

XAML in Xamarin.Forms

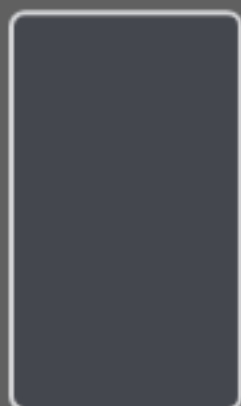
基礎篇



Android



iOS



UWP



Vulcan Lee 著

XAML in Xamarin.Forms 基礎篇

Xamarin.Forms 開發系列叢書

Vulcan Lee

這本書的網址是 <http://leanpub.com/XAML-in-Xamarin-Forms-Basic>

此版本發布於 2018-11-10



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2018 Vulcan Lee (李進興)

Contents

前言	i
關於本書	iii
這本書能提供什麼	iii
誰適合閱讀這本書	iv
如何使用本書	v
意見回饋	v
I 開發前的安裝、設定準備工作	1
1. Visual Studio 2017 for Xamarin 開發環境之安裝與設定	2
1.1 安裝前的準備工作	2
1.1.1 確認作業系統版本	3
1.1.2 確認硬體 BIOS 有啟用虛擬化功能	6
1.1.3 同時安裝支援 Hyper-V 和 Windows Hypervisor 平台服務	9
1.1.4 啟用 UWP 開發人員模式	17
1.2 安裝 Visual Studio 2017	21
1.3 Visual Studio 2017 安裝後的相關設定	35
1.3.1 更新 Android SDK	35
1.3.2 檢查 Android SDK 機碼是否存在	50
1.4 測試與確認開發環境可以進行 Xamarin.Forms 專案開發	56
1.4.1 安裝與啟動 Visual Studio for Android 模擬器	56
1.4.2 若無法正常模擬器進行除錯之修正	64

CONTENTS

1.4.3	測試可以建立 Xamarin.Forms 專案	65
2.	開發時期的 XAML 即時預覽	72
2.1	推薦 XAML 的設計預覽工具大金剛 Gorilla	72
2.1.1	電腦主機軟體安裝	72
2.1.2	裝置端軟體安裝	84
2.1.3	體驗如何使用大金剛 Gorilla 預覽工具	88
2.2	學習 XAML 的最佳良伴 LiveXAML	91
2.2.1	安裝與註冊 LiveXAML 擴充功能套件	92
2.2.2	體驗如何使用 LiveXAML	99
2.3	Xamarin 即時重新載入 Live Reload	107
2.3.1	安裝與註冊 Xamarin 即時重新載入 Live Reload 擴充功能套件 .	108
2.3.2	體驗如何使用 Xamarin 即時重新載入 Live Reload	112
	版權頁	119

前言

Xamarin.Forms 為一個可以針對不同平台上的行動裝置，如：Android, iOS, UWP，協助開發者進行跨平台應用程式；Xamarin.Forms 是一套視覺介面工具組 UI Toolkit，透過宣告式的標記語言 XAML 來宣告與定義應用程式的每個頁面要出現的內容，而關於應用程式需要執行的各類商業邏輯部分，我們則需要使用 C# 程式語言在 .NET 平台上的運作。

我們知道在進行行動裝置應用程式開發的時候，需要進行該應用程式的各種畫面 UI 與商業邏輯程式碼的開發與設計，以往我們要進行開發出一套跨平台的應用程式，應用程式開發者需要精通各 App 平台提供的專屬 SDK 工具，透過 SDK 工具提供的使用者介面設計工具，來進行這些應用程式的頁面要顯示的內容的開發，因此，同樣一個頁面，應用程式開發者需要先使用 Android SDK 的使用者介面設計工具設計這些頁面畫面，接著又需要使用 iOS SDK 的使用者介面設計工具再設計一次同樣的頁面畫面。我們可以看到 Xamarin.Forms 帶來的好處那就是，而是只需要定義一組 XAML 宣告檔案，就已經完成了關於行動應用程式的使用者視覺介面的設計，因為同樣的 XAML 宣告檔案在不同行動作業系統平台下執行時候，會使用原生 SDK 提供的原生使用者介面。

另外一個好處，那就是對於使用原生 SDK 工具來進行動應用程式的開發者而言，必須學習每個平台原生 SDK 指定的專屬程式語言，如：Java, Objective-C, Swift，同樣的，對於同樣的商業邏輯需求，開發者需要使用 Android SDK 的 Java 程式語言來設計一次，接者，需要使用 iOS SDK 的 Objective-C / Swift 語言再進行設計一次，這樣對於想要進行開發出可以在不同行動作業系統平台下的開發者而言，這是一個極為嚴峻的挑戰，因為開發者需要精通這些程式語言，並且也會面臨產生更多錯誤的機會；現在，在 Xamarin.Forms 開發環境之下，我們僅需要使用 C# 程式語言，便可以設計出可以在每個平台下執行的程式碼，而且，絕大多數比例的程式碼，統計上會有高於 80% 以上的程式碼，我們都僅需要撰寫一次，並且同時在不同平台下執行，只有很少比例的程式碼，因為需要呼叫原生平台

SDK 專屬的 API，才需要使用 C# 程式語言，在每個平台下額外撰寫呼叫特定 API 的程式碼。

因此，當您決定採用 Xamarin.Forms 來進行跨平台行動應用程式開發的首選之後，您可以享受到使用一套宣告式的標記語言 XAML 來設計使用者介面 UI 與只需撰寫一次 C# 程式碼，就已經完成了可以同時在 Android, iOS, UWP 平台下可以執行的行動應用程式。

對於想要採用 Xamarin.Forms 開發工具來進行跨平台行動應用程式開發的開發者而言，相信可以在 .NET 環境下執行 C# 程式語言的使用與應用上不會很陌生，可是絕大部分的 .NET 開發者，若本身沒有接觸或者使用過 WPF / UWP 專案開發，面對對於 XAML 這個宣告式的標記語言會感到相當的陌生與恐懼，另外，在本書撰寫的這個時間點，Visual Studio 尚未提供如同 WPF 開發環境中的友善 XAML 開發與設計操作介面，想要學習 XAML 宣告式的標記語言的開發者，必須面對如同 XML 文件中的各種眾多，繁雜的標籤 Tag 與不同的屬性。

作者從事多年 Xamarin.Forms 專案開發與教學工作，有鑑於絕大多數 Xamarin.Forms 新手，都會對 XAML 這套宣告式的標記語言，存在著恐懼，同時市面上並沒有完整與適當的中文書籍，因此，著手撰寫這本書籍。

本書 **XAML in Xamarin.Forms** 將會規畫成為兩本書籍：基礎篇與進階篇，本書為 **XAML in Xamarin.Forms 基礎篇**，對於絕大部分開發者在日常開發 XAML 文件檔案時候，都將會面對這本書中談到的內容，而 **XAML in Xamarin.Forms 進階篇**則會討論到各種 XAML 的客製化、擴充與其他類型的應用主題。

關於本書

這本書能提供什麼

在這本書裡面，將會設計 7 大部分

- 開發前的安裝、設定準備工作

對於 Xamarin.Forms 開發新手，第一個學習卡關將會是如何安裝與設定一個可以進行 Xamarin.Forms 的開發環境；在這個部分將會詳細說明如何安裝與設定 Visual Studio 2017，使其可以順利的進行 Xamarin.Forms 的開發工作，並且也會介紹一個學習 XAML 宣告式標記語言的輔助工具，設計時期預覽工具大金剛 Gorilla Player 與執行時期預覽設計工具 LiveXAML，對於如何安裝與使用進行說明。

- 基本概念

這個部分將會介紹您學習 XAML 語言必須要先瞭解的各種基本知識與觀念，讓您初步綜覽在學習 XAML 語言比須具備的基本技能。

- XAML 共用屬性

我們將說明 XAML 提供的各種類別與其階層關係，這些類別將會被所使用的各種頁面、版面配置、檢視項目所繼承，因此，您可以從這些頁面、版面配置、檢視項目中，使用基底類別所提供的各種屬性、方法與事件。

這些共用的基底類別包含了：可綁定物件 BindableObject，項目 Element，視覺項目 VisualElement，頁面 Page，檢視 View，版面配置 Layout

- 各種不同頁面

頁面是在 Xamarin.Forms 開發上，最為重要的使用者介面項目，當使用者在開啟此應用程式，第一個畫面內容就是一個頁面；原則上，我們進行 Xamarin.Forms 專案

開發，就是要設計出各種不同頁面，並且根據應用程式設計需求，將這些頁面串接起來，提供使用者來操作這些頁面上設計的內容。

Xamarin.Forms 內提供了這些頁面可供選擇：內容頁面、導航頁面、主從階層頁面、標籤頁面、旋轉木馬頁面。

- 各種不同版面配置

我們要把各種 XAML 提供的視覺項目，定位在螢幕上的指定地方，並且可以根據螢幕的大小、解析度、密度來自動調整要顯示出我們所希望的結果，畢竟，要根據不同螢幕的大小、解析度、密度來設計出各種不同的 XAML 文件，這是相當麻煩的，因此，我們將會透過 XAML 中的版面配置來幫助我們自動做到這些事情。

在 Xamarin.Forms 有提供眾多的版面配置，每個版面配置可以針對不同的設計情境來使用，這包括了：框架、內容檢視、堆疊方式版面配置、格線、捲動檢視、絕對版面配置、相對版面配置、彈性方塊版面配置。

- 各種不同檢視

檢視是要提供與使用者互動的使用者介面，例如，讓使用者輸入文字、顯示出各種圖片、文字、請使用者選擇指定的日期與時間等等，在這部分將會帶領大家了解這些檢視的功能與使用方式。

- 各種資料綁定語法

最後，會針對這本書上各種範例中有用到的資料綁定使用方式，進行完整的歸納與整理，讓您清楚了解如何在 XAML 中正確的使用資料綁定，來幫助您進行各種情境的 Xamarin.Forms 應用程式開發。

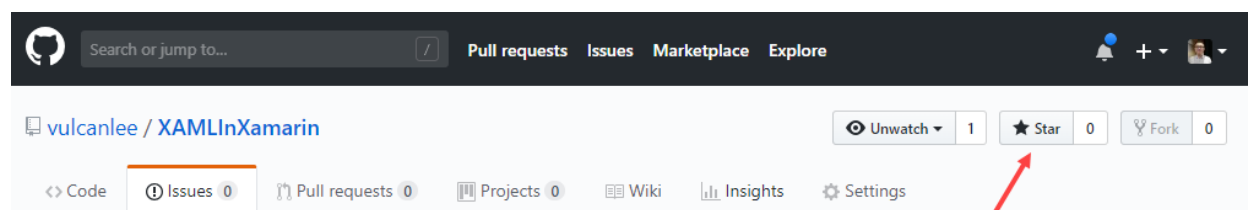
誰適合閱讀這本書

本書適合想要學會各種 XAML 開發技巧的開發者，將會提供基本 XAML 知識與觀念，不同 XAML 項目的使用與應用的介紹，當然，您還是需要擁有基本的 .NET C# 程式語言的開發經驗。

如何使用本書

在書中除了會提供各種學習 XAML 宣告式標記語言的內容，並搭配不同的應用程式範例進行介紹，從書中的第三部份，將會說明這段內容的講解範例是如何製作出來，希望讀者根據書中的說明步驟，搭配講解內容，逐一自我在電腦前面進行演練，可快速增加學習 XAML 的經驗值。

本書中的所有講解範例專案都會放在 Github 上，您可以透過 [Github 的 XAMLInXamarin¹](#) 來取得這些講解範例專案，並且鼓勵大家可以到這個 [XAMLInXamarin Repository²](#) 頁面，在螢幕的右上方，點選 Start 按鈕給予鼓勵，如同下圖箭頭所指向地方。



在 XAMLInXamarin 上建立一個 Issue 操作示意圖

意見回饋

對於在學習 Xamarin.Forms 開發上，有任何的疑問與問題，可以到 Facebook [Xamarin.Forms @ Taiwan³](#) 社團與其他 Xamarin.Forms 開發者進行討論。

也建議加入 Facebook [Xamarin 實驗室⁴](#) 粉絲團，作者會經常在這裡貼出各種

¹<https://github.com/vulcanlee/XAMLInXamarin>

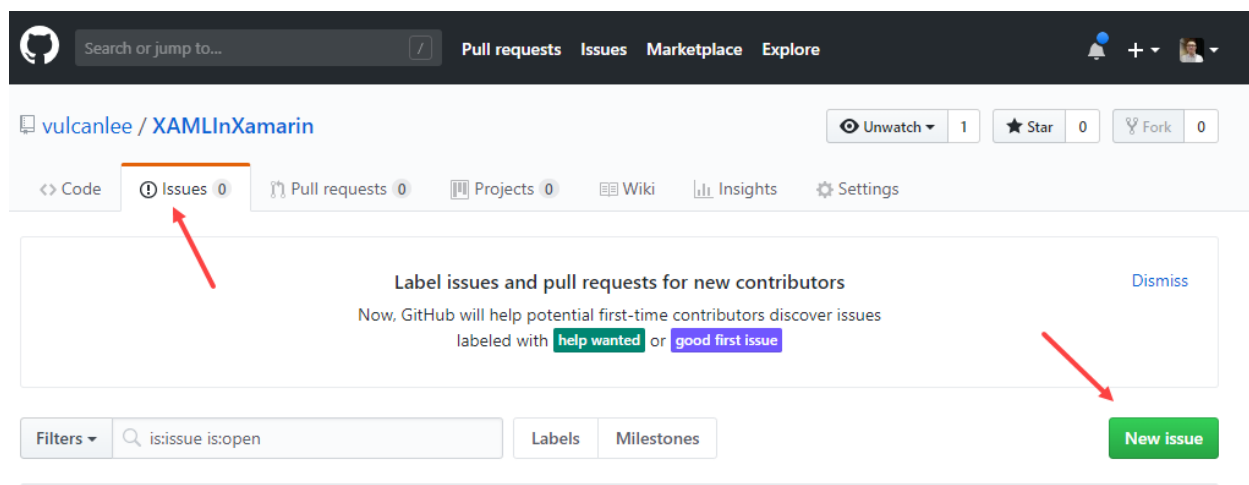
²<https://github.com/vulcanlee/XAMLInXamarin>

³<https://www.facebook.com/groups/XamarinFormstw>

⁴<https://www.facebook.com/vulcanlabtw/>

Xamarin.Forms 的開發新資訊。

若您對於 Github 範例專案有任何問題，可以在這個 Github Repository [XAMLINXamarin](https://github.com/vulcanlee/XAMLINXamarin)⁵ 上，建立一個 Issue，作者將會在這上面與您做討論。



在 XAMLINXamarin 上建立一個 Issue 操作示意圖

⁵<https://github.com/vulcanlee/XAMLINXamarin>

I 開發前的安裝、設定準備工作



特別注意事項

根據作者本身從事多年的 Xamarin.Forms 開發教學經驗，絕大多數的 Xamarin.Forms 開發新手，在開始進行 Visual Studio for Xamarin 開發工具的安裝與設定過程中，就會遇到許多問題，尤其會在某些地方卡關，造成還沒開始學習 Xamarin.Forms 的開發技術，光是建立專案與在不同作業系統平台下執行專案，都無法正常操作。因此，若您是第一次接觸 Xamarin.Forms 的讀者，建議您參考本章內容，逐步進行 Visual Studio for Xamarin 開發工具的相關安裝與設定工作。

1. Visual Studio 2017 for Xamarin 開發環境之安裝與設定

在本書第一章，我們先進行 Xamarin 開發環境的安裝與設定，總共分成 3 部分來說明

- 安裝前的準備工作

這裡將會說明當您準備要進行安裝 Visual Studio 2017 for Xamarin.Forms 開發環境之前，需要確認與準備工作有那些，這裡的內容相當重要，因為，若您的電腦環境一旦規格不符合需求，整個安裝與設定過程，將會產生問題，也會造成某些功能無法使用。

- Visual Studio 2017 安裝後的相關設定

在這裡，我們將會帶領大家，一步一步地進行 Visual Studio 2017 開發工具的安裝，並且在 Visual Studio 2017 安裝完成之後，接下來該如何進行 Xamarin 開發環境的設定。

- 確認開發環境可以進行 Xamarin.Forms 專案開發

最後，這也是最為重要的，在安裝與設定工作都完成之後，我們就需要開始進行檢查與確認，您的電腦環境是否可以真的進行 Xamarin 跨平台應用程式的開發，因此，一定要依據這裡說明內容，逐一進行檢測，確認您的開發環境都正確無誤並且可以運作正常。

1.1 安裝前的準備工作

在進行 Visual Studio 2017 開發工具安裝之前，請務必要進行底下說明的各個檢查與準備工作，否則，在進行安裝 Visual Studio 2017 與 Xamarin 開發工具的時候，可能會發

生莫名的異常問題。

這裡需要的工作有

- 確認作業系統版本
- 確認硬體 BIOS 有啟用虛擬化功能
- 同時啟用 Hyper-V 和 Windows Hypervisor 平台
- 啟用 UWP 開發人員模式

1.1.1 確認作業系統版本

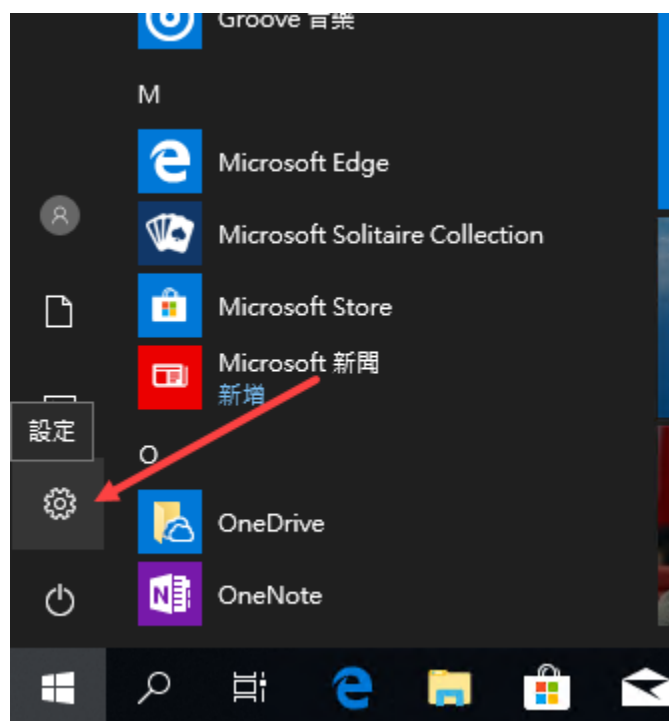
以下說明內容，我們使用的 Windows 10 專業版本作業系統，並且升級到現在最新的組建 1809 版本，原則上，建議您使用當前最新的 Windows 10 作業系統的最新版本來安裝，若您的電腦不是這個版本，請使用 Windows Update 進行更新到組建 1809 版本。



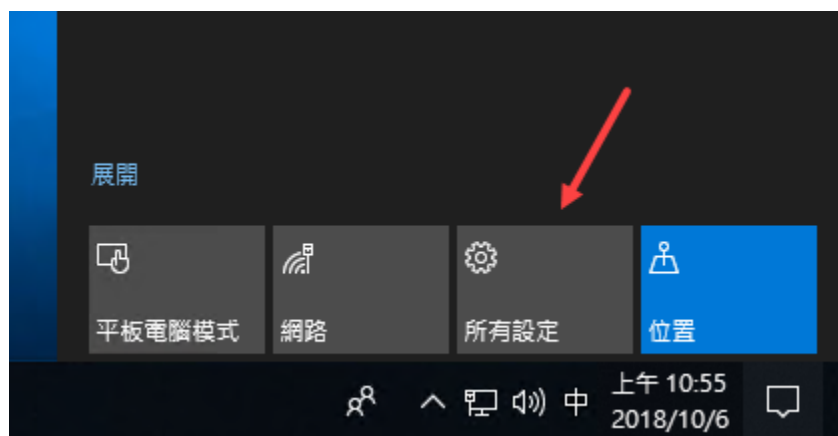
Hyper-V 硬體虛擬化技術的需求

Visual Studio 2017 有提供一套 Android 模擬器，而該模擬器需要使用 Hyper-V 硬體虛擬化技術，由於 Hyper-V 服務僅能夠在 Windows 10 專業版以上版本才能夠啟用，因此，請安裝 Windows 10 專業版作業系統到您的電腦上；若您的 Windows 10 作業系統屬於 Windows 10 Home 家用版本，您將會只有一個選擇，那就是重新安裝 Windows 10 專業版本；若您只是更新 Windows 10 Home 家用版本的授權金鑰為 Windows 10 專業版本，當您在使用 Hyper-V 服務，是會有問題發生的，因此，唯一的解決方法，那就是重新安裝 Windows 10 專業版本到您的電腦上。

欲檢查您的 Windows 10 作業系統版本，請點選左下角的視窗圖示，接著點選左下方的設定齒輪圖示，或者點選右下方通知圖示，選取齒輪圖示的設定按鈕

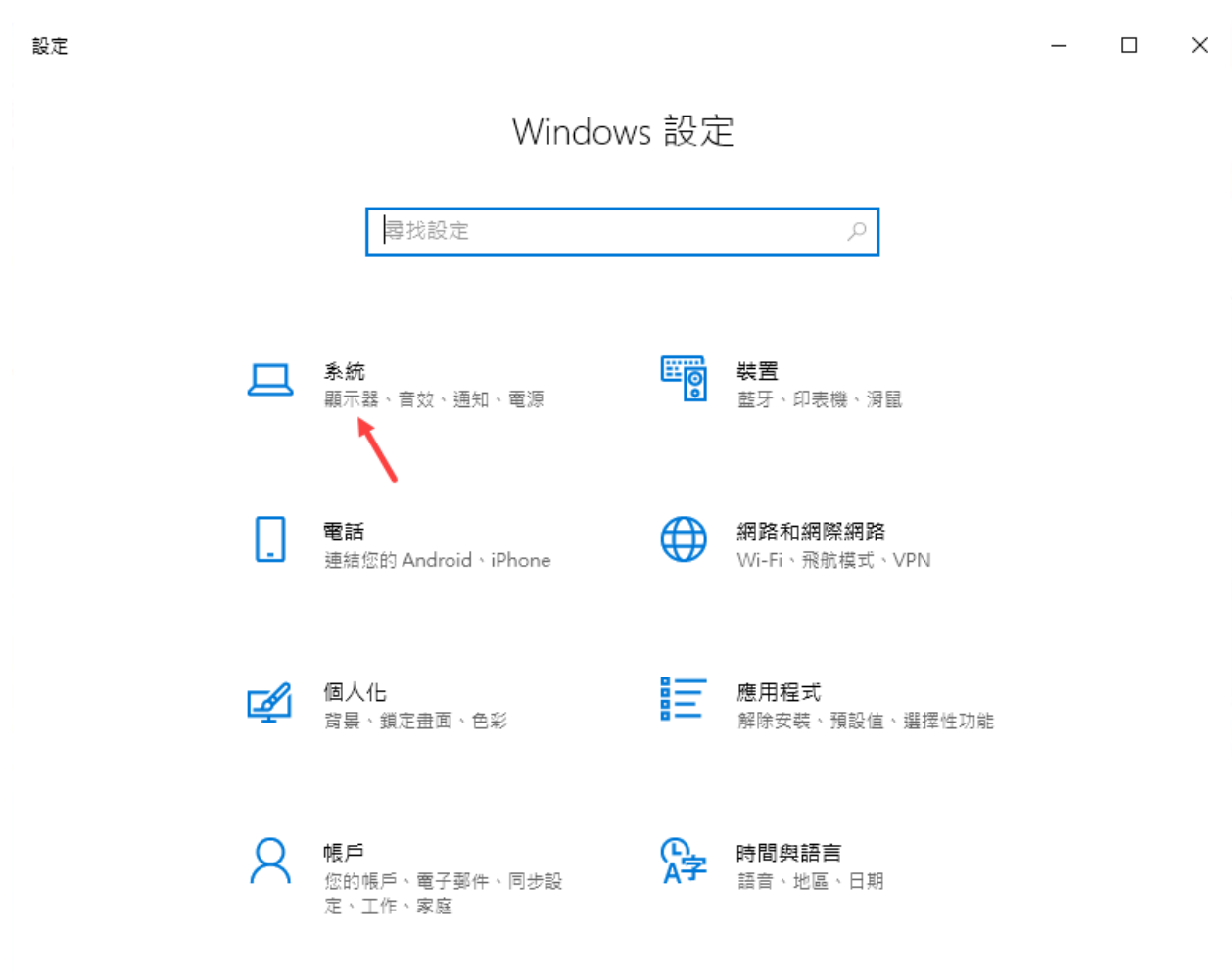


使用開始圖示來開啟 Windows 設定視窗



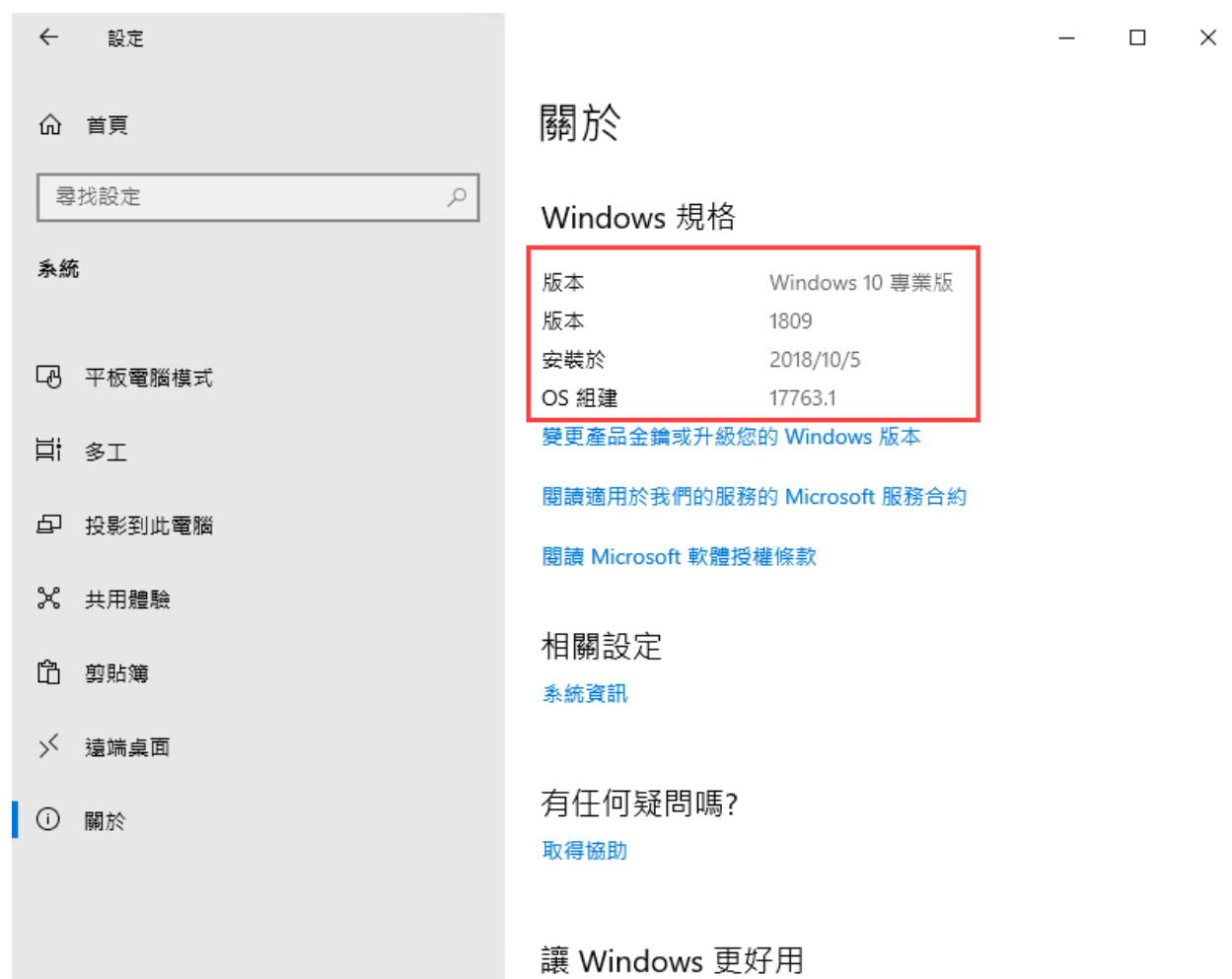
使用通知設定視窗來開啟 Windows 設定視窗

當進入到設定視窗之後，點選上方的系統圖示



在 Windows 設定窗內開啟系統功能視窗

緊接著在左邊的最下方選項，點選關於項目，您就會看到視窗右邊出現的內容，請在視窗右邊區域，使用滑鼠往下捲動，就會查看到 Windows 規格區域，確認您使用作業系統版本至少為 Windows 10 專業版與組件 1809



Windows Setting 設定

1.1.2 確認硬體 BIOS 有啟用虛擬化功能

依照個人喜好，若您想要使用 Visual Studio 2017 所提供的 **Visual Studio Emulator for Android** 模擬器來進行 Android 應用程式的開發與測試，您的電腦 CPU 需要具備支援虛擬化的功能，若您想要啟用您的電腦支援虛擬化功能，請確認您的 CPU 有支援此一功能，並且從 BIOS 啟用此功能。這個影片 [how to enable hyper-v machine in bios¹](https://www.youtube.com/watch?v=EGnv6zyLj-o) 將會展示如何設定 BIOS 可以使用虛擬化的功能。

¹<https://www.youtube.com/watch?v=EGnv6zyLj-o>

關於 Hypervisor 虛擬機器監視器的更多說明與了解，可以參考 [維基百科的 Hypervisor 解釋](#)²

原則上不同的虛擬化產品是無法混合一起使用的，例如 [Hyper-V](#)³ 無法與其他的虛擬化產品 [Intel HAXM](#)⁴ 或 [VMware](#)⁵ 等等，共同使用，您只能夠選擇其一來使用。

不過，Visual Studio 2018 的 15.8 版本之後，您是可以透過 Hyper-V 來執行 Android SDK 提供使用 Intel HAXM 製作出來的原生 x86 模擬器，關於這方面的說明，可以參考 [硬體加速以提升模擬器效能](#)⁶ 文章。

無論如何，若您的電腦硬體有支援 Hypervisor 虛擬化技術，並且您電腦的 BIOS 也有開啟該項功能，就可以在您的作業系統使用工作管理員程式，請先檢查您的作業系統是否可以使用虛擬化的功能；當您打開工作管理員程式之後，點選效能標籤頁次，在這個標籤頁次中，點選 CPU 項目，在 CPU 項目的右方，您將會看到 **模擬欄位**，若標示 **已啟用**，那就表示您電腦的 BIOS 已經開啟了虛擬化功能了。

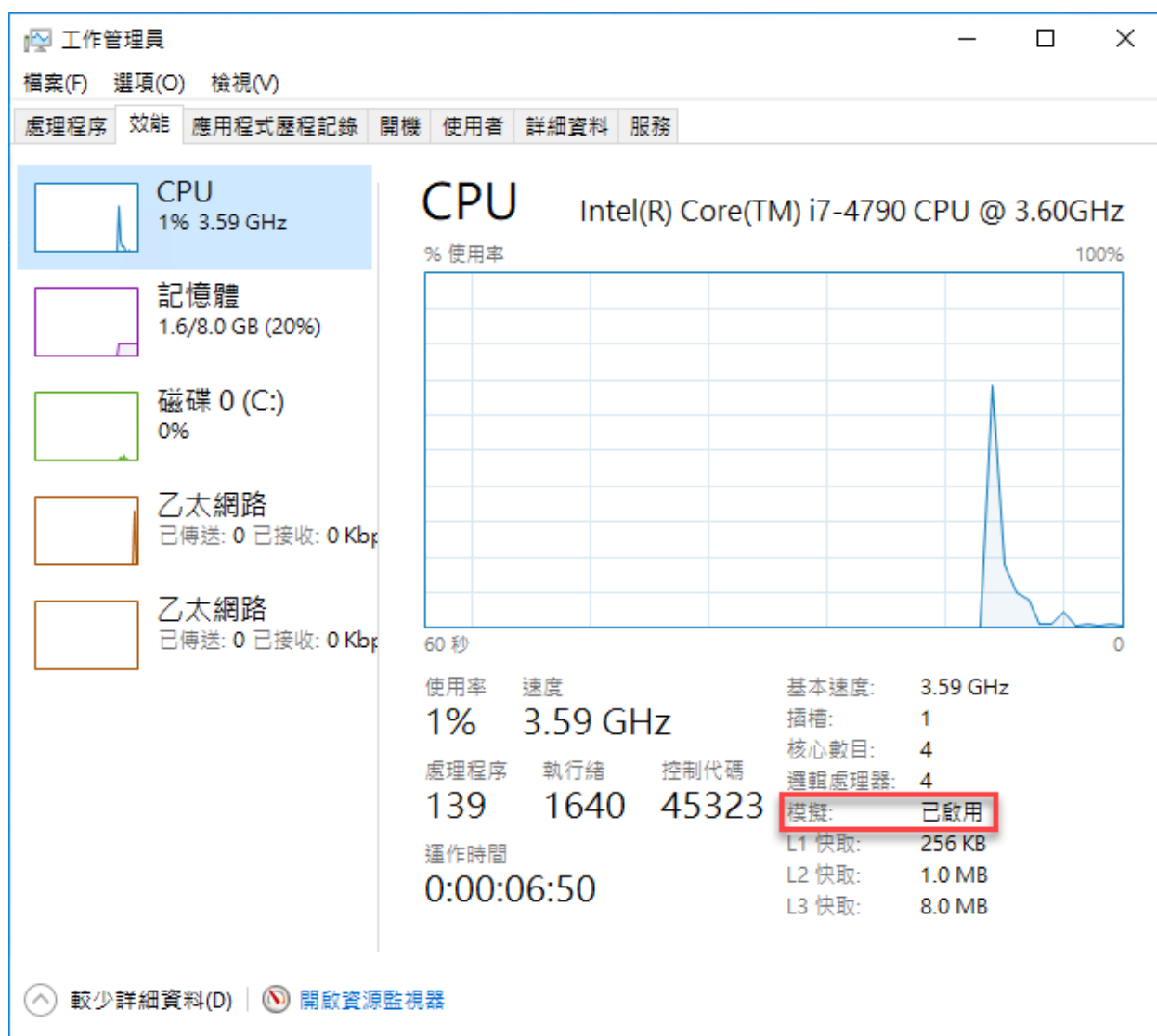
²<https://zh.wikipedia.org/wiki/Hypervisor>

³<https://docs.microsoft.com/zh-tw/virtualization/hyper-v-on-windows/reference/hyper-v-architecture>

⁴<https://software.intel.com/en-us/articles/intel-hardware-accelerated-execution-manager-intel-haxm>

⁵<https://www.vmware.com/tw.html>

⁶<https://docs.microsoft.com/zh-tw/xamarin/android/get-started/installation/android-emulator/hardware-acceleration?tabs=vswin>



使用工作管理員查看虛擬化技術是否已經開啟



我不想或者無法使用 Hyper-V 硬體虛擬化技術

作者本身通常在進行 Xamarin.Forms 專案開發與進行教學授課的時候，都是搭配著使用 Hyper-V 技術的模擬器來進行操作，我是覺得 Visual Studio 2017 本身附上的 Android 模擬器執行速度相當的快與好用。

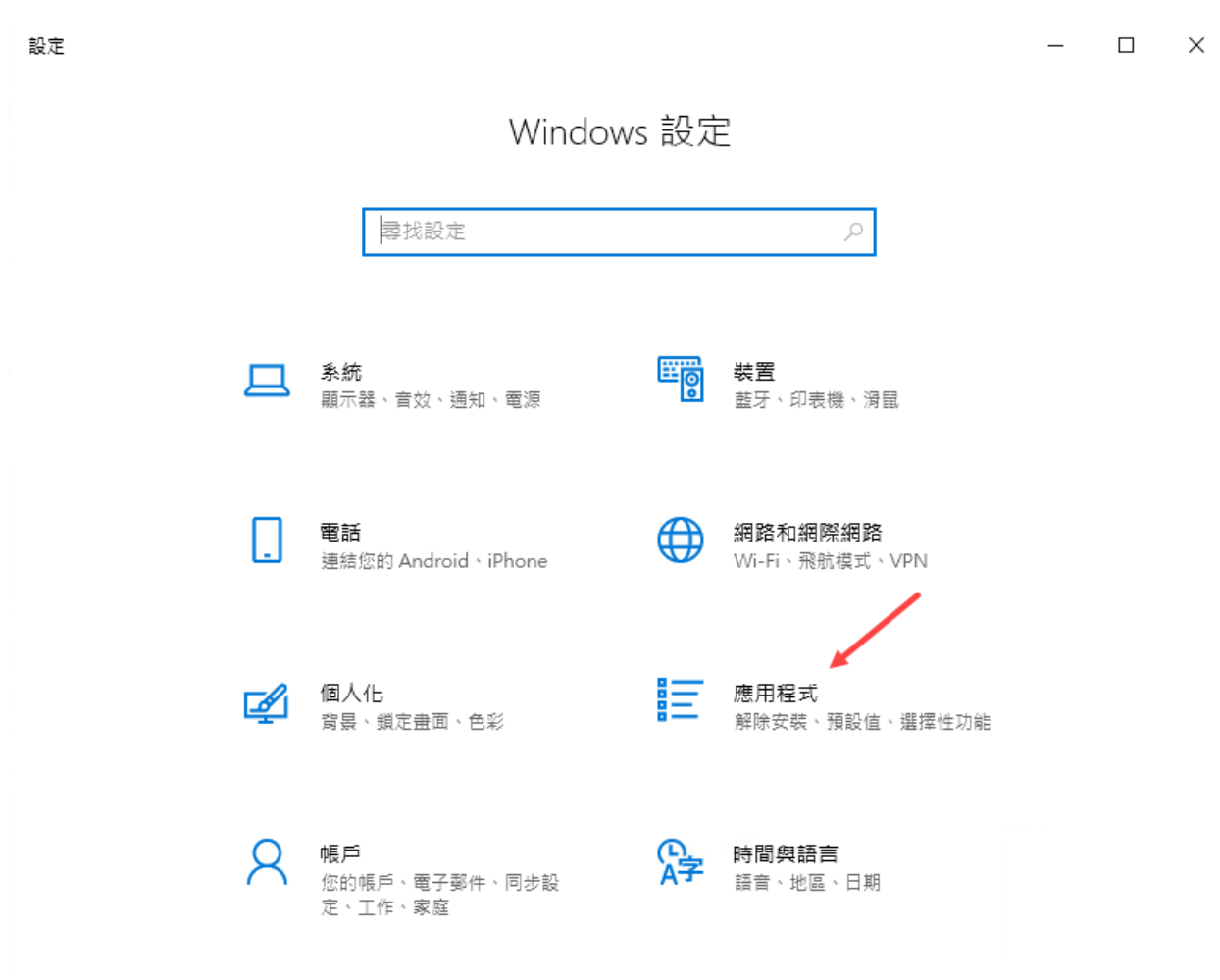
若您的電腦無法從 Windows 10 Home 版本 (因為該版本無法安裝與使用 Hyper-V) 直接重新安裝並升級到 Windows 10 專業版、又或者您的電腦 CPU 沒有支援虛擬化功能、甚至，您的電腦中已經安裝了其他 Hypervisor 的產品，這個時候，您可以選擇不需要 Hyper-V 的模擬器技術，您可以有這兩個選擇：1. 使用實體手機來進行測試與開發 2. 選擇已經安裝的虛擬化技術廠商可用的 Android 模擬器。

