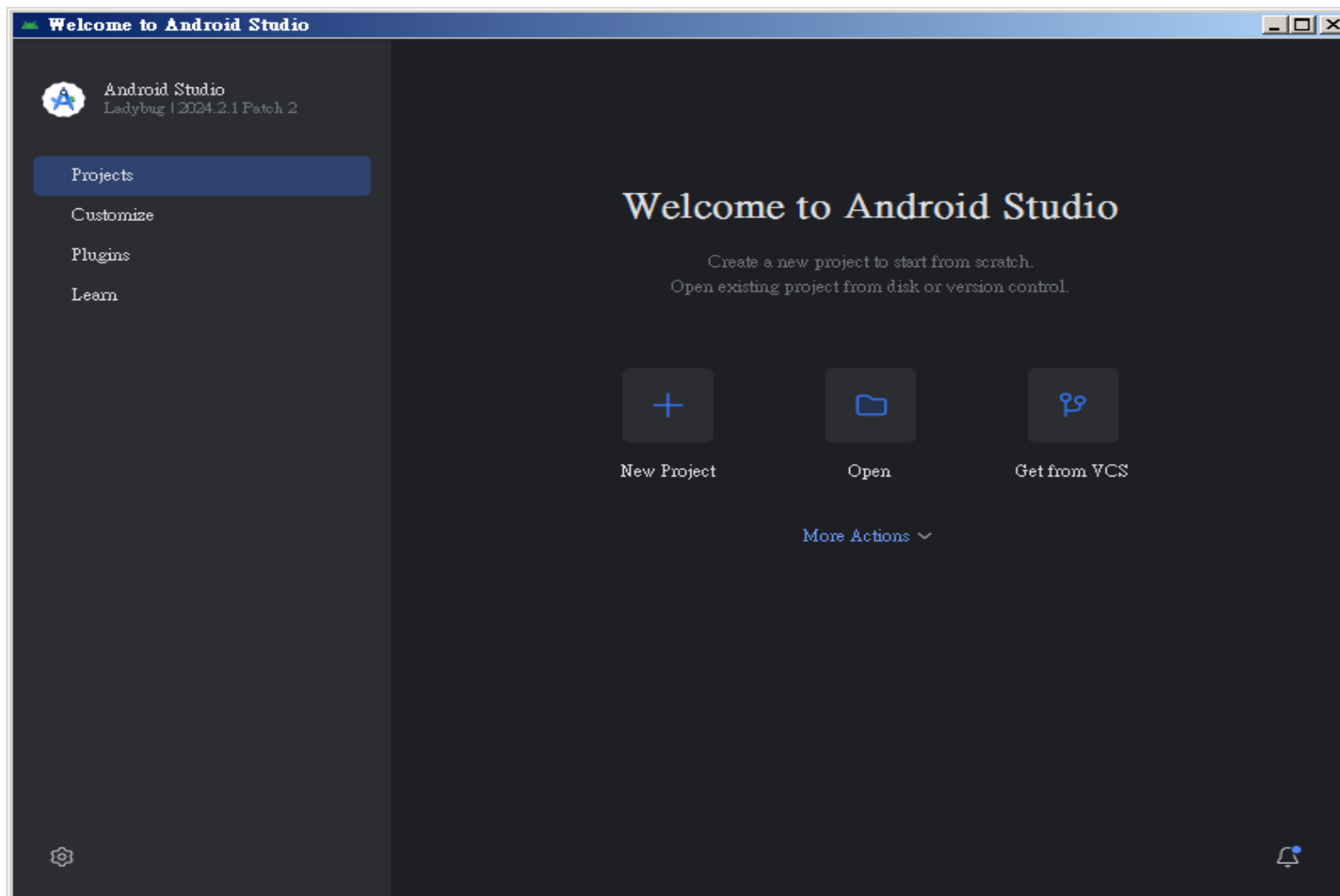
- 下載並安裝 Android SDK 完成，按 Finish 結束





# ANDROID 開發環境與工具

- Android Studio— 啟動畫面

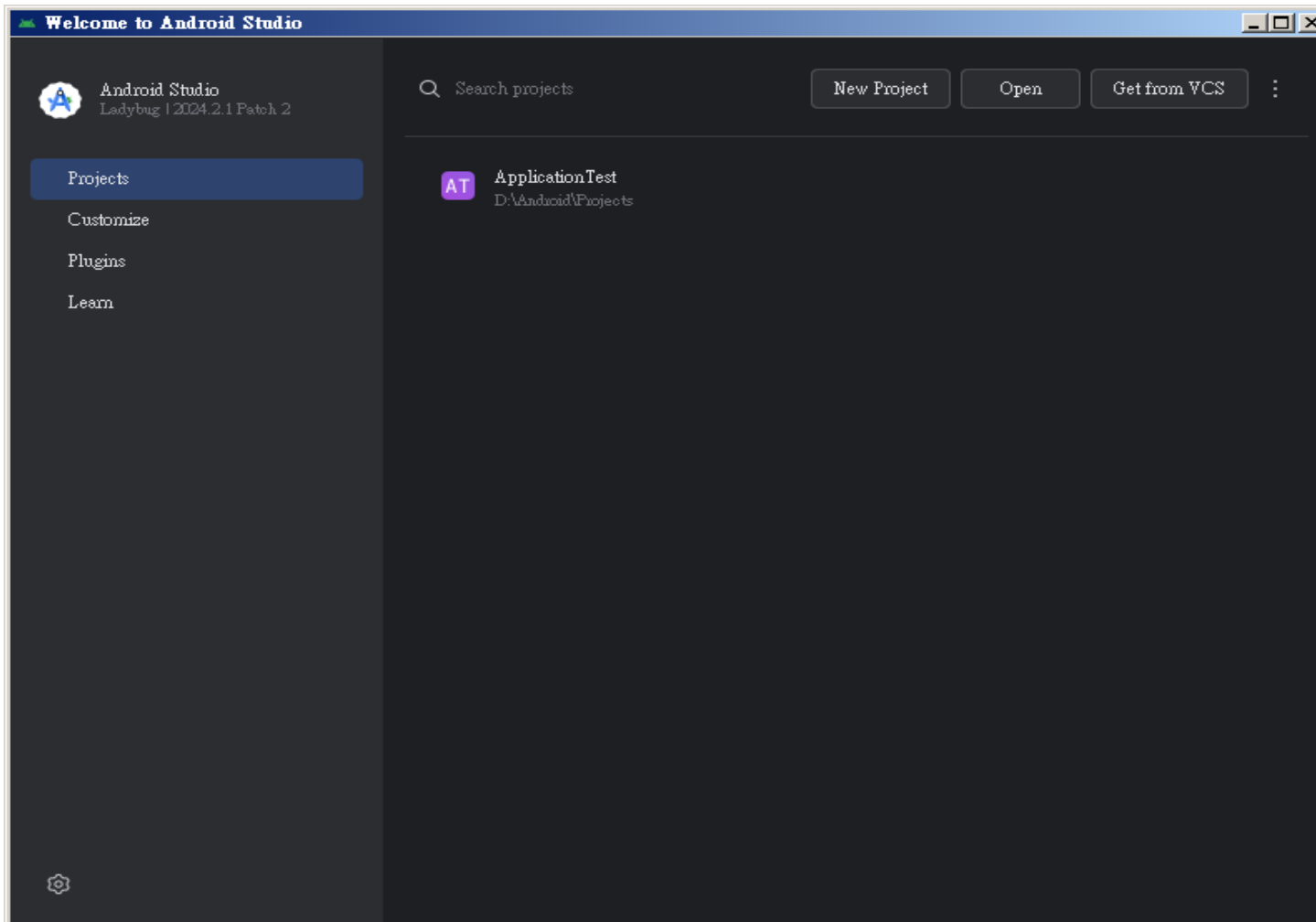






# ANDROID 開發環境與工具

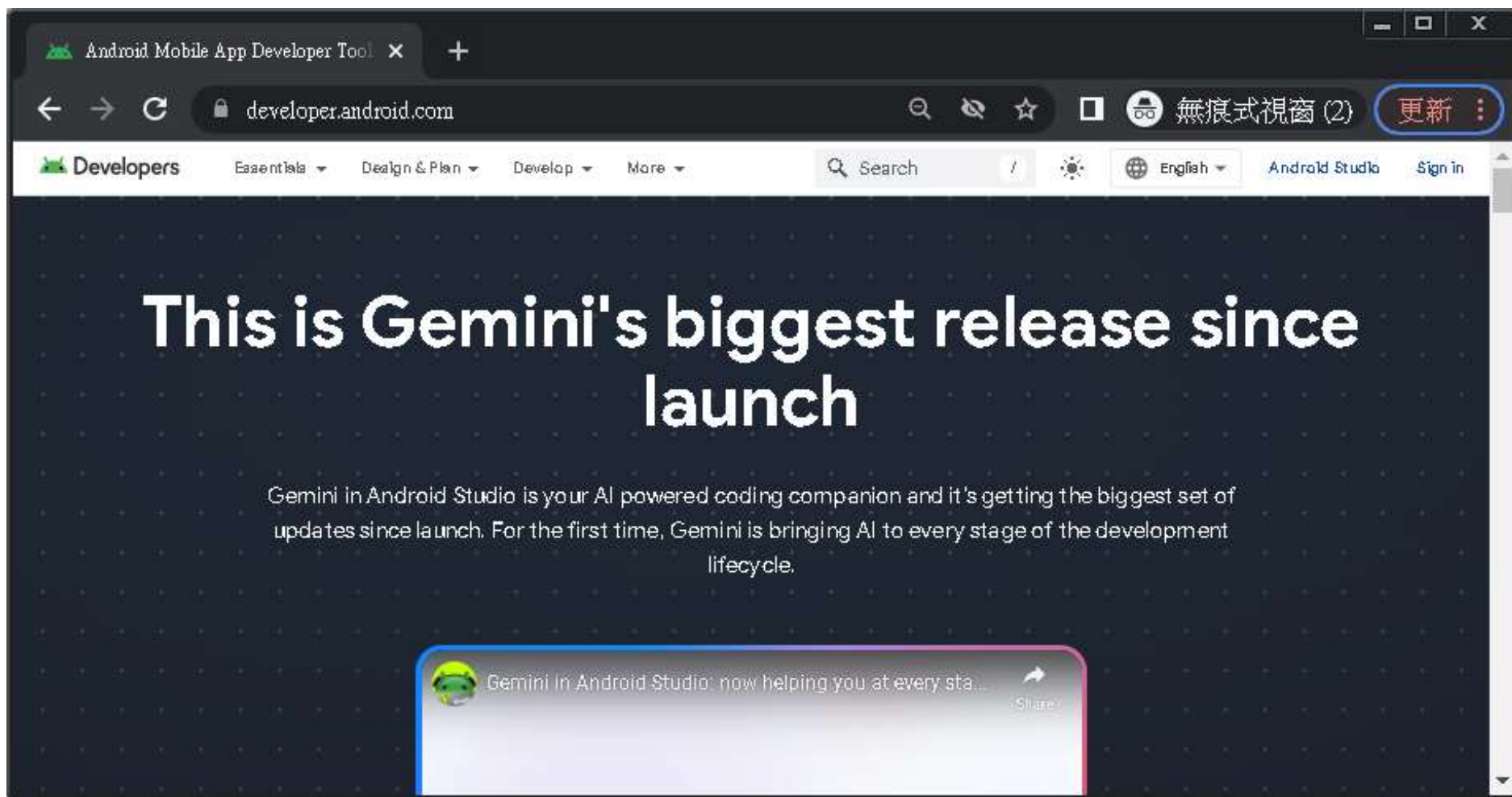
- Android Studio— 啟動畫面





# ANDROID 開發者網站

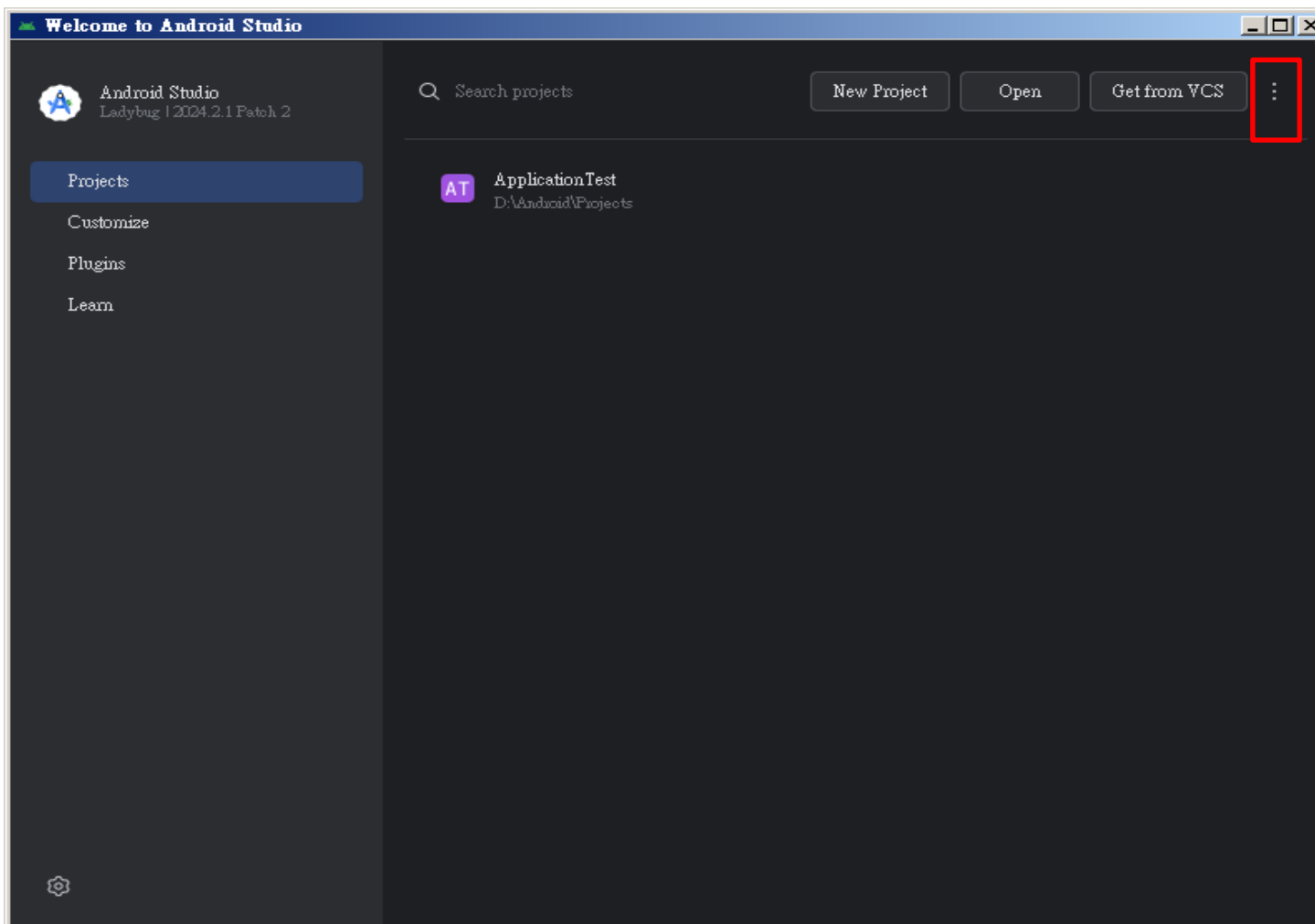
- 網址: [developer.android.com](https://developer.android.com)