再次重新聲明，在進行 Xamarin.Forms 專案開發的時候，沒有說一定只能夠透過 Hyper-V 模擬器或者任何其他家的模擬器來開發，您可以使用實體手機來進行開發，這也是沒有問題。

1.1.3 同時安裝支援 Hyper-V 和 Windows Hypervisor 平台服務

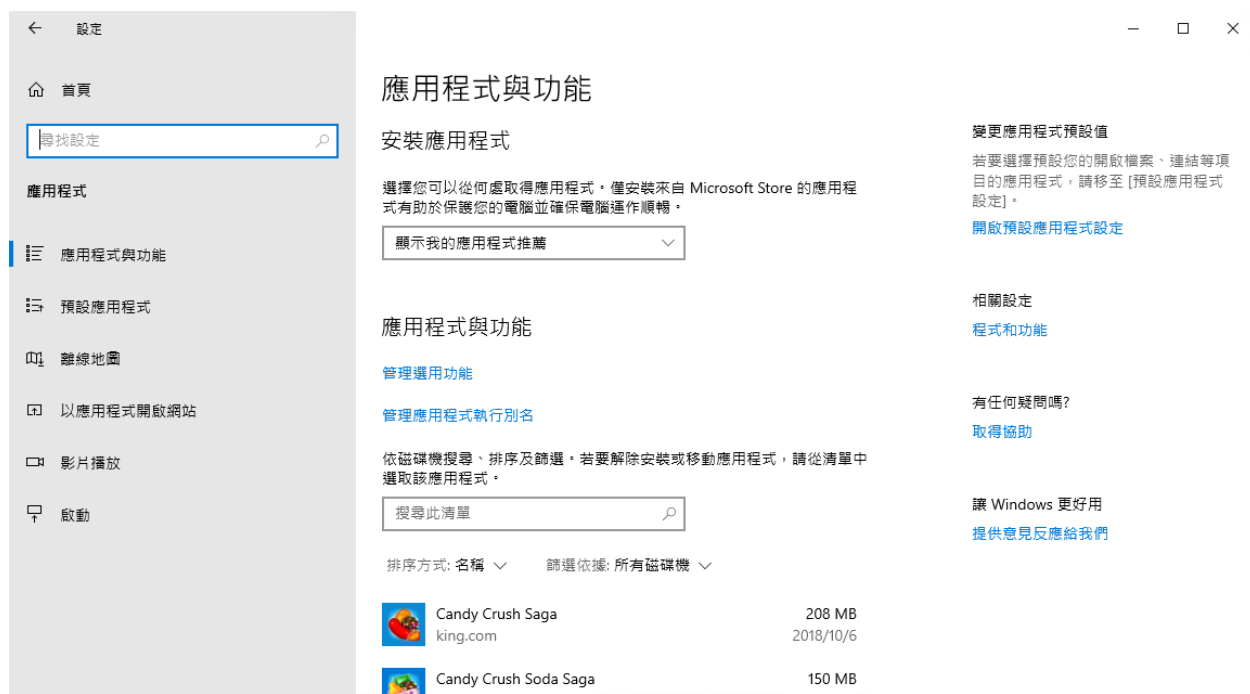
在安裝之前，我們建議您先要啟用該作業系統可以使用 Hyper-V 和 Windows Hypervisor 平台這兩項功能，請根據底下的操作步驟，安裝這兩個服務。

- 首先，依據上面的操作步驟，開啟 Windows 設定視窗 (請點選左下角的視窗圖示，接著點選左下方的設定齒輪圖示，或者點選右下方通知圖示，接著選取齒輪圖示的設定按鈕)



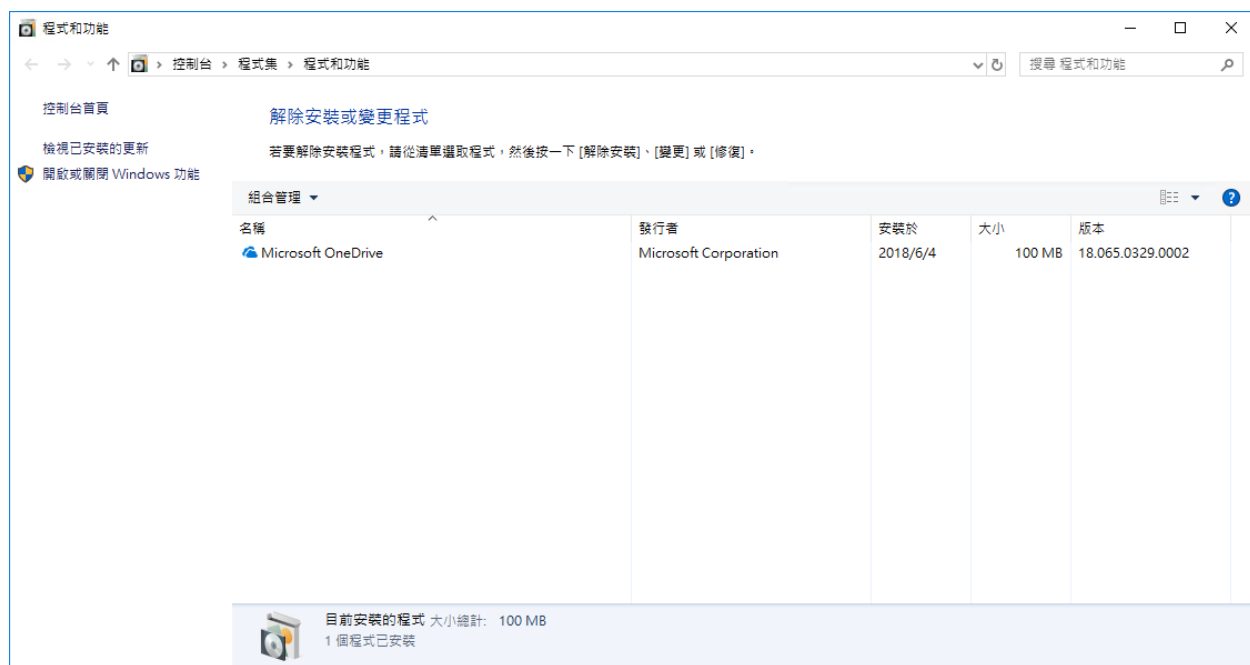
Windows 設定視窗

- 點選應用程式項目，將會出現下圖應用程式與功能視窗畫面



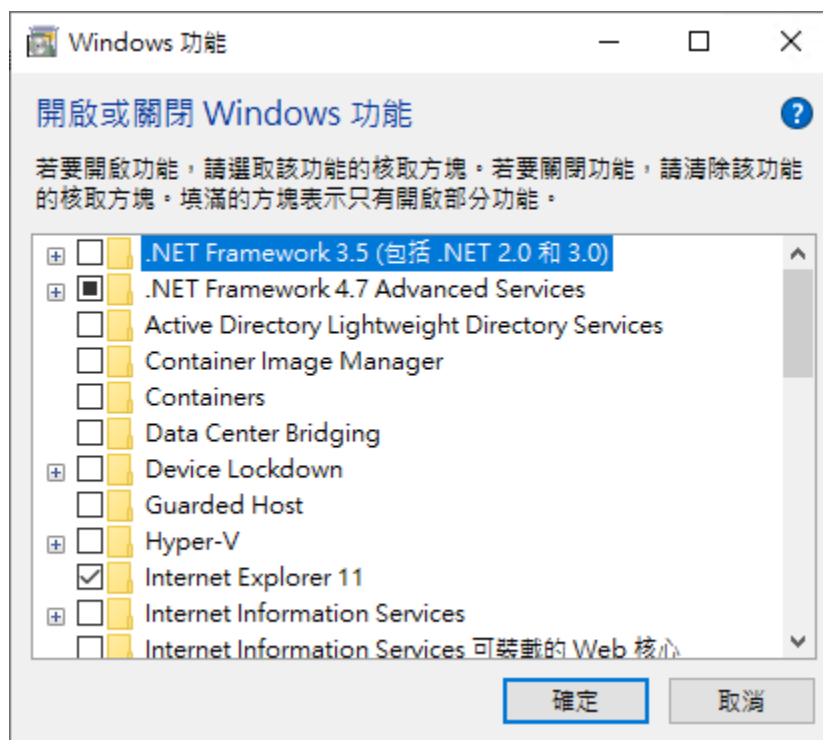
應用程式與功能視窗

- 在應用程式畫面右方，點選相關設定的程式和功能項目



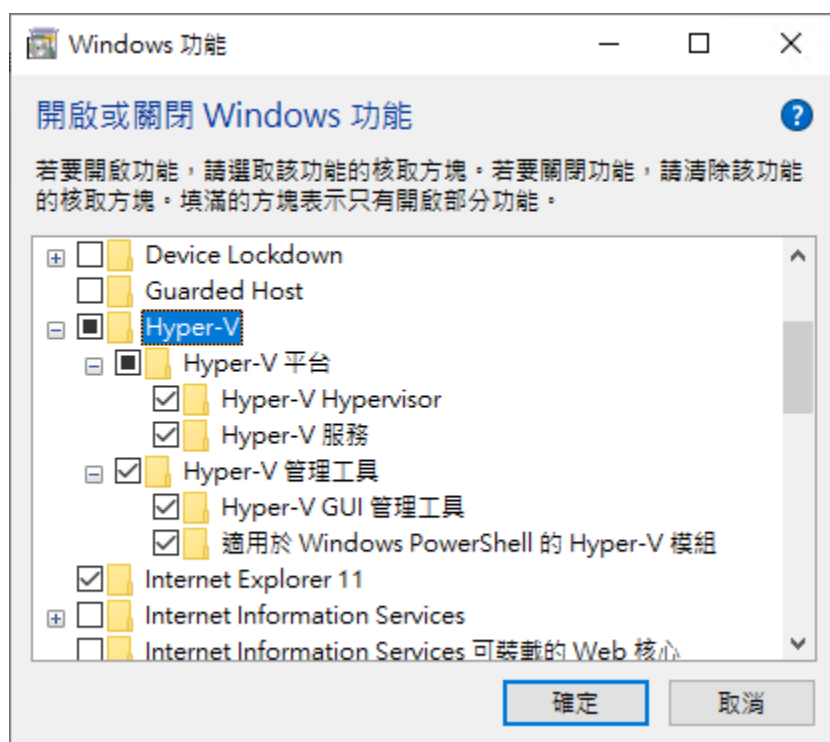
程式和功能視窗

- 當程式和功能視窗開啟之後，點選該視窗左方的開啟或關閉 Windows 功能

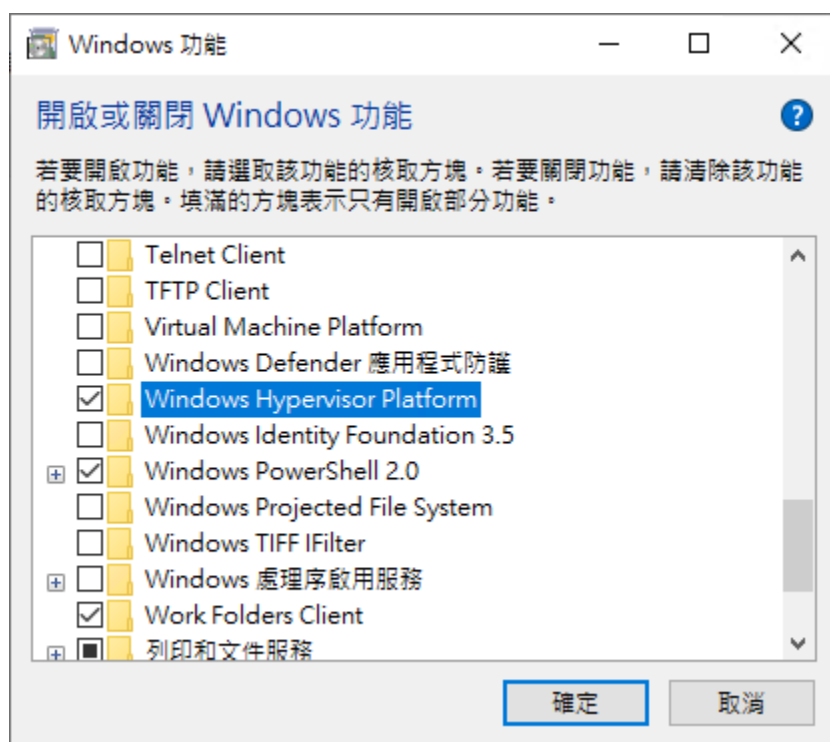


Windows 功能之設定視窗

- 現在，您將會看到 Windows 功能視窗出現在電腦上，請在下方的清單項目，找到 **Hyper-V** 與 **Windows Hypervisor 平台** 這兩個選項，並且勾選這兩個項目與其子項目，完成後請點選確定按鈕。

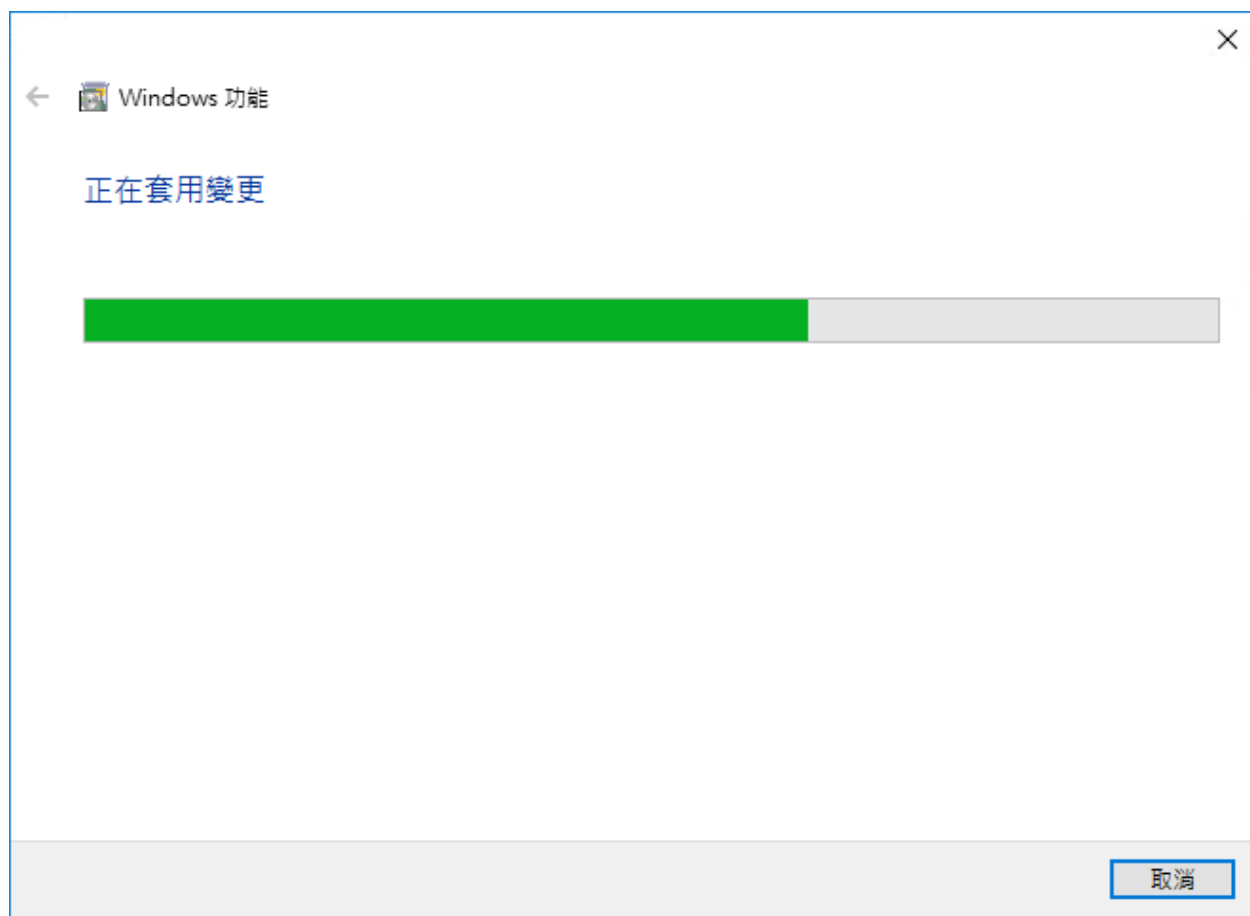


選擇安裝 Hyper-V 的所有功能



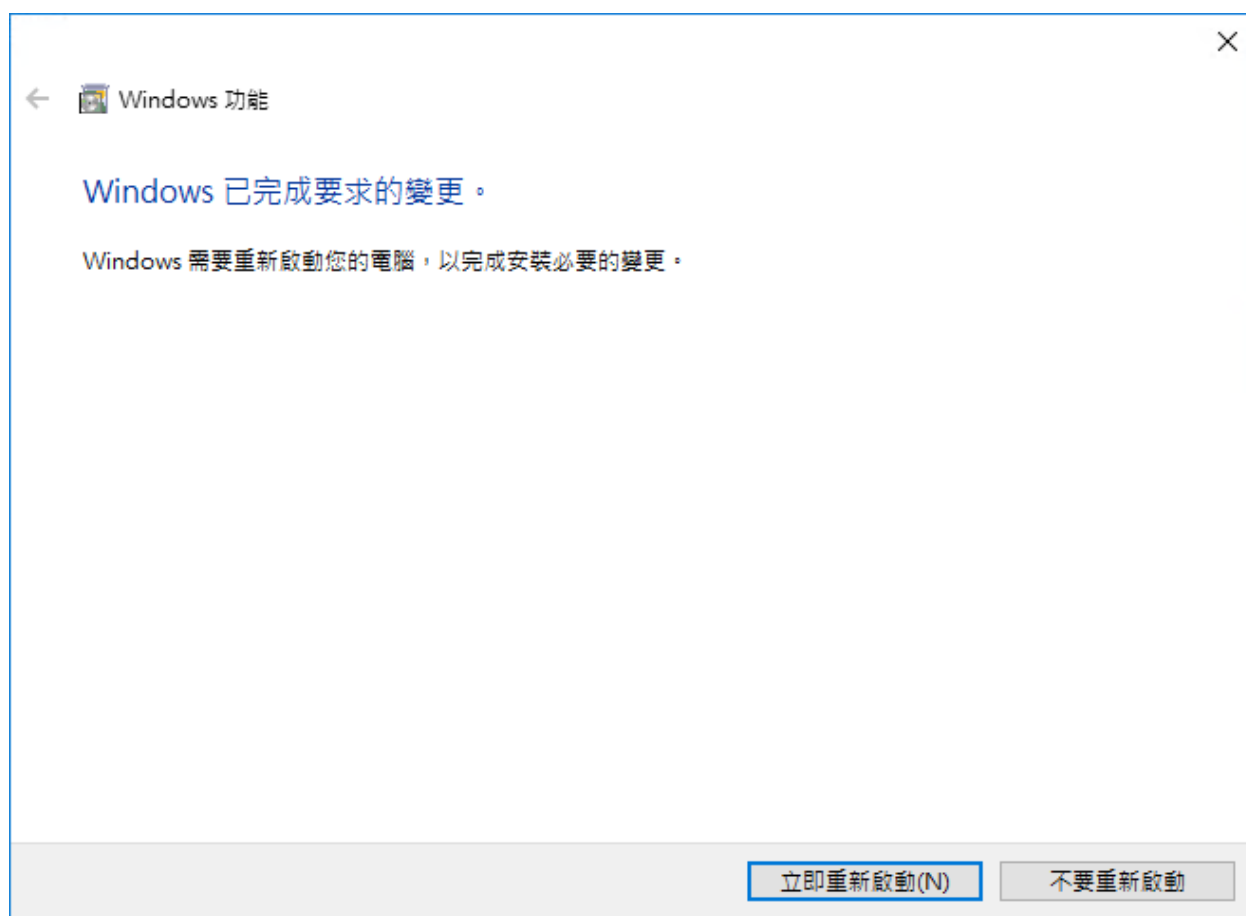
選擇安裝與啟用 Windows Hypervisor 平台

- 現在，Windows 將會開始安裝這兩個功能，請點選確定按鈕。



Windows 正在進行新增加功能安裝

- 當這兩個 Windows 功能安裝完成後，我們需要重新啟動您的作業系統，請點選立即重新啟動按鈕。



Windows 新增加功能安裝完成

1.1.4 啟用 UWP 開發人員模式

若您想要使用 Xamarin.Forms 建立出可以同時在 Android / iOS / UWP 系統下執行的跨平台 App，您就需要參考底下的步驟，先進行 UWP 開發人員模式的啟用

- 首先，還是一樣需要進入到 Windows 設定 (請點選左下角的視窗圖示，接著點選左下方的設定齒輪圖示，或者點選右下方通知圖示，接著選取齒輪圖示的設定按鈕)



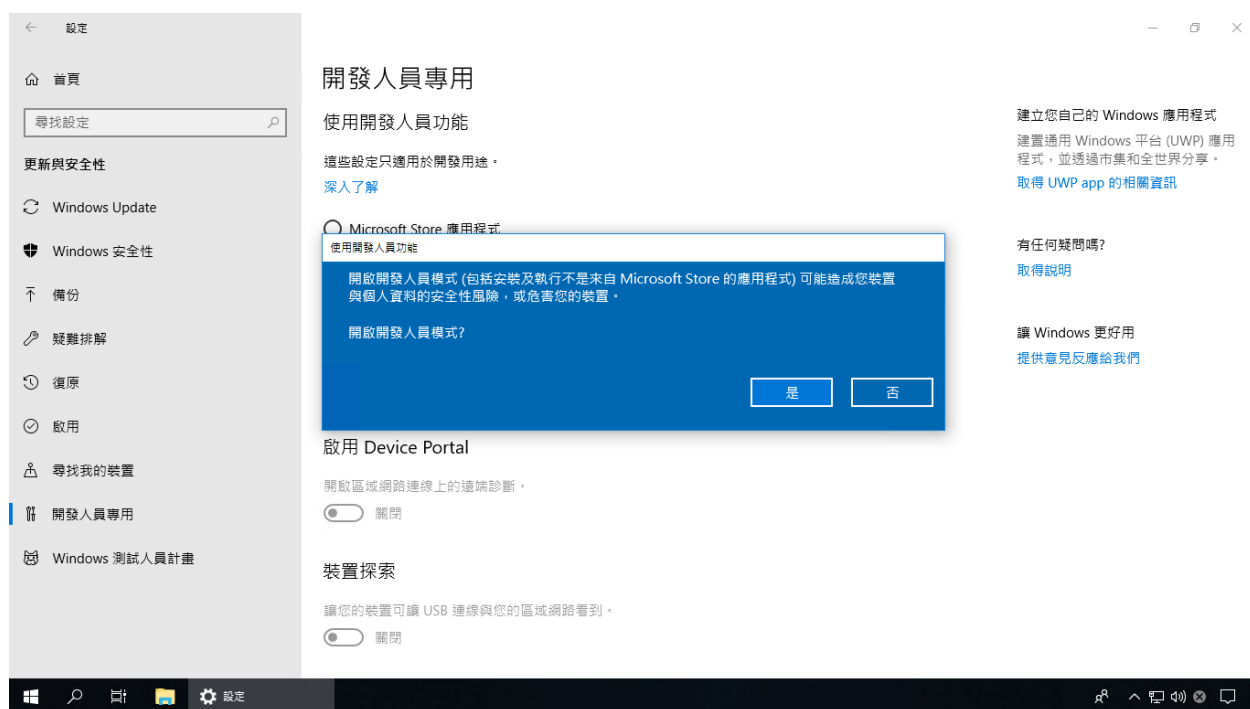
Windows Setting 設定

- 請點選更新與安全性選項，並且點選更新與安全性視窗左方的開發人員專用選項



開發人員專用設定視窗

- 請點選右方內容的開發人員模式選項，現在顯示出使用開發人員功能對話窗，並且說明：開啟開發人員模式 (包括安裝及執行不是來自於 Microsoft Store 的應用程式) 可能造成您裝置與個人資料的安全性風險或危害您的裝置。開啟開發人員模式？請點選是按鈕。



開啟開發人員模式

- 若開發人員模式已經開啟之後，我們可以看到如同下圖畫面，在開發人員模式選下的下方，會顯示出：已安裝「開發人員模式」套件，現已啟用適用於桌面的遠端工具訊息。



已安裝「開發人員模式」套件

1.2 安裝 Visual Studio 2017

若您已經完成 Visual Studio 2017 安裝前的準備工作，現在，我們需要開始進行 Visual Studio 2017 的安裝作業。

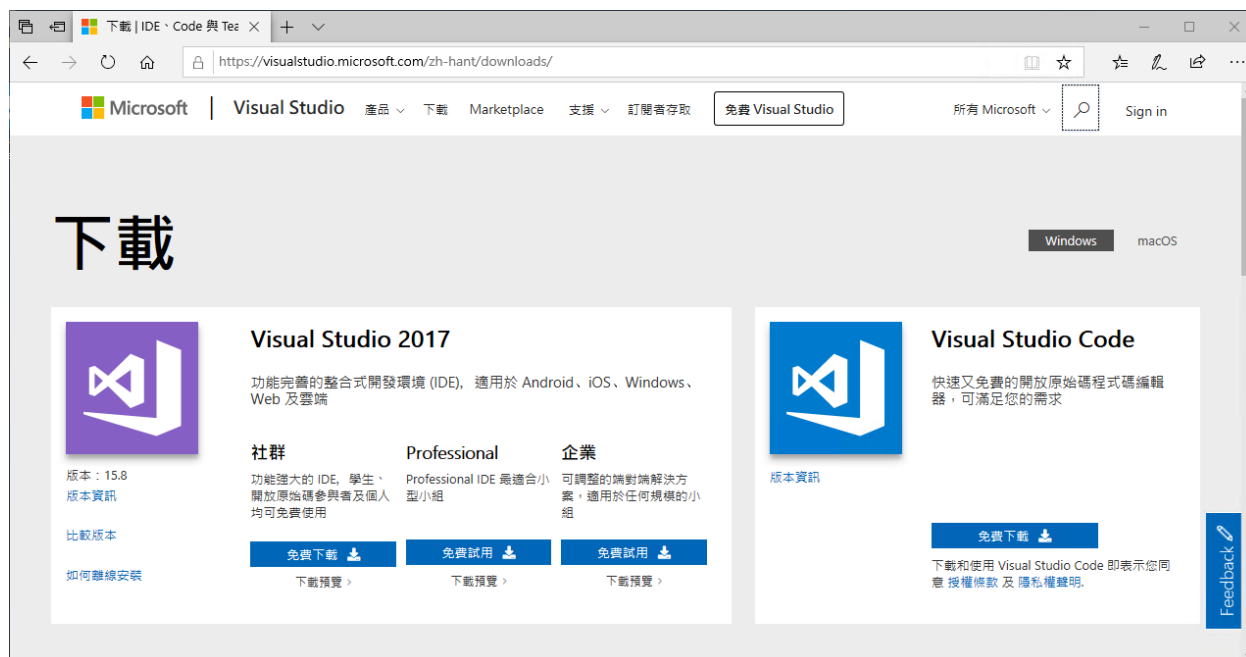
在這裡，我們將會使用 Visual Studio 2017 Community 版本作為安裝說明，不過，任何一種 Visual Studio 2017 的版本，皆可以進行 Xamarin Cross-Platform 跨平台專案的開發，關於 Visual Studio 2017 的不同版本的比較，可以參考 [比較 Visual Studio 2017 IDE⁷](#)

更多關於 Visual Studio 2017 的安裝，可以參考 [安裝 Visual Studio 2017⁸](#)

⁷<https://www.visualstudio.com/zh-hant/vs/compare/>

⁸<https://docs.microsoft.com/zh-tw/visualstudio/install/install-visual-studio>

- 請用瀏覽器打開 [Visual Studio 2017 安裝下載網頁](https://visualstudio.microsoft.com/zh-hant/downloads/)⁹



Visual Studio 2017 安裝下載網頁

- 找到 Visual Studio 2017 下方的社群項目，點選其下方的免費下載按鈕，點選這個按鈕連結，當安裝檔案下載完成後，請點選執行按鈕，開始進行 Visual Studio 2017 社群版本的安裝；當安裝程式下載完成之後，請點選螢幕下方的執行按鈕，開始進行 Visual Studio 2017 社群版的安裝 (如同下面節圖所示)。

⁹<https://www.visualstudio.com/zh-hant/downloads>



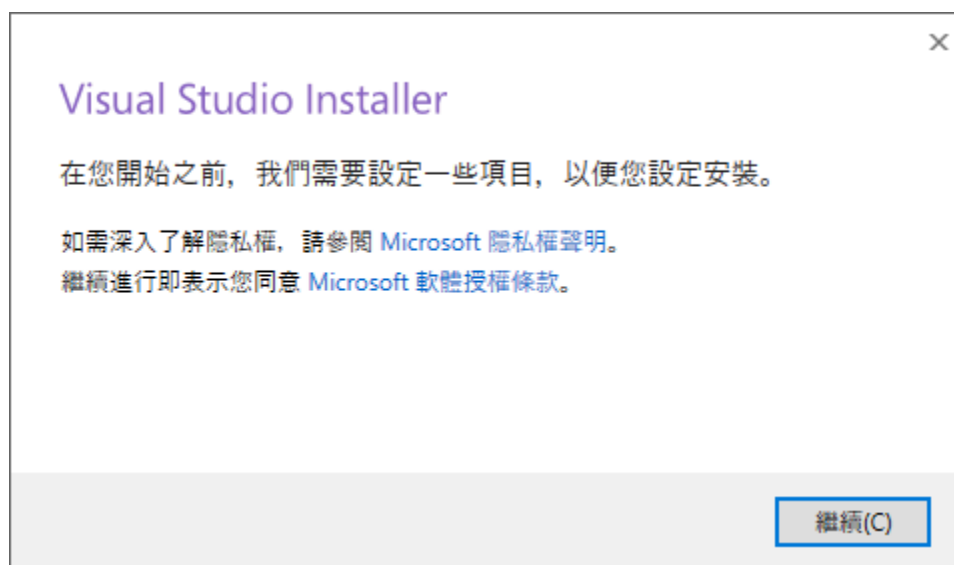
Visual Studio 2017 安裝程式下載完成

- 此時，螢幕會出現使用者帳戶控制對話窗，此對話窗詢問：您是否要允許此 App 變更您的裝置？這個時候，請點選是按鈕。



使用者帳戶控制對話窗

- 現在，看到的對話窗是 Visual Studio Installer，請在該對話窗右下方，點擊繼續按鈕。如此，Visual Studio Install 安裝程式，便會開始下載最新的安裝程式檔案到您的電腦。



Visual Studio Installer 開始安裝前的確認對話窗

- 我們將會看到 Visual Studio Community 2017 - 15.8.6 安裝程式啟動了。

15.8.6 是我寫這篇文章的時候，最新 Visual Studio Community 的版本編號，原則上，您每次進行 Visual Studio 2017 安裝作業的時候，您將會使用最新的 Visual Studio 2017 Installer 來安裝到最新的版本，因此，您將需要比 15.8.6 更新的版本編號出現在您的電腦畫面中。



正在安裝 Visual Studio Community 2017 - 15.8.6 視窗

- 請在工作負載標籤頁次中，至少需要勾選底下項目，才能夠完成 Xamarin 開發工具的安裝
 - 通用 Windows 開發平台
 - 使用 .NET 進行行動開發

當您勾選這兩個工作負載項目之後，在 Visual Studio Community 2017 Installer 視窗的最右方，將會出現準備要安裝的相關元件清單說明



勾選要安裝通用 Windows 開發平台工作負載



勾選要安裝使用 .NET 進行行動開發工作負載

- 請點選個別元件標籤頁次，您將會看到底下的個別元件已經被勾選了



Visual Studio 2017 Installer 個別元件

安裝不同的 Visual Studio 2017 版本，可能會看到不通的套件選項項目內容

- .NET
 - * .NET Framework 4.5 目標套件
 - * .NET Framework 4.6.1 目標套件
 - * .NET Framework 4.6.1 SDK
 - * .NET Native
 - * .NET 可攜式程式庫目標套件
- Code 工具
 - * ClickOnce 發行
 - * Developer Analytics Tools
 - * NuGet 套件管理員
- Compilers, build tools, and runtimes
 - * (無)
- SDK、程式庫和架構
 - * Android SDK 安裝程式 (API 層級 27)

- * Java SE 開發套件 (8.0.1120.15)
- * TypeScript 3.0 SDK
- * Windows 10 SDK (10.0.17134.0)
- SDKs, Libraries, and Frameworks
 - * (無)
- 偵錯和測試
 - * .NET 分析工具
 - * JavaScript 診斷
- 模擬器
 - * Google Android Emulator (API 層級 27)
 - * Intel Hardware Accelerated Execute Manager (HAXM) (全域安裝)
- 程式碼工具
 - * 靜態分析工具
- 編譯器、建置工具和執行階段
 - * C# 與 Visual Basic Roslyn 編譯程式
 - * MSBuild
- 編譯器、建置工具與執行階段
 - * (無)
- 遊戲與圖形
 - * (無)
- 遊戲開發套件
 - * 影像與 3D 模型編輯器
- 開發活動
 - * C# 與 Visual Basic
 - * F# 語言支援
 - * JavaScript 與 TypeScript 語言支援
 - * Xamarin
 - * Xamarin Remoted Simulator

- * SQL Server 的 CLR 資料類型
- 雲端、資料庫和伺服器
 - * 資料來源與服務參考
- 為了要讓你的 Xamarin 開發環境可以正常運作與符合日常開發需求，所以，請依照底下清單，比對您電腦上的個別元件勾選項目，確認底下的項目都已經有勾選到 (額外需要安裝的項目將會以**粗體**文字標示)。
 - .NET
 - * .NET Framework 4.5 目標套件
 - * .NET Framework 4.6.1 目標套件
 - * .NET Framework 4.6.1 SDK
 - * .NET Native
 - * .NET 可攜式程式庫目標套件
 - Code 工具
 - * ClickOnce 發行
 - * Developer Analytics Tools
 - * **Git for Windows**
 - * NuGet 套件管理員
 - * **Visual Studio 的 GitHub 擴充功能**
 - Compilers, build tools, and runtimes
 - * (無)
 - SDK、程式庫和架構
 - * **Android NDK (R138)**
 - * Android SDK 安裝程式 (API 層級 27)
 - * Java SE 開發套件 (8.0.1120.15)
 - * TypeScript 3.0 SDK
 - * Windows 10 SDK (10.0.17134.0)
 - SDKs, Libraries, and Frameworks
 - * (無)
 - 偵錯和測試

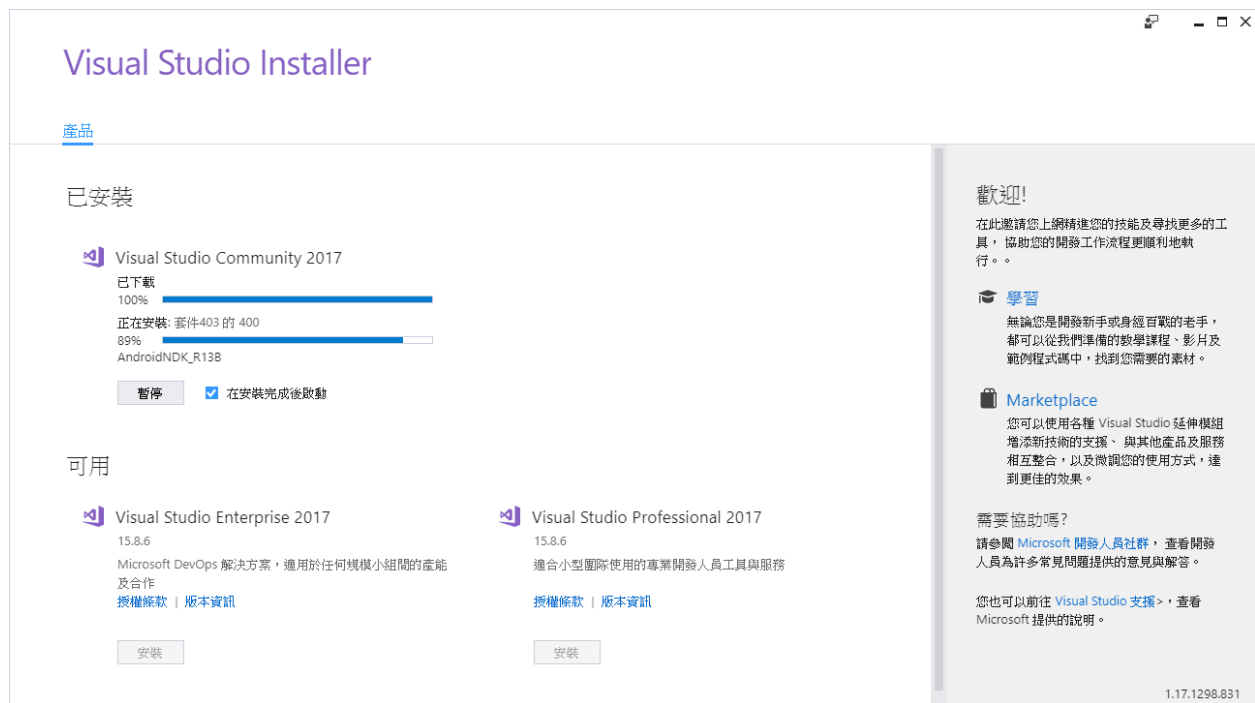
- * .NET 分析工具
- * JavaScript 診斷
- 模擬器
 - * Google Android Emulator (API 層級 27)
 - * Intel Hardware Accelerated Execute Manager (HAXM) (全域安裝)
 - * **Visual Studio 的 Android 模擬器**
- 程式碼工具
 - * 靜態分析工具
- 編譯器、建置工具和執行階段
 - * C# 與 Visual Basic Roslyn 編譯程式
 - * MSBuild
- 編譯器、建置工具與執行階段
 - * (無)
- 遊戲與圖形
 - * (無)
- 遊戲開發套件
 - * 影像與 3D 模型編輯器
- 開發活動
 - * C# 與 Visual Basic
 - * F# 語言支援
 - * JavaScript 與 TypeScript 語言支援
 - * Xamarin
 - * Xamarin Remoted Simulator
 - * SQL Server 的 CLR 資料類型
- 雲端、資料庫和伺服器
 - * 資料來源與服務參考