# ANDROID 開發環境與工具

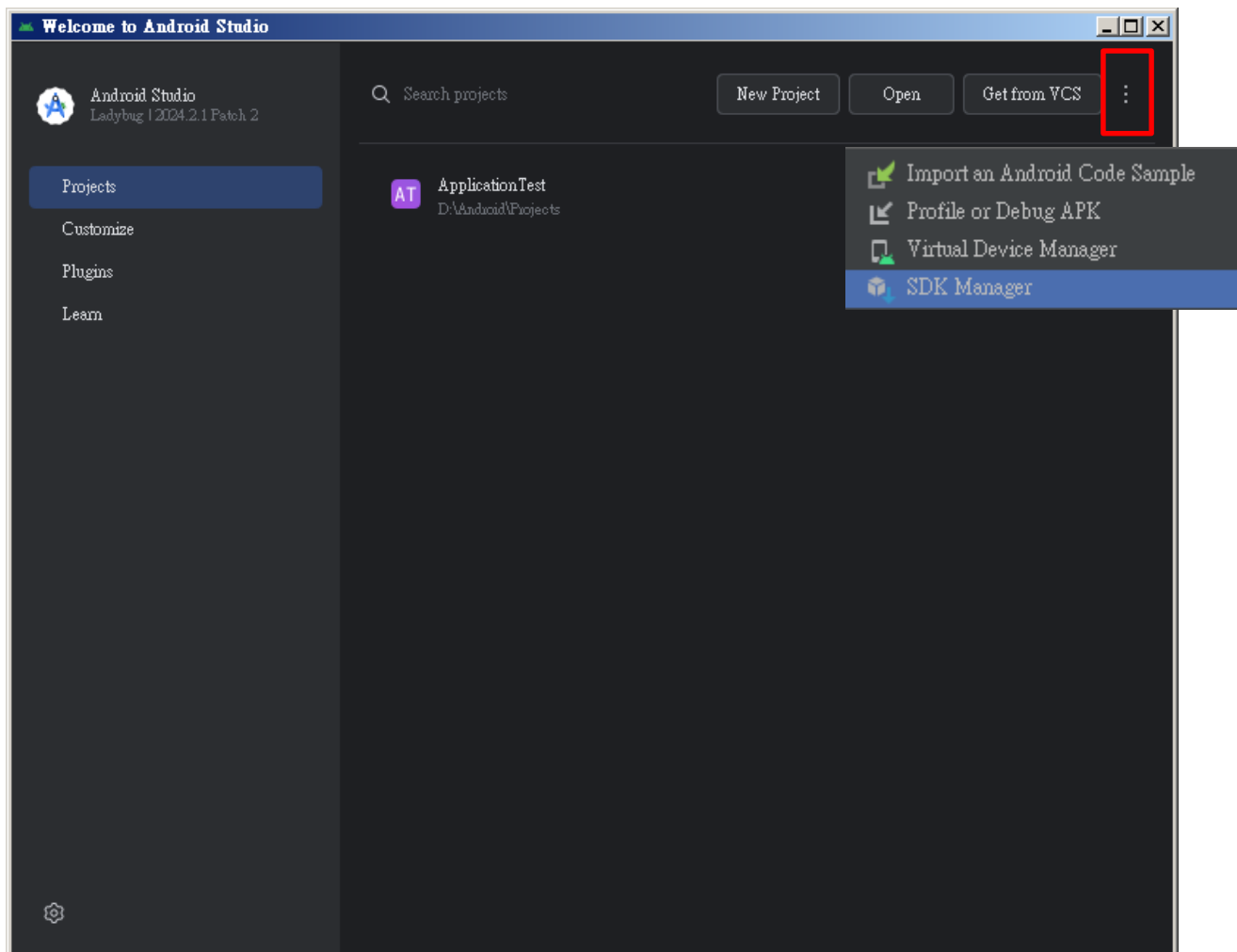
- 建立與使用開發套件管理工具





# ANDROID 開發環境與工具

- 建立與使用開發套件管理工具





# ANDROID 開發環境設定

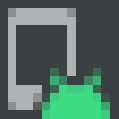
- 建立與使用開發套件管理工具



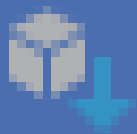
Import an Android Code Sample



Profile or Debug APK



Virtual Device Manager

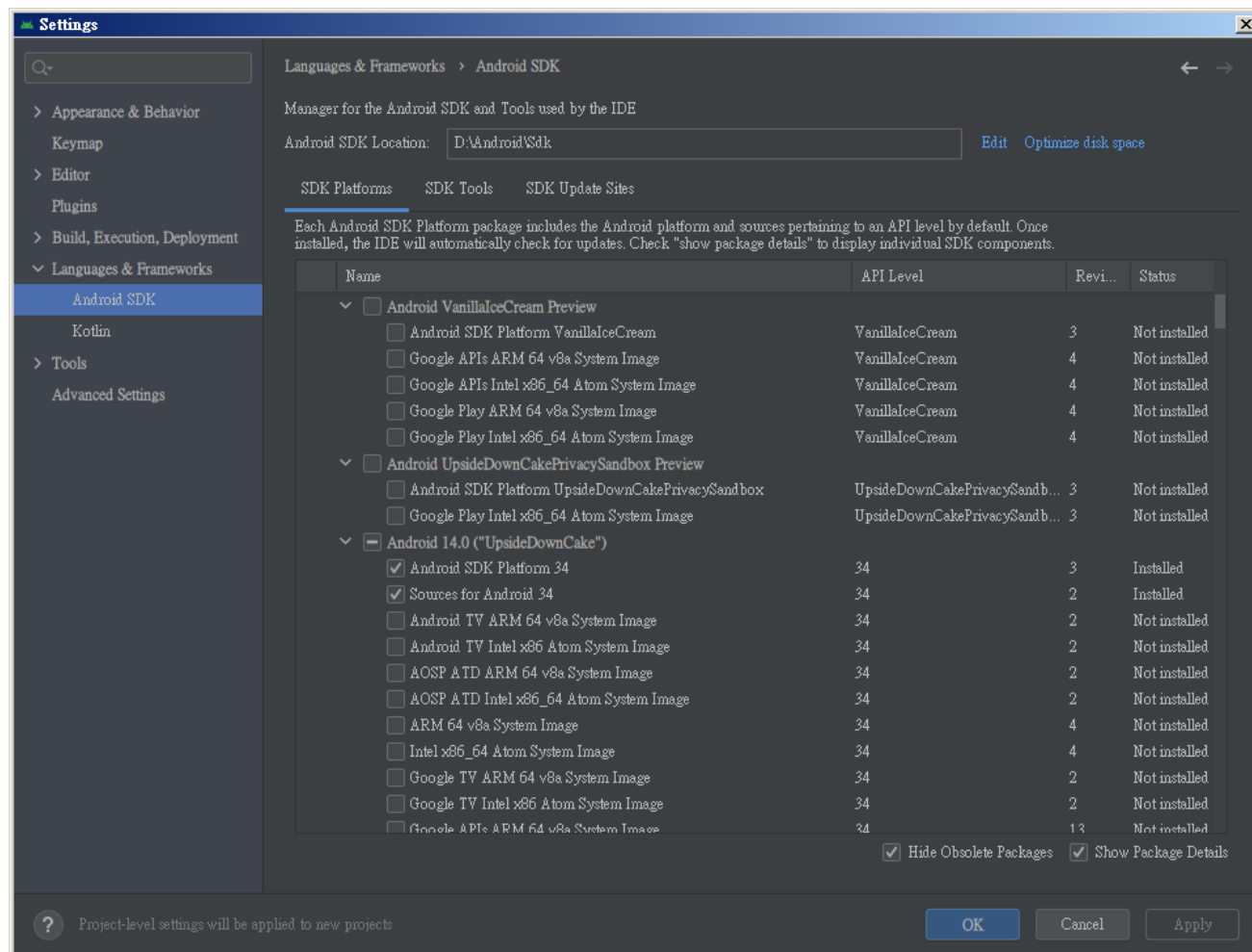


SDK Manager



# ANDROID 開發環境設定

## • SDK Manager 開發套件管理工具





# ANDROID 開發環境設定

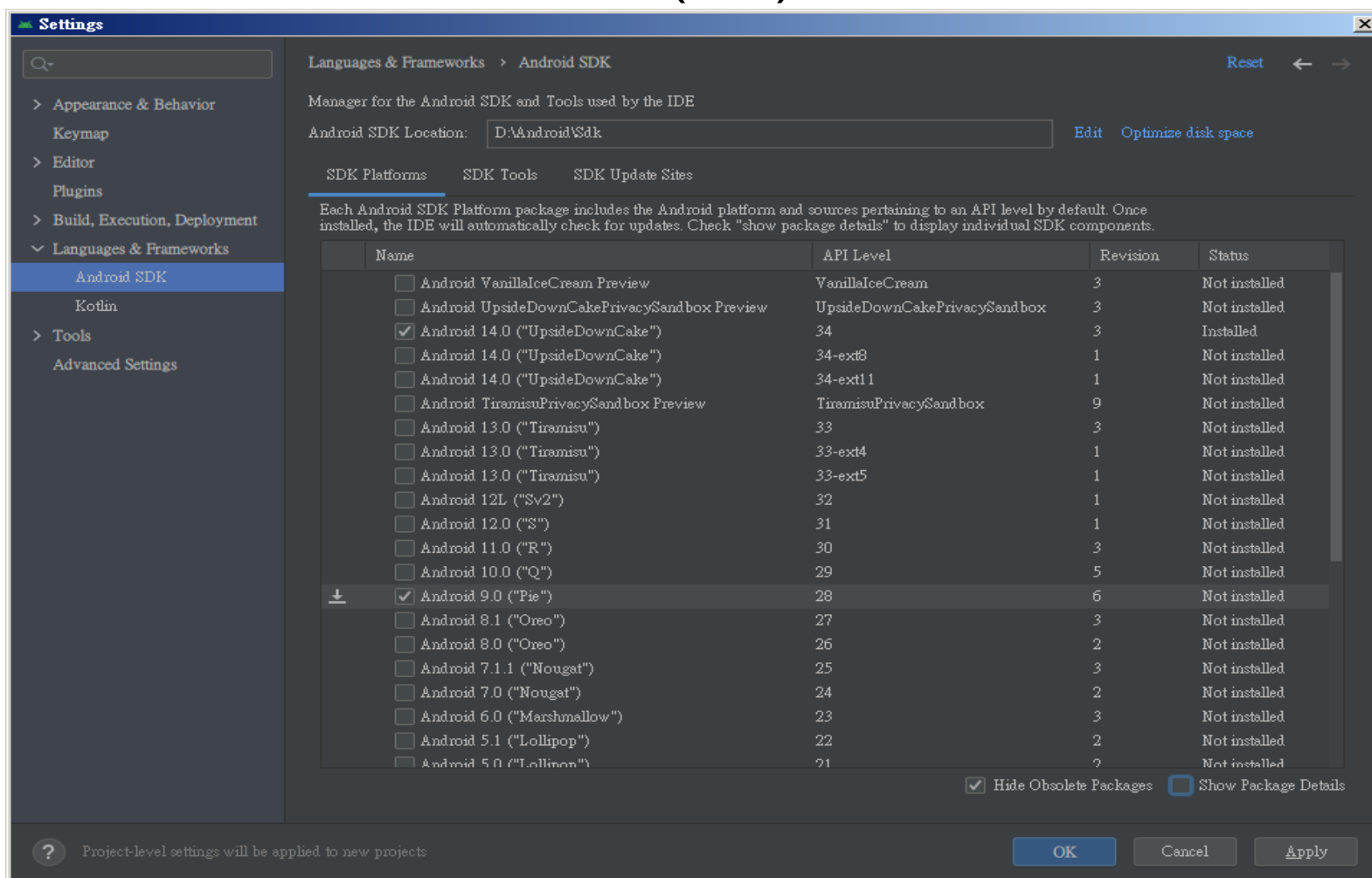
- 開發套件 **API** 版本的選用，開發環境的取捨

版本號+版本名稱	<u>API</u>	用戶分佈
Android 4.4 KitKat	19	0.4 %
Android 5.0 Lollipop - 5.1 Lollipop	21 - 22	0.7 %
Android 6 Marshmallow	23	0.7 %
Android 7.0 Nougat - 7.1 Nougat	24 - 25	2.6 %
Android 8.0 Oreo - 8.1 Oreo	26 - 27	3.5 %
Android 9 Pie	28	6.3 %
Android 10 Q	29	10 %
Android 11 R	30	16.4 %
Android 12 S - S_V2	31-32	12.4 %
Android 13 Tiramisu	33	19.3 %
Android 14 Upside Down Cake	34	25.8 %
Android 15T	35	-



# ANDROID 開發環境設定

• 新增 Android 13 (Pie) , API Level 33

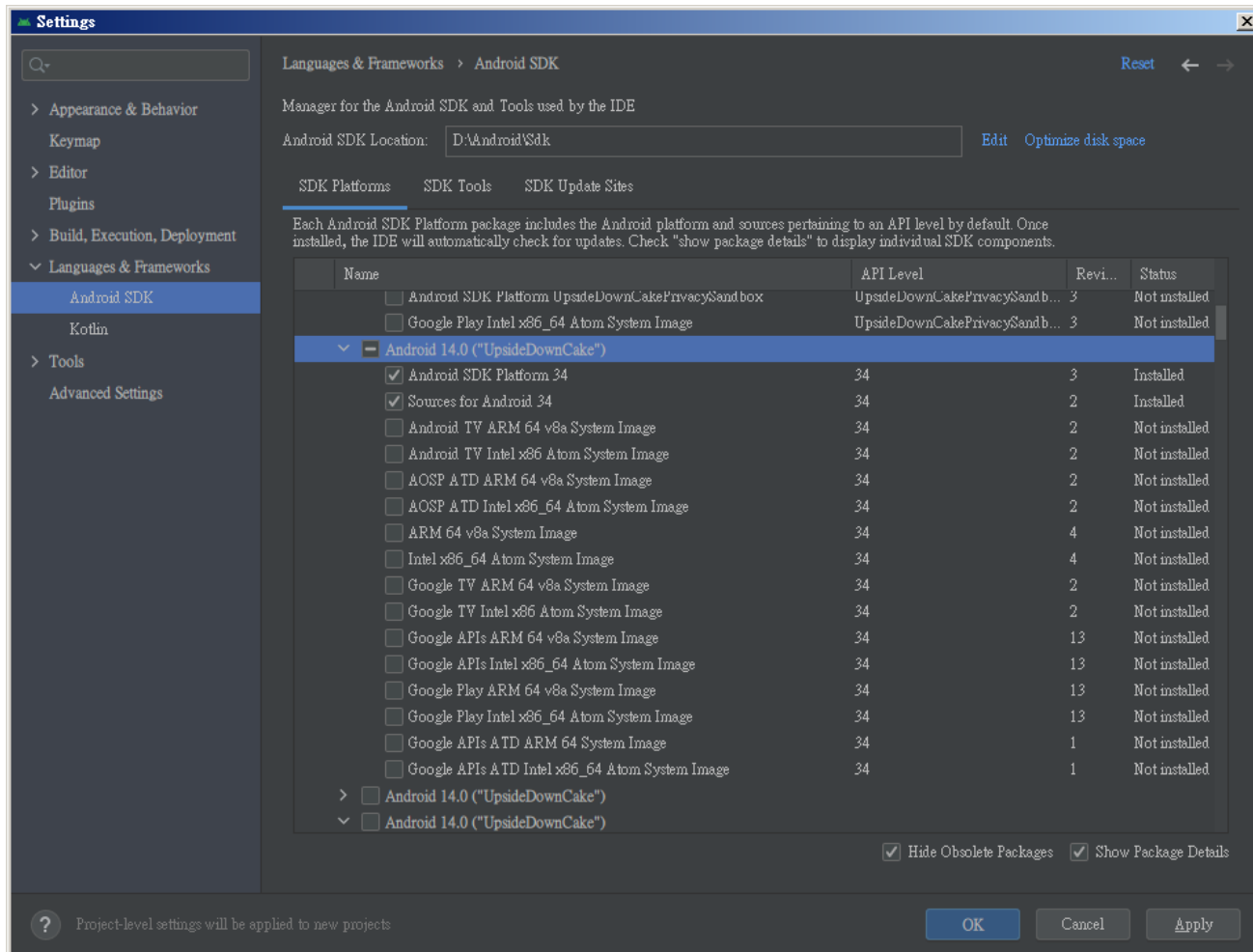






# ANDROID 開發環境設定

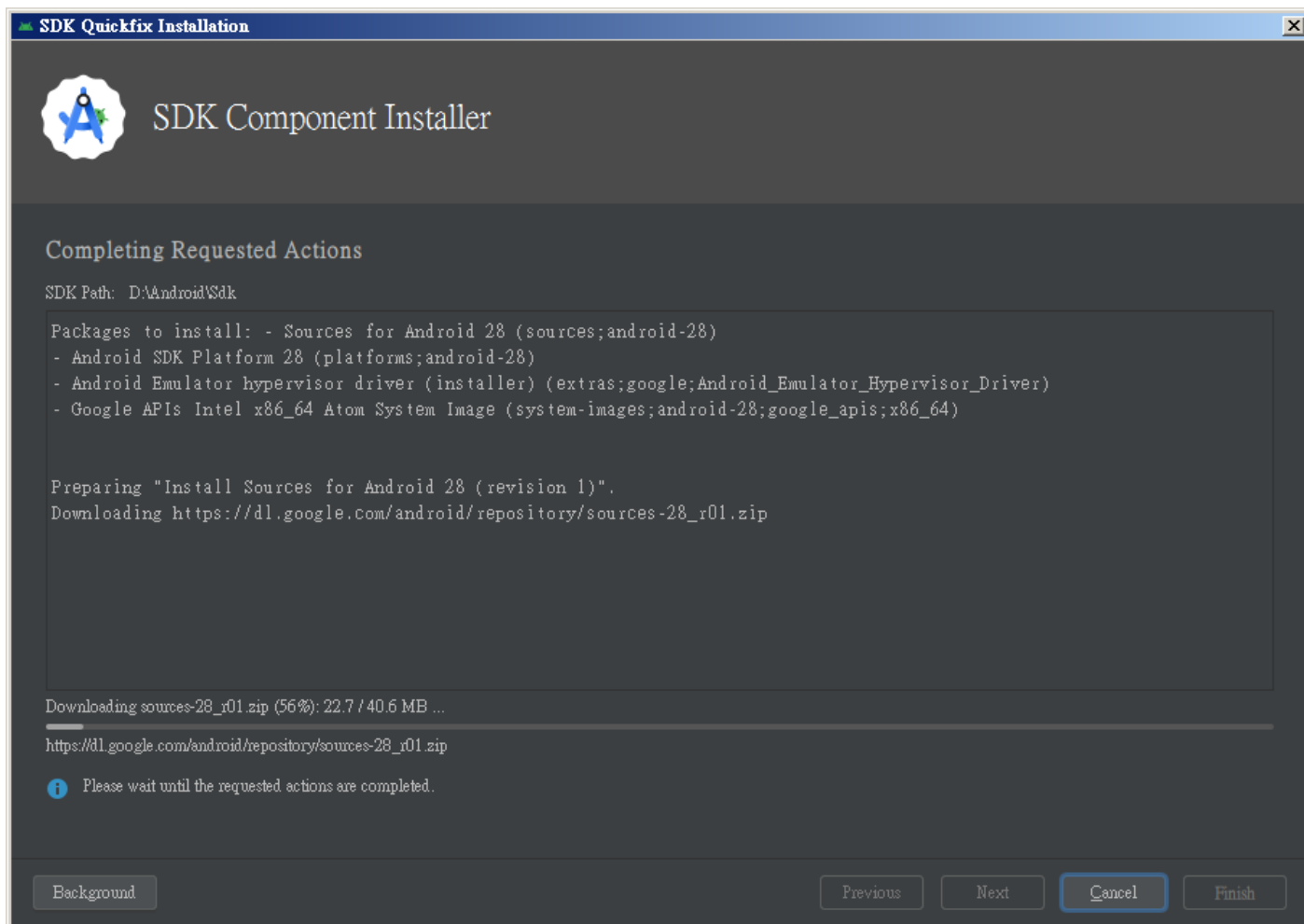
- 按下 Apply，開始更新SDK套件





# ANDROID 開發環境設定

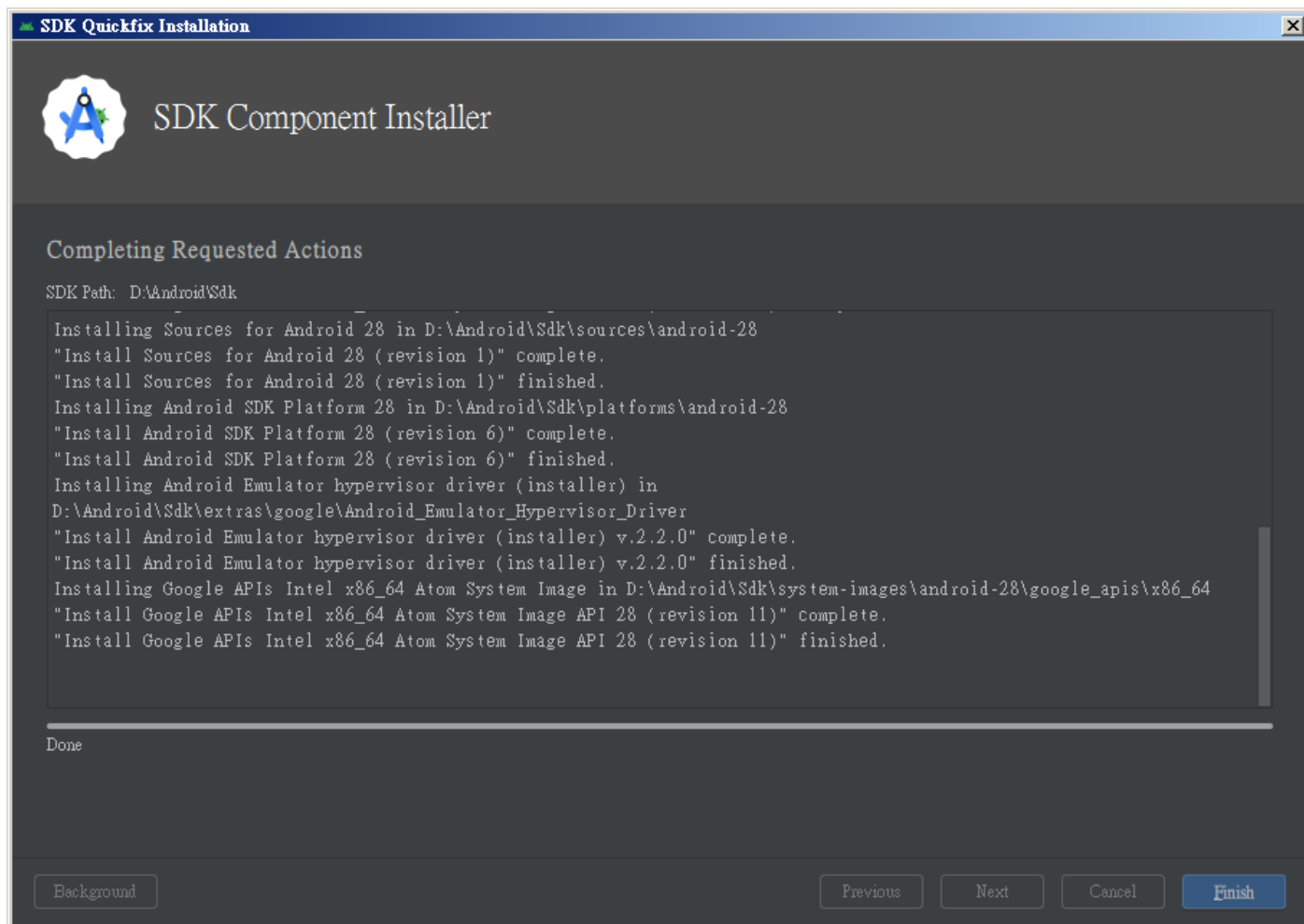
- SDK Manager 開發套件管理工具





# ANDROID 開發環境設定

- SDK Manager 開發套件管理工具





# ANDROID 開發環境設定

- 建立與使用裝置模擬器



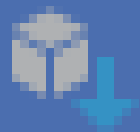
Import an Android Code Sample



Profile or Debug APK



Virtual Device Manager