Visual Studio 2017 Installer 安裝詳細資料

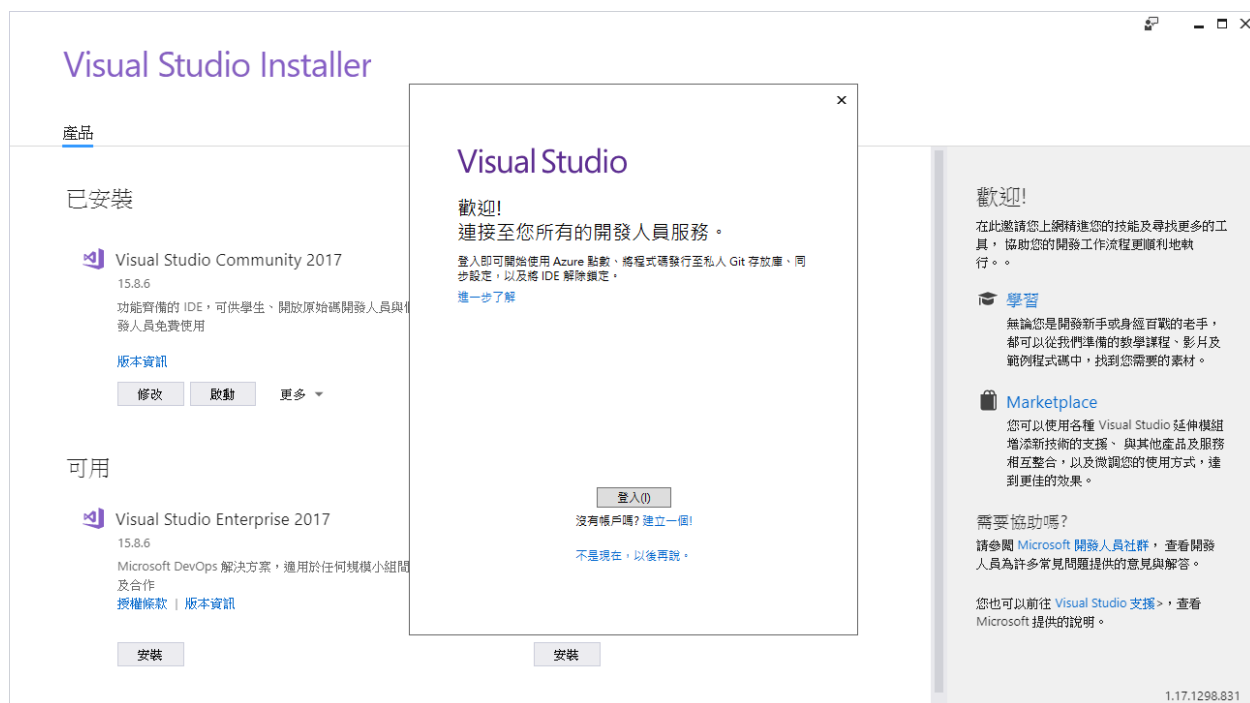
- 若確認無誤，請點選 Visual Studio Community 2017 Installer 程式畫面右下角的安裝按鈕，開始進行 Visual Studio Community 2017 的程式安裝。

根據 Visual Studio Installer 的預估，這樣的安裝過程，大約需要用到 32.73GB 的磁碟空間。



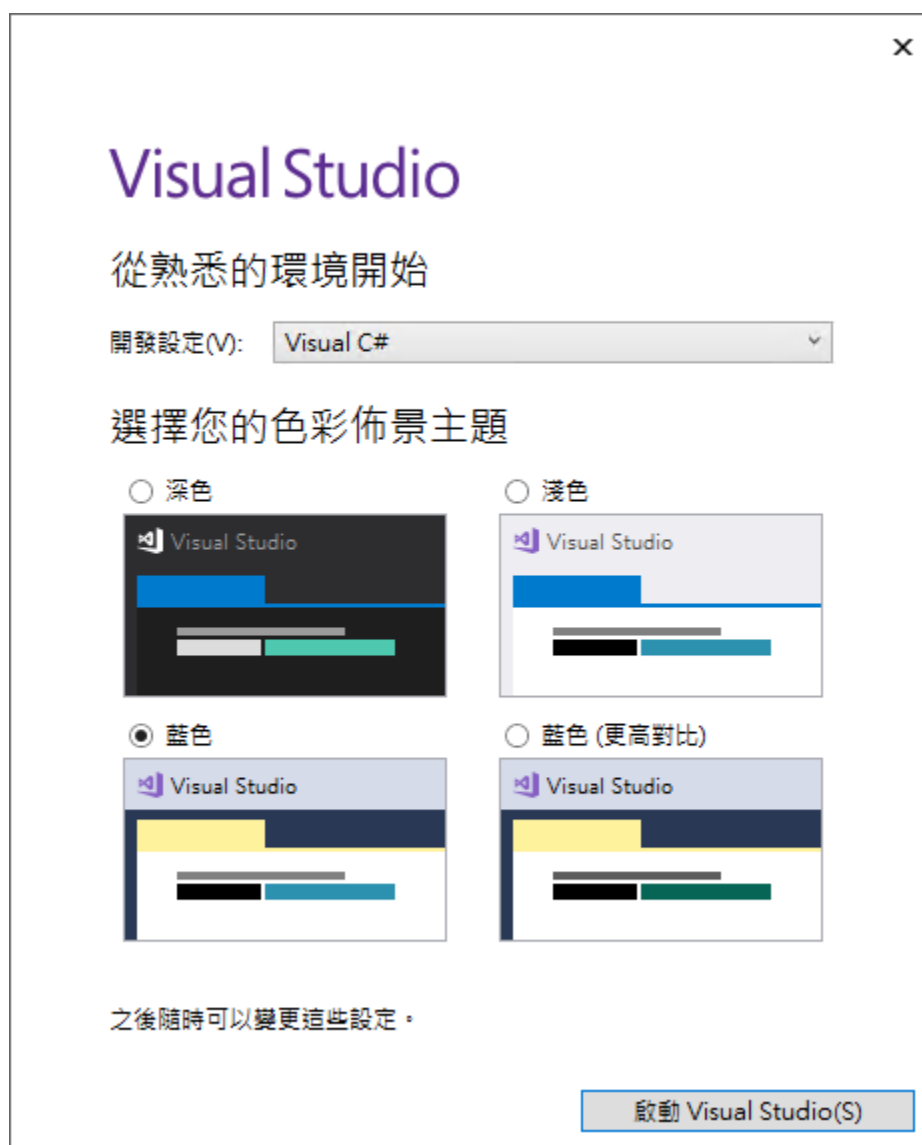
Visual Studio 2017 Installer

- 當安裝作業完成之後，將會看到如下圖的畫面，您可以選擇登入功能，或者點選不是現在，以後再說，在這裡，我們點選後者，也就是點選 **不是現在，以後再說**。



Visual Studio 2017 登入畫面

- 現在，我們需要設定開發設定條件，請在開發設定下拉選單中，選擇 Visual C#，完成後，點選啟動 Visual Studio 按鈕。



Visual Studio 2017 開發設定與主題佈景

- 底下螢幕截圖是 Visual Studio 2017 啟動後的畫面



Visual Studio 2017 第一次啟動畫面

1.3 Visual Studio 2017 安裝後的相關設定

當啟動完成 Visual Studio 2017 之後，我們需要開始進行 Android SDK 的設定與更新作業，確認相關機碼是否有正確產生出來。

在這個階段，我們將會進行底下的相關設定

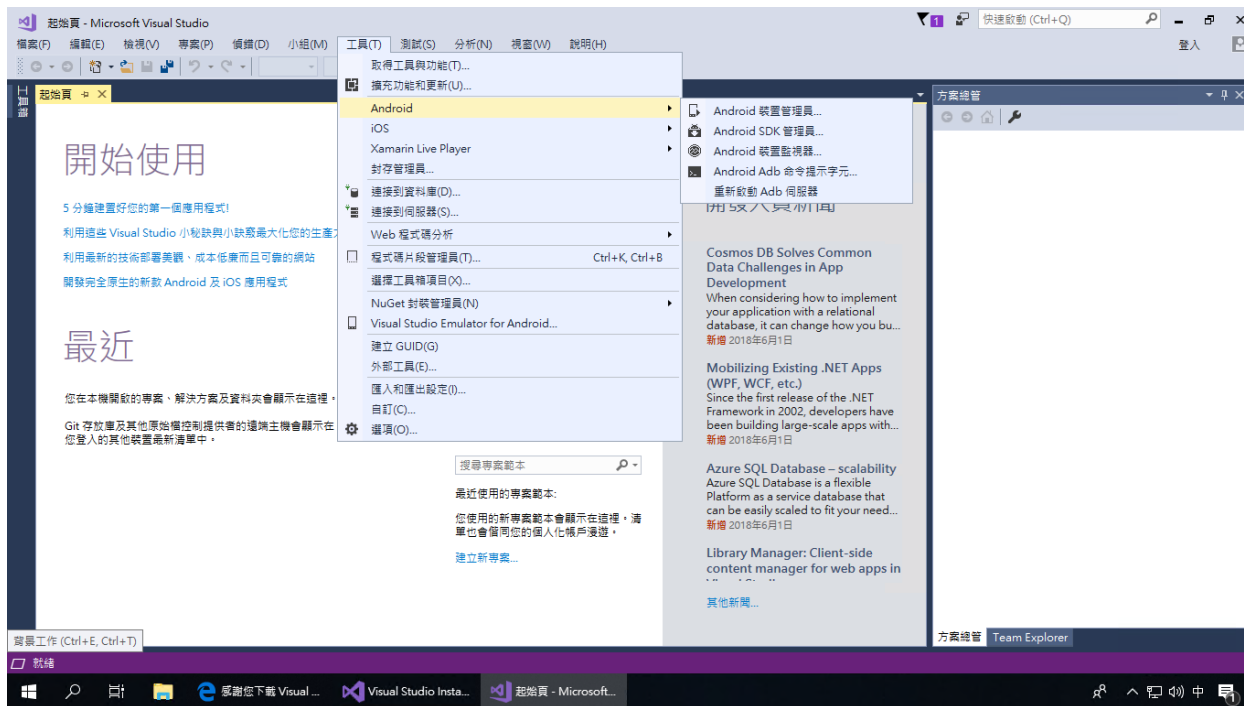
- 更新 Android SDK
- 檢查 Android SDK 機碼是否存在

1.3.1 更新 Android SDK

養成時常確保您的 Android SDK 有更新到最新版本的好習慣，我們需要進行 Android SDK 的設定與更新，確保我們的 Xamarin.Android 專案可以正常順利的建置與執行

Android App。更多關於 Android SDK 的說明，可以參考 [設定 Xamarin.Android 的 Android SDK](#)¹⁰

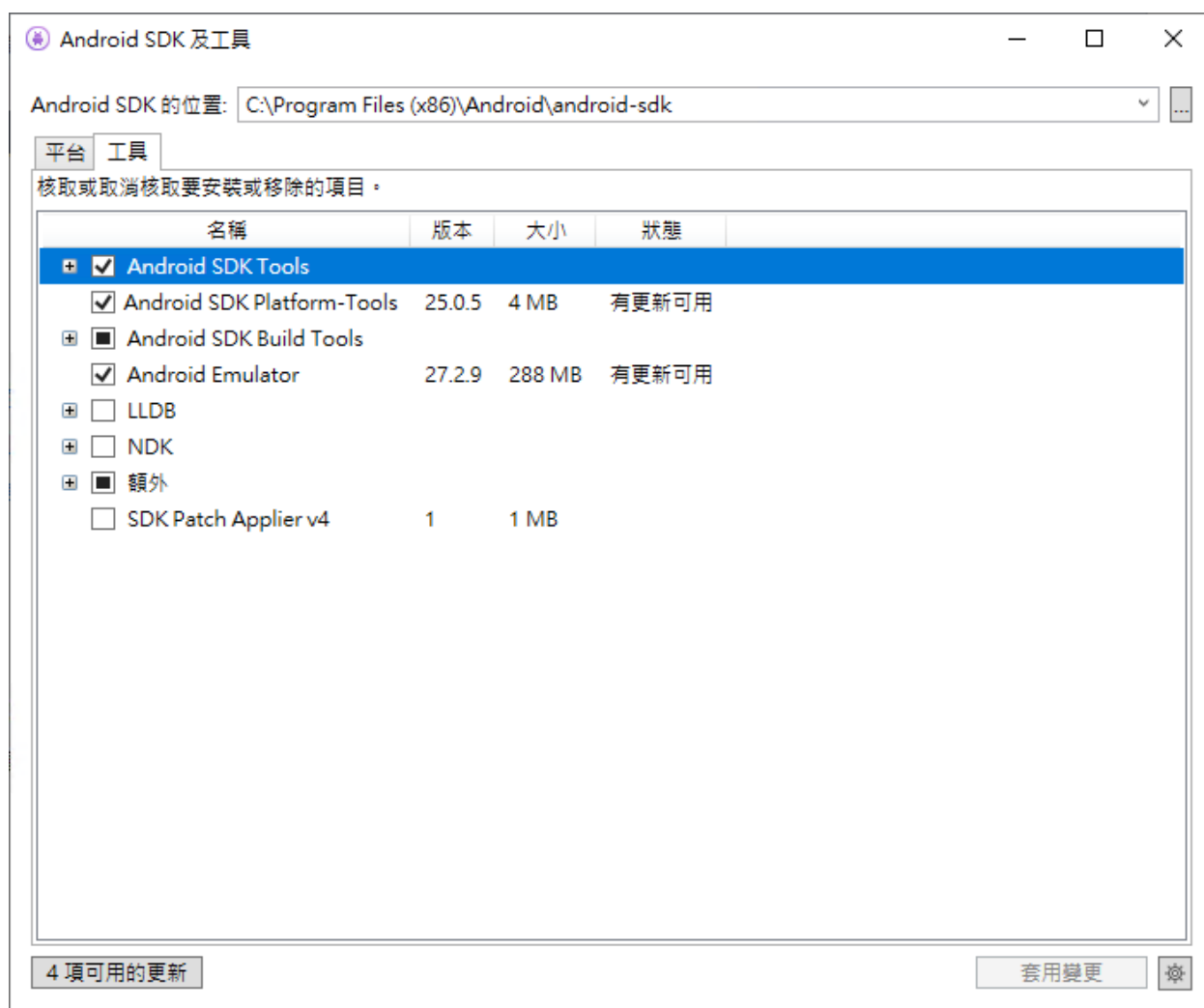
- 請在 Visual Studio 程式中，點選功能表工具 > Android > Android SDK 管理員



啟動 Android SDK 管理員

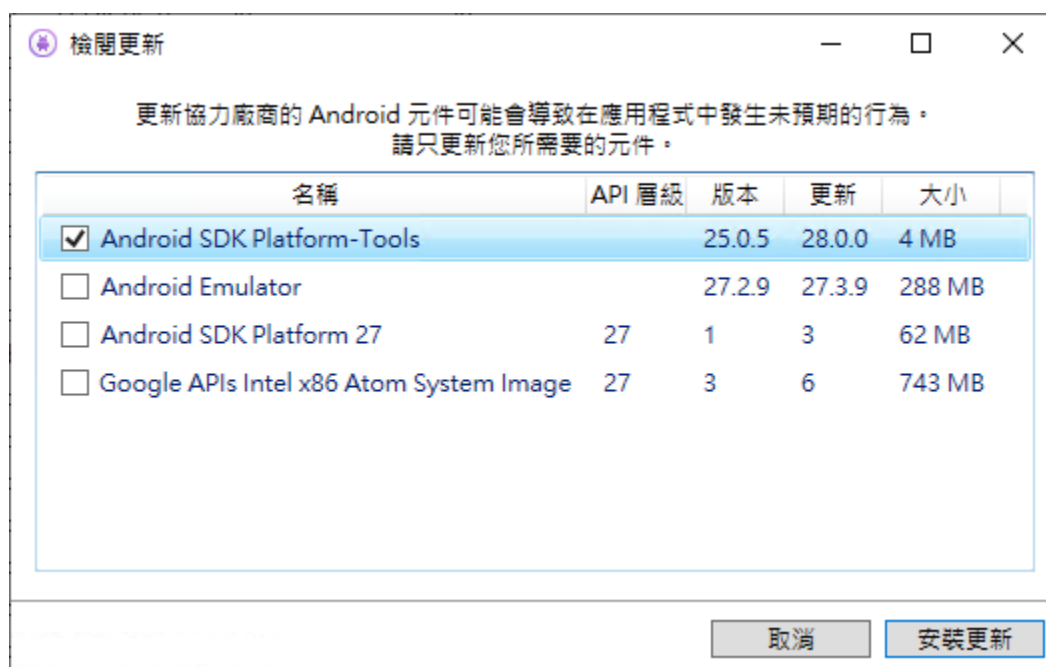
- 當 Android SDK 及工具視窗出現之後，請先點選工具標籤頁次，此時，您會發現到 Android SDK Platform-Tools 的版本是 25.0.5，並且在該項目的狀態欄位，該視窗的左下方，顯示的是有更新可用。

¹⁰<https://docs.microsoft.com/zh-tw/xamarin/android/get-started/installation/android-sdk?tabs=vswin>



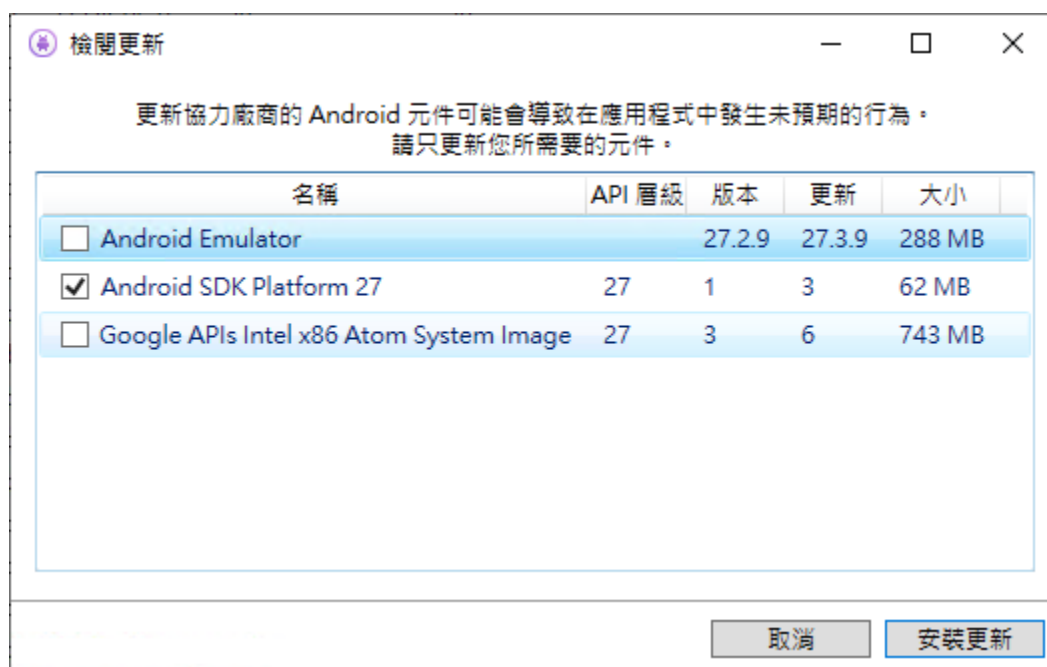
Android SDK 及工具的對話窗

- 我們需要先點選左下角的 4 項可用的更新按鈕，當出現了檢閱更新對話窗的時候，我們只要選取 Android SDK Platform-Tools 這個項目要進行更新，接著，點選安裝更新按鈕。



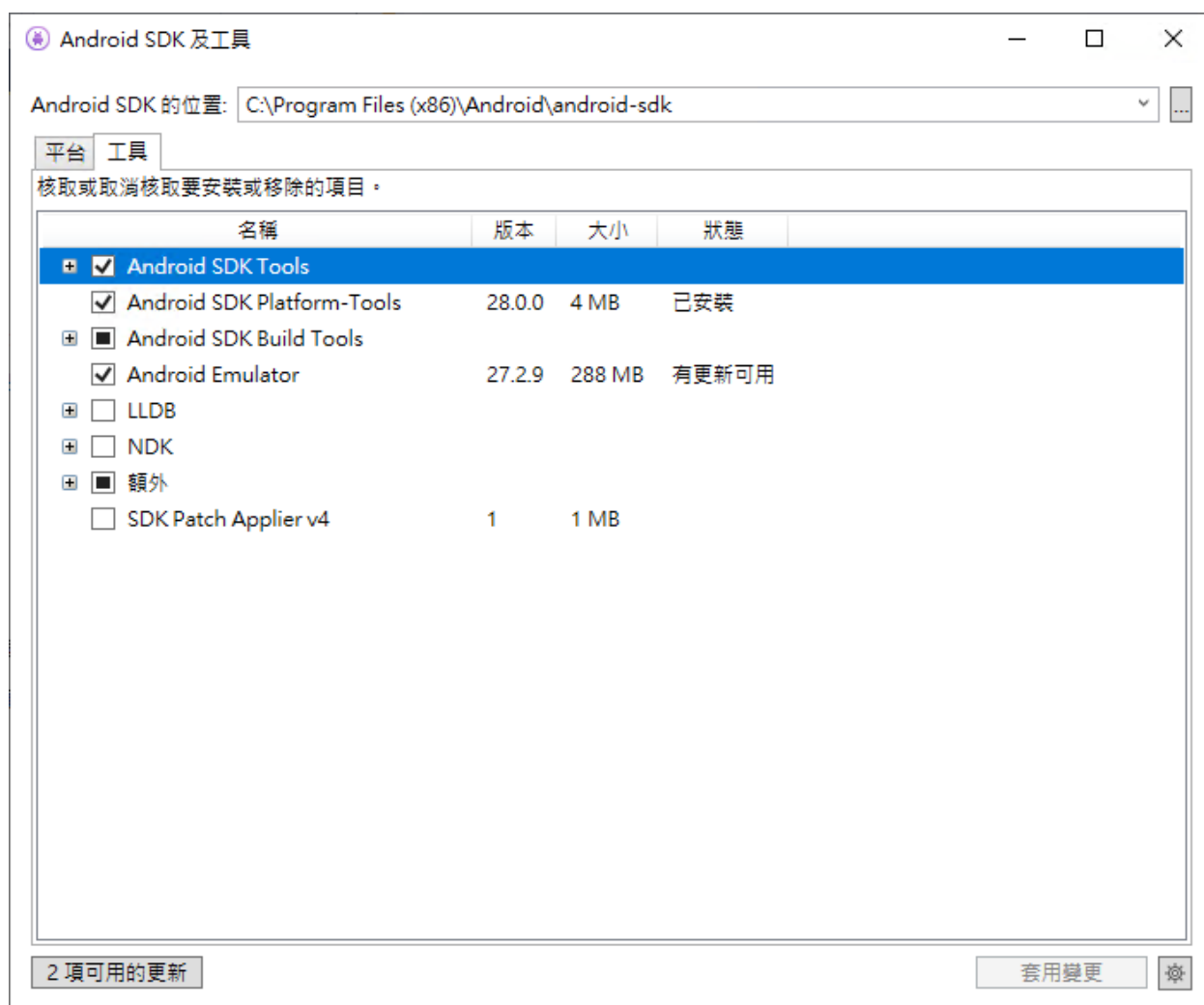
檢閱更新對話窗

- 現在，只剩下 3 個可用更新了，接著，我們需要先點選左下角的 3 項可用的更新按鈕，當出現了檢閱更新對話窗的時候，我們選取 Android SDK Platform 27 這個項目要進行更新，接著，點選安裝更新按鈕。



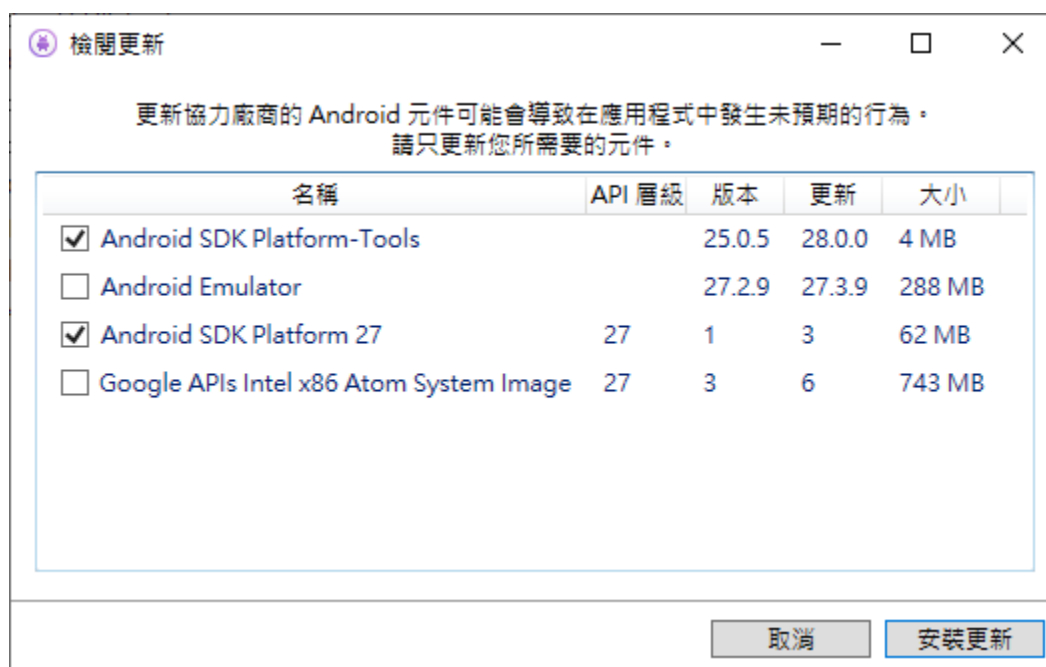
檢閱更新對話窗

- 現在，只剩下2項可用的更新了



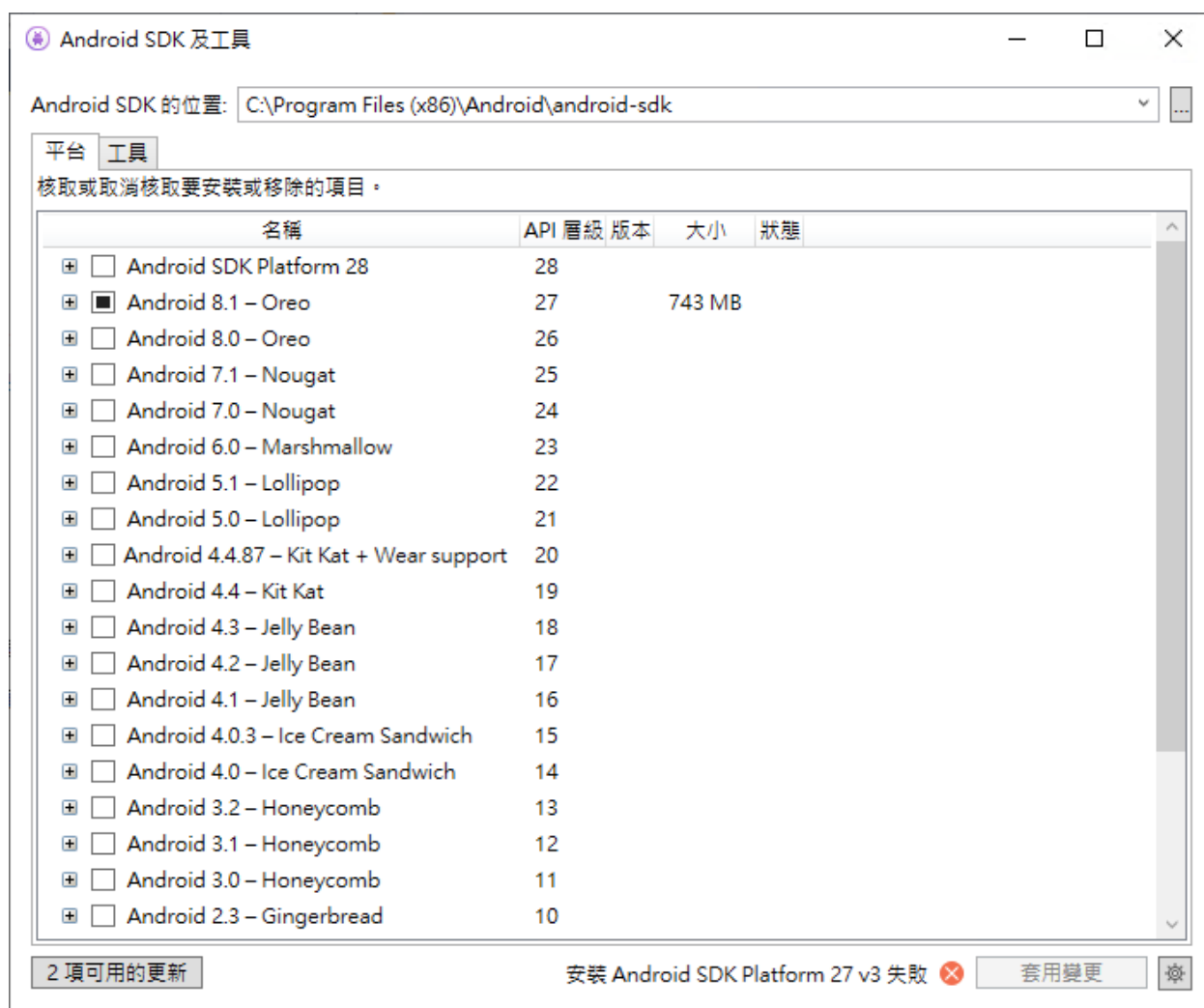
Android SDK 及工具剩下2項可用的更新

- 我們需要使用兩階段的更新方式，這是因為當我們點選左下角的 4 項可用的更新按鈕，當出現了檢閱更新對話窗的時候，我們確定選取 Android SDK Platform-Tools 與 Android SDK Platform 27 這兩個項目要進行更新，接著，點選安裝更新按鈕。

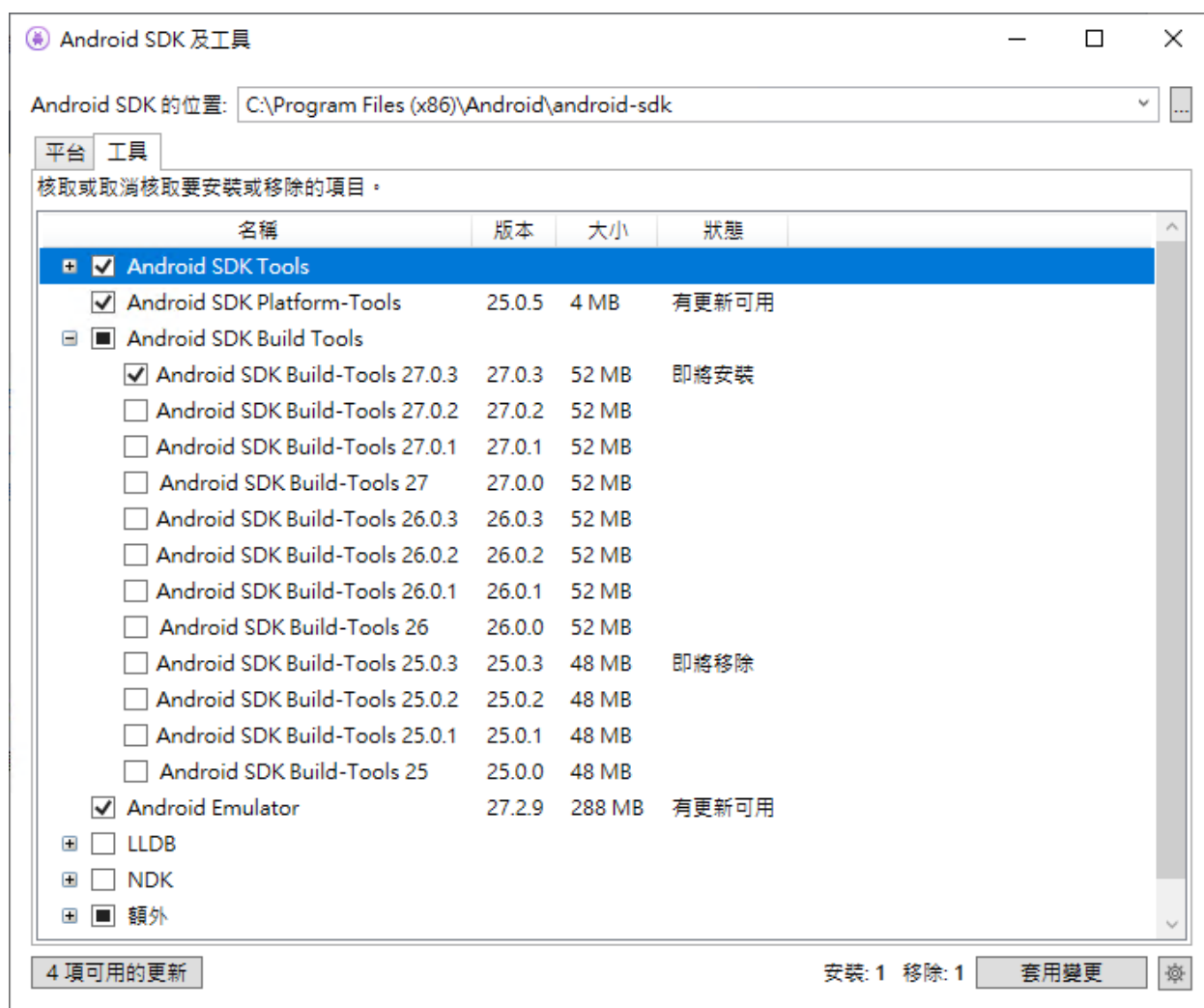


檢閱更新對話窗

可是，當整個更新作業完成後，就會發現在 Android SDK 及工具對話窗的下方，出現了安裝 Android SDK Platform 27 v3 失敗的錯誤訊息，雖然在對話窗的最下方出現了有4項可用的更新，可是此時，我們把 Android SDK 及工具對話窗關閉起來，並且重新開啟 Android SDK 管理員，此時，您會發現到還是有4項可用的更新，剛剛的更新動作並沒有成功。

**Android SDK 及工具**

- 接下來，還是在工具標籤頁次，展開 Android SDK Build Tools 節點，取消勾選 Android SDK Build-Tools 25.0.3，並且勾選 Android SDK Build-Tools 27.0.3
展開額外節點，勾選 Android Support Repository，最後，點選右下方套用變更按鈕