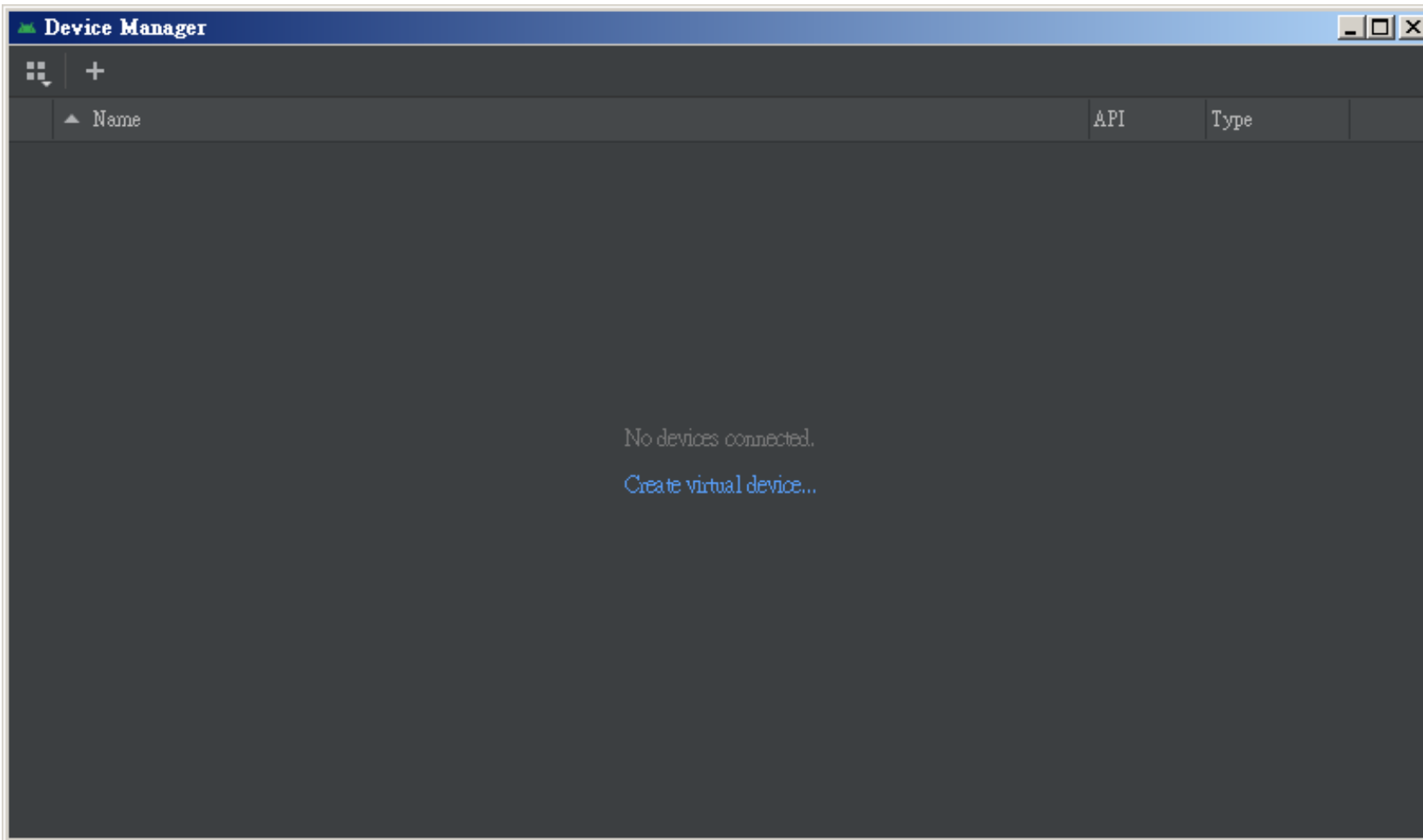
SDK Manager





# ANDROID 開發環境設定

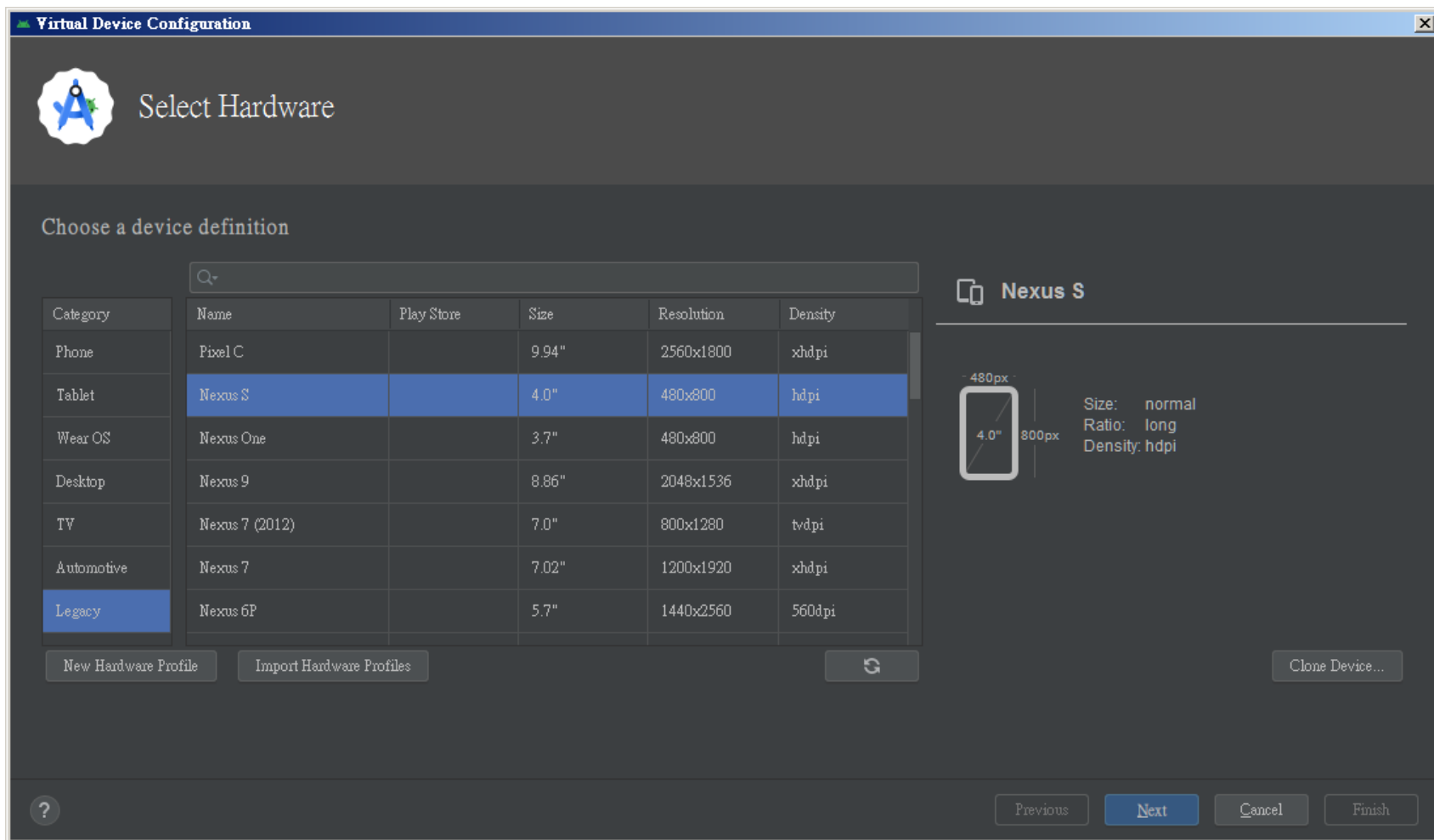
- AVD Manager 模擬器管理工具





# ANDROID 開發環境設定

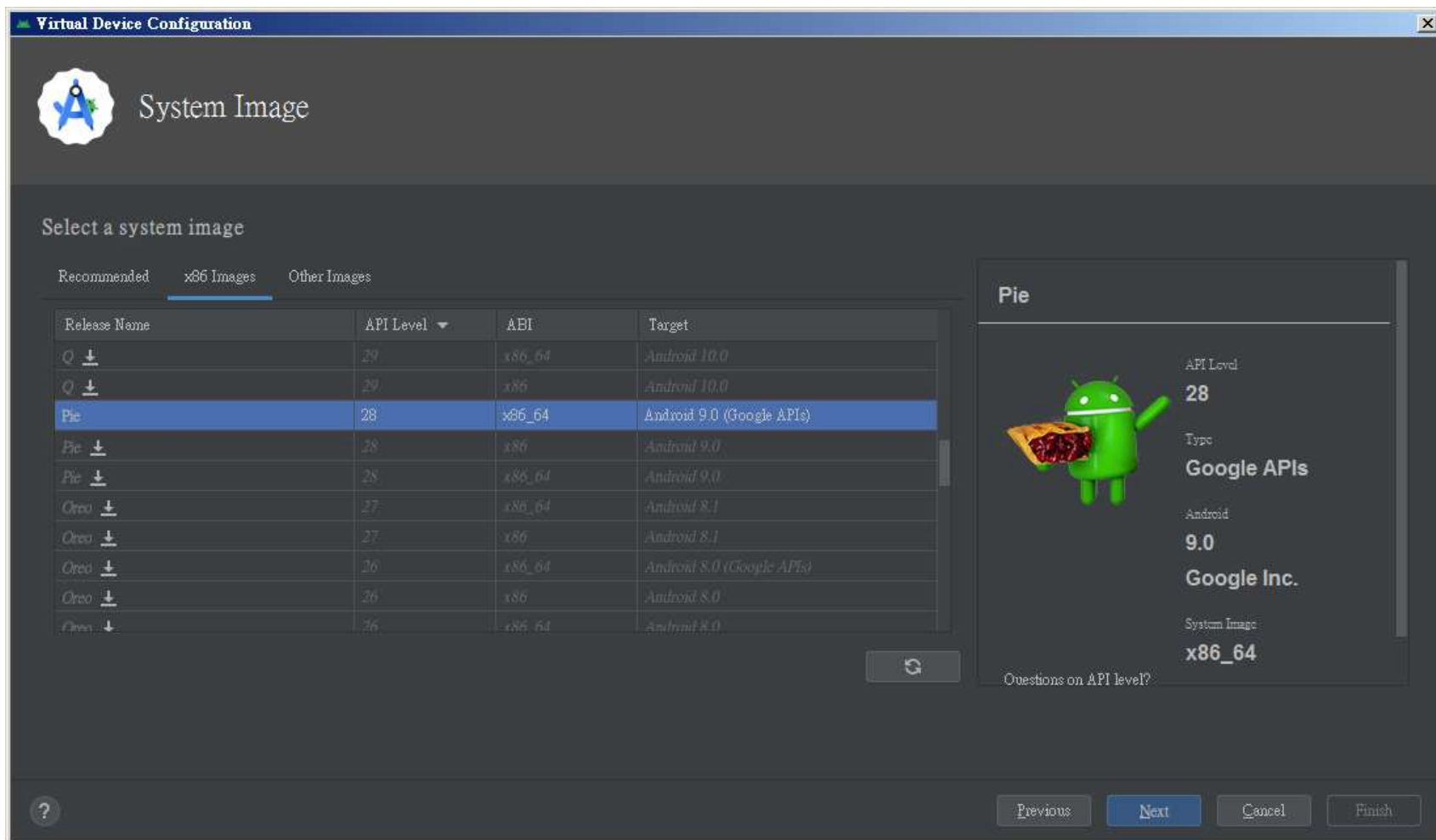
- AVD Manager 模擬器管理工具





# ANDROID 開發環境設定

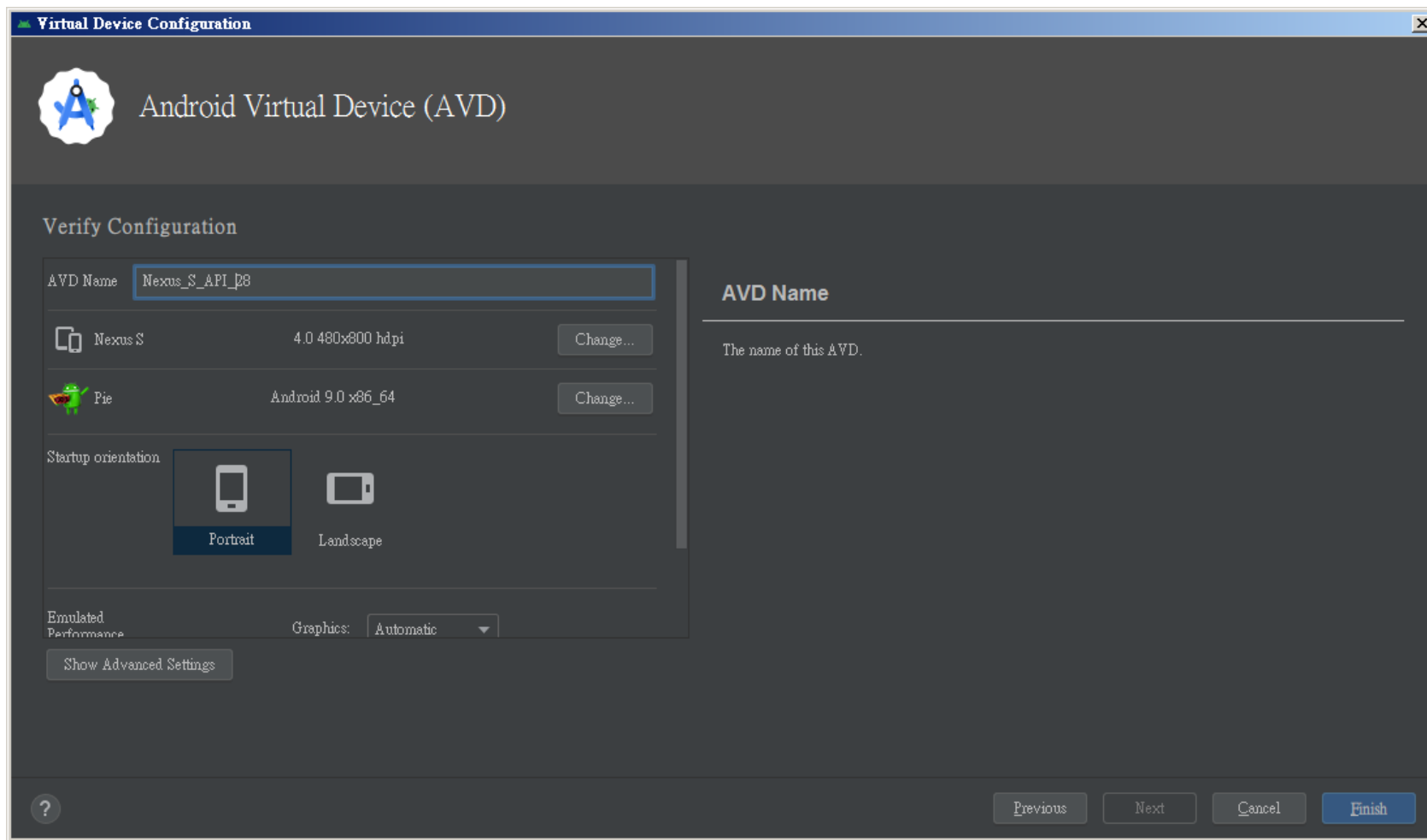
- AVD Manager 模擬器管理工具





# ANDROID 開發環境設定

- AVD Manager 模擬器管理工具







# ANDROID 開發環境設定

- AVD Manager 模擬器管理工具

Virtual Device Configuration

Android Virtual Device (AVD)