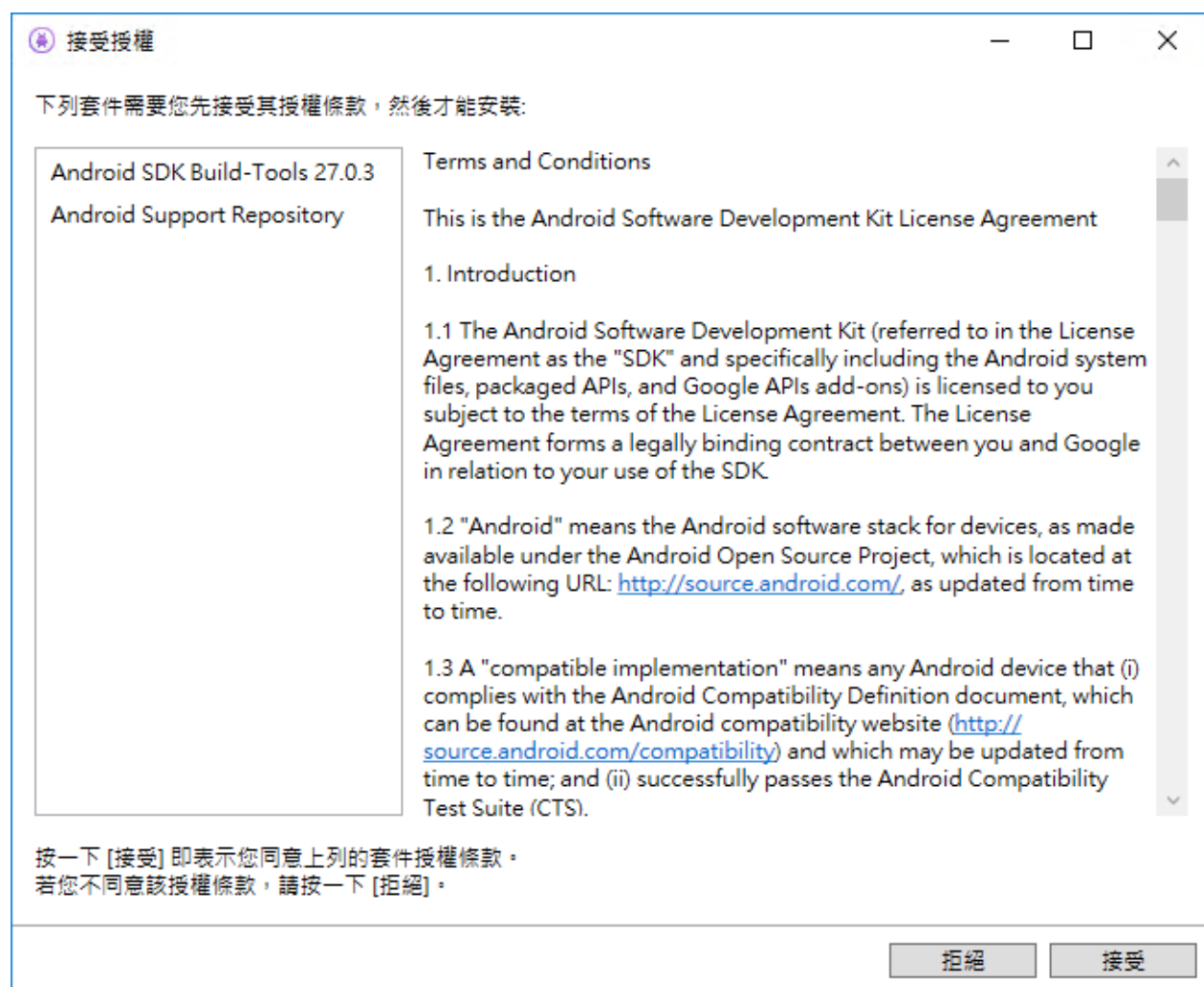


勾選 Android SDK Build-Tools 27.0.3

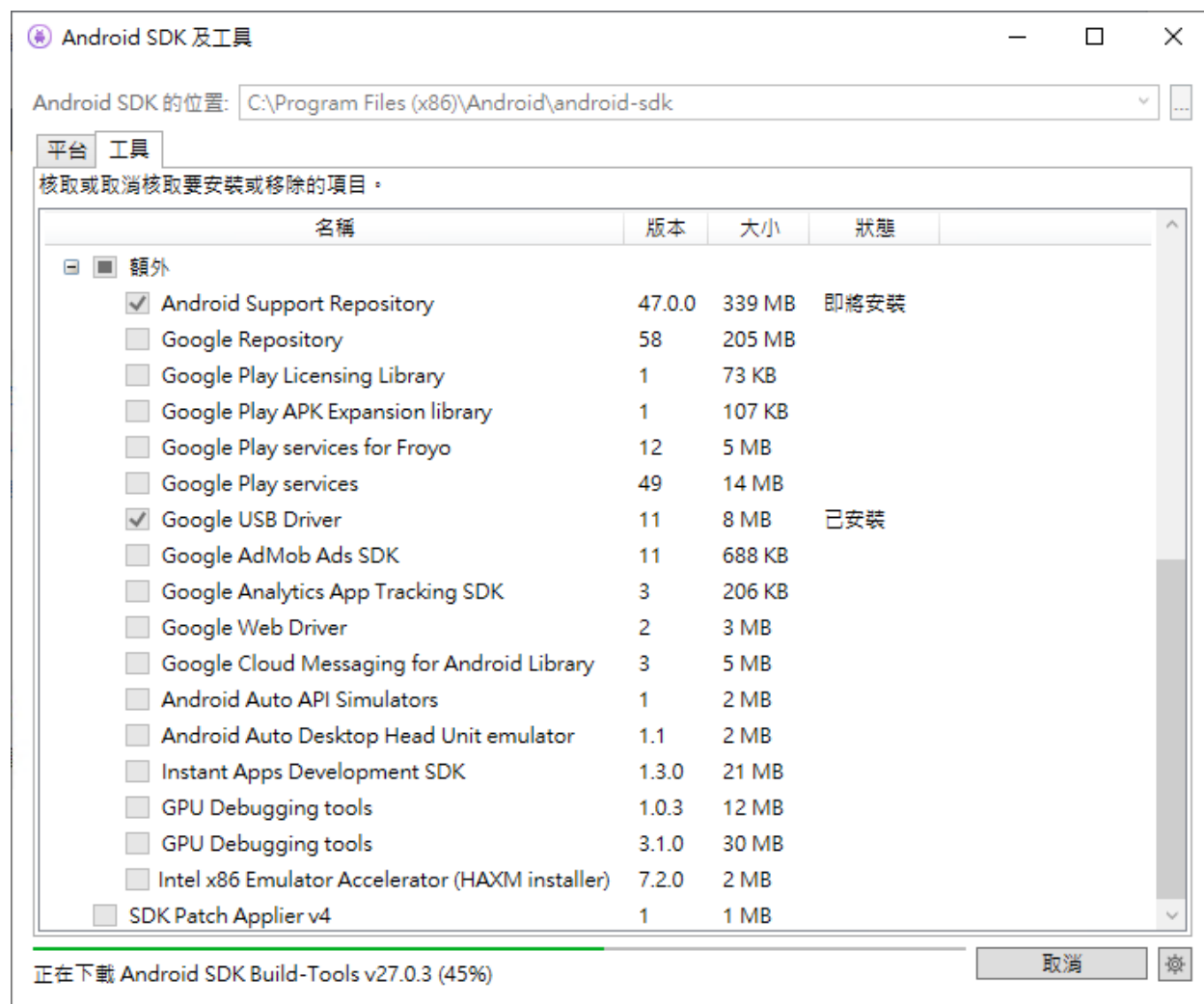


勾選 Android Support Repository

- 出現接受授權對話窗後，直接點選接受按鈕



接受授權對話窗

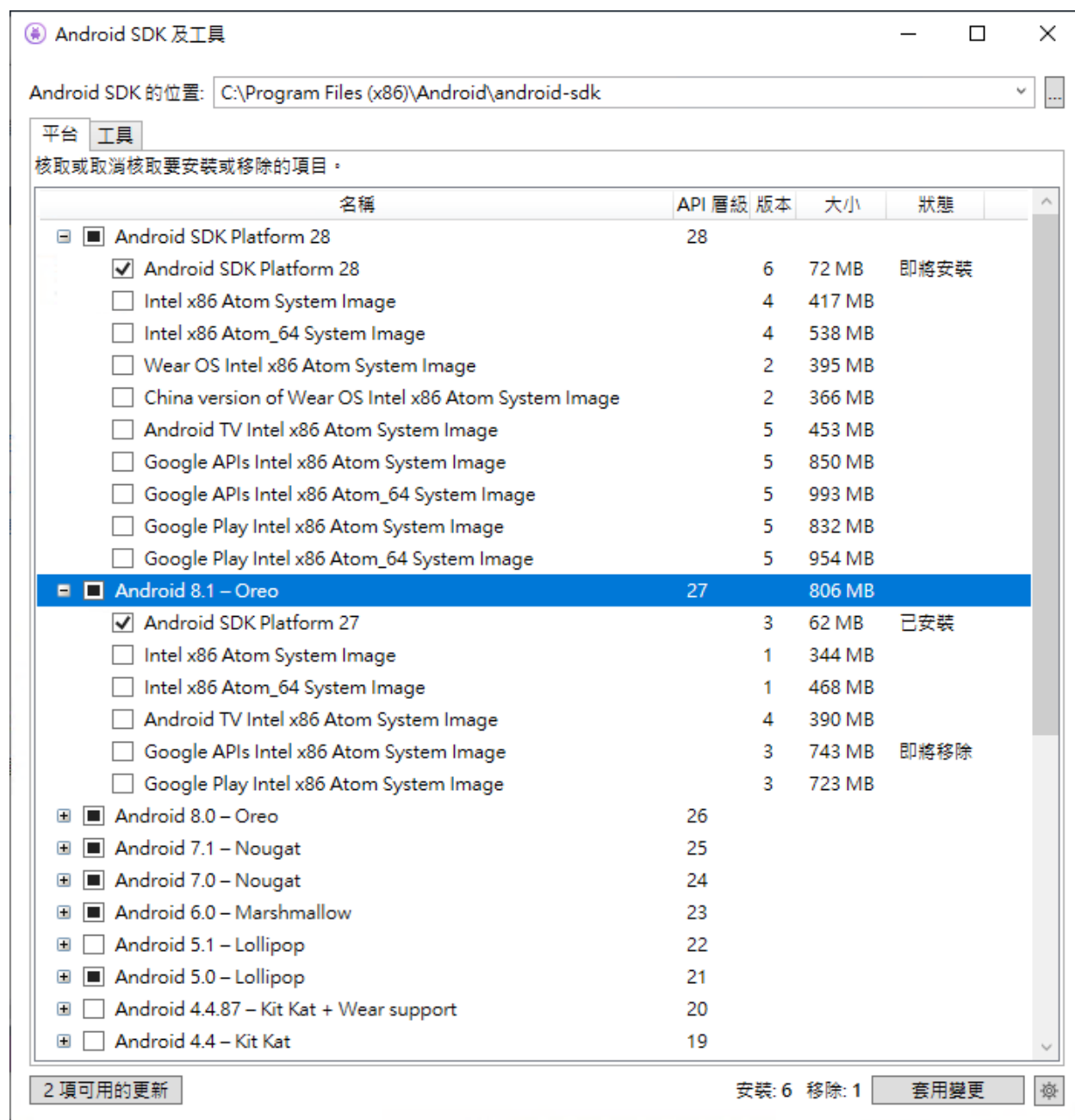


Android SDK 及工具

- 接下來，還是在工具標籤頁次，展開 Android SDK Build Tools 節點，取消勾選 Android SDK Build-Tools 25.0.3，並且勾選 Android SDK Build-Tools 27.0.3
展開額外節點，勾選 Android Support Repository
最後，點選右下方套用變更按鈕
- 現在，我們要來設定平台標籤頁次，請點選標籤頁次
在平台標籤頁次中，會有每個 Android SDK API 的版本項目，請依序展開您有興趣的 Android SDK API 版本，只需要勾選這個項目內的 Android SDK Platform X (X 為該 SDK API 版本編號)，其他的項目可以不用安裝。

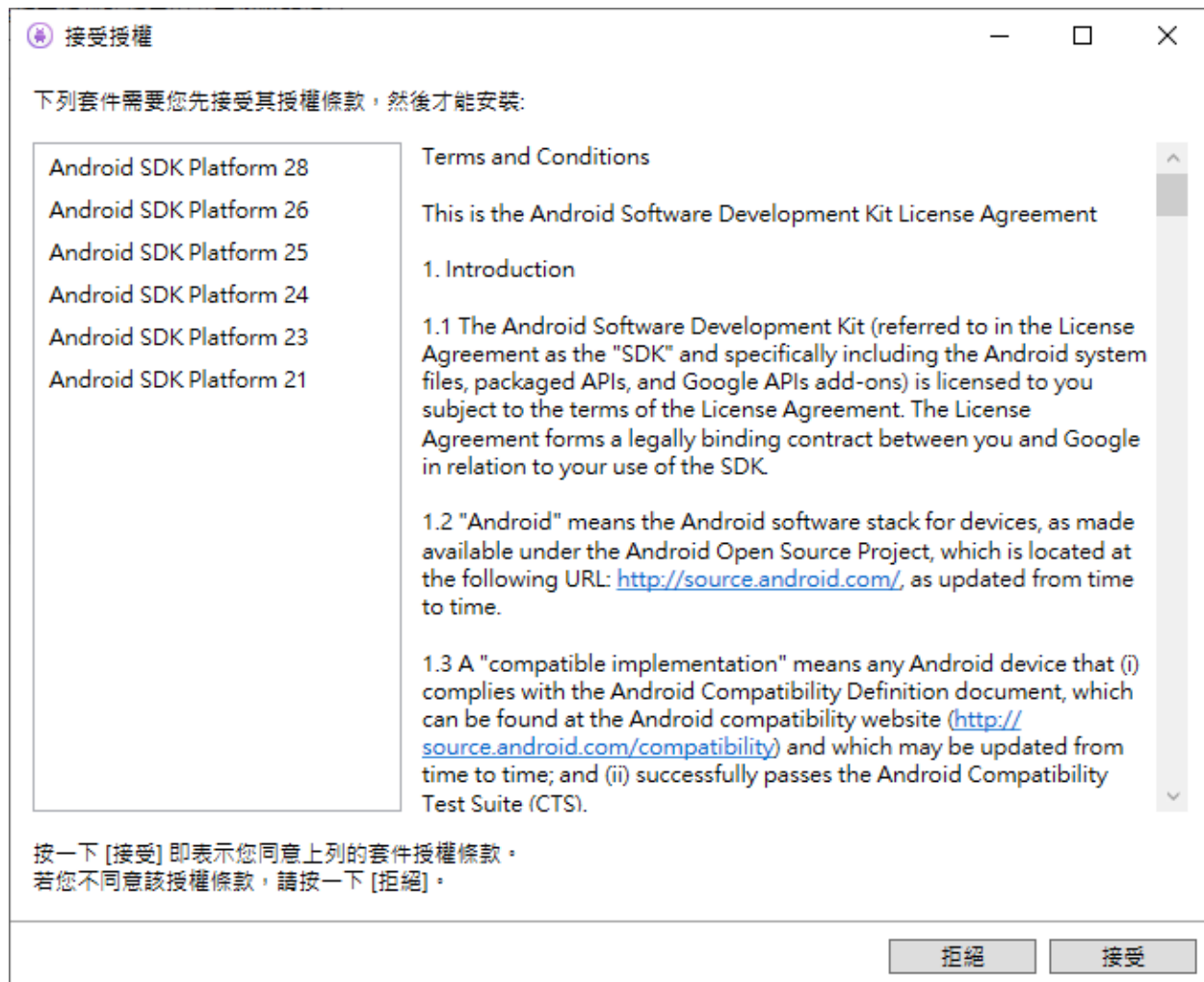
在我們這個安裝過程中，將只會選擇 Android SDK Platform 28 / 8.1 / 8.0 / 7.1 / 7.0 / 6.0 / 5.0 裡面的 Android SDK Platform X 項目。

設定完成後，請點選右下角的套用變更按鈕



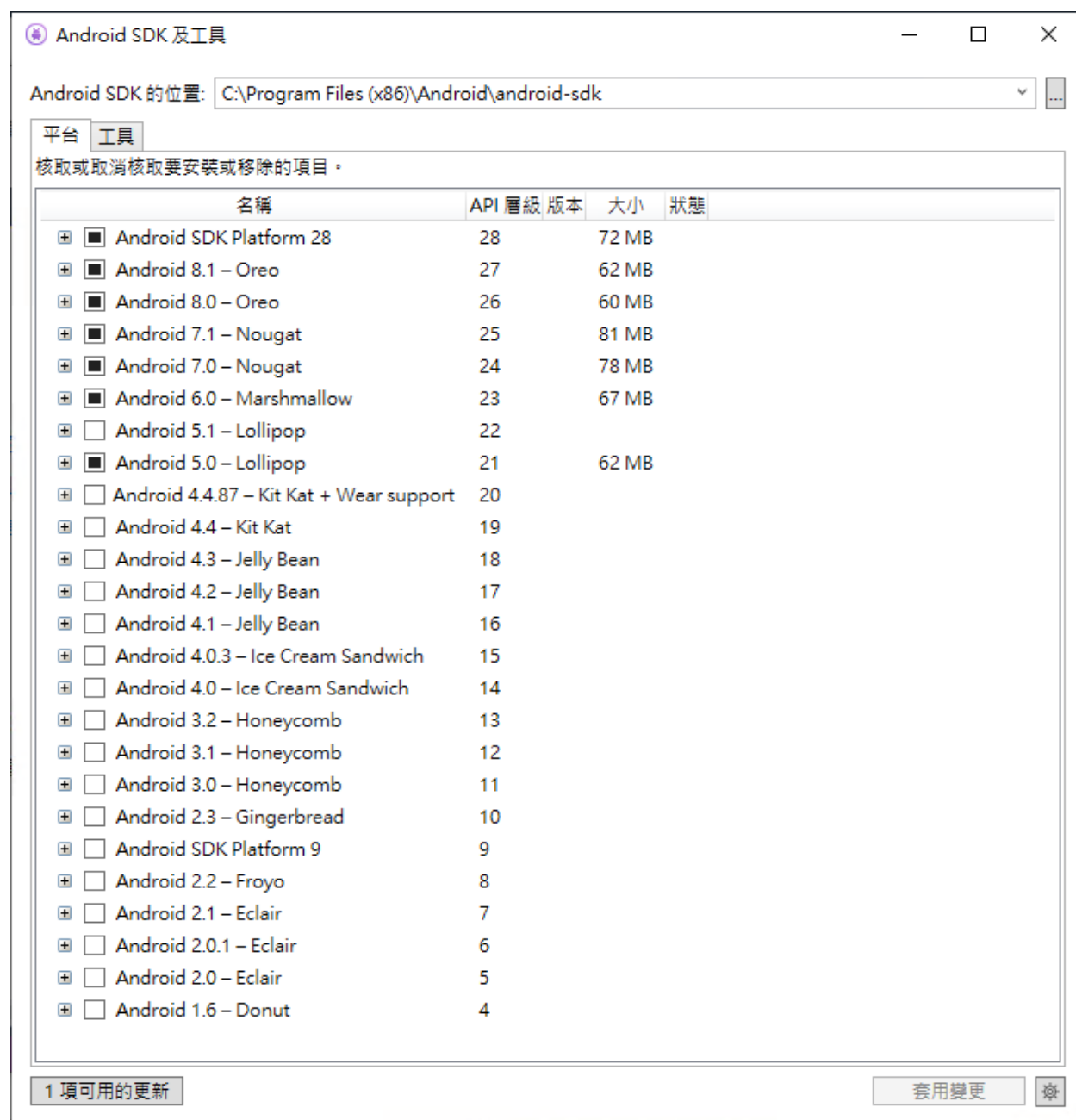
Android SDK 及工具的平台

- 當接受授權對話窗出現之後，請點選右下角的接受按鈕



接受授權對話窗

- 安裝完成之後，就可以把 Android SDK 及工具視窗關閉

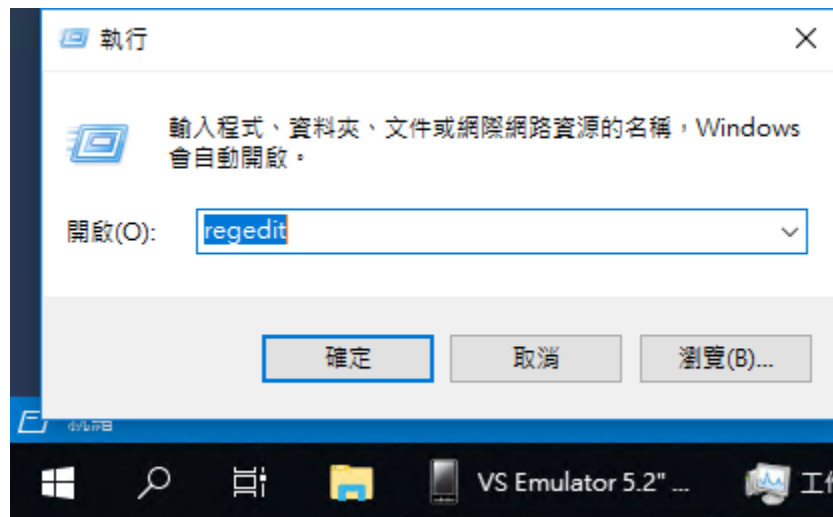


Android SDK 及工具

1.3.2 檢查 Android SDK 機碼是否存在

若沒有進行這個步驟的檢查，將會有可能發生您的 Xamarin.Android 專案無法在 Visual Studio Emulator for Android 模擬器上來執行您的 App。

- 請在左下角的視窗圖示，使用滑鼠右擊該圖示，點選執行項目，並且在執行對話窗的開啟文字輸入盒內，輸入 **regedit**，請打開 Regedit 程式



開啟 Regedit 機碼編輯器

- 請切換到這個機碼下

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node

檢查是否有這個 Android SDK Tools 機碼存在



查看是否有 **Android SDK Tools** 機碼存在

- 若沒有，使用滑鼠右擊 WOW6432Node 這個節點，點選
新增 > 機碼



機碼編輯器的新增 > 機碼

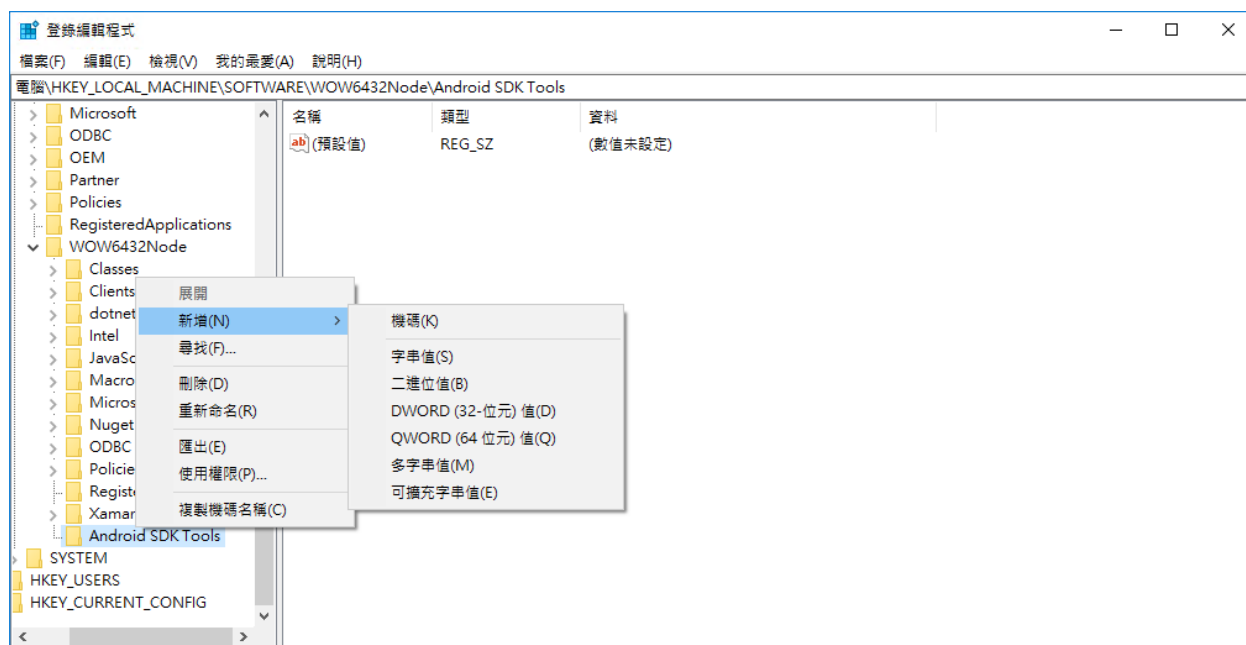
在新增的機碼處，輸入 **Android SDK Tools**



新增 Android SDK Tools 機碼

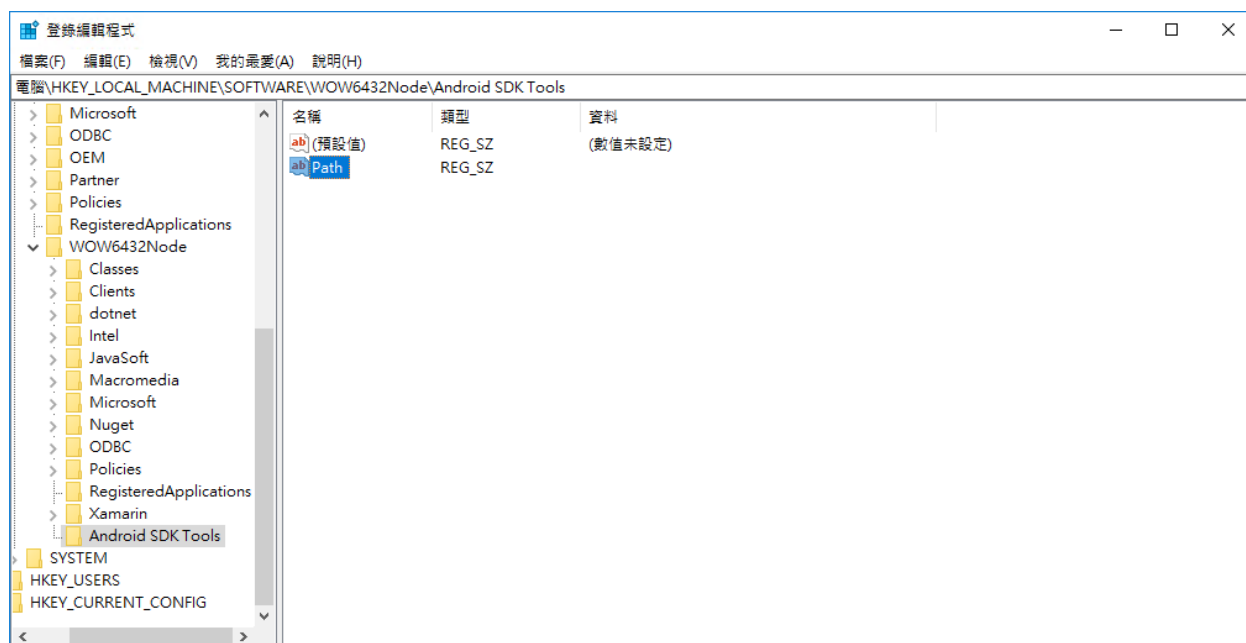
- 使用滑鼠右擊 Android SDK Tools 這個節點，點選

新增 > 字串值



Android SDK Tools 下新增字串

使用 Path 這個字串，取代掉剛剛建立的新數值 #1 字串文字

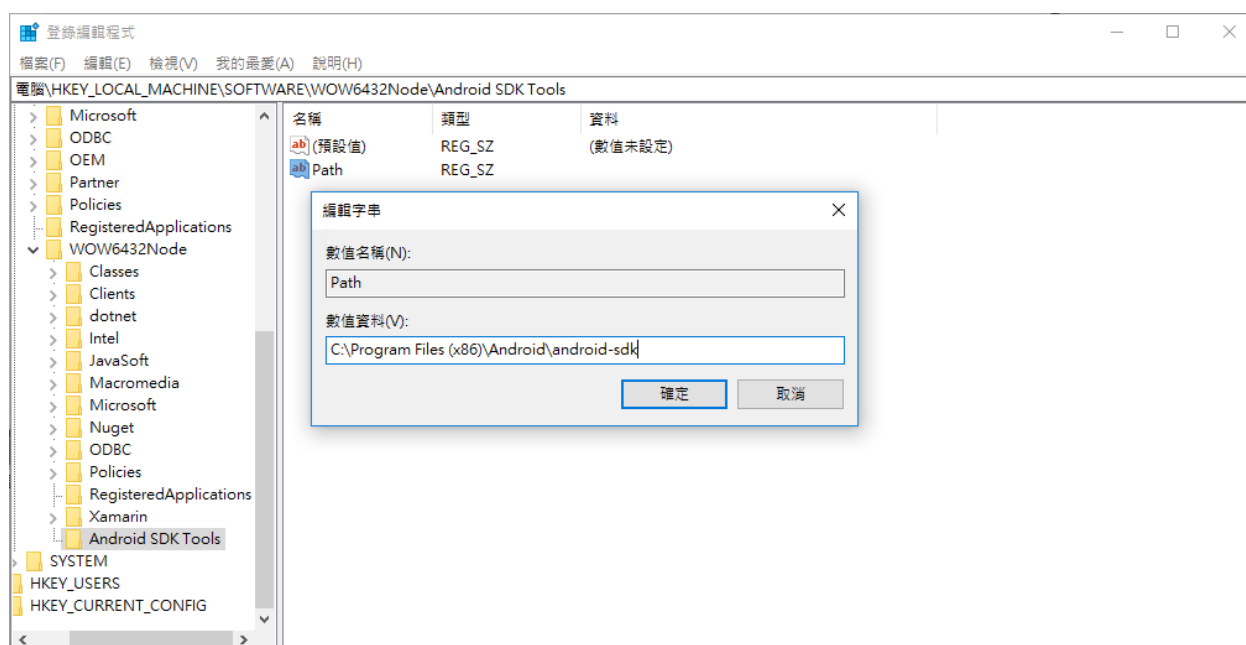


修正字串名稱為 Path

- 使用滑鼠雙擊 Path 項目

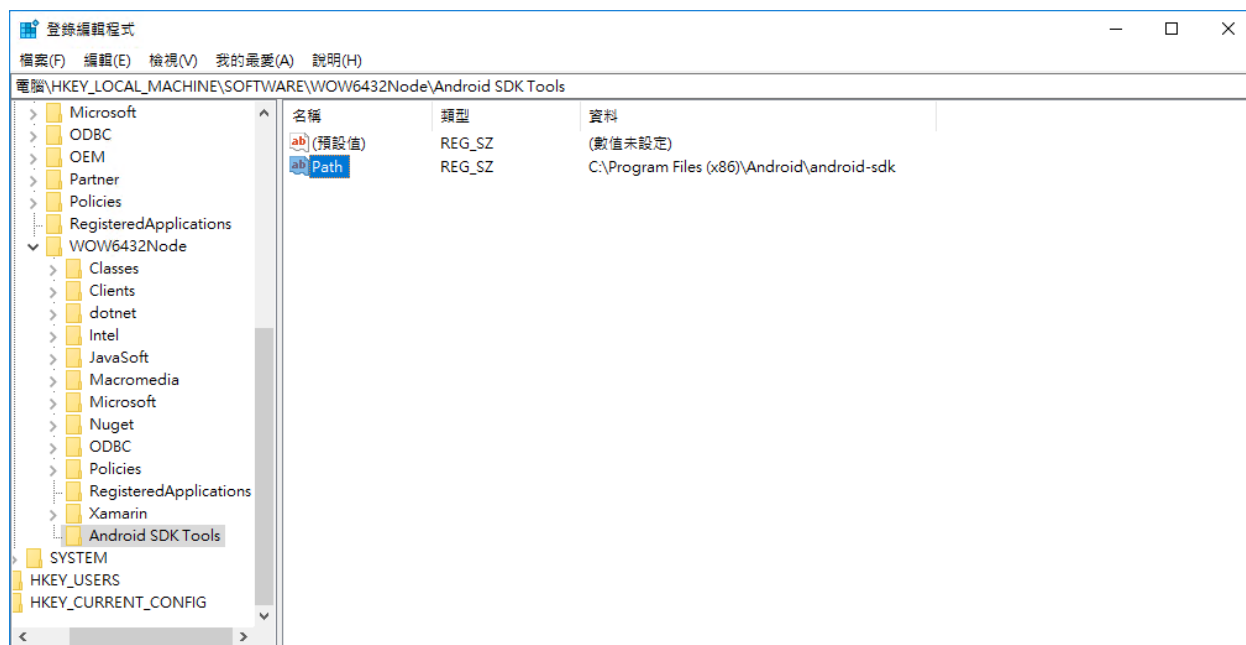
當出現編輯字串對話窗，請在數值資料欄位，輸入底下路徑

C:\Program Files (x86)\Android\android-sdk



修正 Path 的字串值

- 最後，結果如下圖，您可以關閉登錄機碼程式



完成 Android SDK Tools 機碼新增工作

1.4 測試與確認開發環境可以進行 Xamarin.Forms 專案開發

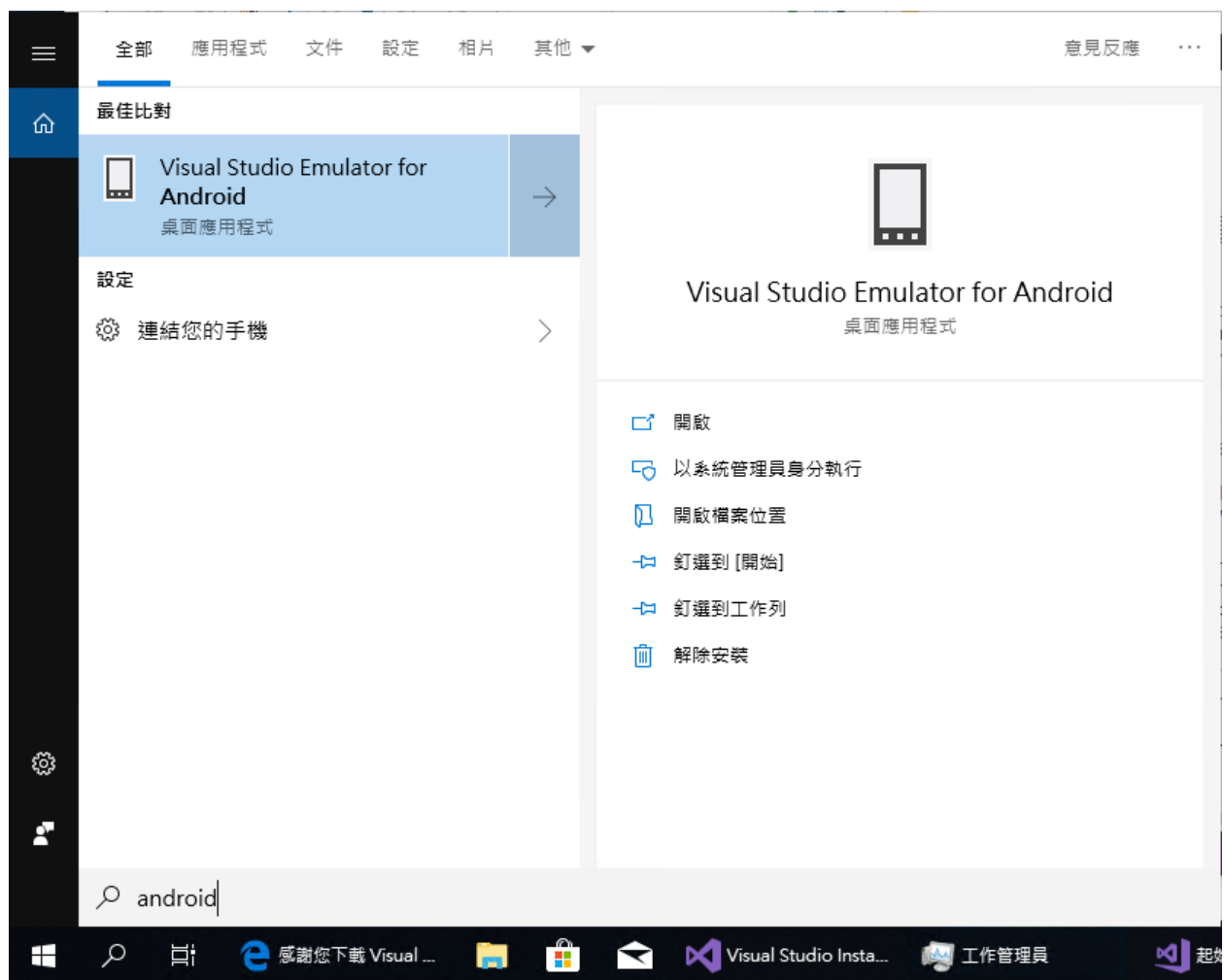
您必須要確實進行這些檢查與測試工作，確保您的 Xamarin.Forms 開發環境可以正常運作：

- 安裝與啟動 Visual Studio for Android 模擬器
- 測試可以建立 Xamarin.Forms for Prism 專案
- 確認程式碼片段可以正常運作

1.4.1 安裝與啟動 Visual Studio for Android 模擬器

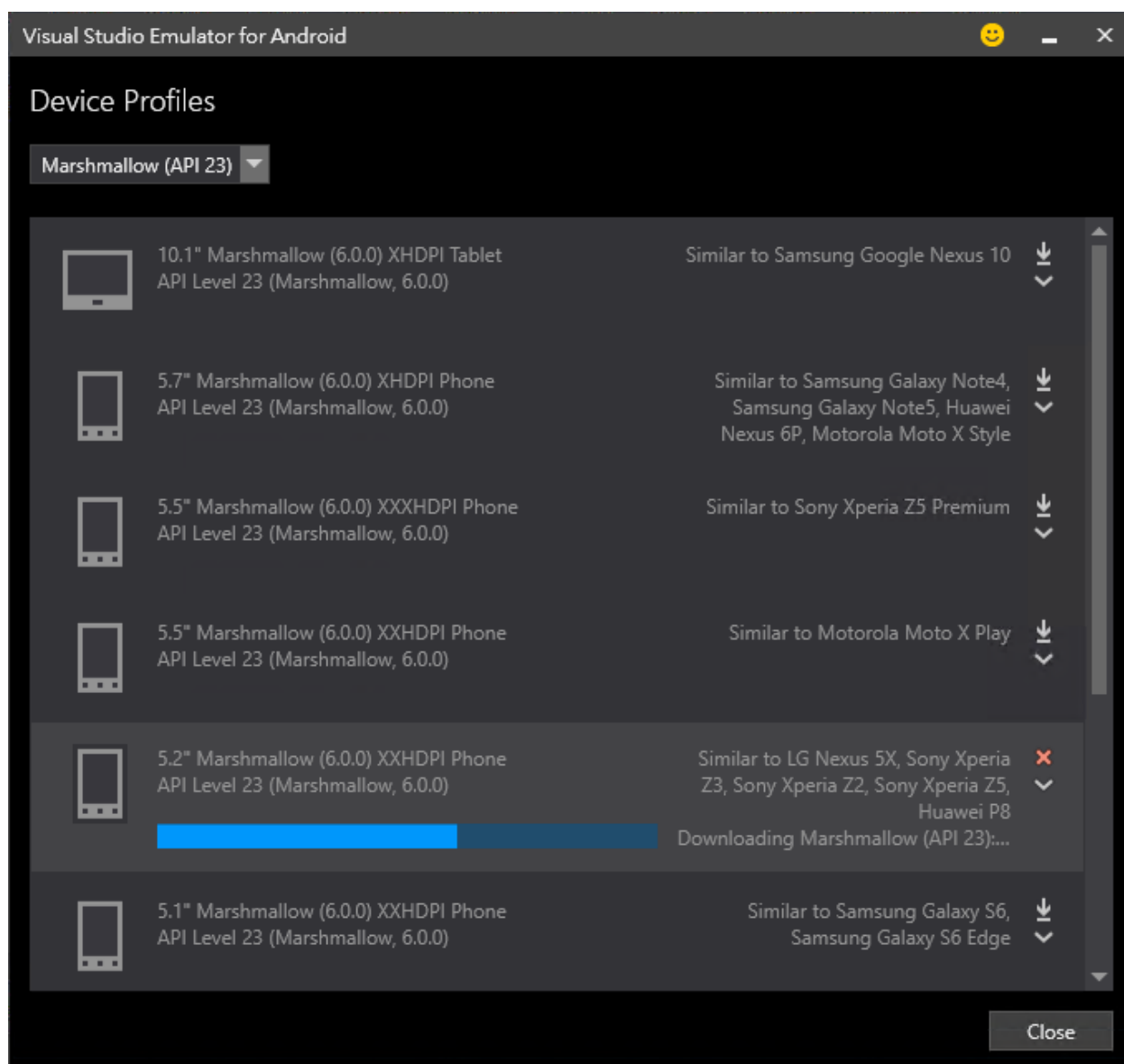
- 點選 Windows 作業系統畫面左下方的放大鏡，輸入 android 文字

現在，Windows 作業系統，會顯示出 Visual Studio Emulator for Android 這個應用程式，請點選這個應用程式，開始執行它



搜尋 Visual Studio Emulator for Android 程式

- 在 Visual Studio Emulator for Android 視窗中
切換 Device Profiles 下方的下拉選單為 Marshmallow (API 23)



安裝指定的 API 模擬器

- 作者本身習慣會下載與安裝這個模擬器，當然，您可以選擇任何您喜歡使用的模擬器
5.2" Marshmallow (6.0.0 XXHDPI Phone) Similar to LG Nexus 5X, Sony Xperia Z3, Sony Xperia Z2, Sony Xperia Z5, Huawei P8
API Level 23 (Marshmallow 6.0.0)
因此，請點選這個項目 (該畫面中的倒數第二個) 右方的向下箭頭圖示，該使下載與安

裝這個模擬器。

- 若有出現下方的使用者帳戶控制對話窗，請點選是按鈕

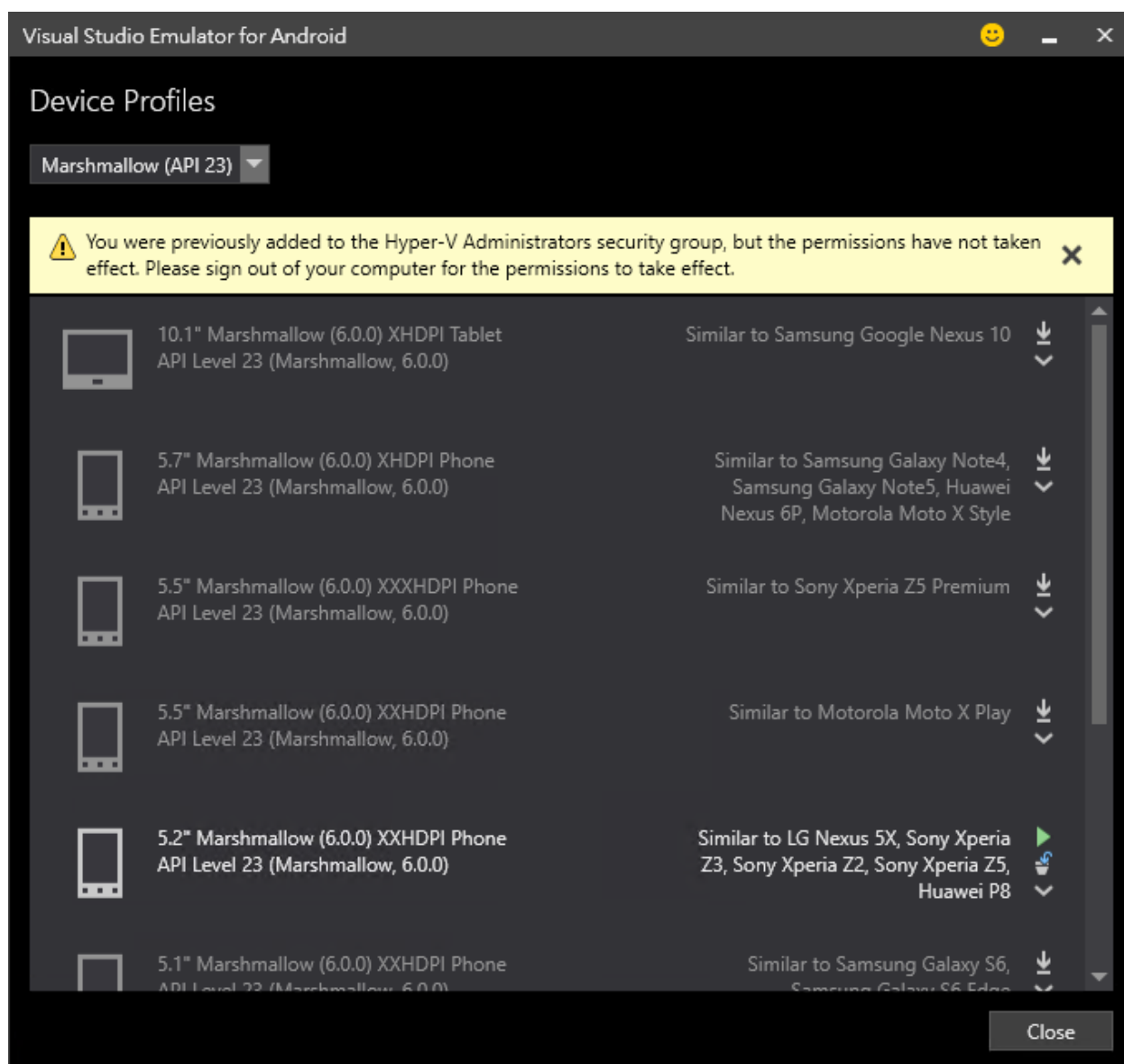


Visual Studio Emulator for Android

- 現在，若您在 Visual Studio Emulator for Android 視窗中，看到下列文字

You were previously added to the Hyper-V Administrators security group, but the permissions have not taken effect. Please sign out of your computer for the permissions to take effect.