Verify Configuration

Emulated Performance

Graphics: Automatic

Boot option: ☒ Cold boot  
☐ Quick boot  
☐ Choose from snapshot  
(no snapshots)

☒ Multi-Core CPU 2

Memory and Storage

RAM:	1024	MB
VM heap:	128	MB
Internal Storage:	2048	MB

Hide Advanced Settings

Internal Flash

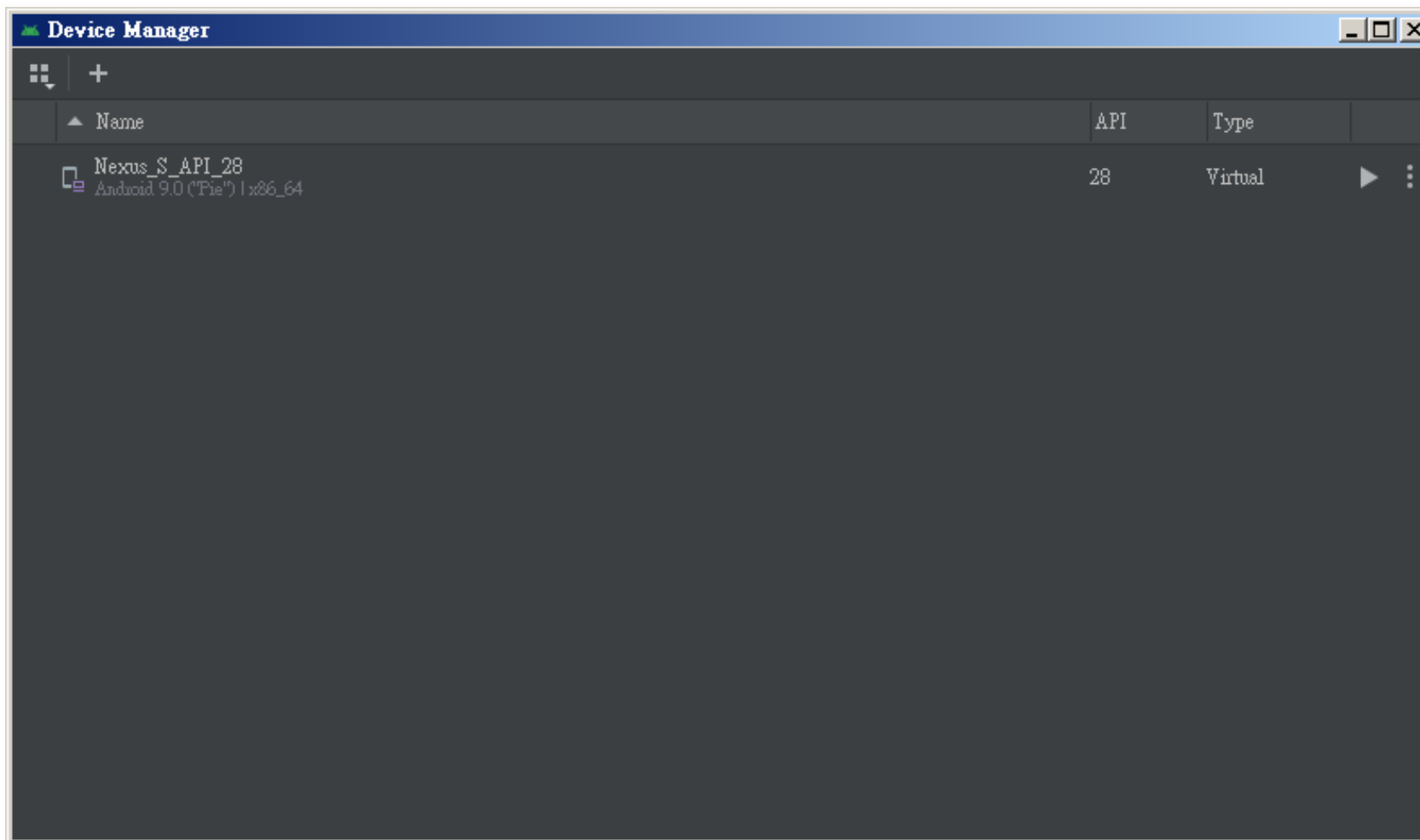
The amount of non-removable space available to store data on the device.

? Previous Next Cancel Finish



# ANDROID 開發環境設定

- 完成新增一個新模擬裝置





# ANDROID 開發環境設定

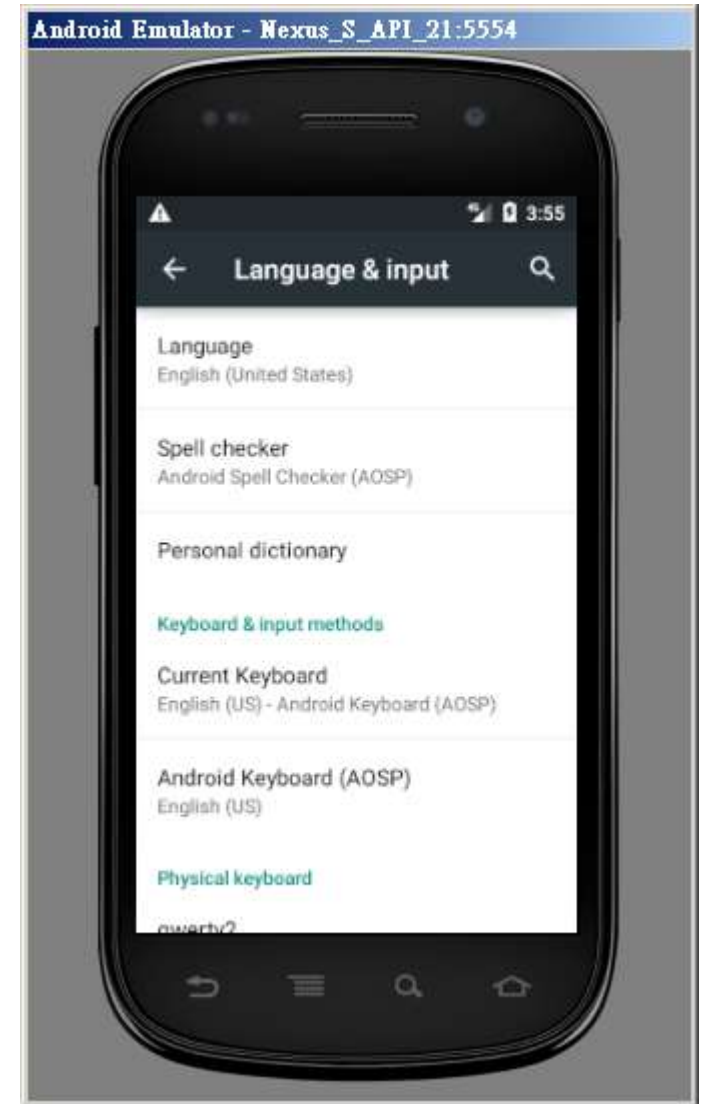
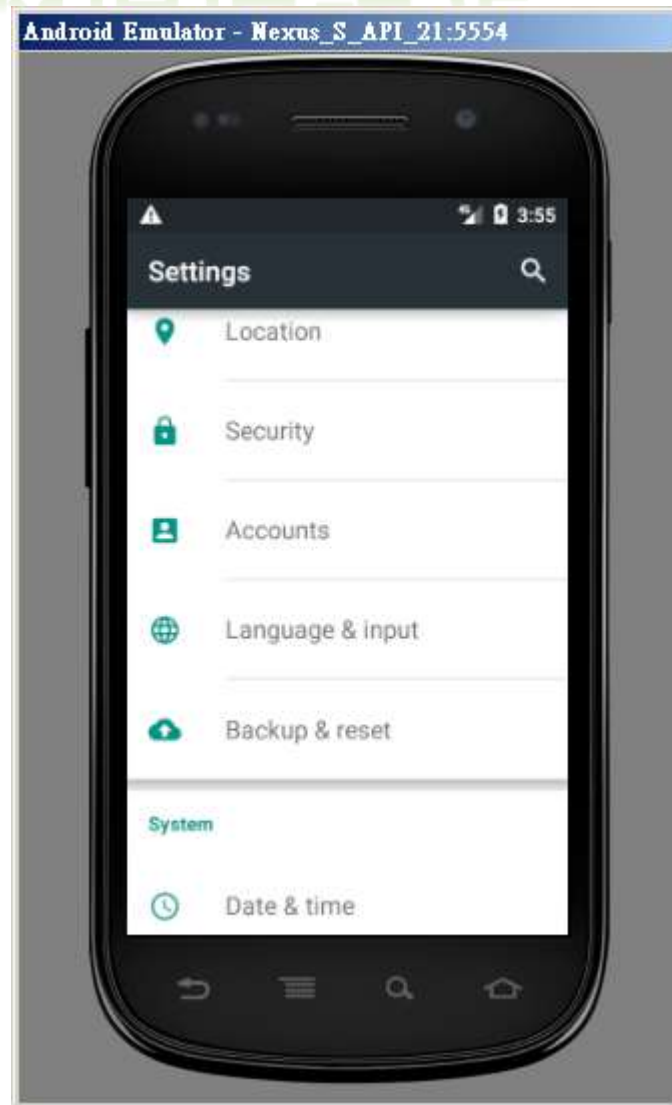
- 啟動模擬裝置，並測試相關功能
- 點選畫面下方中間按鈕，可顯示出所有已安裝的**APP**，如右圖
- 可切換語言介面至繁體中文
- 可上網，可使用地圖





# ANDROID 開發環境設定

- Setting
- Language&input
- Language
- 切換語言介面至繁體中文





# ANDROID 開發環境設定

- 畫面右下方  
啟動瀏覽器  
功能
- 可模擬上網





# ANDROID 開發環境設定

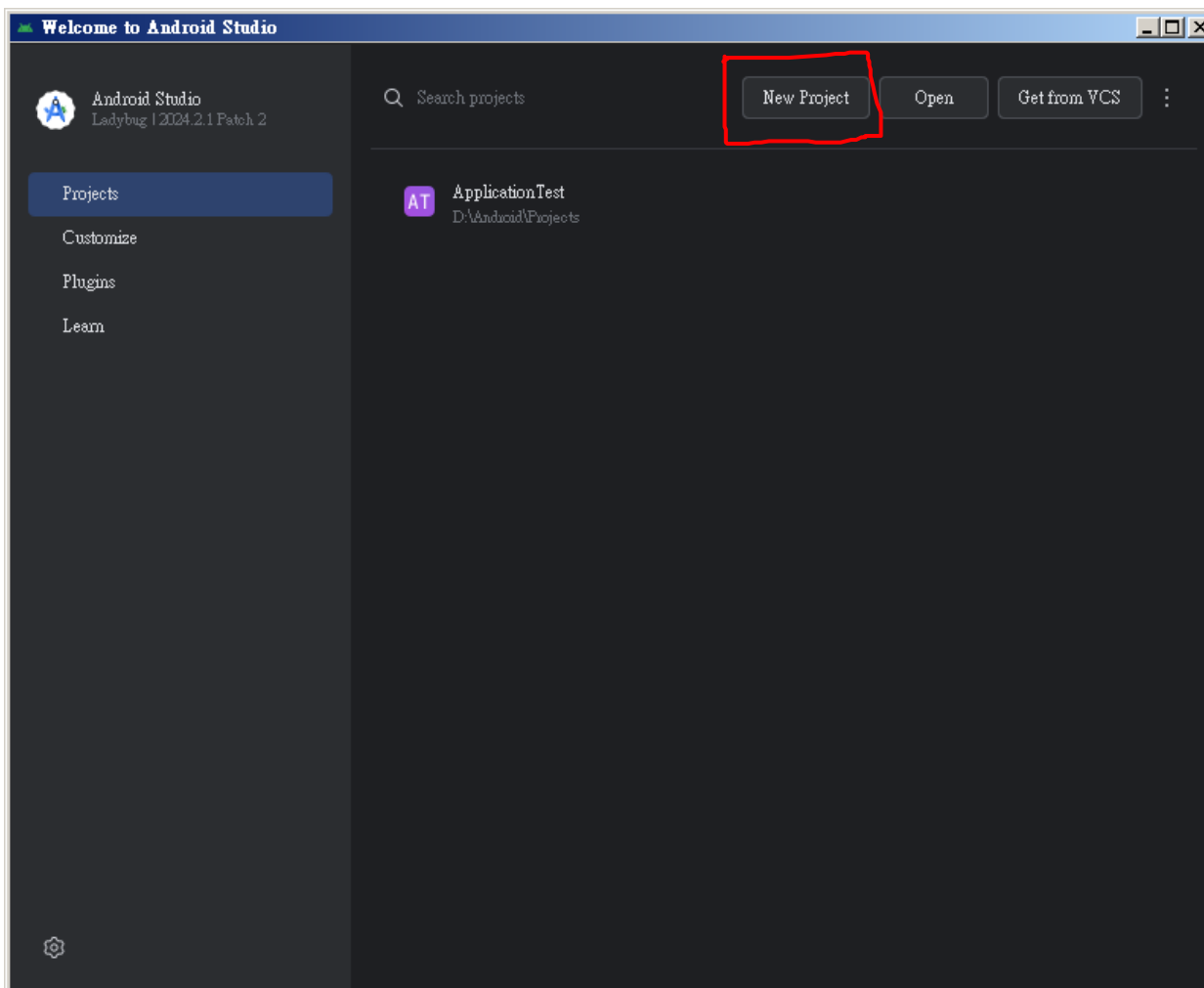
- 畫面下方中間按鈕，可顯示出已安裝 APP
- 點選左上方 Google Map APP
- 可模擬手機 Google Map APP 功能





# ANDROID 程式開發初探

- 開始使用 Android Studio 開發平台

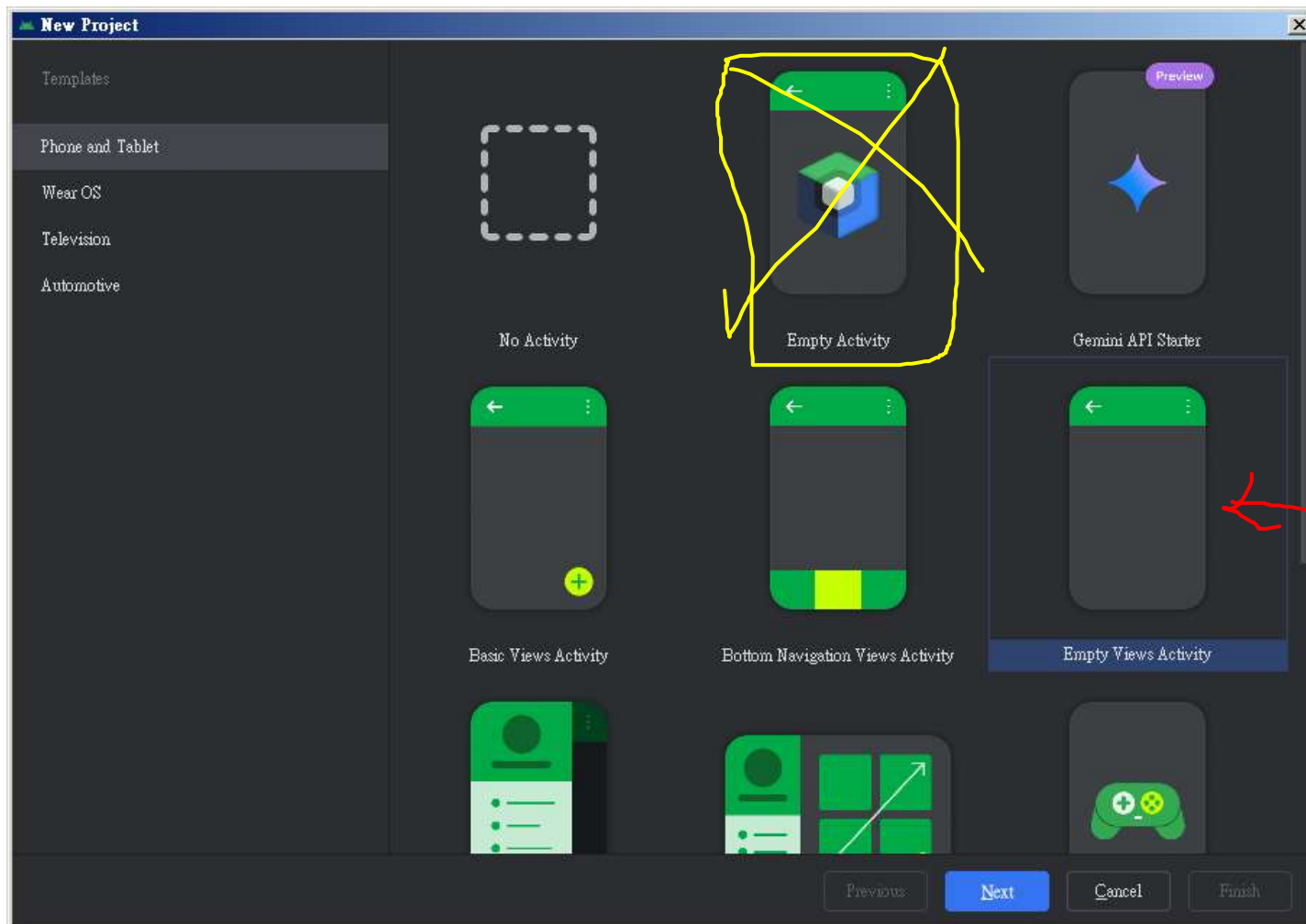






# ANDROID 程式開發初探

- 選擇畫面模板，請選：Empty Views Activity







# ANDROID 開發環境設定

- 首先建立一個新專案

A screenshot of the "New Project" dialog box in an IDE. The dialog has a title bar "New Project" with a close button. The main area is dark gray. At the top, it says "Empty Views Activity" and "Creates a new empty activity". Below this are several fields: "Name" with the value "Application1", "Package name" with "com.example.application1", "Save location" with "D:\Android\Projects\Application12" and a folder icon, "Language" with a dropdown menu showing "Java", and "Minimum SDK" with a dropdown menu showing "API 24 ('Nougat'; Android 7.0)". Below these is a blue information icon and text: "Your app will run on approximately 96.3% of devices. Help me choose". At the bottom is a "Build configuration language" field with a dropdown menu showing "Groovy DSL (build.gradle)". At the very bottom are four buttons: "Previous", "Next", "Cancel", and "Finish".

New Project

Empty Views Activity

Creates a new empty activity

Name Application1

Package name com.example.application1

Save location D:\Android\Projects\Application12

Language Java

Minimum SDK API 24 ("Nougat"; Android 7.0)

! Your app will run on approximately 96.3% of devices.  
Help me choose

Build configuration language ? Groovy DSL (build.gradle)

Previous Next Cancel Finish



# ANDROID 程式開發初探

- **Application Name :**
  - 應用程式的名稱，使用一個易於辨識的名子。
- **Company Domain :**
  - 輸入公司或個人的網域名稱。
  - 代表應用程式主要套件的名稱。
  - 採用Java應用程式的套件規則。
  - Android應用程式需要套件名稱來進行識別，最少要有兩層。
  - 在Google Play上的Android應用程式，不允許有同樣的套件名稱。
- **Package Location :**
  - 代表應用程式儲存的位置。



# ANDROID 開發環境設定

- 設定最低支援的 API LEVEL (SDK 版本)

The screenshot shows the 'New Project' dialog box in Android Studio. The dialog is titled 'New Project' and has a close button in the top right corner. It contains the following fields and options:

- Empty Views Activity**: A section header.
- Creates a new empty activity**: A description of the project type.
- Name**: A text field containing 'Application1'.
- Package name**: A text field containing 'com.example.application1'.
- Save location**: A text field containing 'D:\Android\Projects\Application12'.
- Language**: A dropdown menu set to 'Java'.
- Minimum SDK**: A dropdown menu showing a list of API levels. The selected option is 'API 24 ("Nougat"; Android 7.0)'. The list includes:
  - API 17 ("Jelly Bean"; Android 4.2)
  - API 18 ("Jelly Bean"; Android 4.3)
  - API 19 ("KitKat"; Android 4.4)
  - API 20 ("KitKat Wear"; Android 4.4W)
  - API 21 ("Lollipop"; Android 5.0)
  - API 22 ("Lollipop"; Android 5.1)
  - API 23 ("Marshmallow"; Android 6.0)
  - API 24 ("Nougat"; Android 7.0)
- Build configuration language**: A text field with a help icon (?) next to it.

At the bottom of the dialog, there are four buttons: 'Previous', 'Next', 'Cancel', and 'Finish'.



# ANDROID 開發環境設定

- 設定 Activity 畫面名稱，建議設定具有代表性的文字

**New Android Activity**

Empty Views Activity  
Creates a new empty activity

Activity Name  
MainActivity2

☒ Generate a Layout File

Layout Name  
activity\_main2

☐ Launcher Activity

Package name  
com.example.application1

Source Language  
Java

Target Source Set  
main

Previous Next Cancel Finish



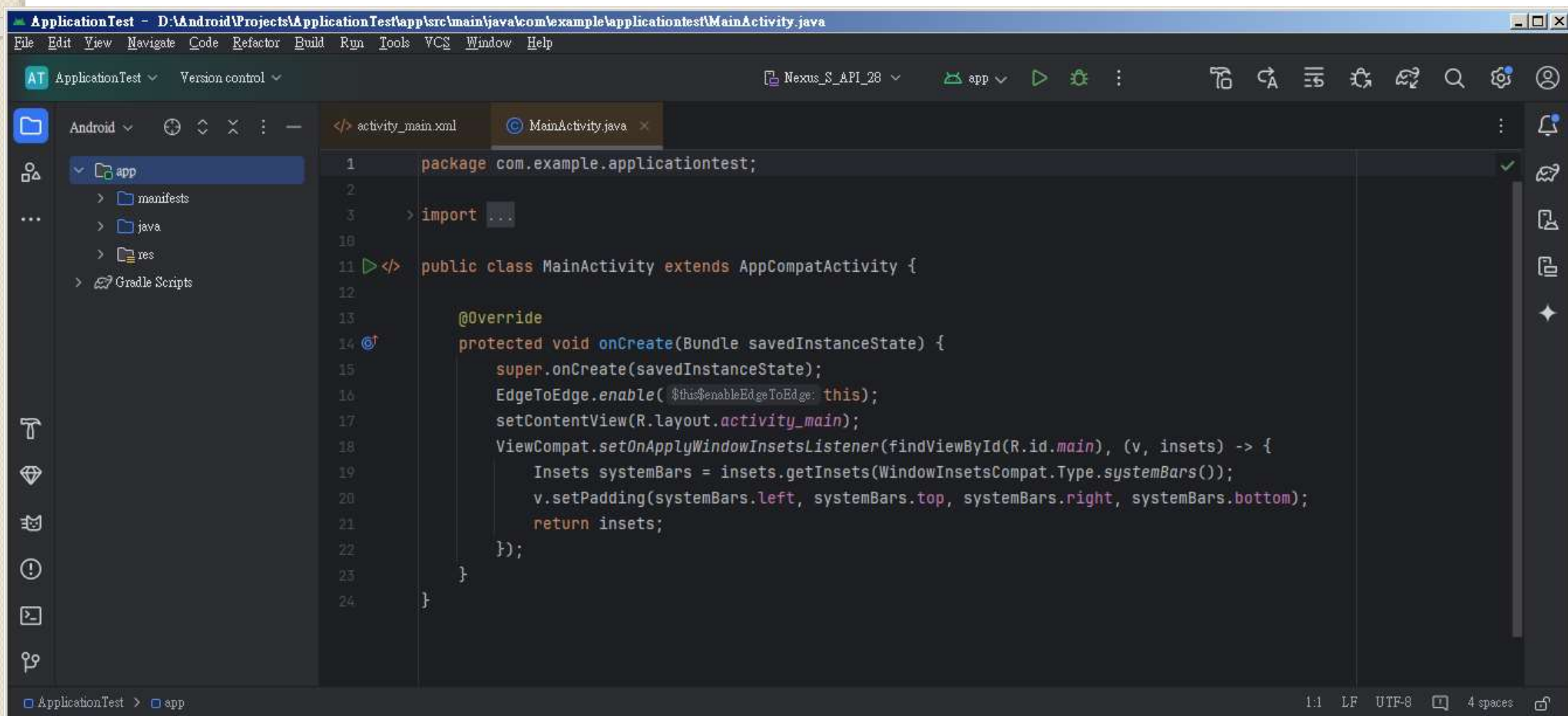
# ANDROID 程式開發初探

- Activity Name
  - Activity是Android系統中的使用者元件，通常對應於一個畫面
  - 需符合Java 類別名稱的命名規則定義
- Layout Name
  - 需符合Java 變數名稱的命名規則定義