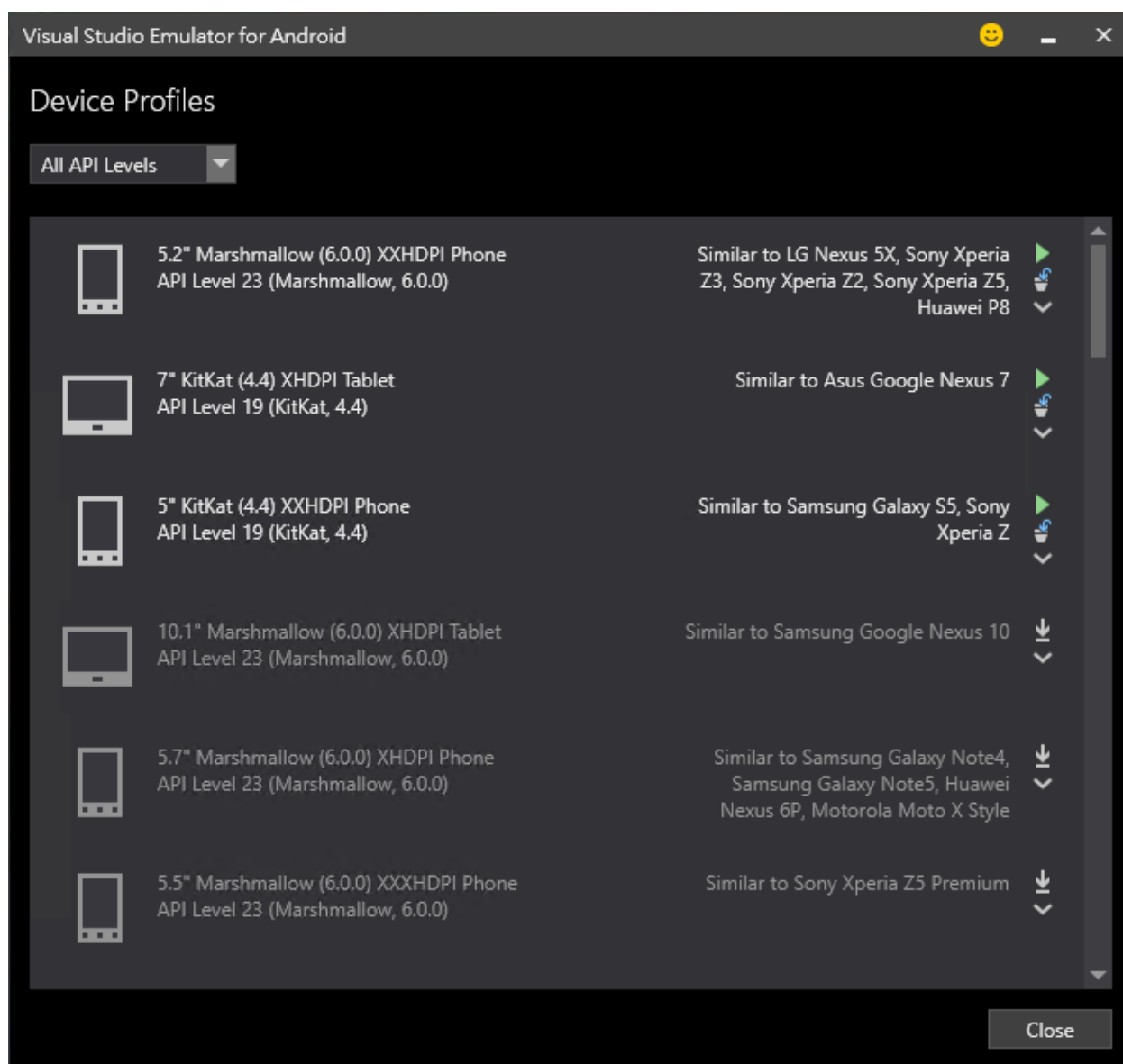
建議您重新啟動您的作業系統



Hyper-V 管理者安全群組有異動，比較建議重新開機

- 在我們重新啟動 Windows 作業系統之後，請重新開啟 Visual Studio Emulator for Android 應用程式

現在，您可以看到我們剛剛下載與安裝的 6.0 模擬器已經準備好了，請點選該項目最右方的綠色啟動三角形



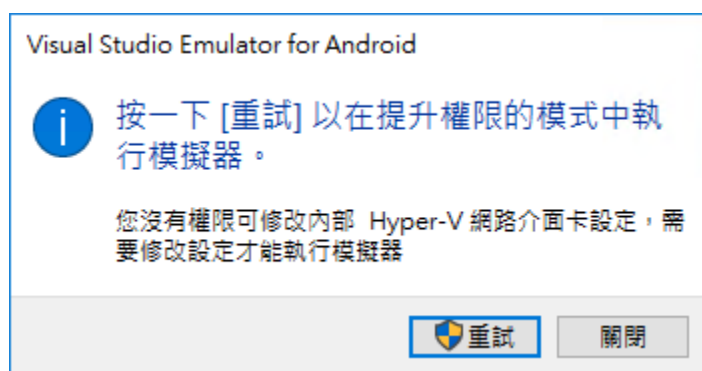
Visual Studio Emulator for Android

- 若您的環境有看到底下對話窗

按一下 [重試] 以在提升權限的模式中，執行模擬器

您沒有權限可以修改內部 Hyper-V 網路介面卡設定，需要修改設定才能執行模擬器

請點選重試按鈕



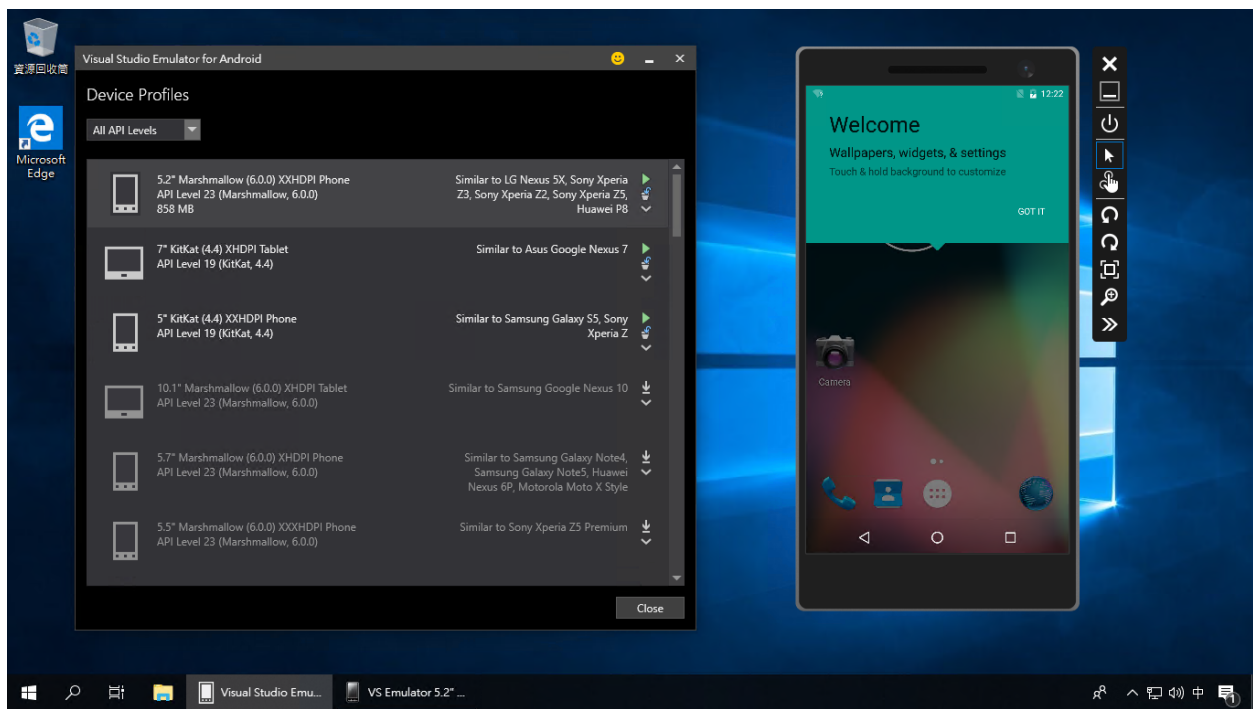
Visual Studio Emulator for Android 警告對話窗

接著會出現使用者帳戶控制對話窗，請點選是按鈕

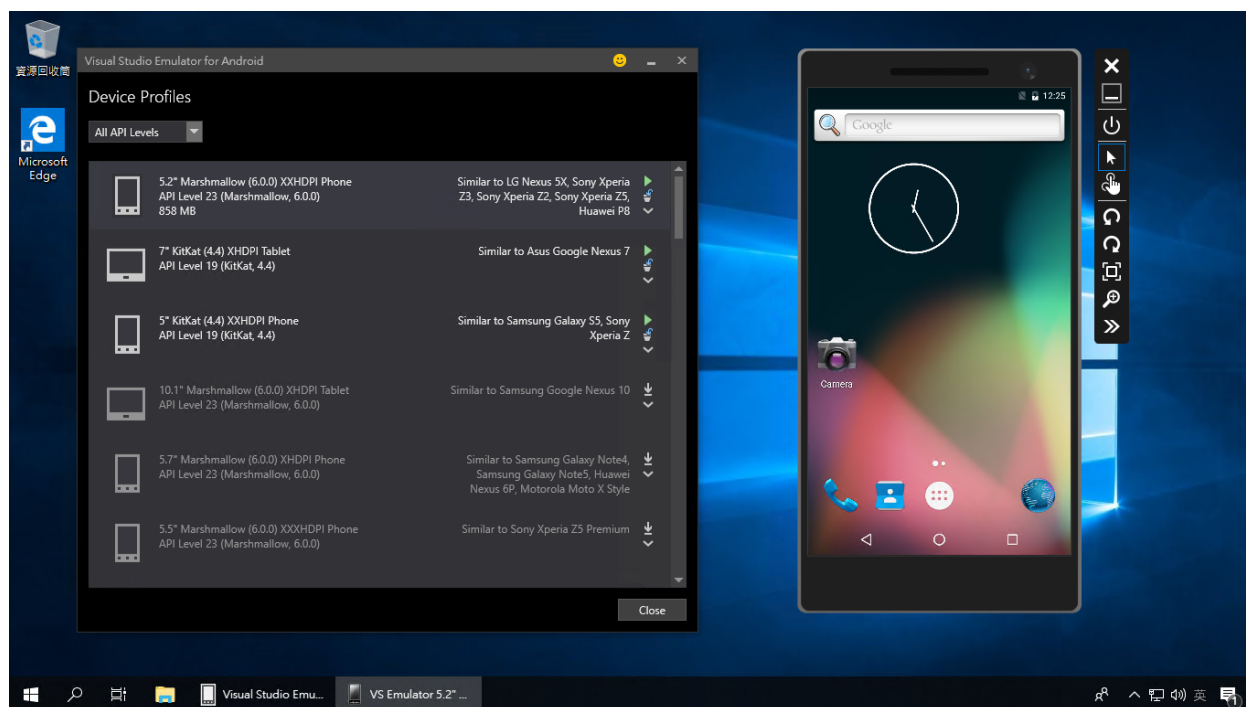


使用者帳戶控制對話窗

- 現在，若您看到畫面如下所示，那就表示您的 Visual Studio Emulator for Android 模擬器可以正常執行，您可以點選模擬器上 GOT IT 按鈕

**Visual Studio Emulator for Android**

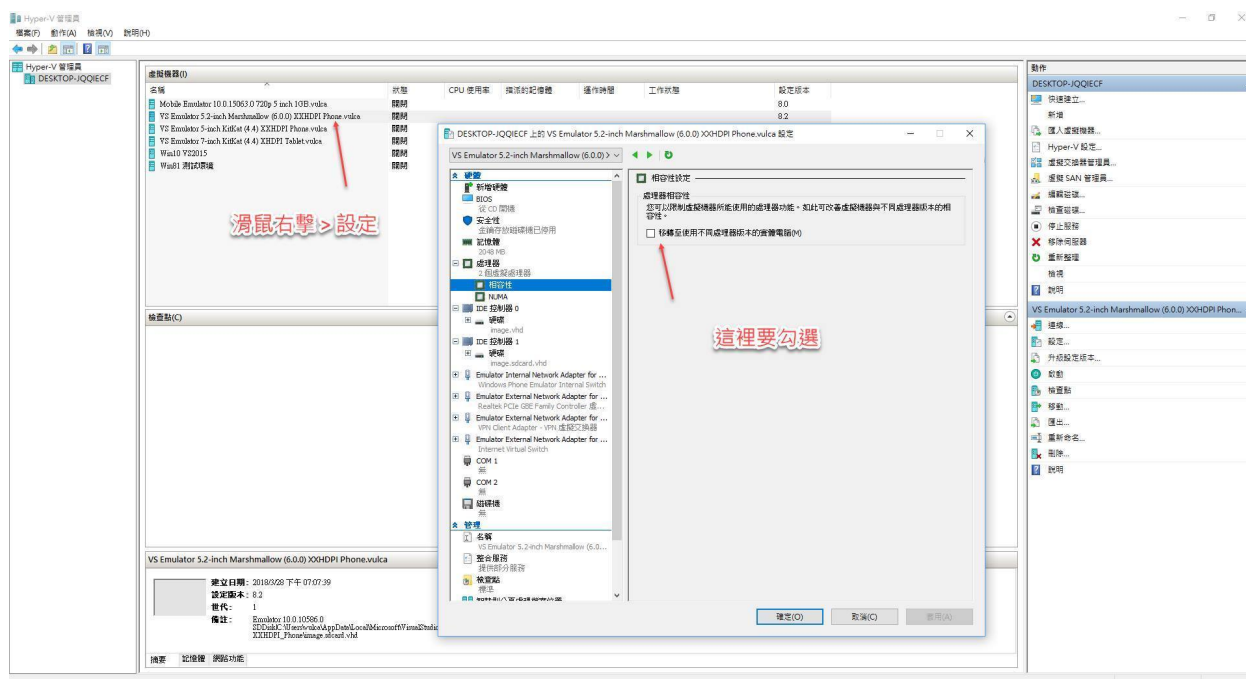
現在，您的電腦畫面將會如下所示



Visual Studio Emulator for Android

1.4.2 若無法正常模擬器進行除錯之修正

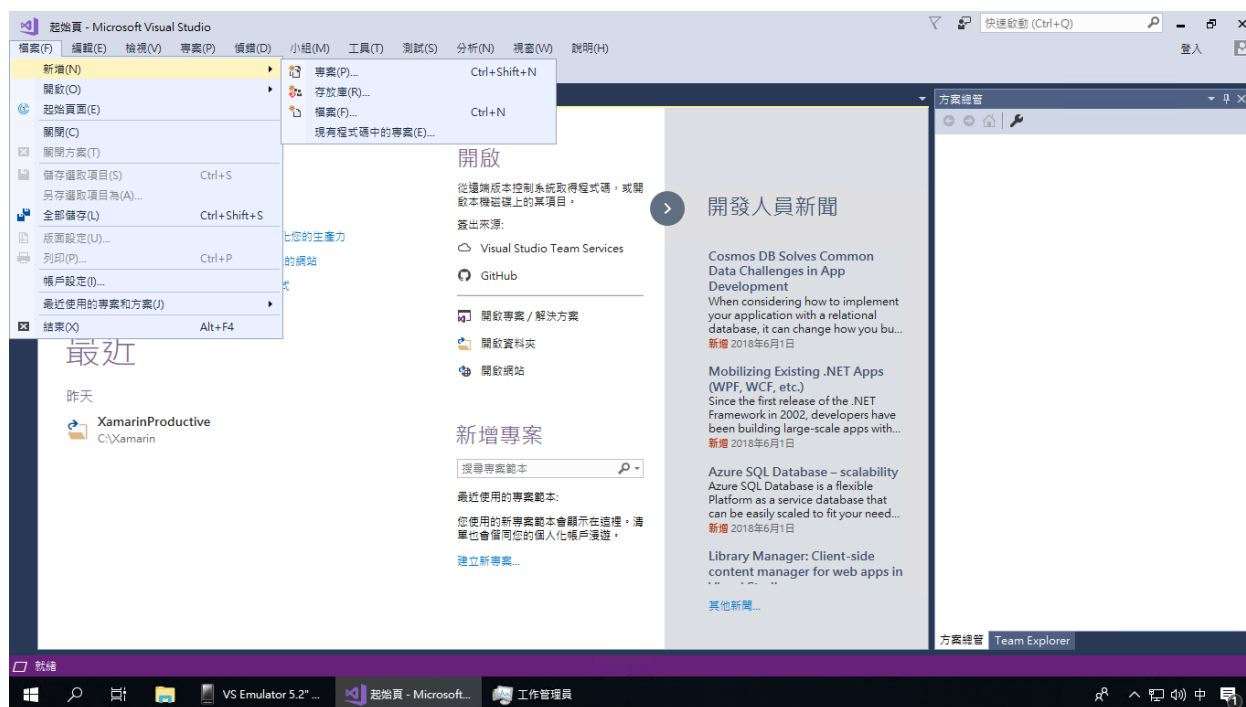
請打開 Hyper-V 管理員，找到您要連線的模擬器，並且記得要先關閉該模擬器，接著，滑鼠右擊該模擬器，設定處理器 > 相容性，要將 **移轉至使用不同處理器版本的實體電腦** 選項打勾，之後，可以重新啟用該模擬器，就可以正常進行除錯了。



修正模擬器之相容性

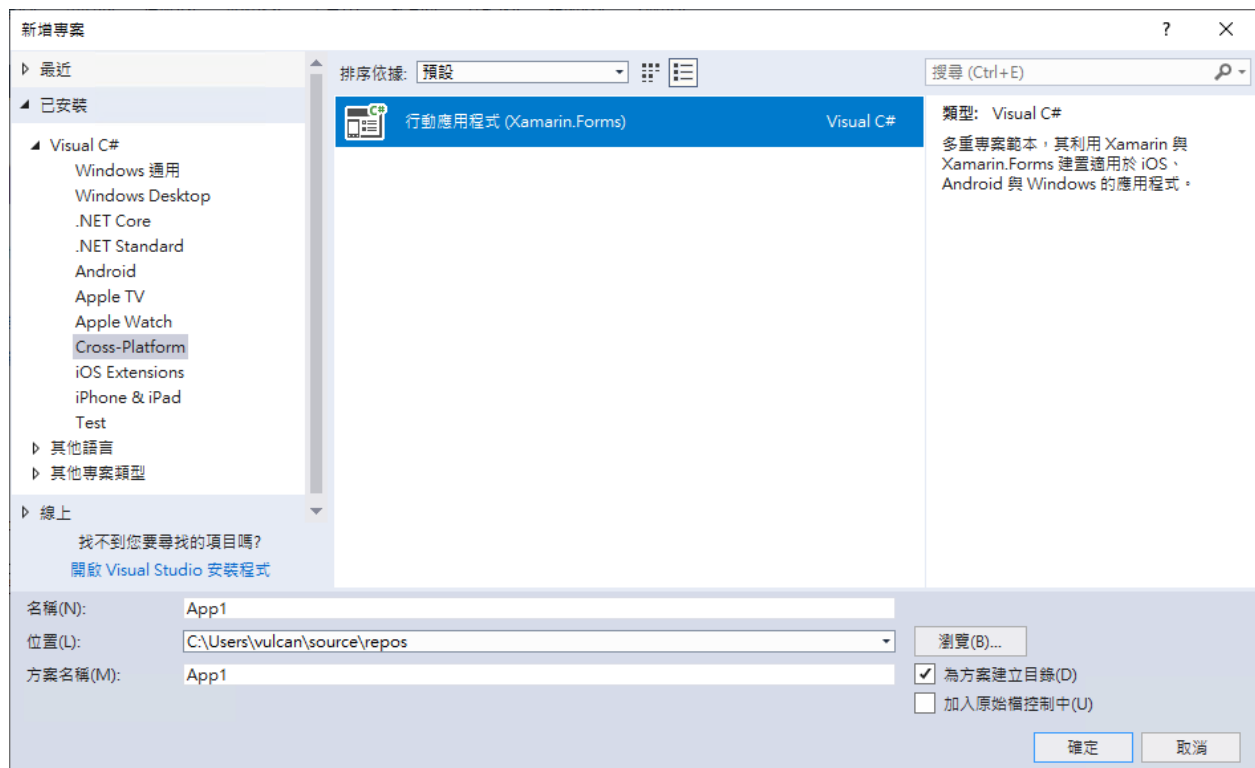
1.4.3 測試可以建立 Xamarin.Forms 專案

- 請在開啟 Visual Studio 2017 程式，並且，點選功能表
檔案 > 新增 > 專案



Visual Studio 2017 建立新的專案

- 在新增專案對話窗中，請點選該對話窗左方的 Cross-Platform 項目
在中間選項中，點選行動應用程式 (Xamarin.Forms)，最後點選對話窗右下方的確定按鈕



新增專案對話窗

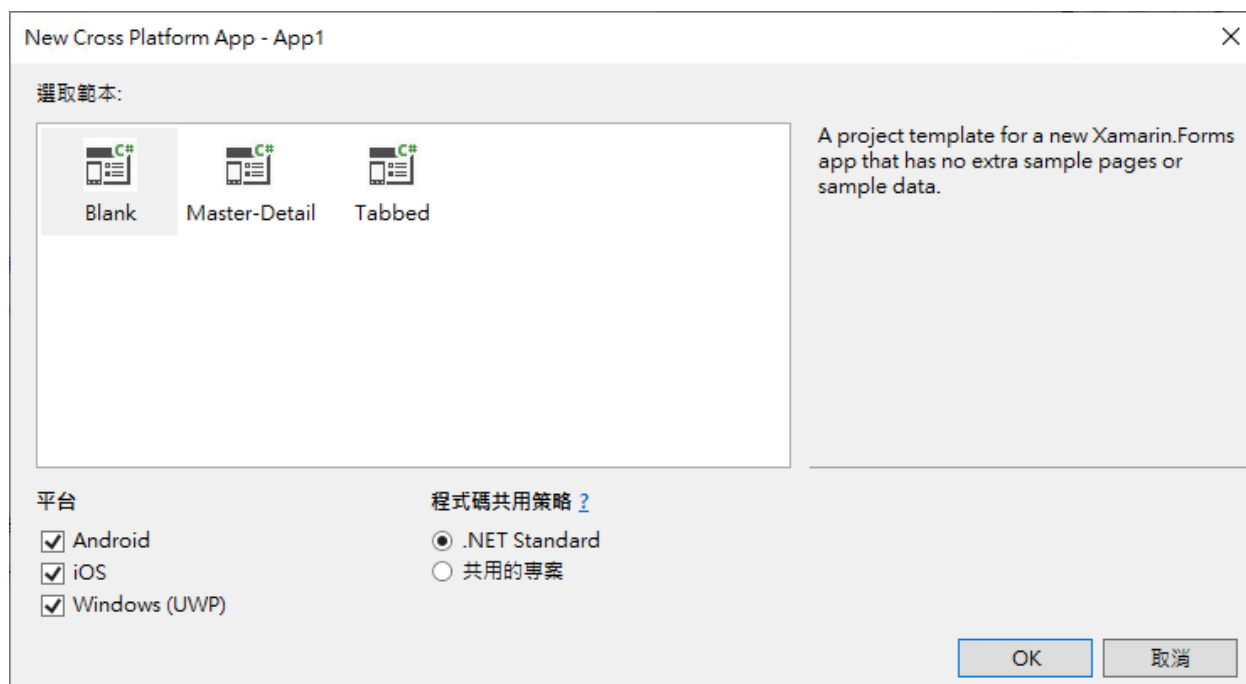
- 出現了 New Cross Platform App 對話窗

請確認在選取範本中，點選 Blank

下方的平台選項，請記得 Android / iOS / Windows (UWP) 請都要勾選

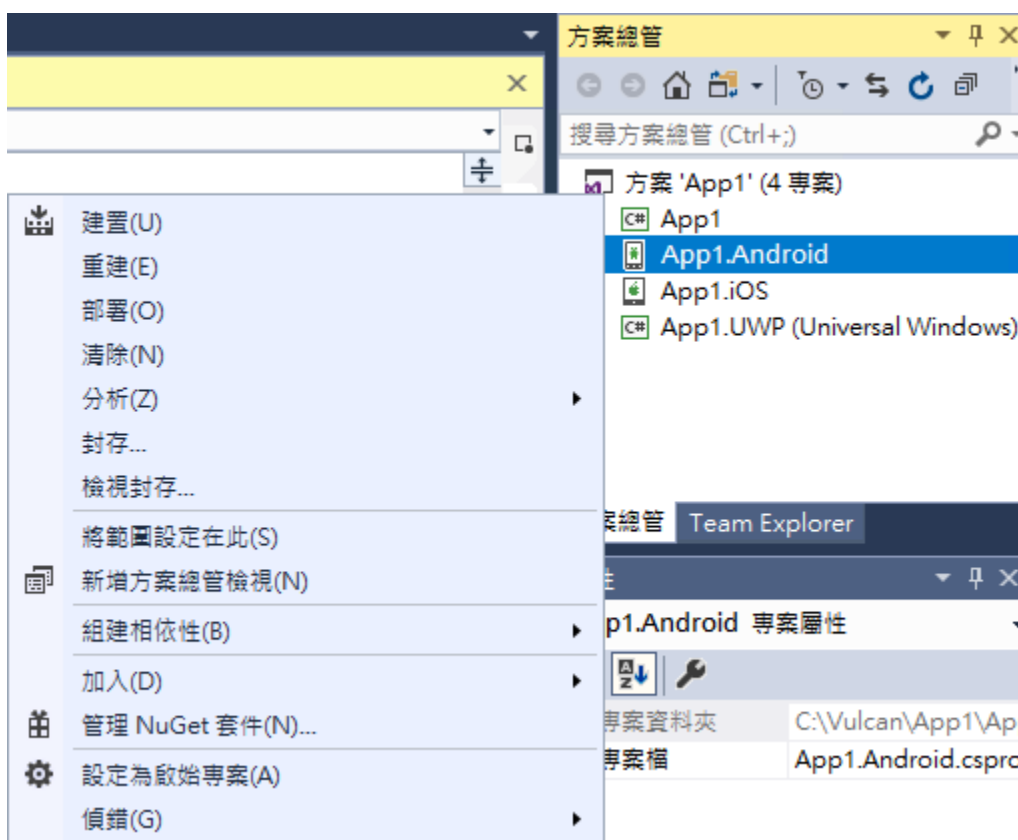
在程式碼共用策略的下方，請選擇 .NET Standard

請點選該對話窗右下方的 OK 按鈕



New Cross Platform App 對話窗

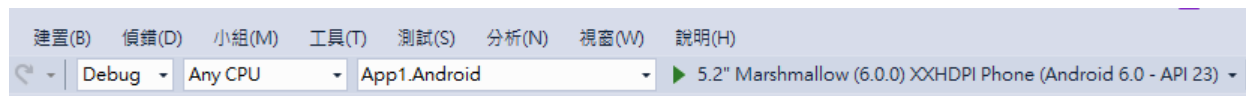
- 當 Xamarin.Forms for Prism 專案建立完成後，請使用滑鼠右擊 Android 專案選擇設定為起始專案



在方案總管中設定預設起始專案

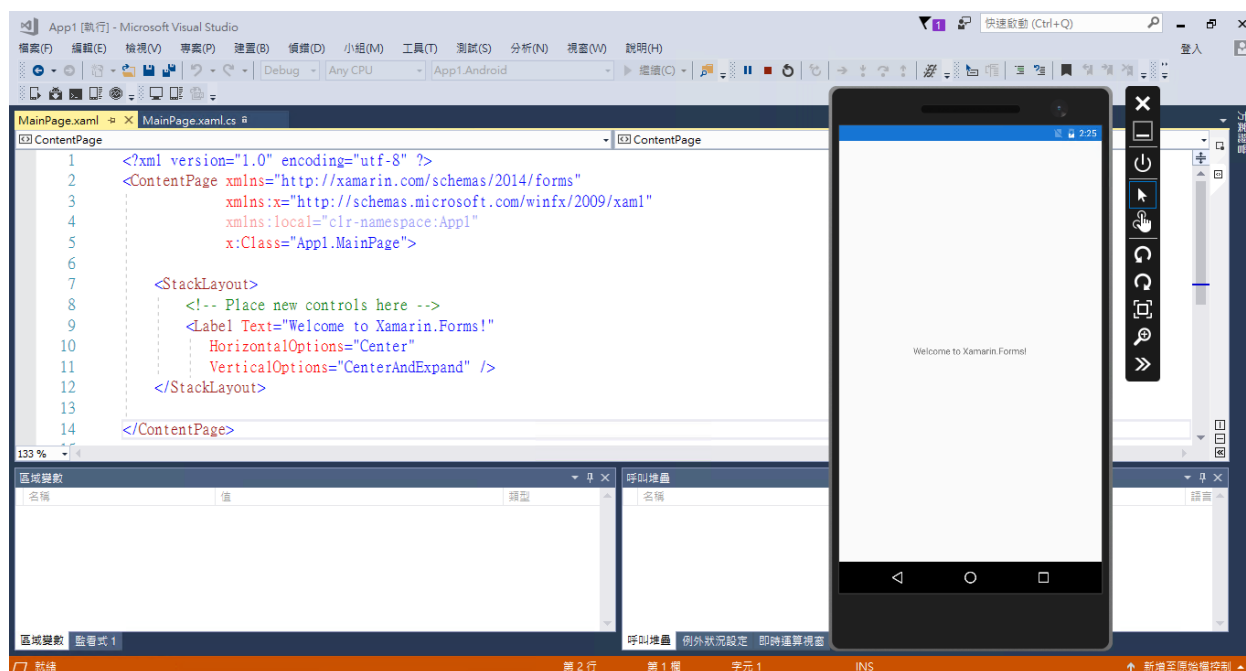
- 此時，您會在 Visual Studio 2017 中間上方區域，看到啟動專案已經設定為 Android 專案，並且要執行的裝置為 5.2" Marshmallow (6.0.0) XXHDPI Phone (Android 6.0 API 23) 這個模擬器

請點選有綠色三角形的模擬器項目，開始建置與在這個模擬器上來執行



執行 Android 專案

- 由於第一次進行專案建置，需要下載許多 NuGet 套件，因此，需要花費一些時間，請耐心等候一下，底下是成功執行完成的畫面

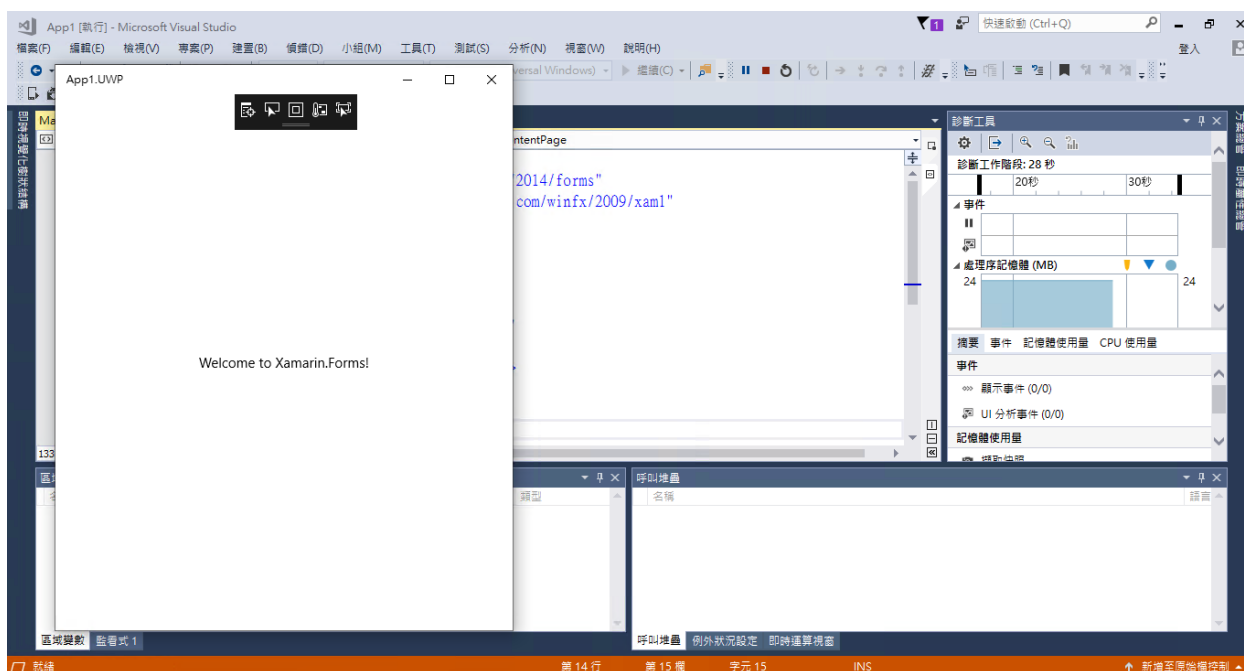


Xamarin.Forms 的 Android 專案成功執行結果

- 停止這個 Android 專案的執行 (點選工具列上的紅色正方形按鈕)
- 使用滑鼠右擊 UWP 專案

選擇設定為起始專案

- 點選 Visual Studio 2017 最上方工具列綠色三角形的本機電腦
底下將會是成功執行 UWP 專案的畫面



Xamarin.Forms 的 UWP 專案成功執行結果

- 停止 UWP 專案的執行

若您的 Visual Studio 2017 可以成功建立 Xamarin.Forms 專案，可以建置在 Android / UWP 平台下執行，那麼，恭喜您，您的 Visual Studio 2017 開發環境，已經可以正常進行 Xamarin.Forms 跨平台專案開發了

2. 開發時期的 XAML 即時預覽

在這個章節，我們來看看市面上對於設計時期的預覽工具，這些工具可以幫助我們進行 XAML 語言學習並且增加應用程式開發的效率，分別是：大金剛 Gorilla、LiveXAML、即時重新載入 Live Reload；而除了 LiveXAML 為付費產品之外，其餘兩套產品都是免費的。

2.1 推薦 XAML 的設計預覽工具大金剛 Gorilla

大金剛撥放器 Gorilla Play 提供在進行 XAML 文件設計時期的即時預覽能力，當您完成安裝與設定的步驟之後，您無須透過 Visual Studio 執行 Xamarin.Forms 專案到模擬器或者實體裝置上，就可以進行線上直接看到您的 XAML 文件檔案的在這些模擬器與實體裝置上的執行結果。

大金剛撥放器是完全免費的，它是由這幾個成員所組成：桌機代理程式、服務器元件、Visual Studio IDE 擴充功能、模擬器或者實體裝置上的撥放器。現在讓我們來了解如何進行安裝與使用這個大金剛撥放器所提供的 XAML 預覽功能。