# ANDROID 開發環境設定

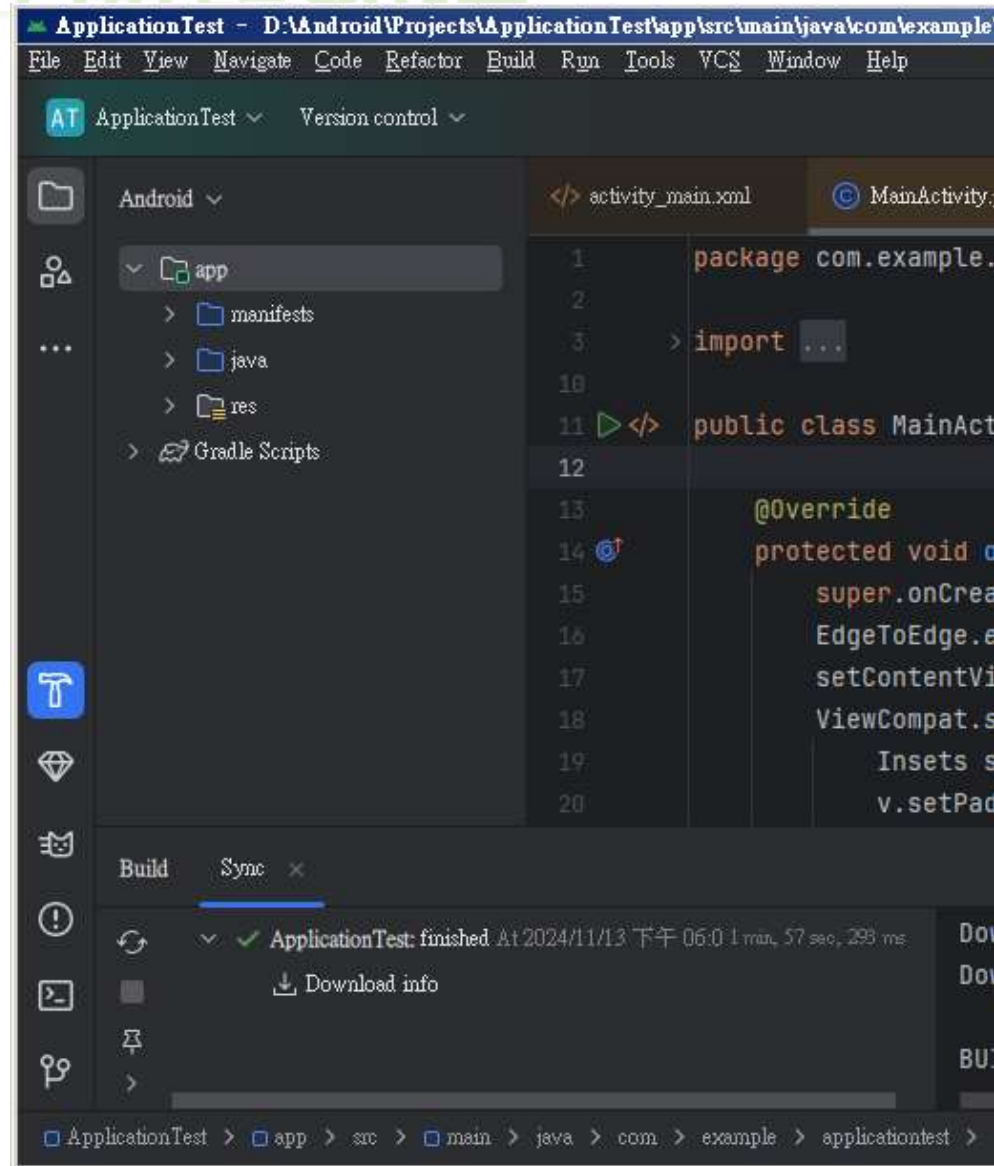
## • Android studio IDE 主畫面





# ANDROID 程式開發初探

- Android 專案，主要結構包含3 個目錄
  - Manifests
  - java
  - res





# ANDROID 程式開發初探

- **Android Studio** 產生一個應用程式專案時，會建立預設的目錄與檔案結構。以下是我們在**HelloAndroid**應用程式目錄下的結構：
  - **manifests**
    - 存放應用程式的主要設定檔「**AndroidManifest.xml**」
  - **java**
    - 存放應用程式需要的**Java**原始程式碼
    - 展開時會看到在建立專案時決定的主要套件名稱
    - **HelloAndroid** 套件下有一個預設元件類別**MainActivity.java**





# ANDROID 程式開發初探

- **Android Studio** 產生一個應用程式專案時，會建立預設的目錄與檔案結構。以下是我們在**HelloAndroid**應用程式目錄下的結構：
  - **res**
    - 存放各式資源檔案的目錄，用以分組資源類型
    - 應用程式需要的資源，例如圖形（**png**）與音效（**mp3**）檔案
    - 各種**XML**格式的資源檔案也都放在這個目錄
    - **Layout**內存放定義使用者介面版面配置的 **XML** 檔案
    - **mipmap**內存放各種不同啟動器的圖示檔案
    - **Values** 內存放各種簡單值 (字串、整數、浮點數) 的 **XML** 檔案
    - **drawable**內存放繪圖資源子類型(**.png**、**.jpg**、**.gif**的點陣圖檔案



# ANDROID 程式開發初探

- **Gradle** Scripts目錄，儲存與 Gradle 建置系統環境相關的設定檔
- Gradle 是 Android Studio 一種全新應用程式建置系統
- 當使用 Android Studio開發Android應用程式，一個應用程式可以有多個模組（Module），一般有三個模組，每一個模組都可以被建置成一個獨立的App
  - ✓ 行動電話 TV ✓
  - ✓ 平板電腦 Desktop
  - ✓ 穿戴式 Auto
- Gradle Scripts目錄兩個主要的設定檔：
  - build.gradle(Project:專案名稱)：應用程式最頂端的Gradle設定檔
  - build.gradle(Module:模組名稱)：每一個模組的Gradle設定檔



# 甲骨文(Oracle)與Google之Java侵權事件

- 2010年8月開始，甲骨文公司對Google提起訴訟
  - 主張 google 無授權使用Java語言，侵犯專利
  - 表示原本任職 Sun 公司 的 Java 開發人員在轉職Google後所使用 Android 平台 Java 環境使用 Oracle 的專利而沒有 付費
  - 在美國加州北區地方法院提起訴訟
  - 要求 90億美元的賠償
- 2016年8月開始內建 Open JDK
  - Google 在 Android 7.0 (Nougat - API 24) 開始，將原本 Sun 公司的 JDK 替換成開源社群 Opensource 的 Open JDK
  - 後續並接著推出 Kotlin 語言，以徹底解決Java的專利問題



# Hello Android 程式

- MainActivity.java
  - package com.example.application1;
  - import androidx.appcompat.app.AppCompatActivity;
  - import android.os.Bundle;
  - public class MainActivity extends AppCompatActivity {
    - @Override
    - protected void onCreate(Bundle savedInstanceState) {
    - super.onCreate(savedInstanceState);
    - setContentView(R.layout.activity\_main);
    - }
    - }



# Hello Android 程式

- activity\_main.xml

- <TextView
- android:layout\_width="200dp"
- android:layout\_height="200dp"
- android:text="Hello World 我的第一個 Android APP!"
- android:textSize="24sp"
- app:layout\_constraintBottom\_toBottomOf="parent"
- app:layout\_constraintLeft\_toLeftOf="parent"
- app:layout\_constraintRight\_toRightOf="parent"
- app:layout\_constraintTop\_toTopOf="parent"
- app:layout\_constraintHorizontal\_bias="0.133"
- app:layout\_constraintVertical\_bias="0.063" />



# Hello Android 程式

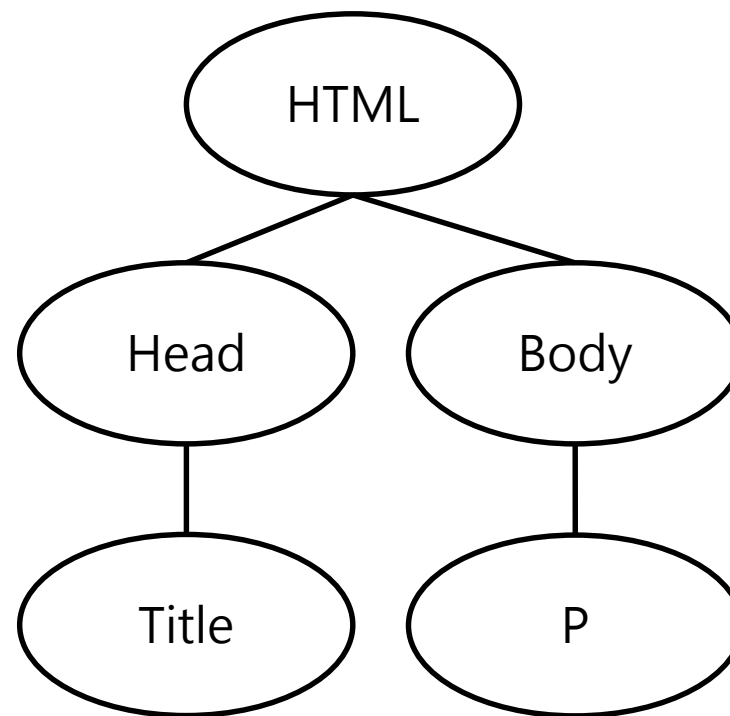
- Xml 格式檔案基礎
- eXtensible Markup Language
- 一種標籤，適合人類、機器共同閱讀
- 具有良好可攜性
- 跨平台
- 提供長期資料保存的效果
- 一種理想的檔案交換格式



# Hello Android 程式

- Xml格式及語法
- 樹狀結構
- 語法類似於HTML

- <html>
- <head>
- <title>
- My Web Site
- </title>
- </head>
- <body>
- This is my web site
- </body>
- </html>





# Hello Android 程式

- Xml格式及語法
  - 由標籤組成
    - `<?xml version = "1.0"?>`
    - `<!--` 這裡是註解 `-->`
    - `<message id="8942020">`
    - `<data>Hello! XML!!</data>`
    - `</message>`
  - 沒有固定的tag和elements
    - `<java></java>`
    - `<Cpp></Cpp>`





# Hello Android 程式

- Xml操作的要件

- Parser :

- 一種用來解析XML文件的軟體

- Well-formed :

- 滿足XML語法，能順利被parser解讀

- 讀寫：

- XML文件必須格式正確，Parser才能順利存取



# Hello Android 程式

- Well-formed XML語法的規則

✱ 起始標籤要有個相對應的結束標籤

- `<data></data>`

- `<br></br>`

✱ 元素與元素不可以重疊使用

- `<a><b></a></b>`

✱ 屬性值需用引號包圍

- `<person id="9935">Frank</person>`

✱ 元素或屬性裡不能出現<或&字元符號

- `<SA&P></SA&P>`

- `<GG&amp;D> </GG&amp;D>`



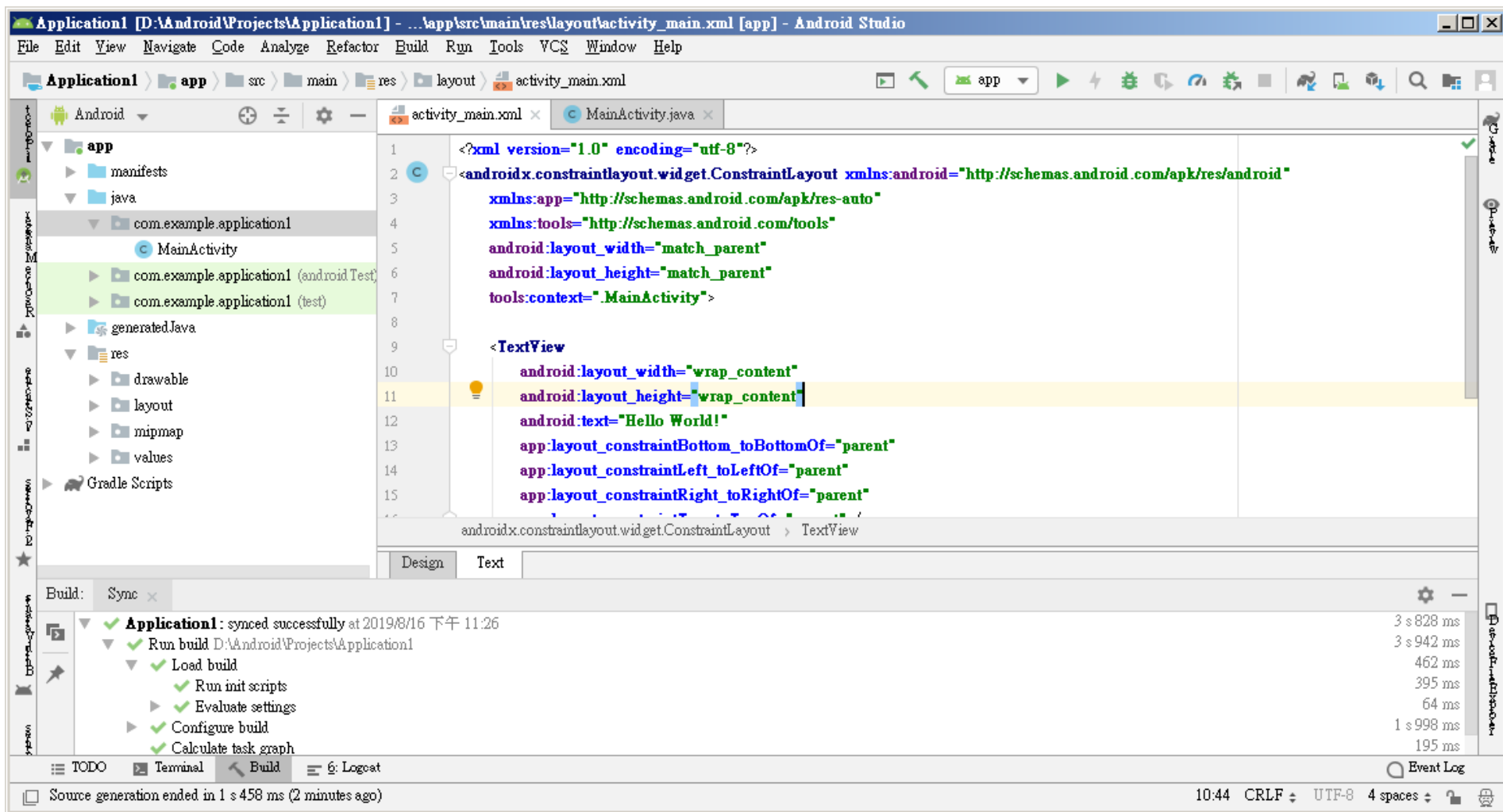
# Hello Android 程式

- XML常見語法與註解
  - `<元素> character data </元素>`
  - `<Space>`
  - `< hasID = "simple" />`
  - `<!-- hasID = " i20" / -->`
  - `</Space>`
- 具大小寫差異性
  - `<NAME>陳念</name>`
- 註解寫法
  - `<!-- 註解內容 -->`
  - `--->` 不可用 作結束符號