2.1.1 電腦主機軟體安裝

首先，使用瀏覽器打開 [下載 Gorilla 網址¹](https://grialkit.com/gorilla-player/) 網頁，您將會看到右上角提供許多功能，如同下面螢幕截圖畫面。


¹<https://grialkit.com/gorilla-player/>



下載 Gorilla 網頁

我們先要取得大金剛 Gorilla 預覽工具的使用帳號，因此，請點選網頁右上方的 REGISTER 連結，此時，便會顯示如同下方要進行註冊的內容。請填入網頁需要的資料，最後點選 Register 按鈕之後，將會完成大金剛 Gorilla 預覽工具使用帳號。

Gorilla Registration



Please complete the form below

Gorilla 使用帳號註冊

我們接著需要安裝大金剛 Gorilla 預覽工具的桌機代理程式，請同樣點選大金剛 Gorilla 預覽工具首頁右上方的 DOWNLOAD 連結，您將會看如同下面的畫面，大金剛 Gorilla 預覽

工具的桌機端代理程式支援可以在 Windows / Mac 作業系統下執行，在這裡我們將會使用 Windows 作業系統作為講解，所以，請點選 Windows 按鈕，下載安裝程式。



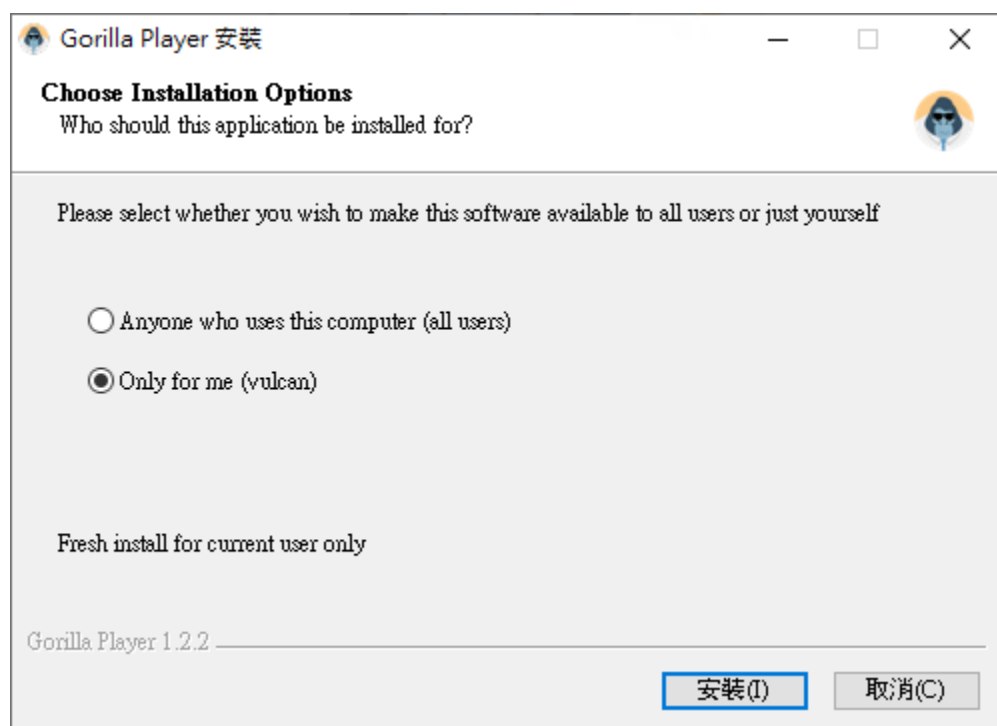
Download Gorilla Player 1.2.2.0

 WINDOWS

 MAC OS

選擇 Gorilla 需要執行作業系統

安裝程式下載完成後，請啟動這個安裝程式，此時 Gorilla Player 安裝對話窗將會出現，我們可以直接點選安裝按鈕進行安裝作業。



進行 Gorilla 安裝

當安裝完成之後，會出現如下圖的畫面，請點選完成按鈕，將會緊接著啟動與執行 Gorilla Player。



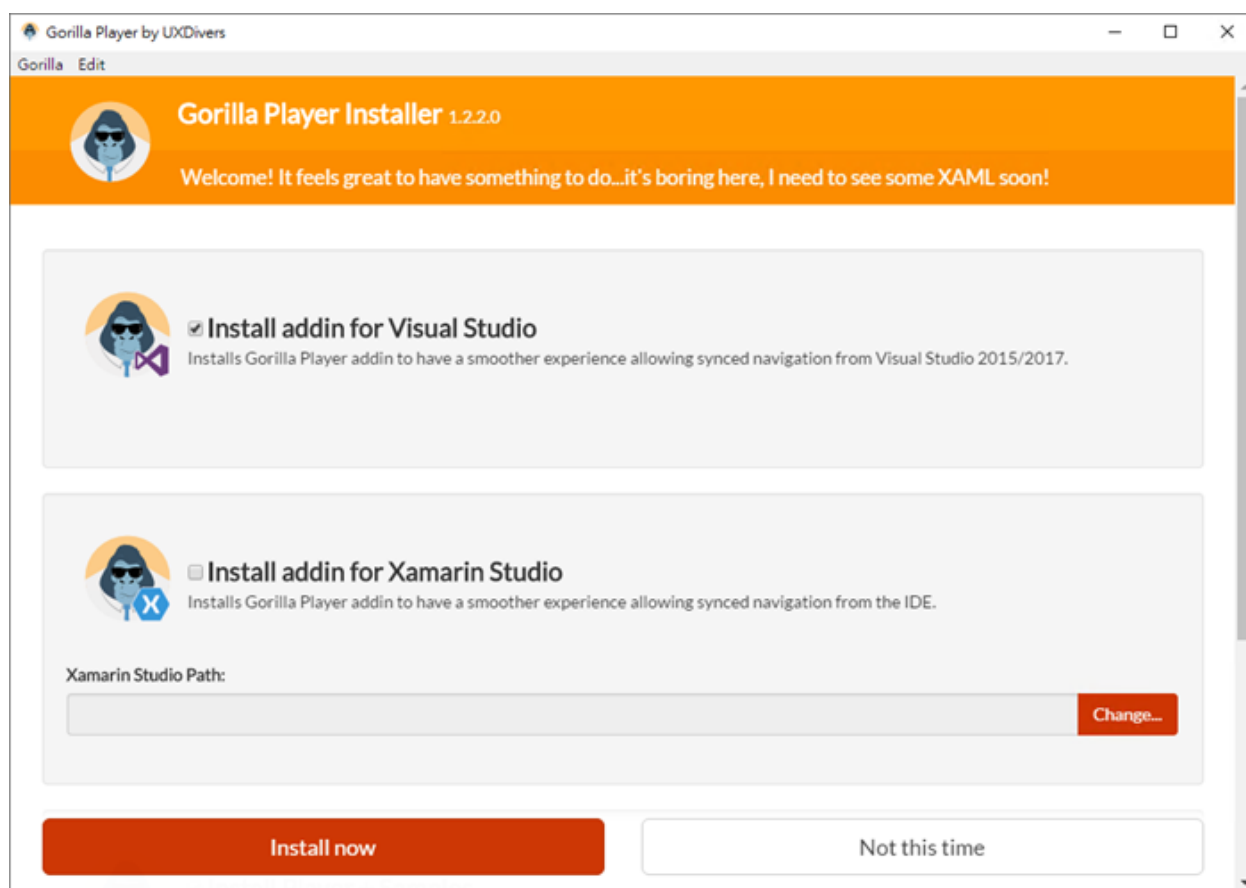
完成 Gorilla 安裝並且啟動 Gorilla Player

您需要先同意這個大金剛 Gorilla 預覽工具軟體使用授權，因此，請點選 I accept this EULA 按鈕。



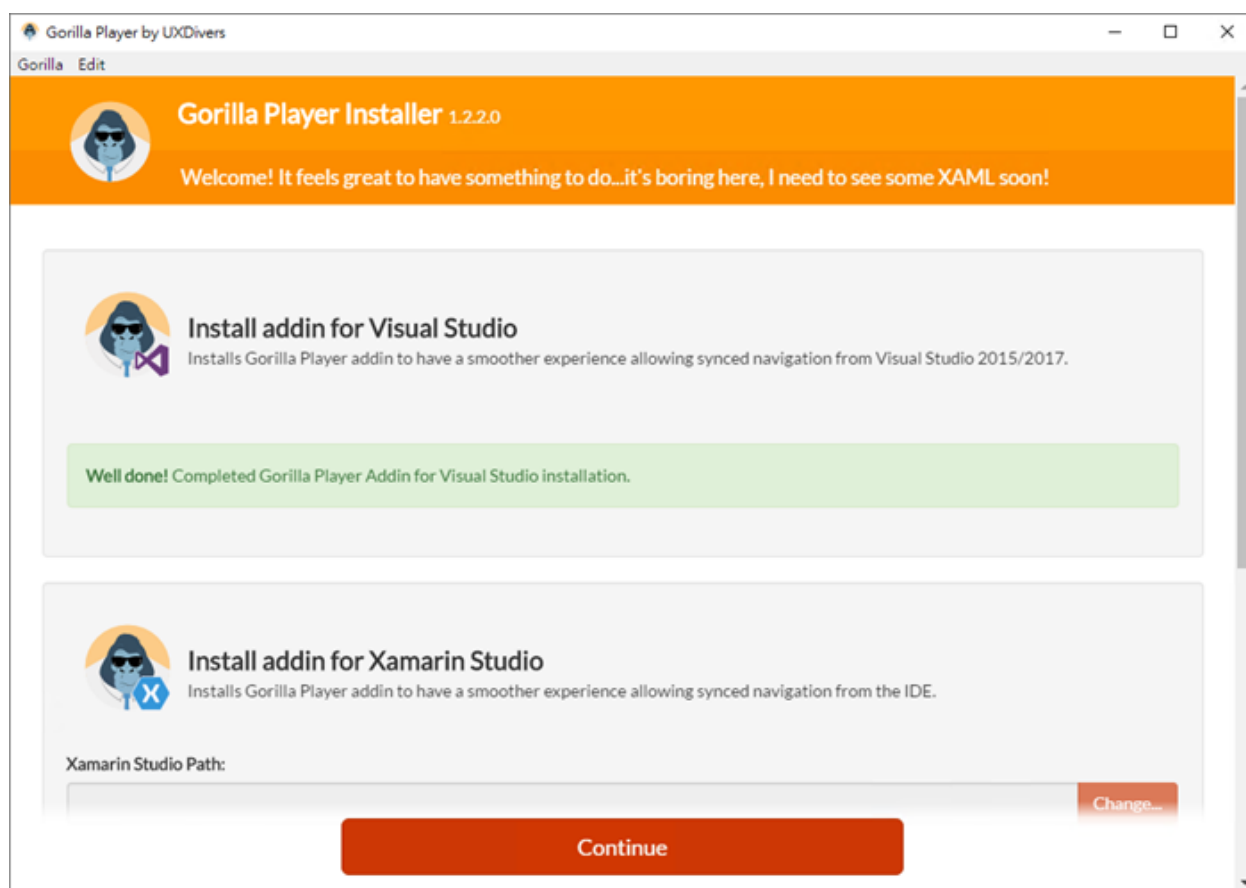
接受使用者授權

在這裡，我們需要點選 Install Now 按鈕，以便將大金剛 Gorilla 預覽工具的 Visual Studio 擴充功能安裝到 Visual Studio IDE 內。

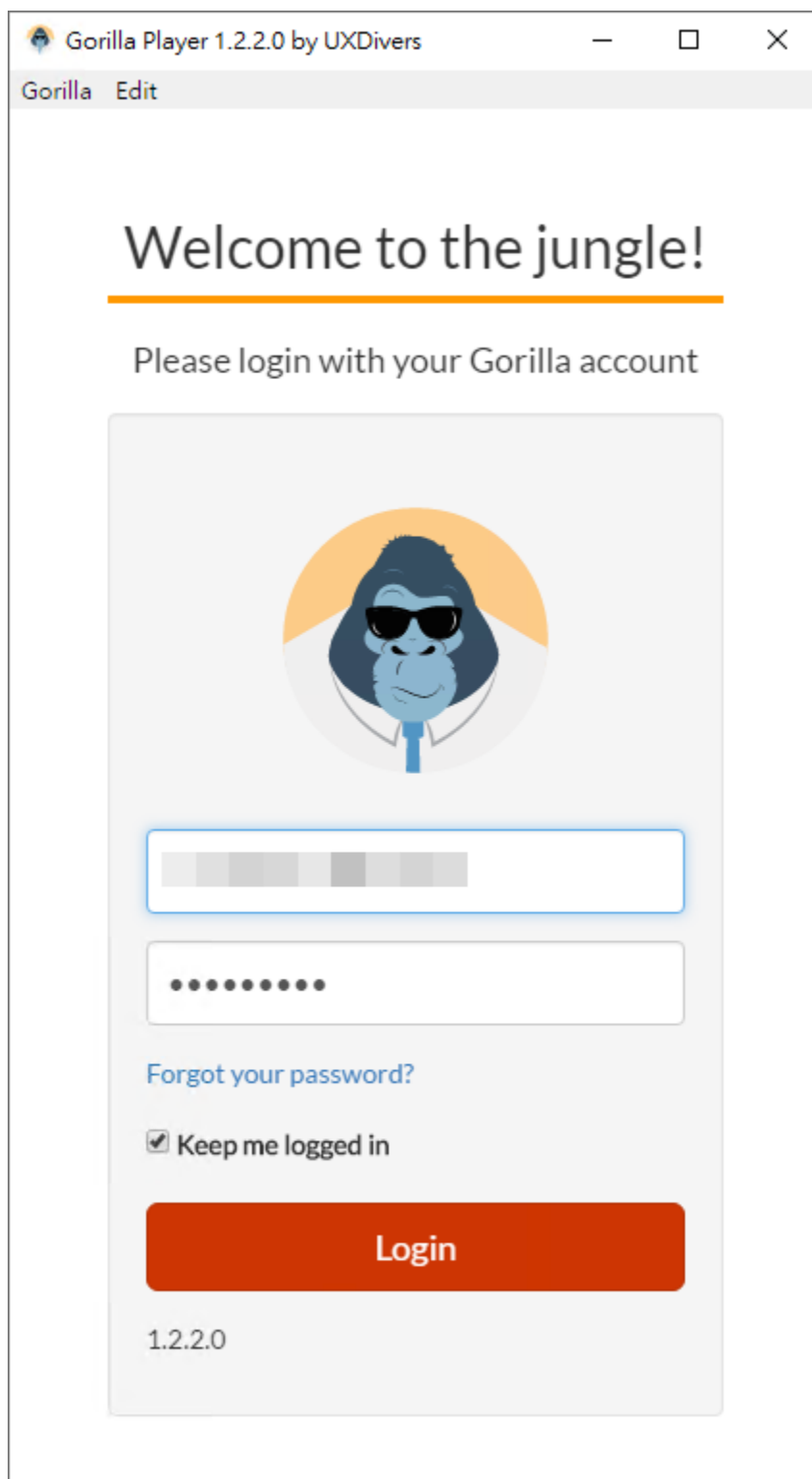


安裝 Visual Studio 擴充功能

Visual Studio 擴充功能之後，您就可以點選 Continue 按鈕。

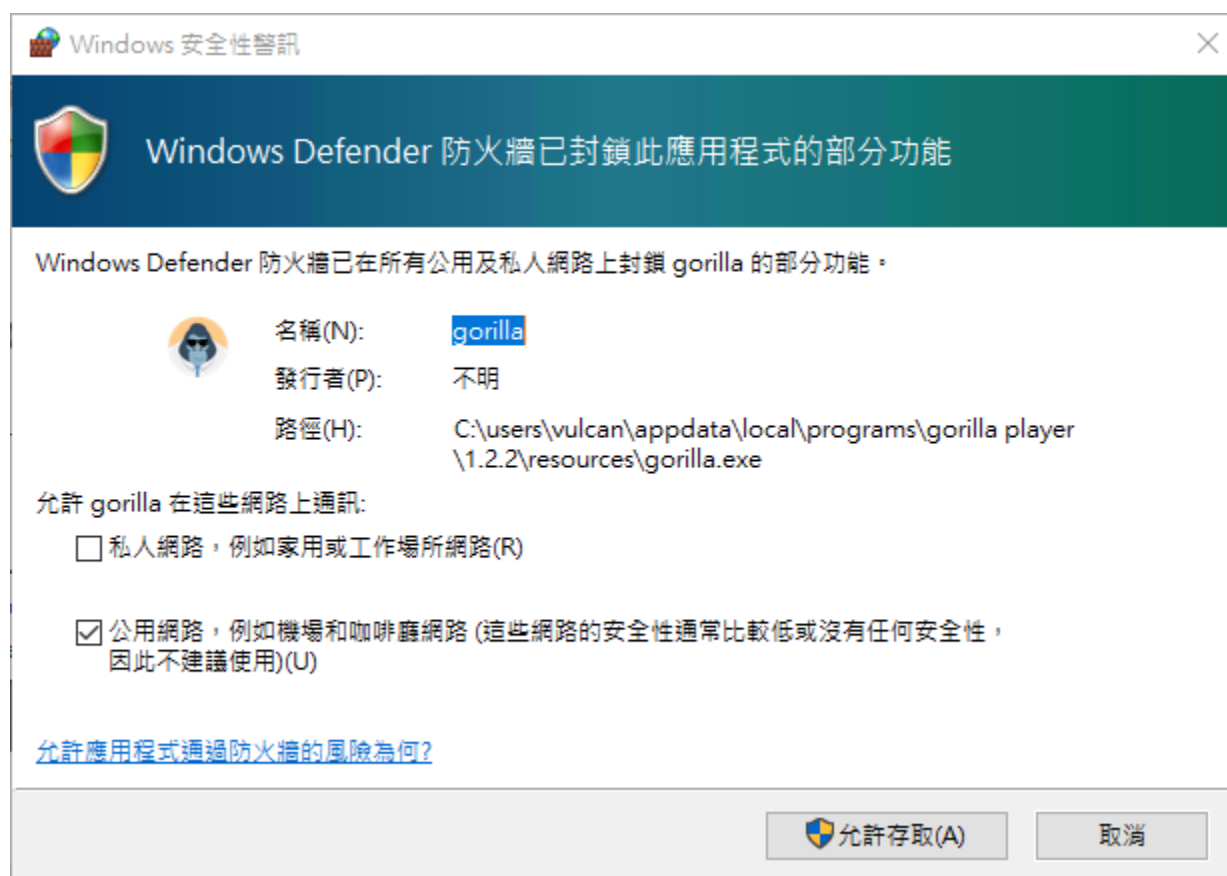
**Visual Studio 擴充功能安裝完成**

現在，Gorilla Player 登入畫面將會出現，我們需要在這裡先登入到 Gorilla Player 系統內，因此，請在這個畫面上填入剛剛申請到的帳號與密碼，接者點選 Login 按鈕。



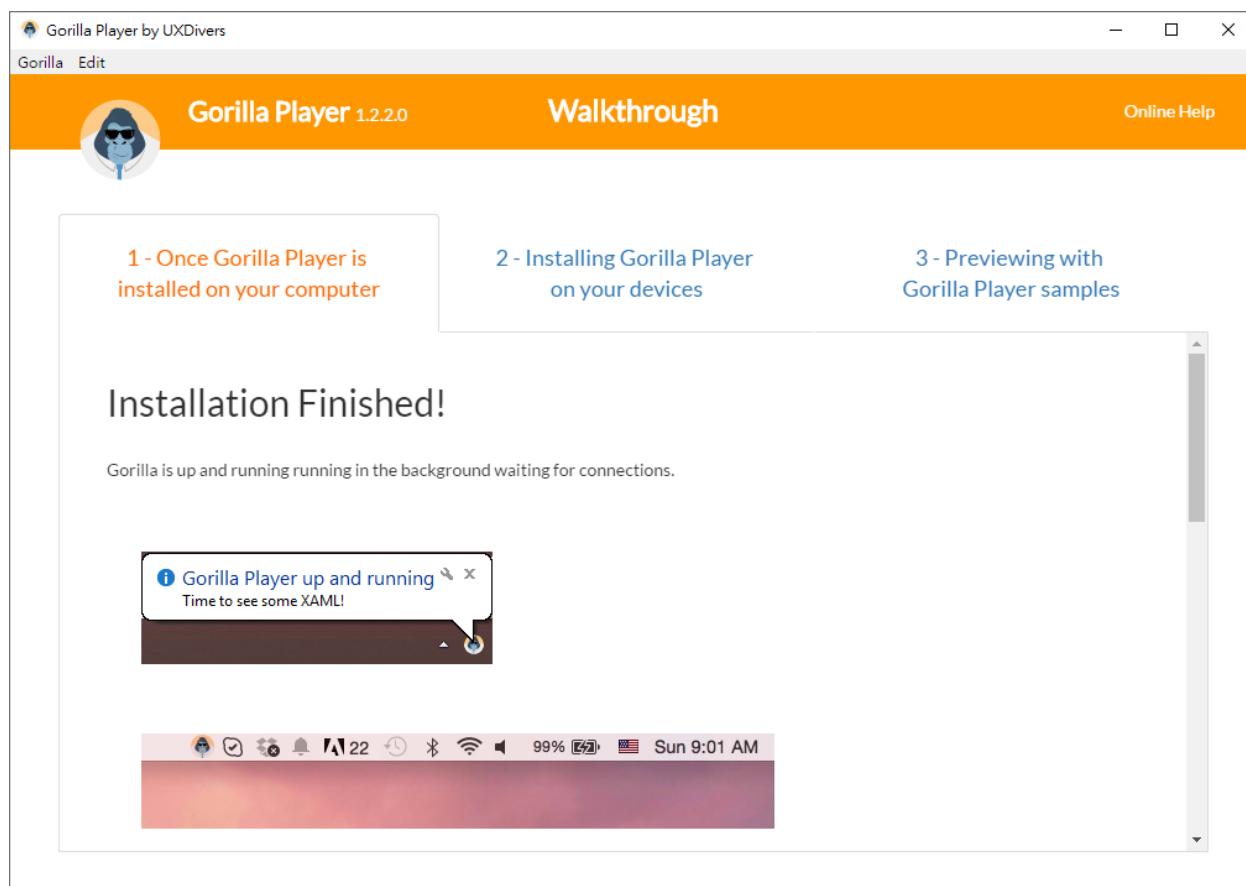
登入 Gorilla 帳號

當登入驗證成功之後，將會要求您授權防火牆存取授權，這裡一定要點選允許存取按鈕，否則，大金剛 Gorilla 預覽工具整個系統將不會正常運作。



授權防火牆存取授權

現在，您將會看到如下圖的畫面，您會看到它會告訴我們說已經把 Gorilla Player 桌機代理程式安裝到這台電腦上了，您可以桌面右下方的圖示，看到 Gorilla Player 代理程式已經成功啟動了。



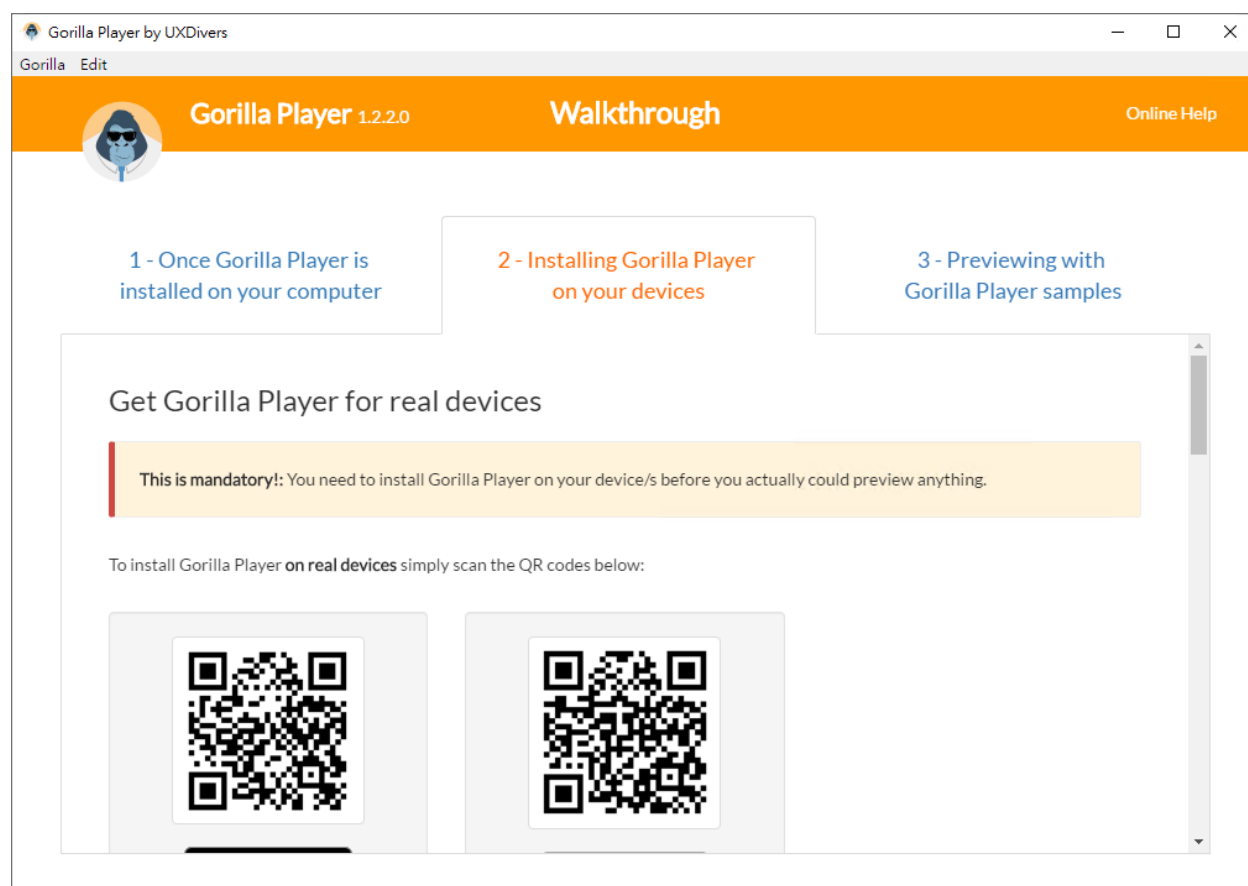
Gorilla Player by UXDivers 程式



Gorilla 桌機代理程式

2.1.2 裝置端軟體安裝

現在，我們需要把 Gorilla Player 在模擬器或者實體裝置上的軟體安裝好，您可以透過從 [Apple Store](#)² 或者 [Google Play](#)³ 上來下載 Gorilla Player 程式。

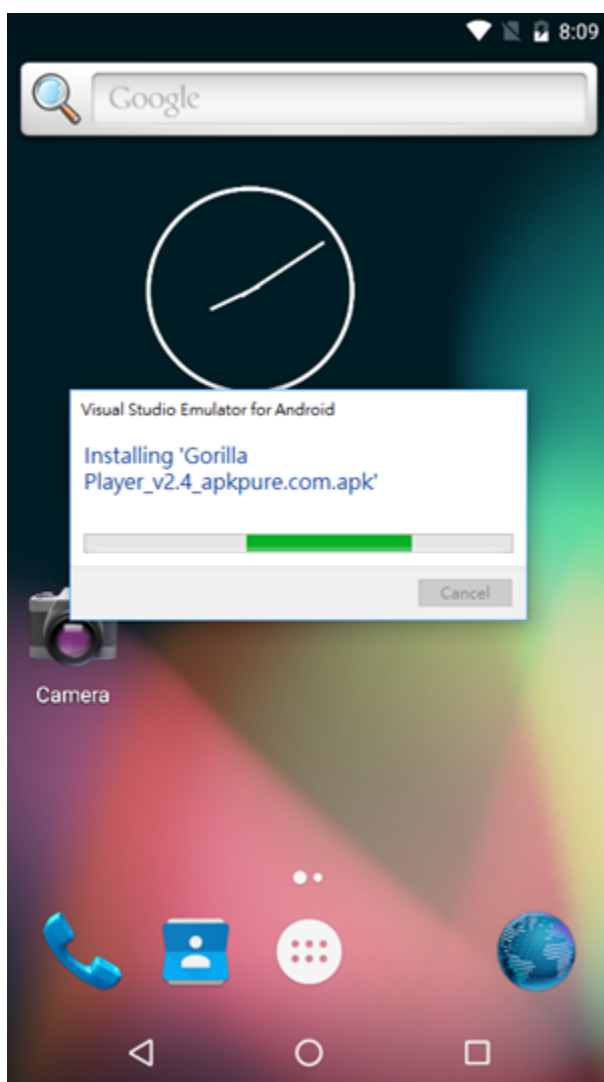


安裝 Gorilla 撥放器 Player 到模擬器或裝置上

對於模擬器而言，您可以透過 Google 搜尋引擎來搜尋 [gorilla player apk] 關鍵字，找到可以下載的 APK 檔案，並且將這個 APK 檔案拖拉到 Android 模擬器上，此時，您會看到如下圖畫面，知道 Gorilla Player 正在安裝到模擬器上。

²<https://goo.gl/G72G9p>

³<https://goo.gl/6Bbbtv>



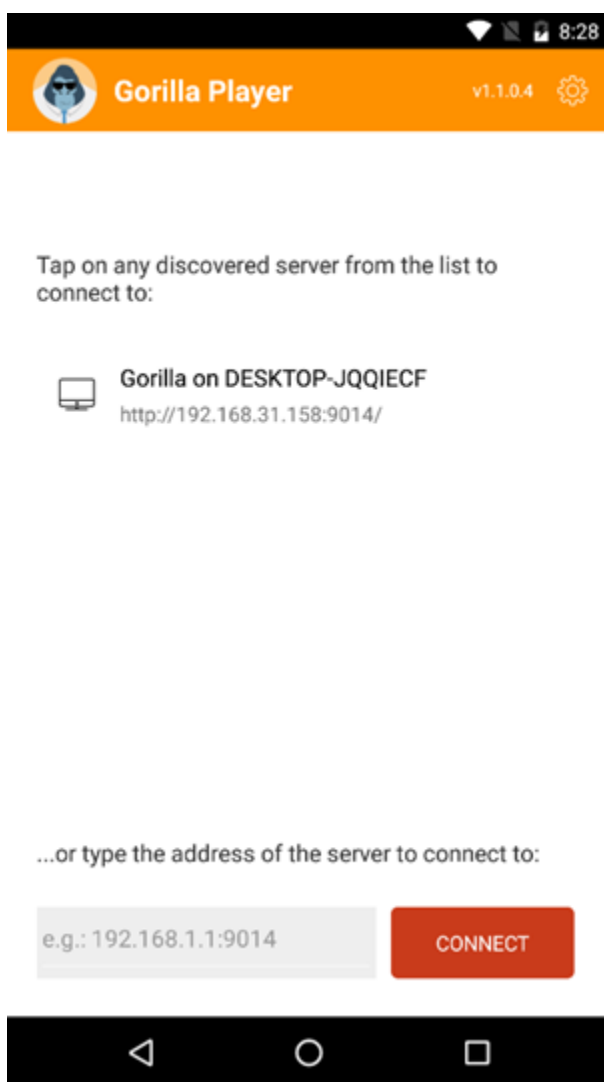
安裝 Gorilla 撥放器 Player 到模擬器

當安裝完成後，您會看到模擬器或者實體裝置上會出現一個 Gorilla Play 這個應用程式，請開啟這個應用程式。



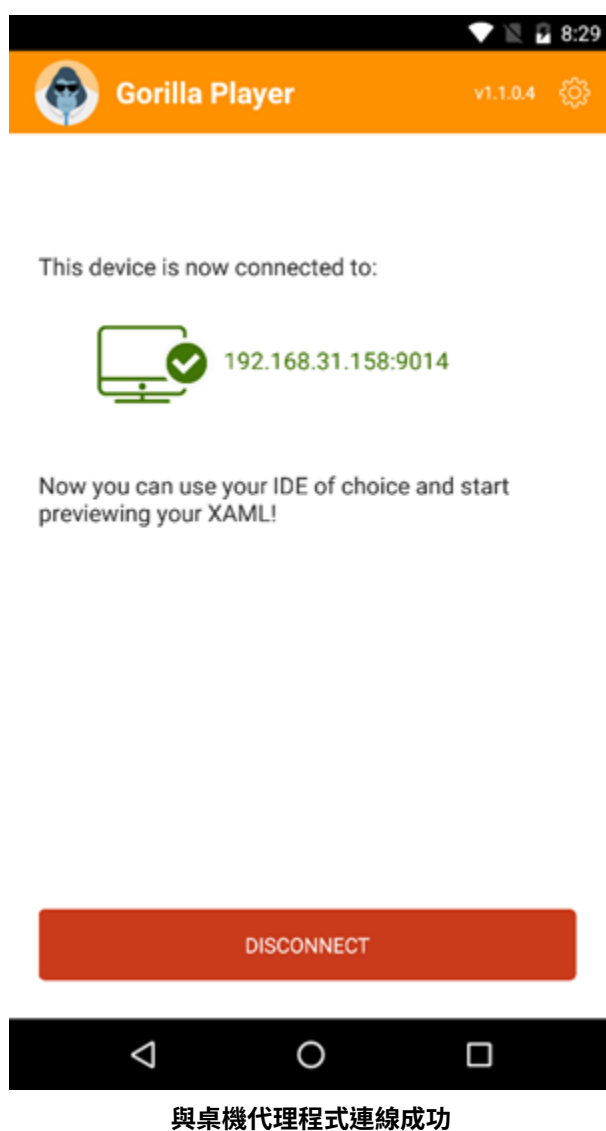
啟動 Gorilla Player App

您將會看到模擬器或者實體裝置上執行的 Gorilla Player 程式已經偵測到您電腦上運行的代理程式，請點選這個項目。



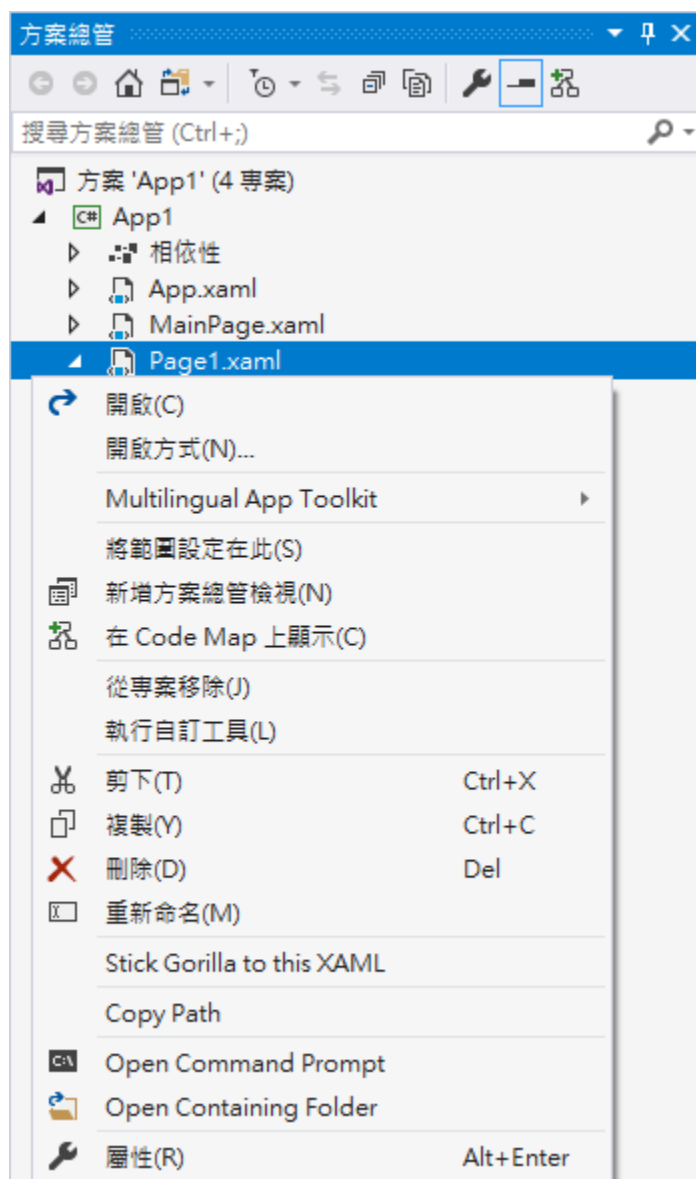
與桌機代理程式連線

一旦可以成功連線，將會看到如下圖畫面。



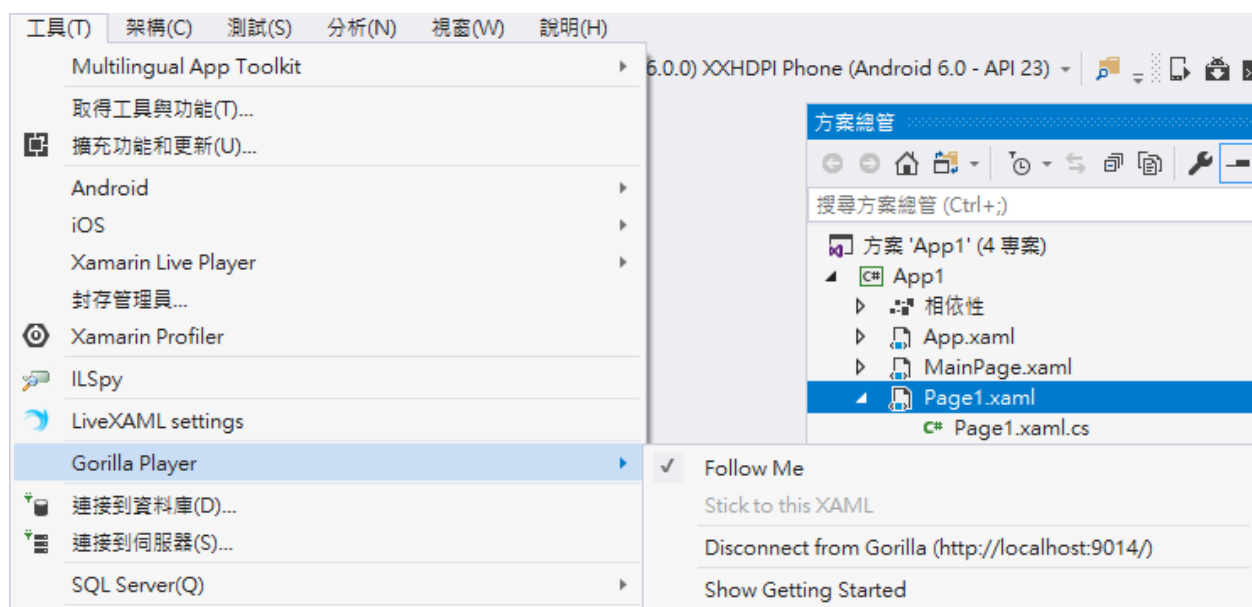
2.1.3 體驗如何使用大金剛 Gorilla 預覽工具

現在，您可以建立一個 Xamarin.Forms 專案或者打開舊的 Xamarin.Forms 專案。當您要預覽這個 XAML 最後執行效果，您可以使用個別指定要預覽 XAML 文件檔案的方式，您僅需要使用滑鼠右擊該 XAML 文件檔案，接著在彈出功能表上選擇 **Stick Gorilla to this XAML** 選項。