# Hello Android 程式

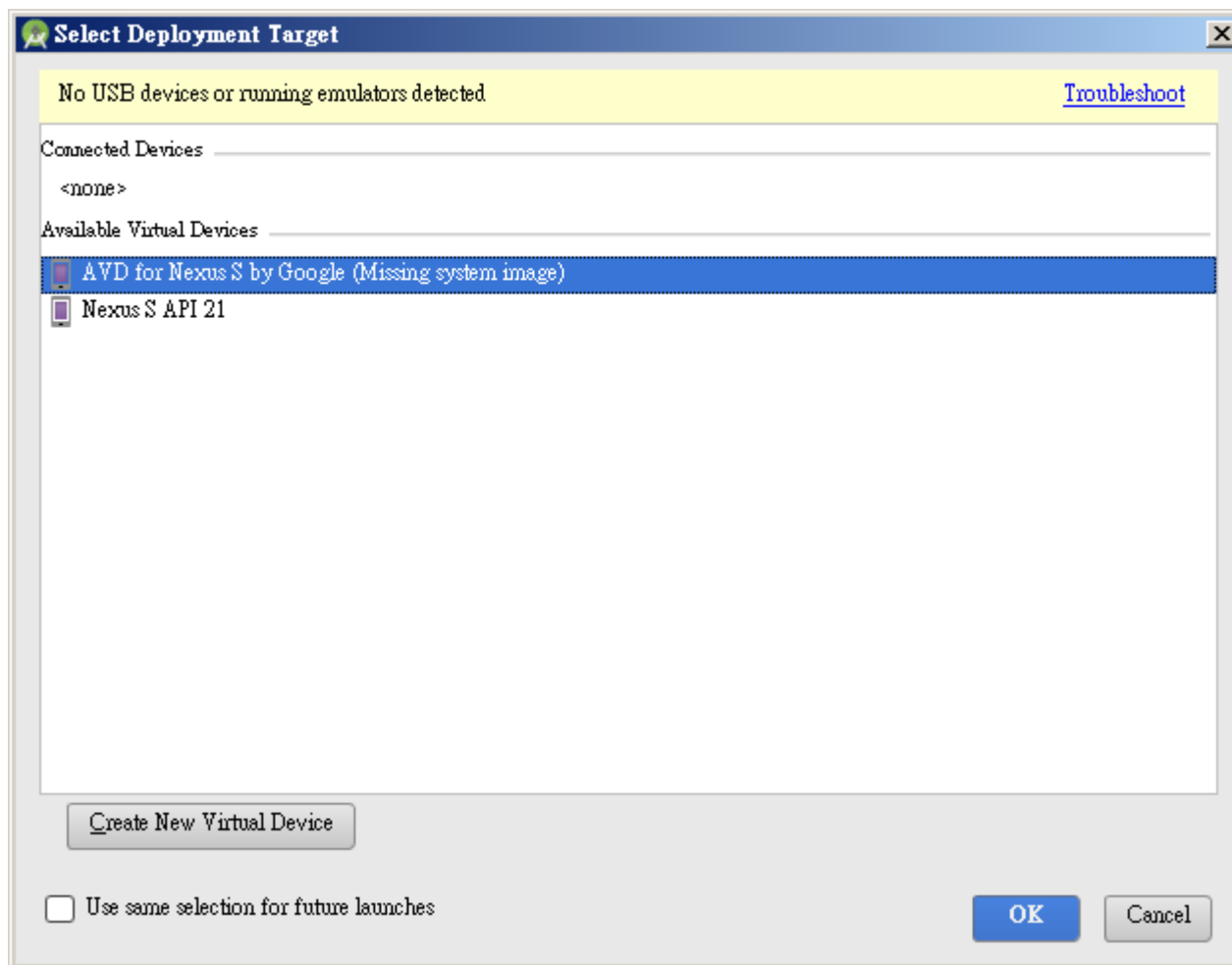
- 編輯程式內容並執行





# Hello Android 程式

- 編輯程式內容並執行





# Hello Android 程式

- 實際執行結果





# Hello Android 程式

- 直接使用實體手機開發





# Hello Android 程式

- 直接使用實體手機開發

神腦官網-下載專區

https://www.senao.com.tw/download.php?m=&cs=&model=6040

PortableApps.com 金融 歌 浪跡 百家講壇 紀連海 - Y... 上市櫃股票一覽 - ... (4906) 正文 財報 - 2... 其他書籤

檔案類型 產品品牌 輸入產品型號 搜尋

序號	檔案類型	產品型號	項目說明	版本	大小	語言	更新日期	附件
1	使用手冊	6040D	6040D 使用手冊	v1.0	3.44MB	中文		
2	同步軟體	6040D	同步軟體	V1.1.2	120.21 MB	繁體中文	2013-11-12	

1

投資人專區 | 下載專區 | 隱私權保護 | 聯絡我們 | 菁英招募 | 經銷商專屬網站 | 防詐騙公告

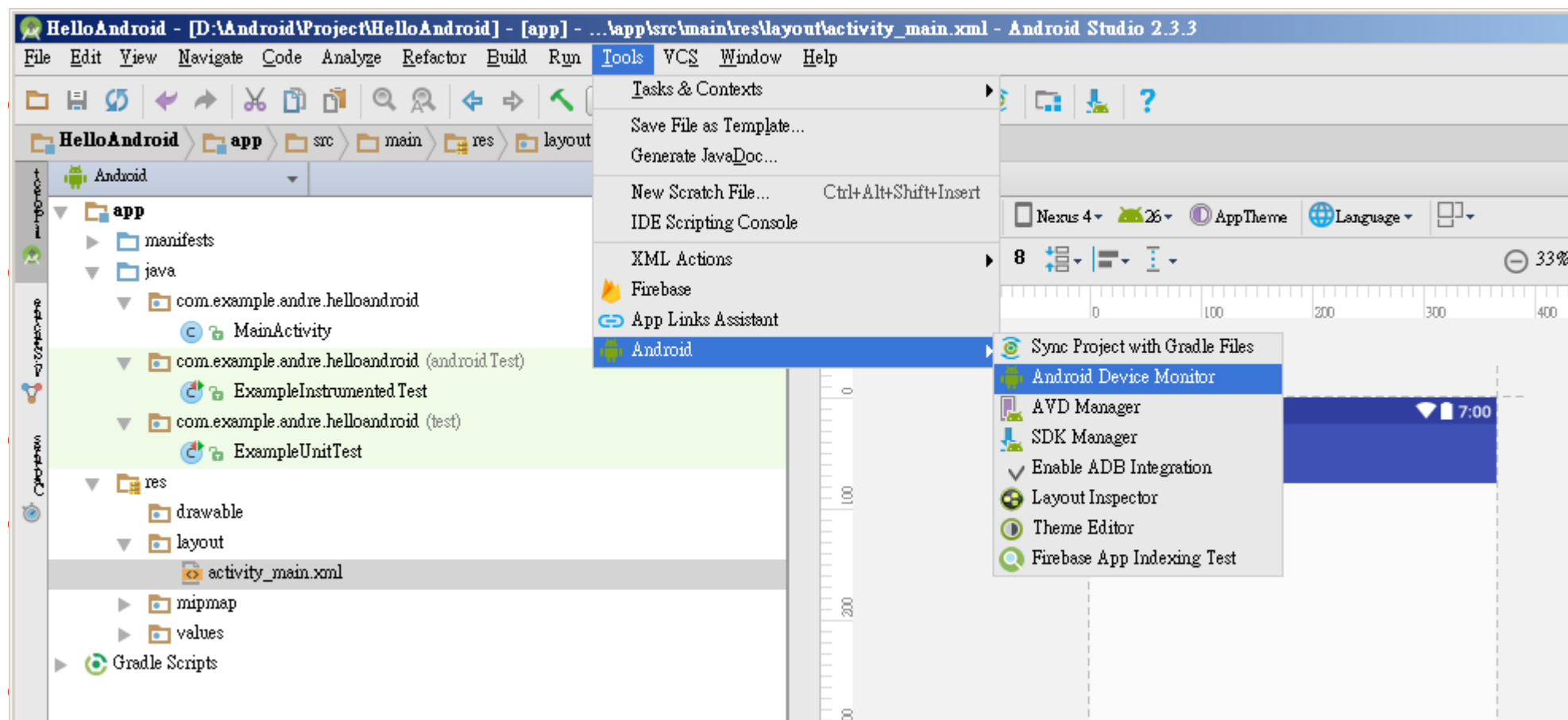
關於我們 商品資訊 維修服務 更多資訊 服務與支援





# Hello Android 程式

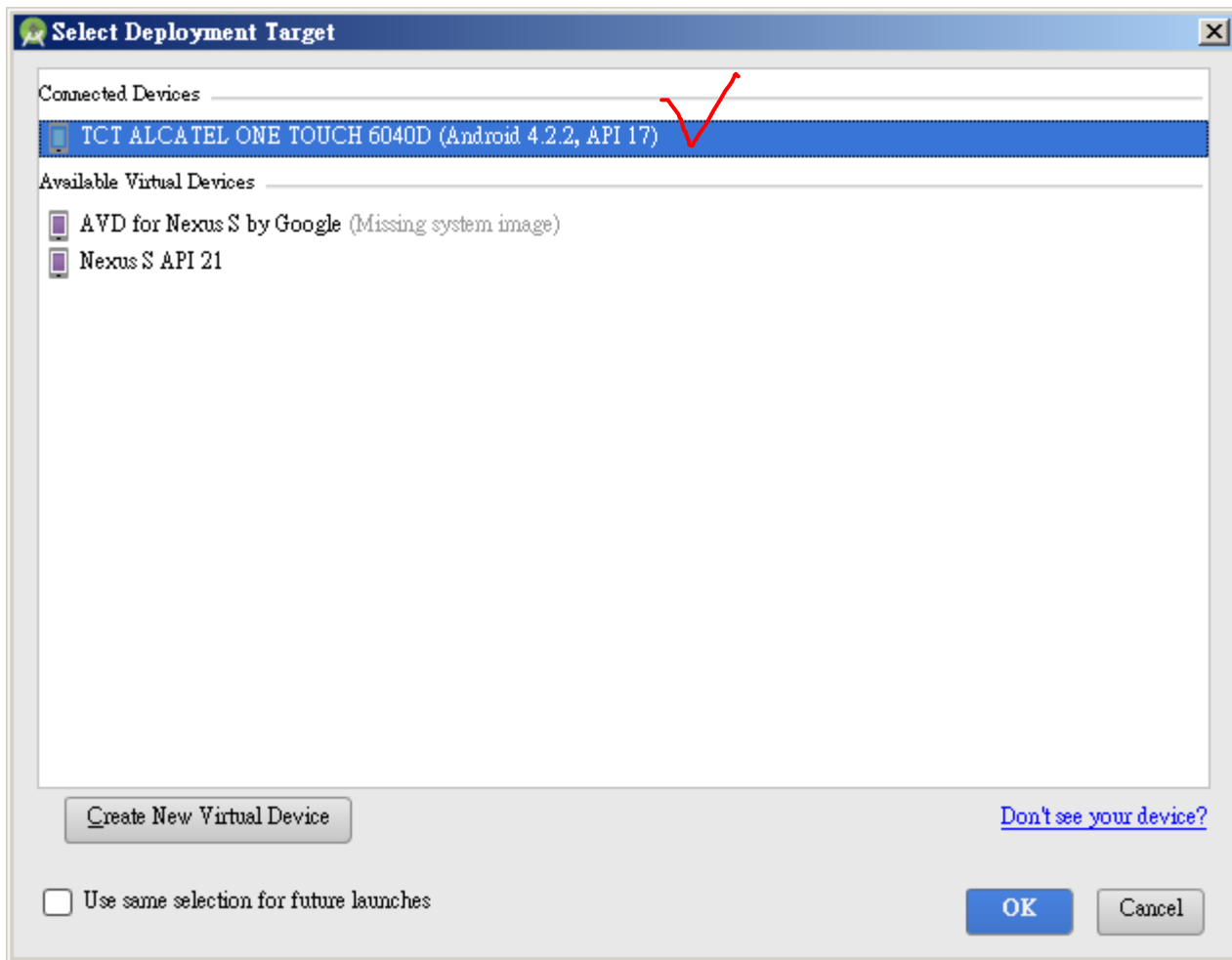
- 直接使用實體手機開發





# Hello Android 程式

- 直接使用實體手機開發





# Hello Android 程式



HelloAndroid

Hello World! 我的第一個 Android APP!!



 **THE END**