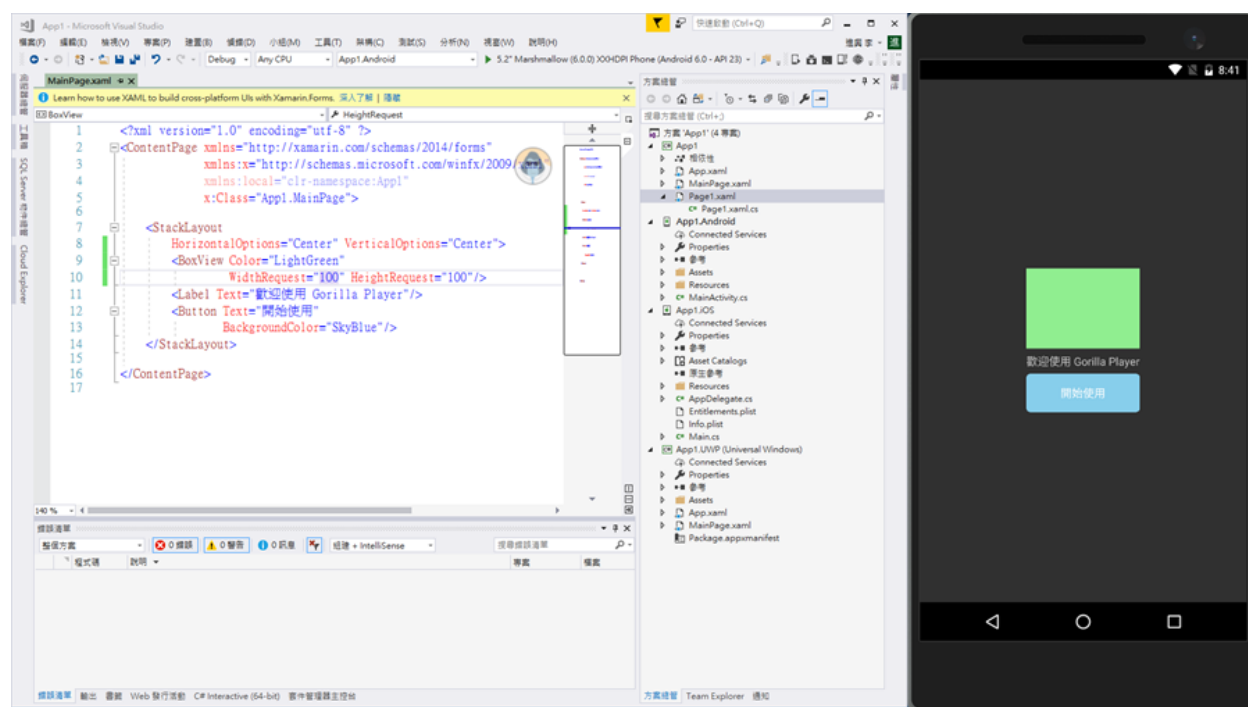
指定特定 XAML 文件檔案啟用預覽功能

或者，您希望只到打開一個 XAML 文件檔案之後，就直接可以進行預覽功能，此時，請點選 Visual Studio 功能表的 [工具] > [Gorilla Player] > [Follow Me] 選項即可。



啟用所有 XAML 文件檔案自動預覽

現在，您將會在模擬器或者實體裝置上看到這個 XAML 文件真實執行結果。



使用大金剛 Gorilla 預覽工具效果

2.2 學習 XAML 的最佳良伴 LiveXAML

關於 LiveXAML 這個工具，在 [LiveXAML 官方網頁](http://www.livexaml.com/)⁴ 上是這麼描述的：LiveXAML 不僅僅只是另一套編輯 XAML 的預覽工具，更是協助在開發 App 時 XAML 就能直接活靈活現的好幫手！

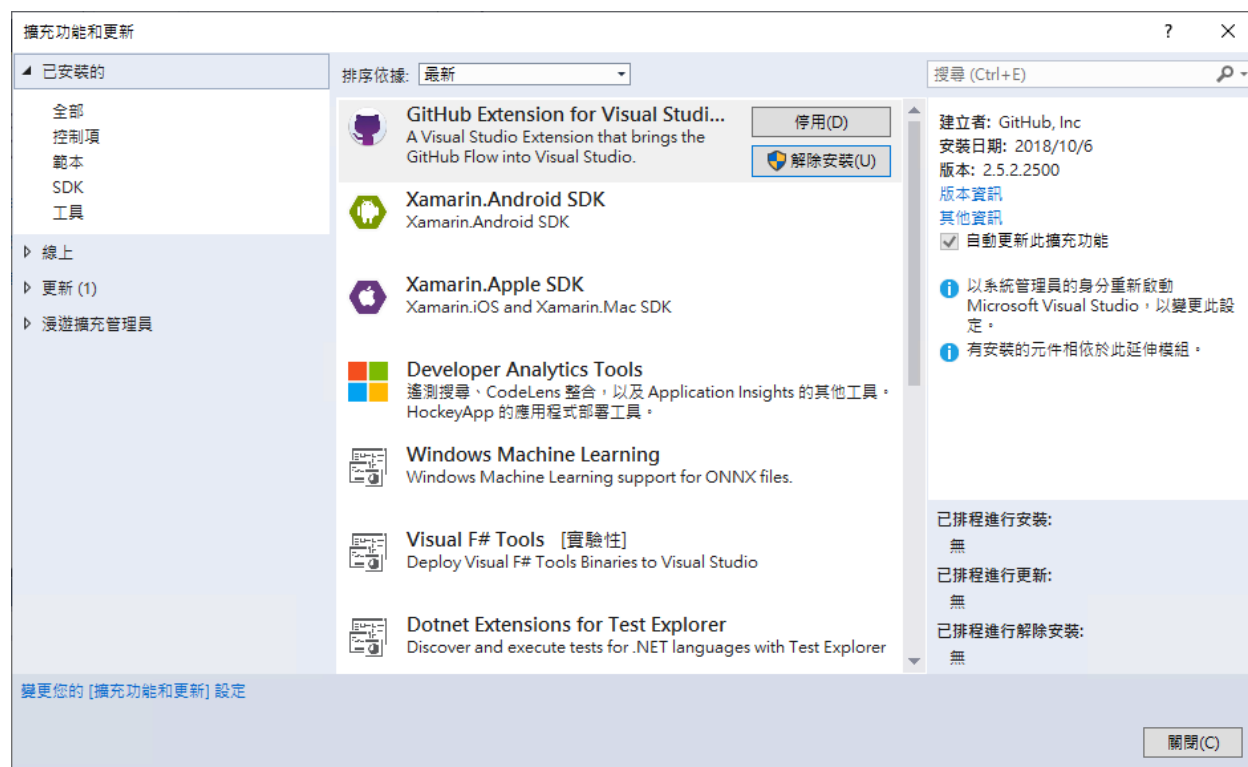
不過，我要先在此提出聲明，這個擴充功能套件不是免費的軟體，這是需要付費的產品，但是，當您安裝與設定完成之後，可以擁有 14 天的完整功能使用權；當超過 14 天之後，若您沒有購買 LiveXAML 的 license 授權，則您僅能夠在單一專案中編輯不超過 3 個 XAML 檔案，然而，這樣的限制並不會對於您使用這個擴充功能來學習這本書讓的相關知識與技能，因為，我們大部分的範例專案中使用到的 XAML 檔案數量都是小於 3 個，您可以放心使用試用版本所提供的功能。

⁴<http://www.livexaml.com/>

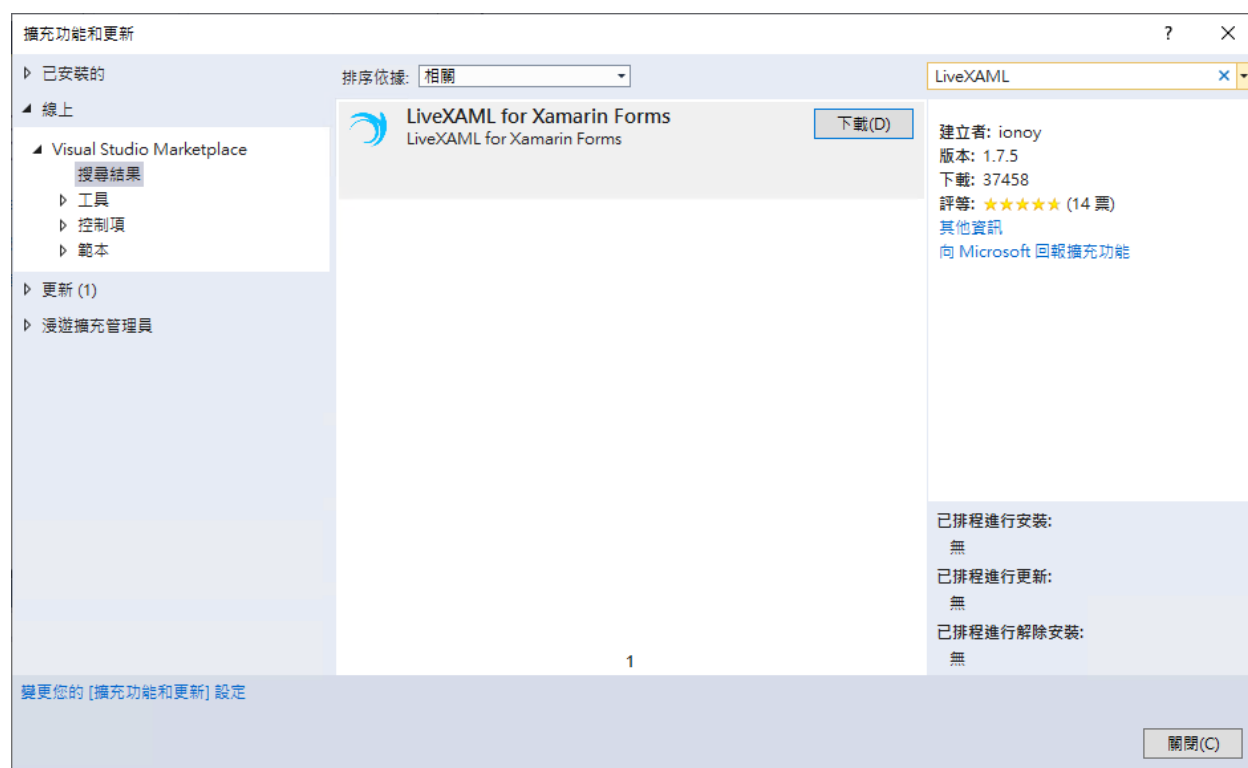
2.2.1 安裝與註冊 LiveXAML 擴充功能套件

現在，讓我們來將 LiveXAML 擴充功能套件，安裝到 Visual Studio 2017 內，體驗看看他有甚麼好處的？

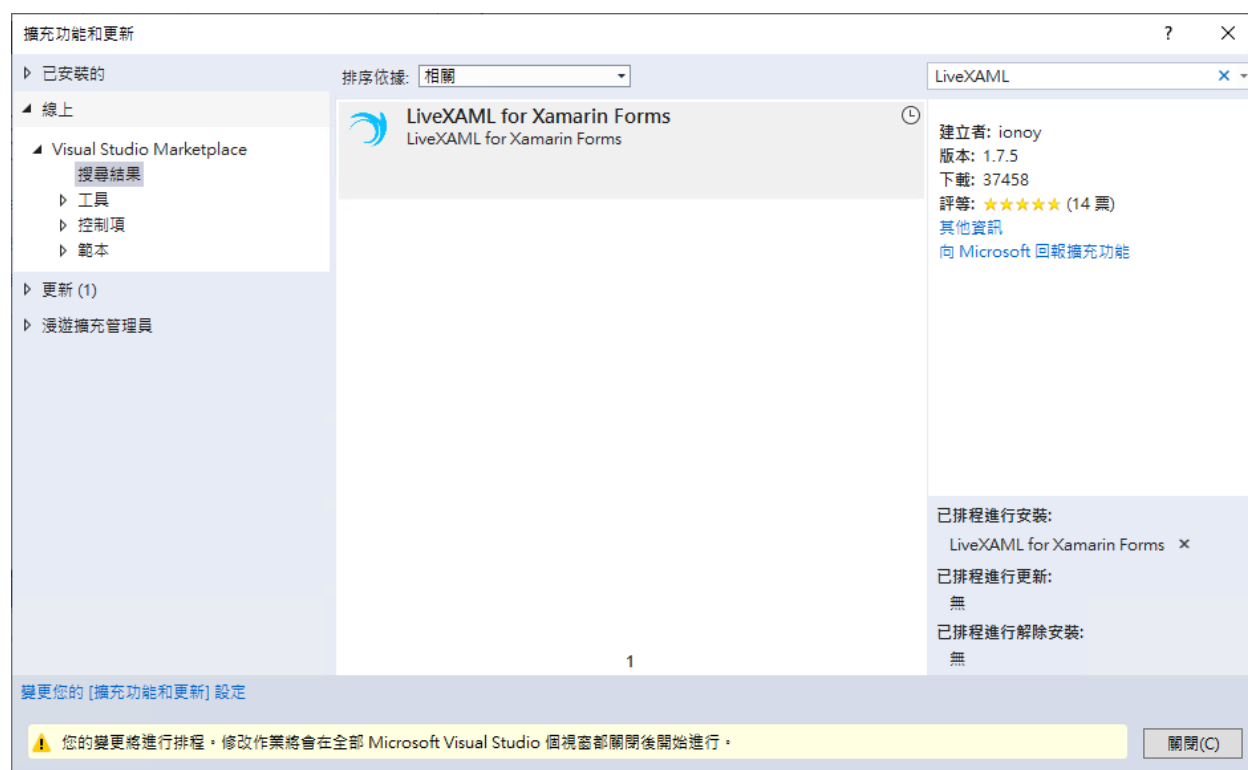
- 請啟動 Visual Studio 2017
- 點選 Visual Studio 2017 功能表的 [工具] > [擴充功能和更新] 選項
- 此時擴充功能和更新對話窗將會出現



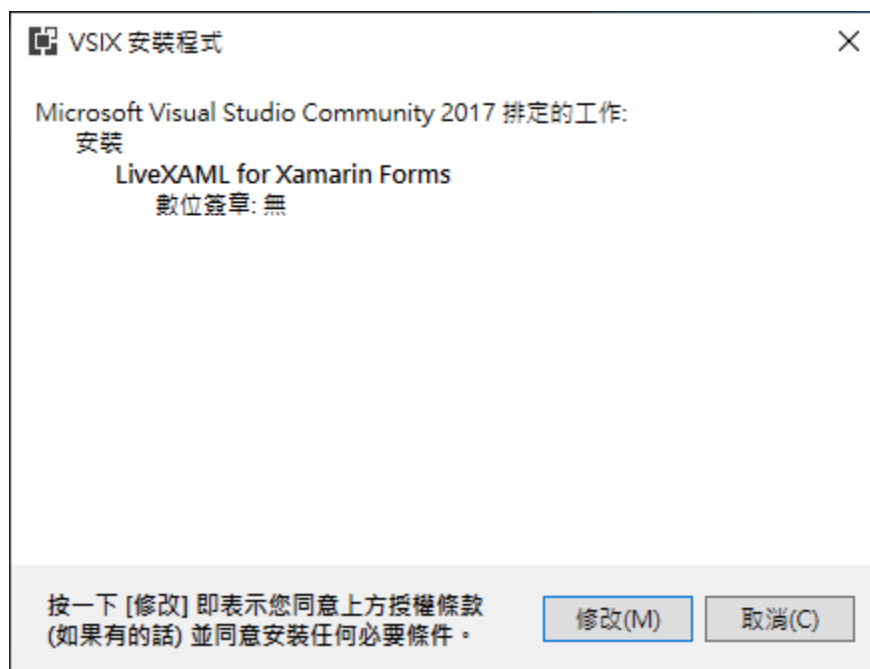
- 請點選擴充功能和更新對話窗左方的線上選項，接著在右上方的搜尋文字輸入盒內輸入 **LiveXAML** 關鍵字



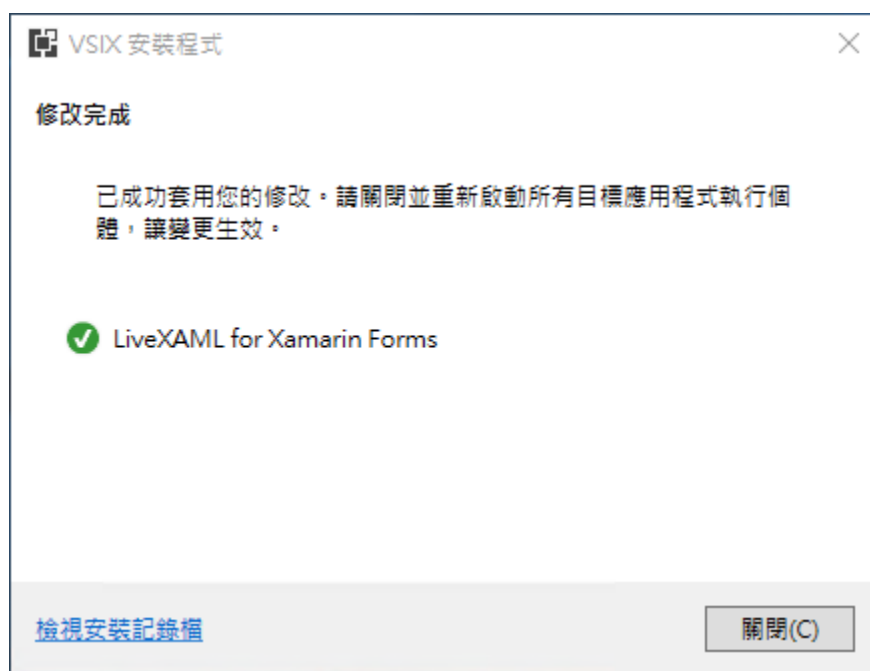
- 此時，請點選搜尋出來的 **LiveXAML for Xamarin.Forms** 右邊的下載按鈕，進行下載與安裝這個 Visual Studio 擴充功能。
 - 安裝完成後，您將會看到該對話窗的最下方會顯示一段提示訊息：您的變更將進行排程，修改作業將會再全部 Microsoft Visual Studio 個視窗都關閉後開始進行。
- 因此，請將 Visual Studio 程式關閉起來



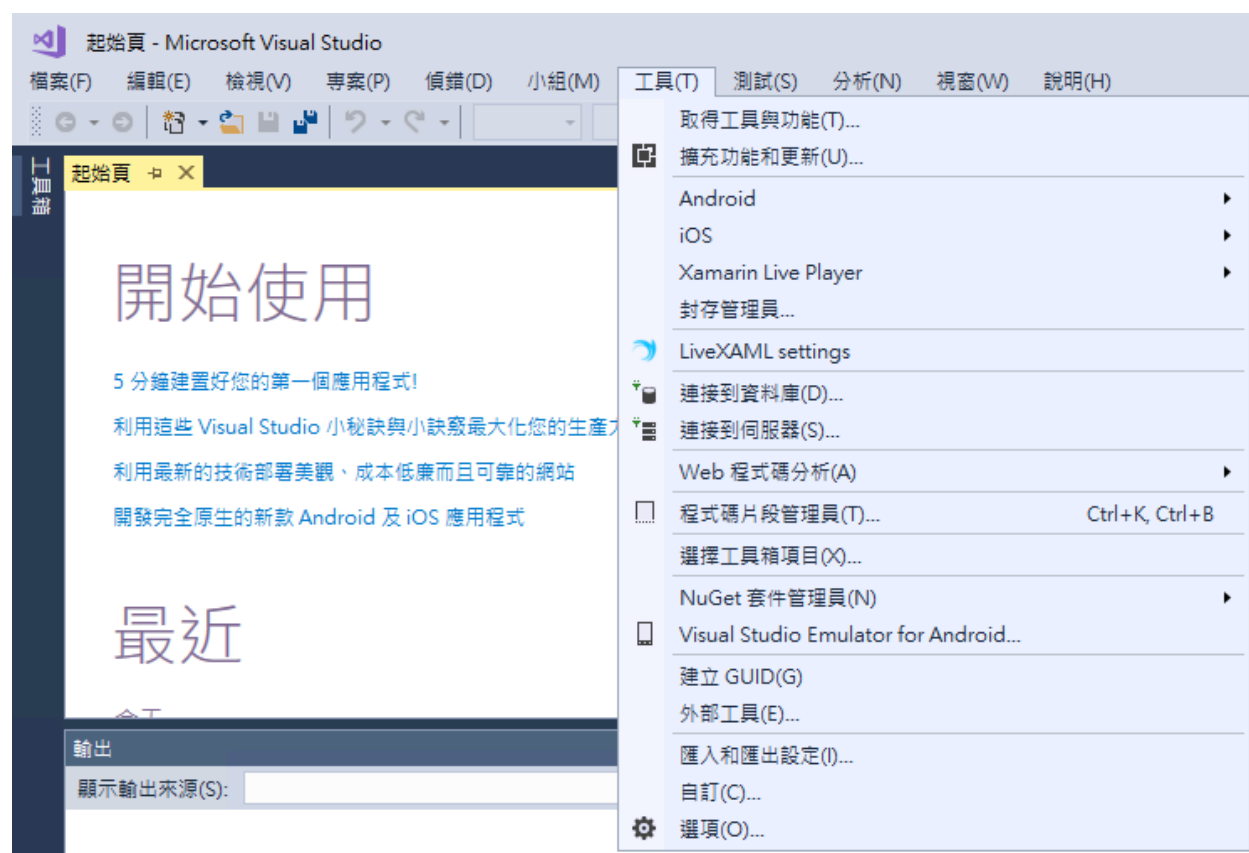
- 當您關閉 Visual Studio 2017 程式之後，將會顯示底下 VSIX 安裝程式對話窗，請點選修改按鈕，以便開始進行安裝這個擴充功能。



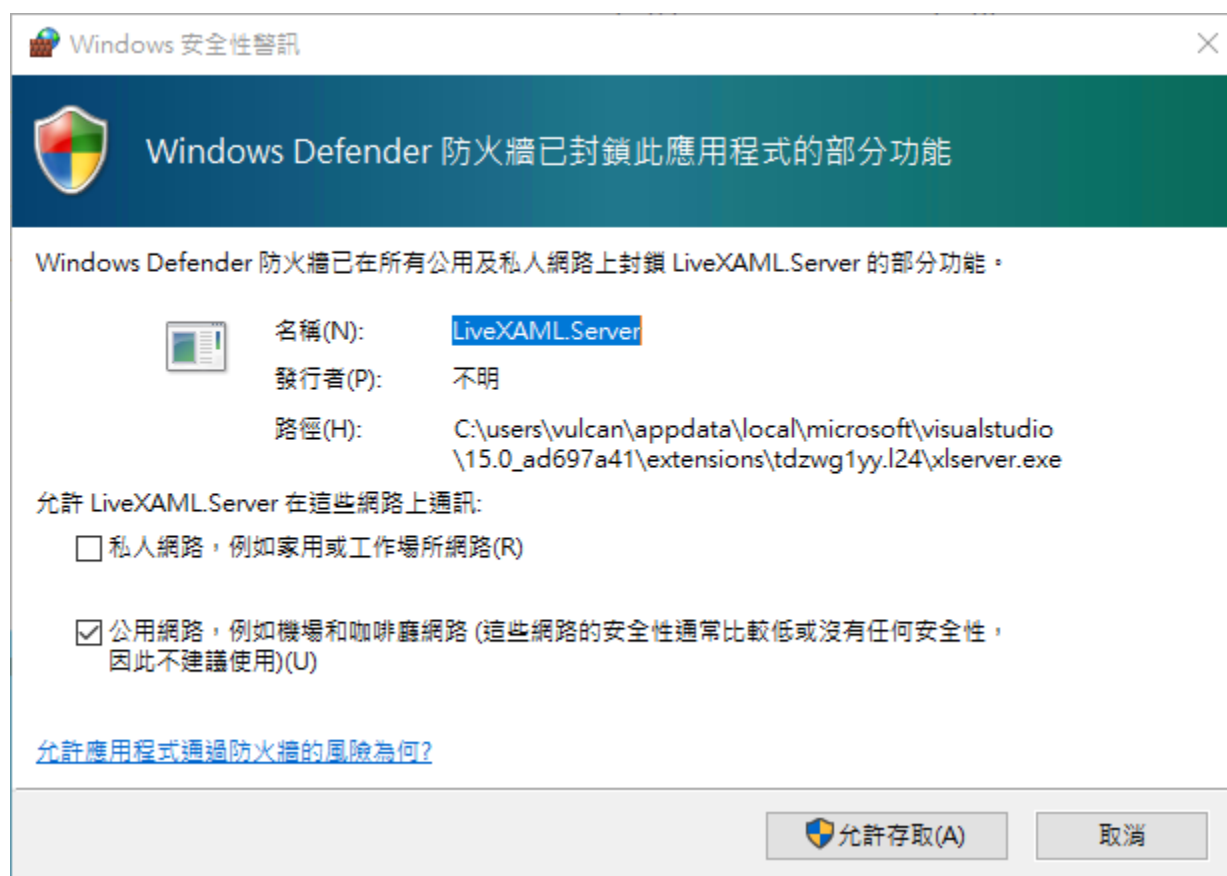
- 擴充功能安裝完成之後，將會看到底下對話窗，告訴您 LiveXAML for Xamarin Forms 這個擴充功能套件已經安裝好了，請點選右下方的關閉按鈕，並且重新開啟 Visual Studio 2017。



- Visual Studio 2017 重新開啟後，請點選功能表的 [工具]，您將會看到工具子功能表清單內有個 LiveXAML settings 這個選項，請點選這個選項來準備進行 LiveXAML 的設定與註冊工作。



- 首先會先看到 Windows 安全性警訊對話窗，您需要授權 LiveXML Server 的網路存取功能，否則，您準備要使用 LiveXML 將無法正常運作，因此，請點選該對話窗右下角的允許存取按鈕。



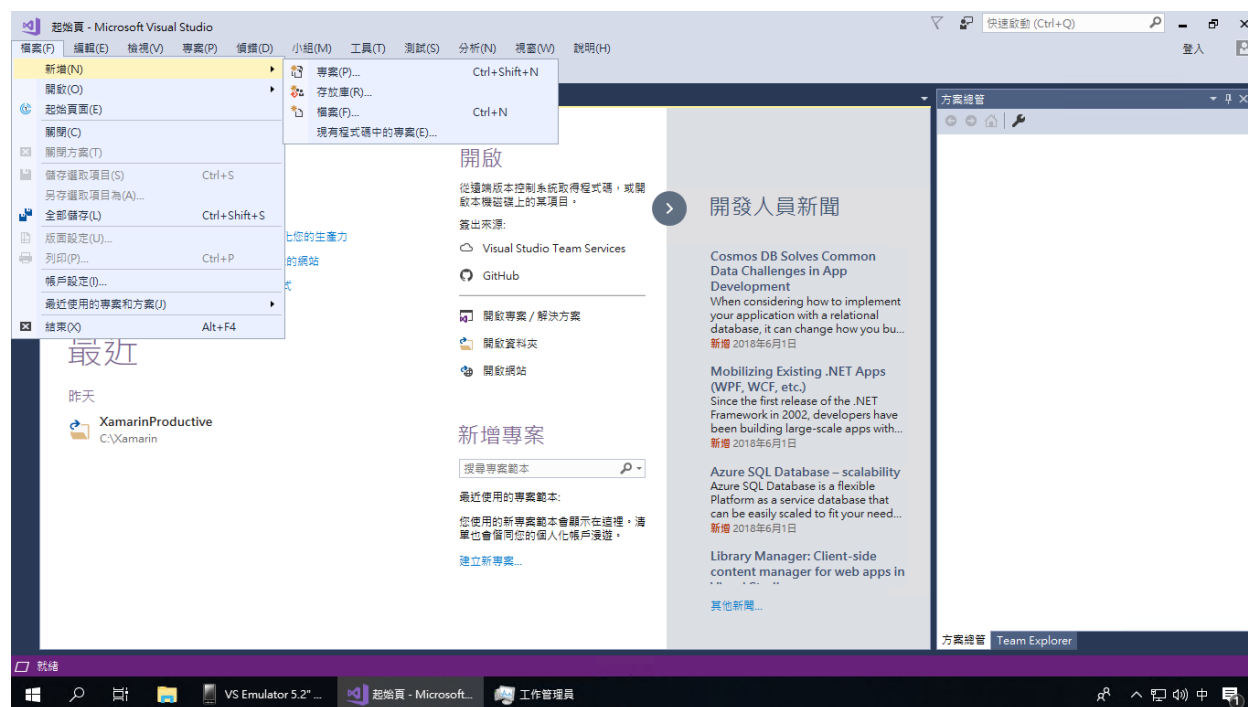
- Visual Studio 2017 重新開啟後，請點選功能表的 [工具]，您將會看到工具子功能表清單內有個 LiveXAML settings 這個選項，請點選這個選項來準備進行 LiveXAML 的設定與註冊工作。

只要您看到這個畫面，就代表 LiveXAML 這個擴充功能已經安裝好了；若您有購買這套 LiveXAML 的使用授權，您將會收到啟用這個擴充功能的 License Email 與 Password，您只需要輸入在左下方藍色方塊內，最後點選取得 License 按鈕，如此，就不會有 14 天可以試用的限制了。



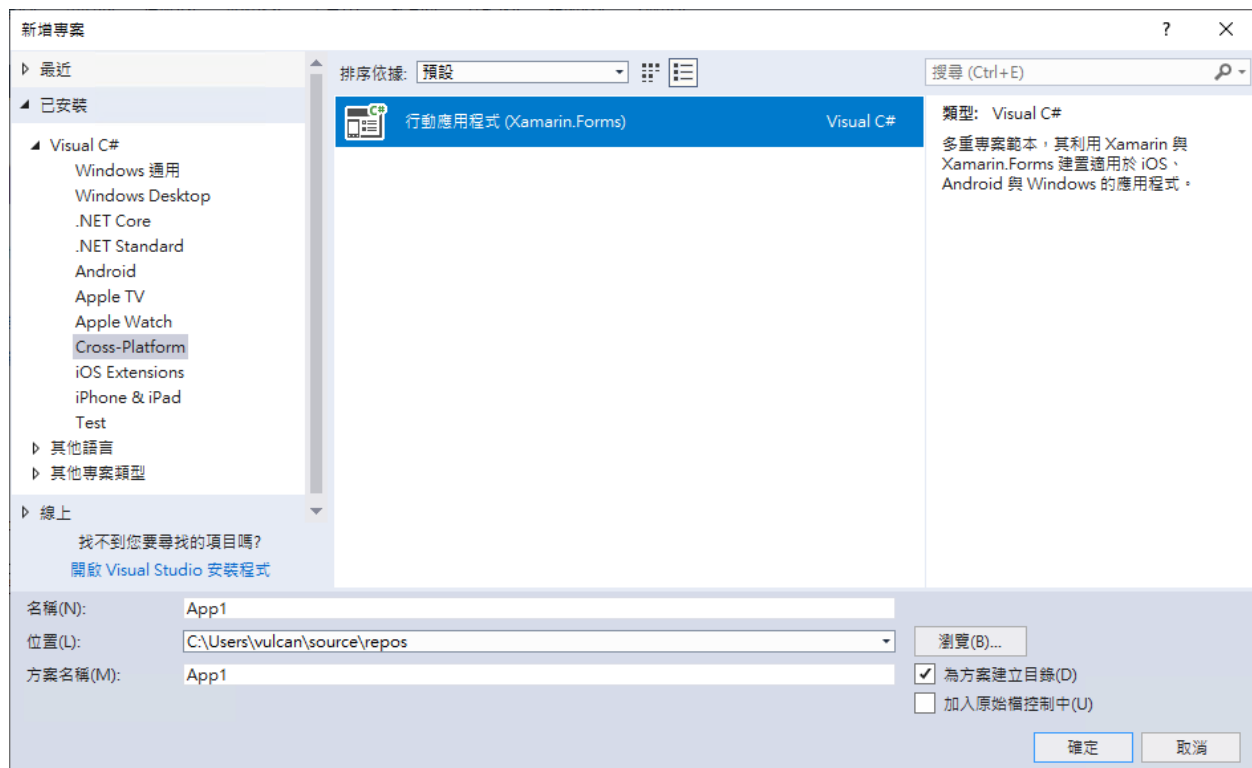
2.2.2 體驗如何使用 LiveXAML

- 現在，讓我們看看要如何使用 LiveXAML 所帶來的神奇功能，請先關閉剛剛開啟的 LiveXAML 對話窗
- 請在開啟 Visual Studio 2017 程式，並且，點選功能表
檔案 > 新增 > 專案



Visual Studio 2017 建立新的專案

- 在新增專案對話窗中，請點選該對話窗左方的 Cross-Platform 項目
在中間選項中，點選行動應用程式 (Xamarin.Forms)，最後點選對話窗右下方的確定按鈕



新增專案對話窗

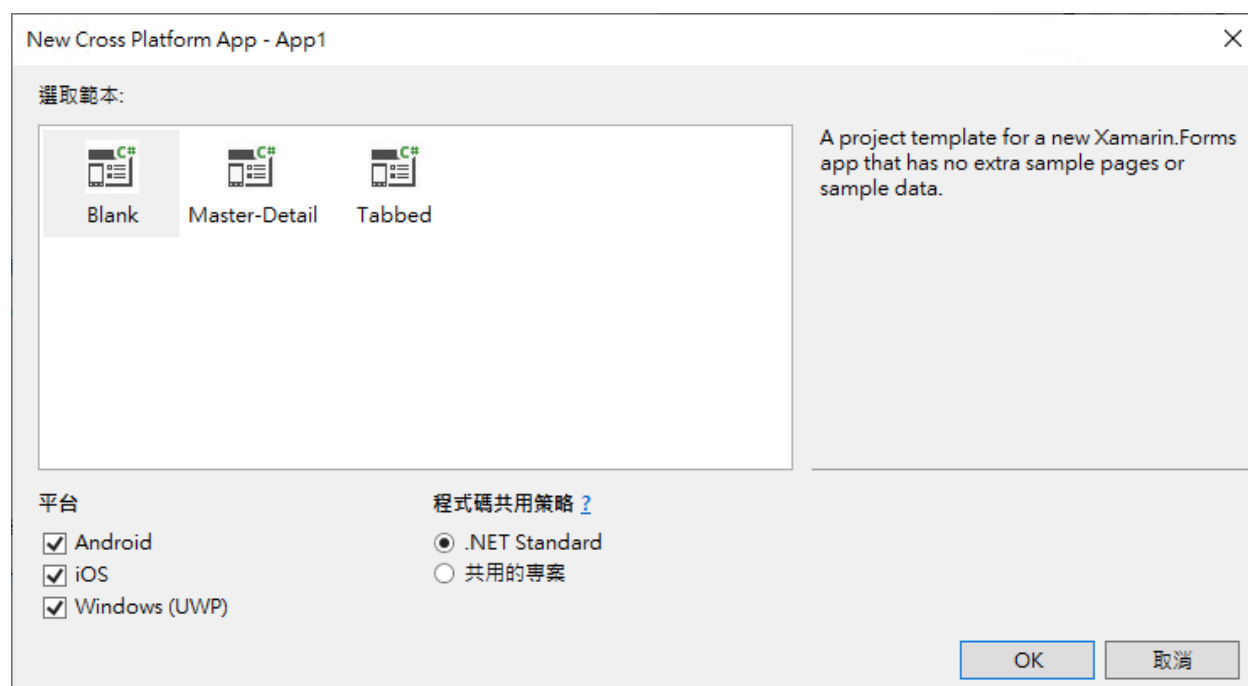
- 出現了 New Cross Platform App 對話窗

請確認在選取範本中，點選 Blank

下方的平台選項，請記得 Android / iOS / Windows (UWP) 請都要勾選

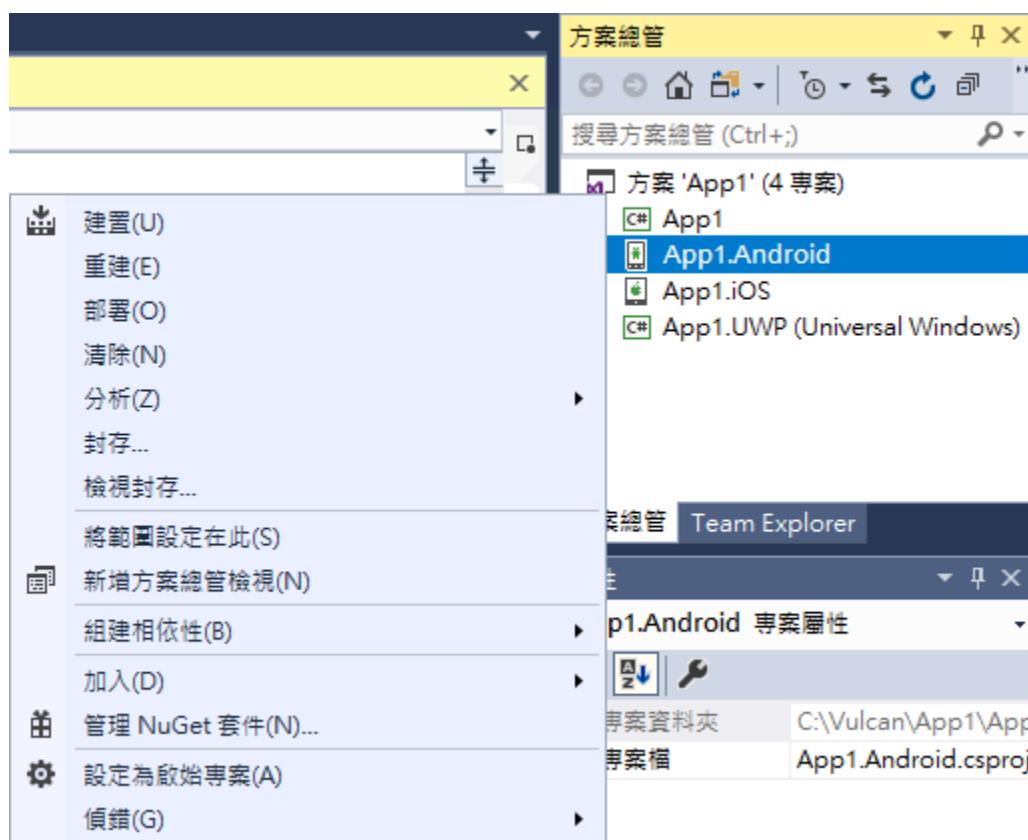
在程式碼共用策略的下方，請選擇 .NET Standard

請點選該對話窗右下方的 OK 按鈕



New Cross Platform App 對話窗

- 當 Xamarin.Forms for Prism 專案建立完成後，請使用滑鼠右擊 Android 專案選擇設定為起始專案



在方案總管中設定預設起始專案

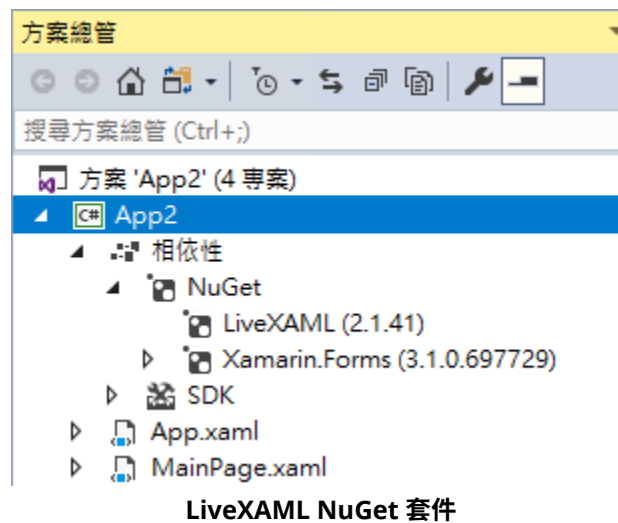
- 請在 Visual Studio 2017，點選功能表的 [工具] > [LiveXAML settings] 這個選項，此時，LiveXAML 對話窗會顯示在螢幕上。

請點選啟用自動模式，針對 App1 這個綠色按鈕，讓 LiveXAML 擴充功能幫我們安裝 LiveXAML 需要用到的 NuGet 套件，到這個 Xamarin.Forms 專案內



LiveXAML 設定對話窗

- 請關閉 LiveXAML 對話窗，到方案總管視窗中，針對 Xamarin.Forms 專案展開其相依性這個節點，接著展開 NuGet 節點，此時，您會看到 LiveXAML 這個 NuGet 套件已經安裝到 Xamarin.Forms 的專案內了。

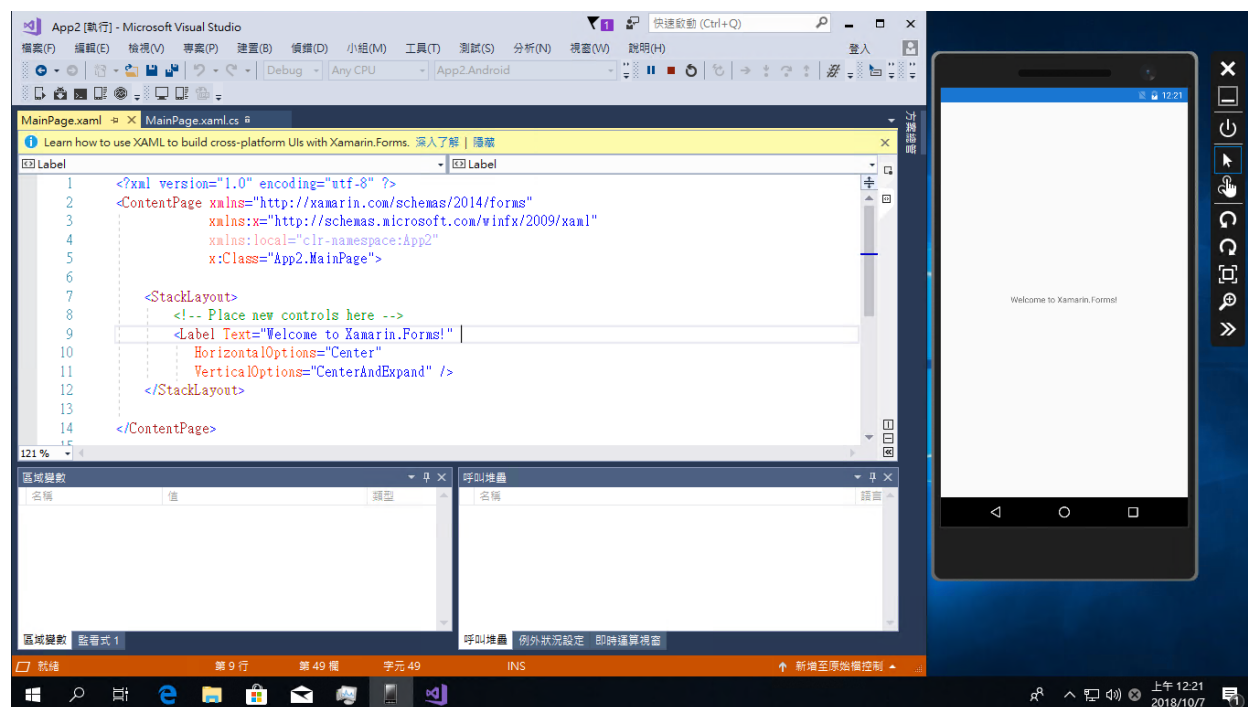


- 現在，您會在 Visual Studio 2017 中間上方區域，看到啟動專案已經設定為 Android 專案，並且要執行的裝置為 5.2" Marshmallow (6.0.0) XXHDPI Phone (Android 6.0 API 23) 這個模擬器

請點選有綠色三角形的模擬器項目，開始建置與在這個模擬器上來執行

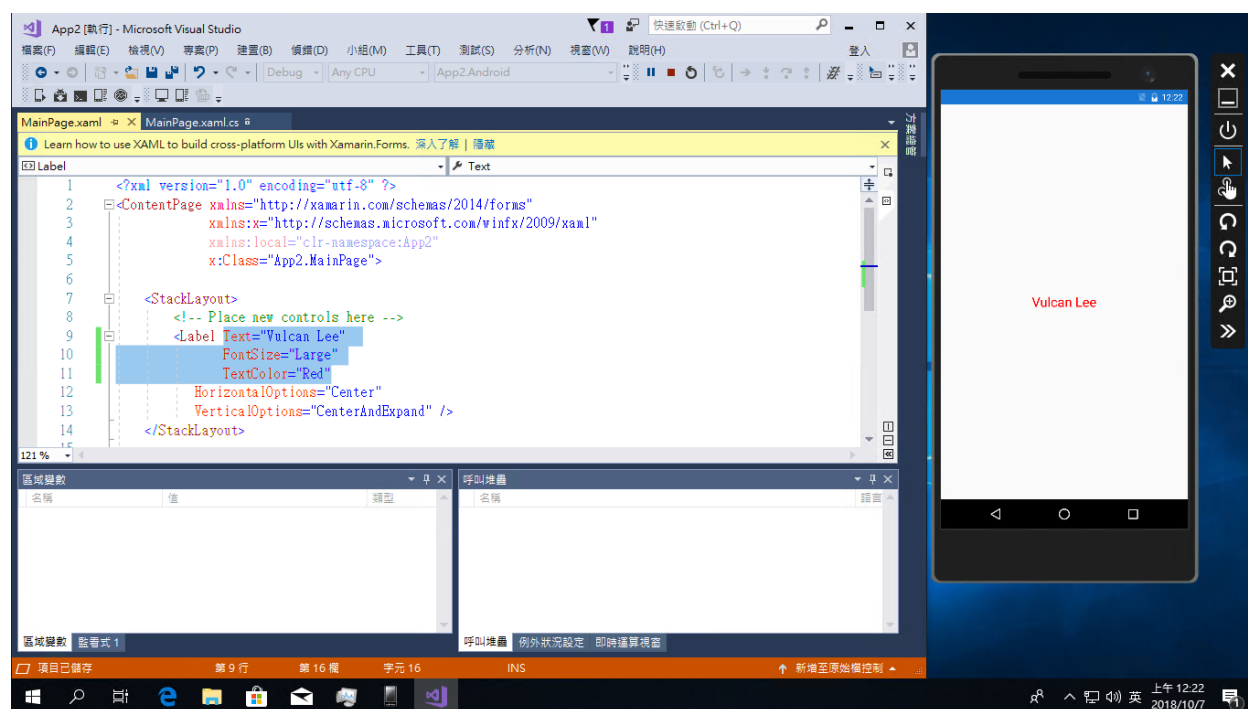


- 我們將會看到這個應用程式在模擬器上執行的結果



Xamarin.Forms 的 Android 專案成功執行結果

- 現在，我們不要停止執行這個專案，我們將要針對文字標籤 Label 這個檢視項目進行修正：
 - 直接修正 Text 的屬性值為 "Vulcan Lee" 這個字串
 - 增加 FontSize 這個屬性與設定該屬性值為 "Large" 字串
 - 增加 TextColor 這個屬性與設定該屬性值為 "Red" 字串
- 修改後的內容，將會如同底下截圖的圈選區域所示
- 請在 Visual Studio 針對 MainPage.xaml 這個 XAML 文件檔案按下 Ctrl + S，把這個文件儲存起來，此時，您將會看到螢幕右方的 Visual Studio Emulator for Android 模擬器上面的內容，已經更新成為我們剛剛針對 MainPage.xaml 這個 XAML 文件檔案所做的修正了。



Xamarin.Forms 的 Android 專案修正更新結果

所以，LiveXAML 對於我們要學習 Xamarin.Forms 中的 XAML 語言與法與用法扮演著相當重要的腳色，當您在執行這個應用程式的時候，可以隨時進行各個頁面的 XAML 文件檔案的編輯，並且當您把這個 XAML 文件儲存起來的時候，我們可以馬上在這個專案執行的模擬器或者實體手機螢幕上，看到修正的結果，讓您學習 XAML 語言的用法，更加的輕鬆容易與方便。不過，再次強調，這個 LiveXAML 擴充功能套件，是需要付費購買的。

2.3 Xamarin 即時重新載入 Live Reload

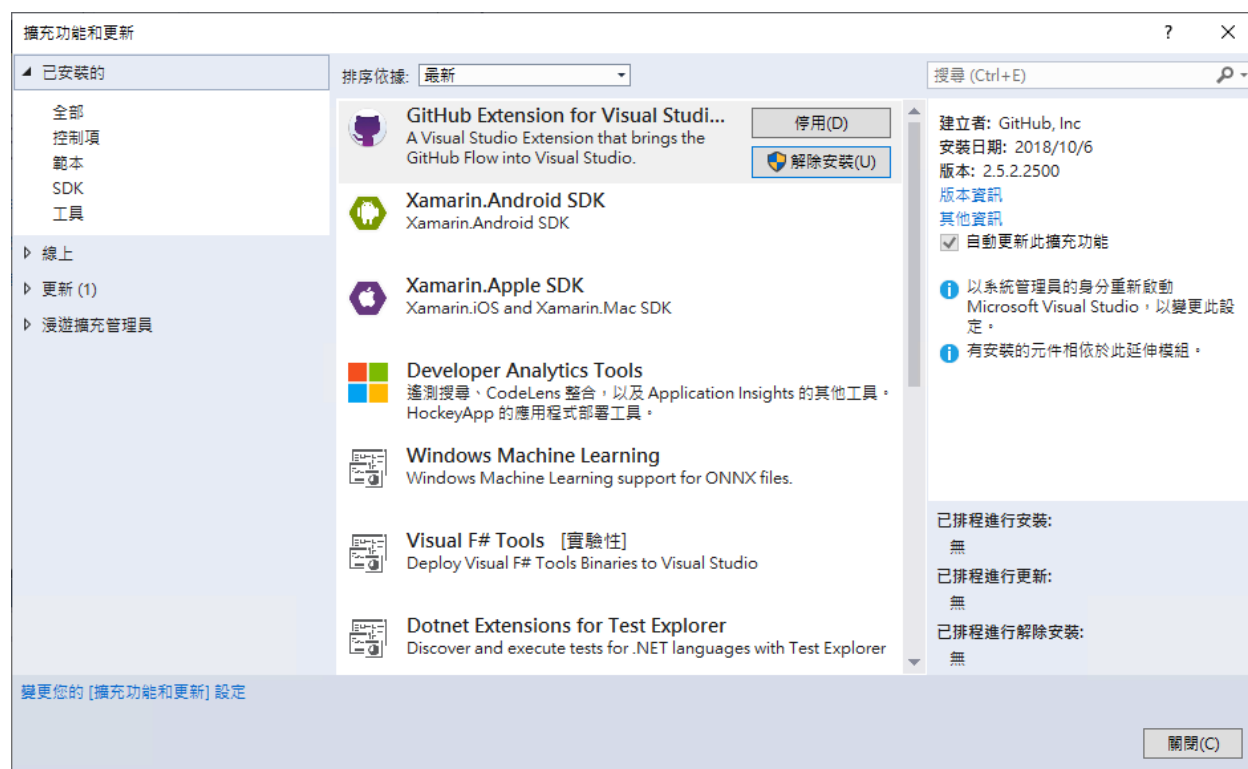
Xamarin Live Reload 這項服務是由 Xamarin 官方所提供的一個免費服務，現在還是 Preview 版本，他可以讓您在進行 Xamarin.Forms 專案開發的時候，讓您在除錯執行模式來進行 XAML 文件檔案的變更，並且在進行儲存變更的時候，便可以立即在執行的裝置上看到修正後的結果，也就是說，我們在修改 XAML 文件檔案的過程中，並不需要終止除錯，重新編譯專案與佈署專案到行動裝置上，才能夠看到修正的結果。

當使用 XAML 即時重新載入功能的時候，可以適用於所有的程式庫和協力廠商控制項，也能夠在 Android, iOS, UWP 專案下運行並且包含佈署到模擬器與實體裝置上。

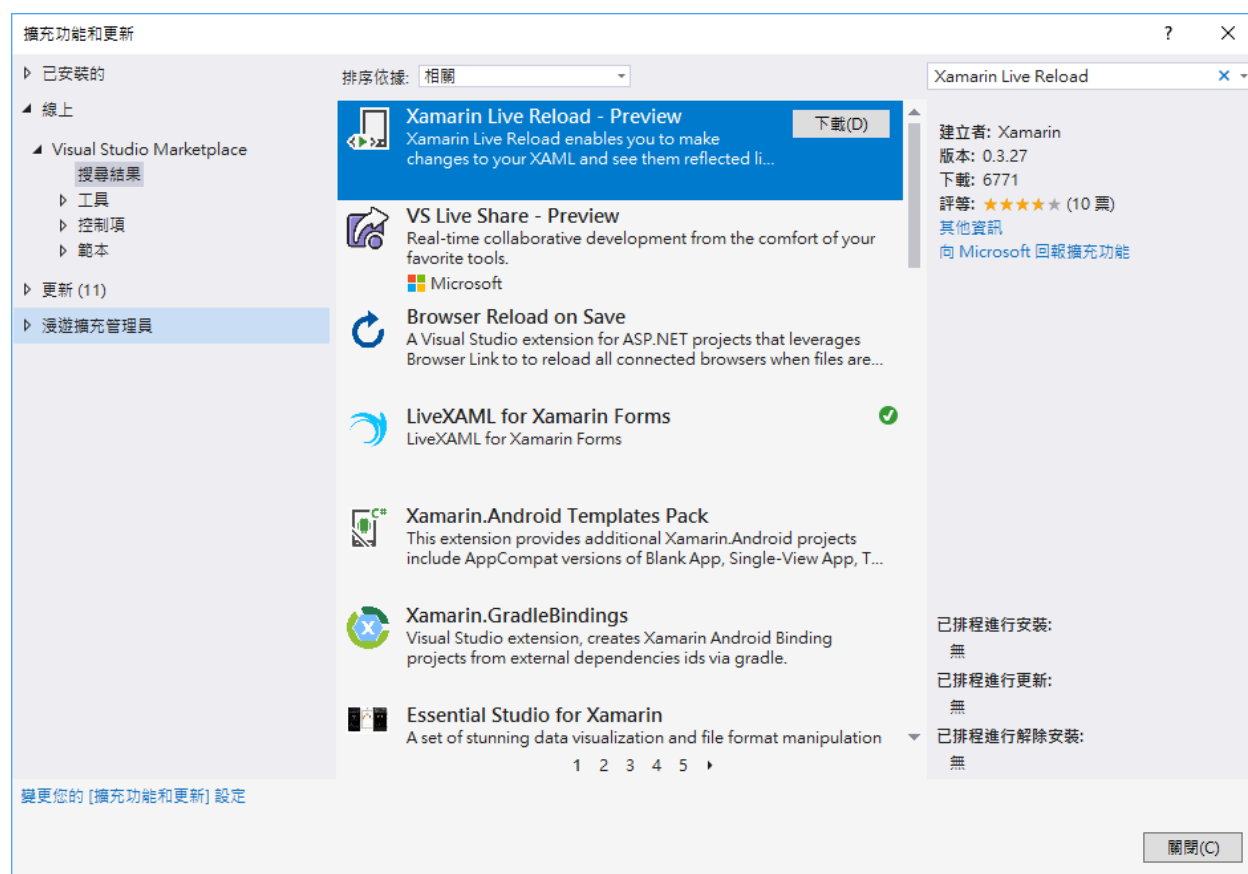
2.3.1 安裝與註冊 Xamarin 即時重新載入 Live Reload 擴充功能套件

現在，讓我們來將 Xamarin 即時重新載入 Live Reload 擴充功能套件，安裝到 Visual Studio 2017 內，體驗看看他有甚麼好處的？

- 請啟動 Visual Studio 2017
- 點選 Visual Studio 2017 功能表的 [工具] > [擴充功能和更新] 選項
- 此時擴充功能和更新對話窗將會出現



- 請點選擴充功能和更新對話窗左方的線上選項，接著在右上方的搜尋文字輸入盒內輸入 **LiveXAML** 關鍵字



- 此時，請點選搜尋出來的 **Xamarin Live Reload** 右邊的下載按鈕，進行下載與安裝這個 Visual Studio 擴充功能。
 - 安裝完成後，您將會看到該對話窗的最下方會顯示一段提示訊息：您的變更將進行排程，修改作業將會再全部 Microsoft Visual Studio 個視窗都關閉後開始進行。
- 因此，請將 Visual Studio 程式關閉起來



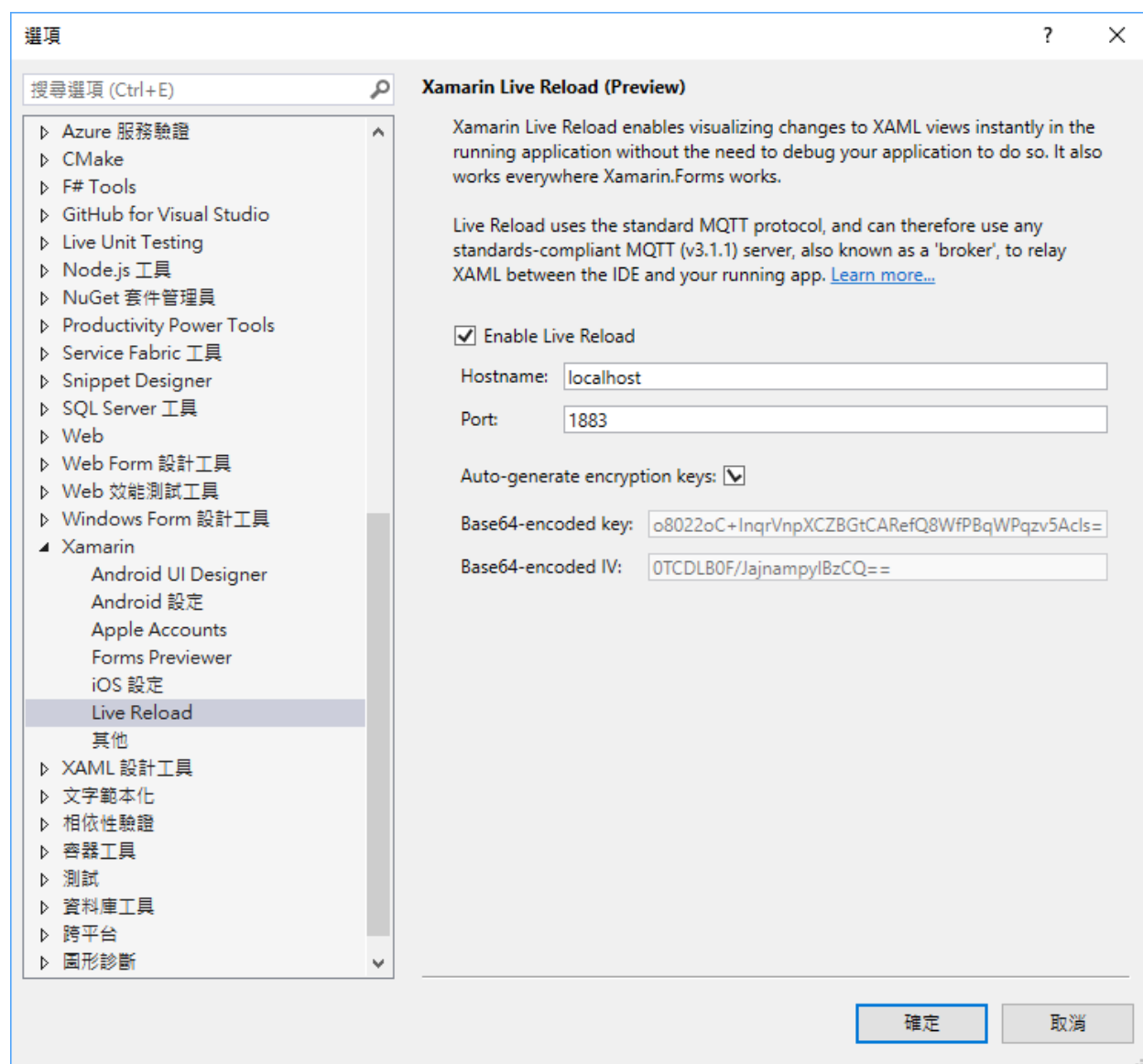
- 當您關閉 Visual Studio 2017 程式之後，將會顯示底下 VSIX 安裝程式對話窗，請點選修改按鈕，以便開始進行安裝這個擴充功能。



- 擴充功能安裝完成之後，將會看到底下對話窗，告訴您 LiveXAML for Xamarin Forms 這個擴充功能套件已經安裝好了，請點選右下方的關閉按鈕，並且重新開啟 Visual Studio 2017。



- Visual Studio 2017 重新開啟後，請點選功能表的 [工具] > [選項]，在選項對話窗的左半部捲動到 Xamarin 項目上，並且該項目展開，就會看到有個 Live Reload 項目，這表示您已經安裝好這個擴充功能了。

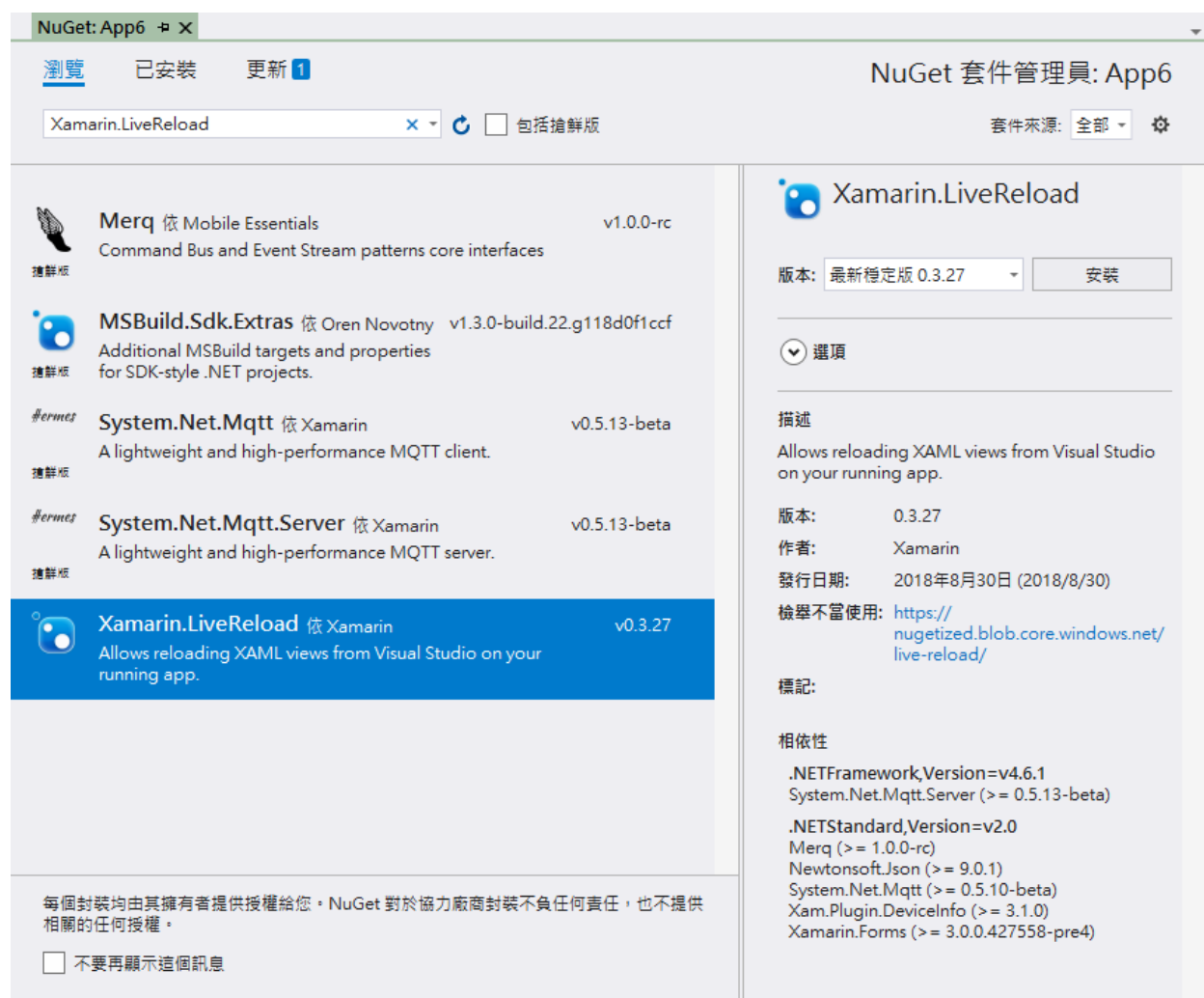


2.3.2 體驗如何使用 Xamarin 即時重新載入 Live Reload


設定您的應用程式使用即時重新載入即時重新載入加入現有的行動裝置應用程式可在三個步驟：請確定所有的專案已更新為使用 Xamarin.Forms 3.0.0 以上或更新版本。新增 Xamarin.LiveReload NuGet 套件：a. .NET standard –安裝 Xamarin.LiveReload 到您的.NET Standard 2.0 程式庫的 NuGet。這不需要安裝在您的平台專案。請確認套件

來源設為所有。b. 共用專案-安裝 Xamarin.LiveReload NuGet 的所有平台專案（例如 Android、iOS、UWP 等）。請確認套件來源設為所有。將 Xamarin 即時重新載入 NuGet 使用 NuGet 封裝管理員新增新增 LiveReload.Init(); 建構函式中 Application 類別，如下列程式碼片段所示：

- 現在，讓我們看看要如何使用 Xamarin 即時重新載入 Live Reload 所帶來的神奇功能
- 請在開啟 Visual Studio 2017 程式，並且，點選功能表
檔案 > 新增 > 專案
- 在新增專案對話窗中，請點選該對話窗左方的 Cross-Platform 項目
在中間選項中，點選行動應用程式 (Xamarin.Forms)，最後點選對話窗右下方的確定按鈕
- 出現了 New Cross Platform App 對話窗
請確認在選取範本中，點選 Blank
下方的平台選項，請記得 Android / iOS / Windows (UWP) 請都要勾選
在程式碼共用策略的下方，請選擇 .NET Standard
請點選該對話窗右下方的 OK 按鈕
- 當 Xamarin.Forms for Prism 專案建立完成後，請使用滑鼠右擊 Android 專案
選擇設定為起始專案
- 滑鼠右擊 Xamarin.Forms 專案，選擇 [管理 NuGet 套件]
- 當 NuGet 對話視窗出現後，請點選右上角的 [套件來源] 下拉選單，選擇 [全部] 選項
- 點選 [瀏覽] 標籤頁次，在搜尋文字輸入盒上，輸入 **Xamarin.LiveReload** 這個套件名稱
- 當出現如下圖畫面，請選擇 **Xamarin.LiveReload** 這個套件，並且點選 [安裝] 按鈕

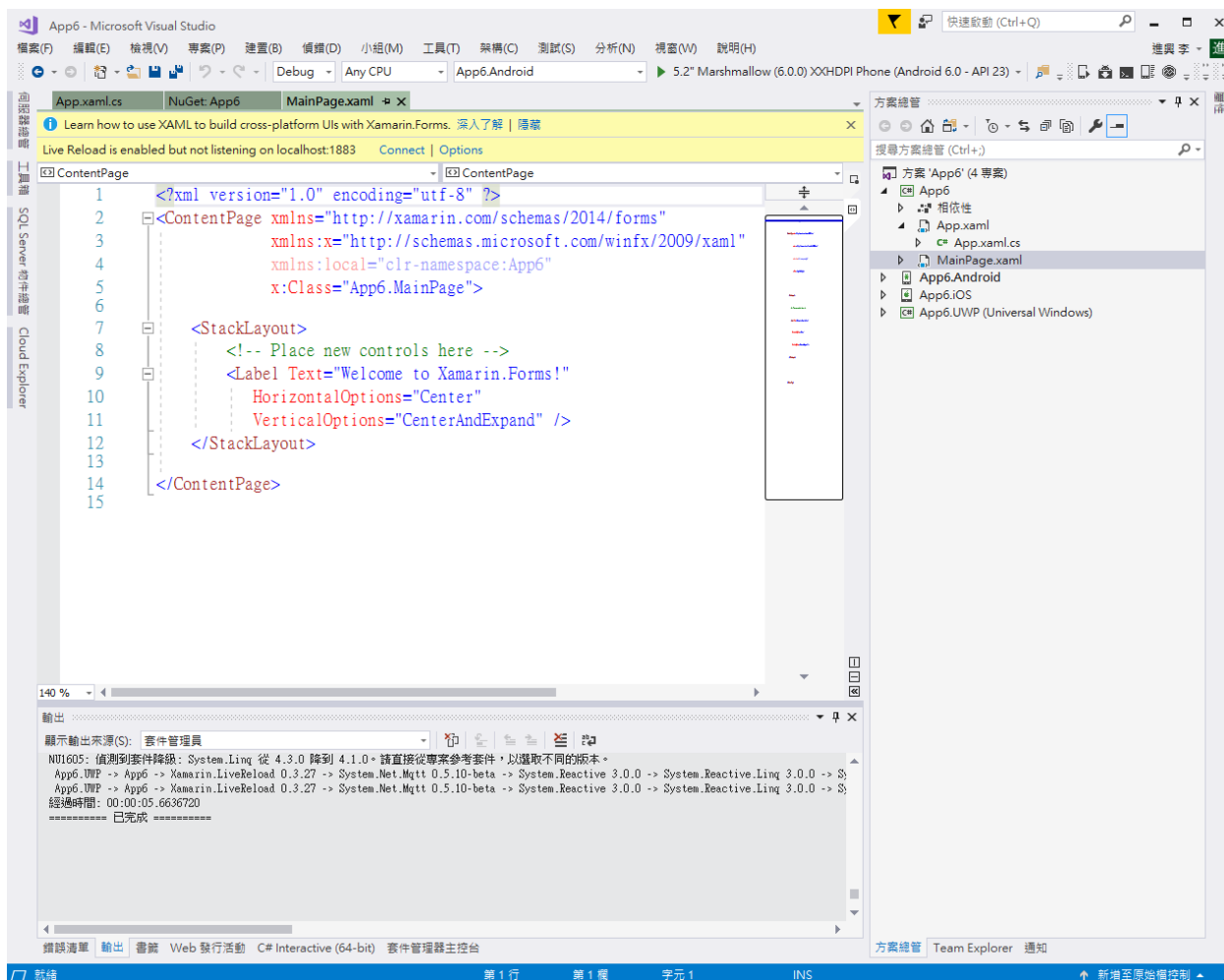


- 請在 Xamarin.Forms 專案內找到 App.xaml.cs 節點，打開這個內容，在 App 類別的建構函式內的 InitializeComponent(); 敘述前，加入底下程式碼 LiveReload.Init();。

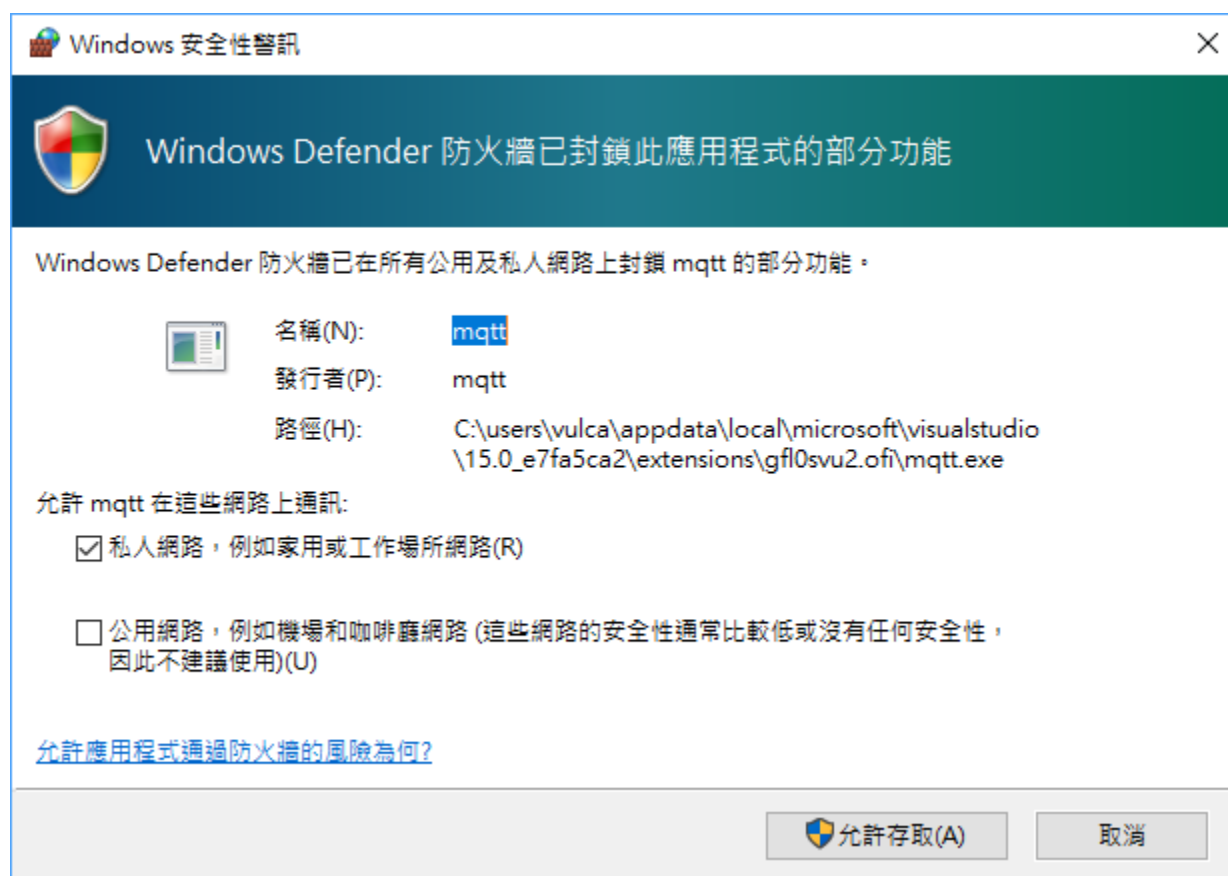


```
namespace App6
{
    7 個參考
    public partial class App : Application
    {
        3 個參考
        public App()
        {
            // Initialize Live Reload.
            #if DEBUG
                LiveReload.Init();
            #endif
            InitializeComponent();
            MainPage = new MainPage();
        }
    }
}
```

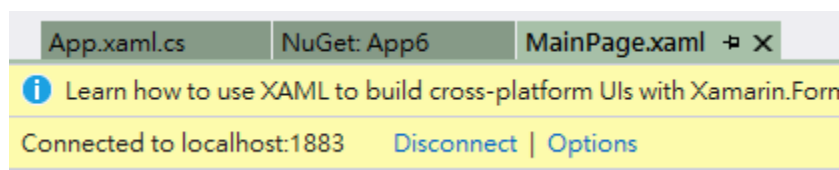
- 請在 Xamarin.Forms 專案內找到 MainPage.xaml 這個檔案，請打開這個檔案，此時，您將會看到在上方會出現一條黃色底色的訊息 **Live Reload is enabled but not listening on localhost:1883 Connect | Options**，現在，請點選 [Connect] 這個文字。



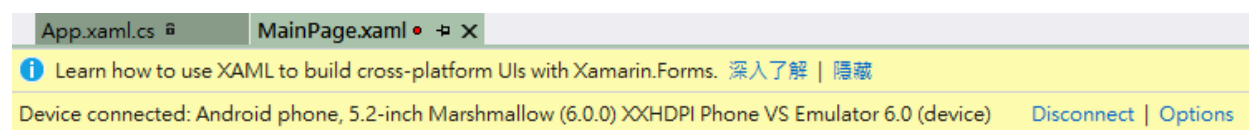
- 若有出現 Windows 安全性警訊對話窗，如下圖所示，請點選允許存取按鈕。



- 當出現了 **Connected to localhost:1883** 就表示 Xamarin 即時重新載入已經準備好了。



- 現在您可以執行這個專案，接下來的操作，就如同前面提到的 LiveXAML 產品相同，打開該頁面的 XAML 文件檔案，進行修改，接著按下儲存按鈕，此時，該頁面就會即時更新在模擬器上了。另外，您也會看到如下圖的黃色底色訊息，告訴您現在正在連線上的裝置是甚麼？



版權頁

XAML in Xamarin.Forms 基礎篇: Xamarin 開發系列叢書

檔案格式：EPUB3 、PDF 、MOBI

版本： 1.0

日期： 2018.11

定價： US\$ 44.99

作者： Vulcan Lee 李進興

版權所有，請勿非法複製、散佈。