Preparation before you come to the Xamarin Hands On Labs workshop

To get the most out of the Xamarin Hands On Labs workshop, it is crucial that you have at least one emulator or simulator or a physical mobile testing device working with Visual Studio, so you can test and debug your applications.

Detailed setup instructions and troubleshooting tips can be found on the GitHub page of our Hands On Labs workshop: https://github.com/xpiritbv/xamarin-hands-on-labs#getting-started

This document is intended to help you get your setup working before coming to the workshop.

Fastest debugging and testing options

Debugging **Android** apps is easiest using an Android device (phone or tablet) that is *developer unlocked*. Make sure to bring a USB cable to the workshop and unlock your device for development. Steps: https://www.wikihow.tech/Unlock-Developer-Options-on-Android. You may have to install a device driver, depending on the brand and type of your device. If you don't have an Android device, you will find setup and verification steps for an emulator in this document.

Debugging **iOS** apps is easiest using the iOS Simulator on your Mac because this will save you the trouble of having to create provisioning profiles.

Depending on your computer hardware there are several options for running emulators. Please look at the options carefully and select the one that applies to you:

You have an Apple laptop like a MacBook, MacBook Pro or a MacBook Air

Make sure your device runs the latest version of the operating system and Visual Studio for Mac. Please check for system updates on both the Apple AppStore, so you have the latest version of XCode that matches the operating system. After installing the XCode update, launch XCode at least once to confirm any copyright messages and/or confirm any additional downloads. Not doing so may block compilation or debugging of your iOS apps. Also check for updates on Visual Studio for Mac by launching Visual Studio for Mac and checking in the menu: **Visual Studio** > **Check for Updates…**

Install **Visual Studio for Mac (https://www.visualstudio.com/vs/mac/)** This should take care of all prerequisites to build and run iOS applications. Please note that Visual Studio for Mac **is not the same as Visual Studio Code!** Xamarin development is not possible with Visual Studio Code.

For Android Development you can run either on a Mac or on a Windows machine.

If you are running on a Mac, you can create a new solution in Visual Studio for Mac. Select *Android* > *App* > *Android App*. The first time you create such a project, you will be prompted to download the Android SDK on your machine. Please accept this download and all the license agreements. It will get you all the software you need to build and run Android applications on your Mac.

You have a Windows PC running Windows 7, 8 or 10

Make sure you have *Admin* privileges on your development machine, otherwise you will not be able to install and configure Android emulators and SDK's.

On Windows 10

Best experience when you run Windows 10 build 1809, because this Windows build supports the new Google Android emulators based on Hyper-V.

For iOS development on your PC

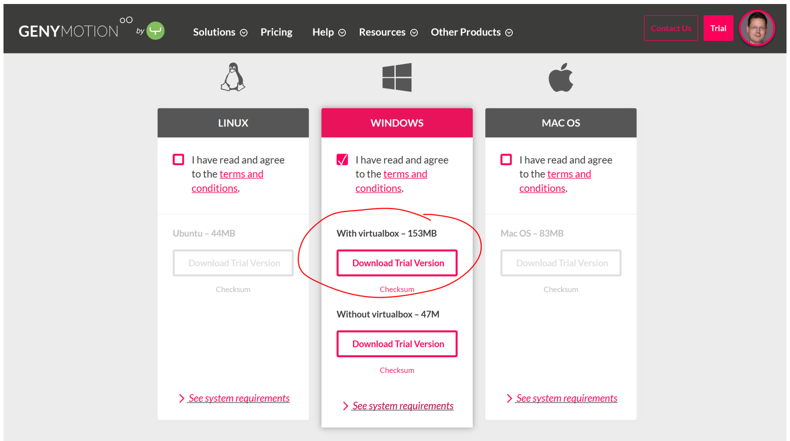
If you want to develop applications for iOS, you need an Apple Device because of Apple licensing restrictions on the iOS SDK. You need either a MacBook or a MacBook Air that you can connect to over the network. Don’t worry if you don’t have a Mac, you can still come to the workshop and do all labs on Android.

For Android Development on your PC

When you run on a computer that has Hyper-V capabilities, then you are best off downloading the Microsoft Android Hyper-V emulator. You can find this download here: https://www.visualstudio.com/vs/msft-android-emulator/

You can either select to download the emulator there or use the new preview emulator. For this workshop we prefer that you use the proven version and not the preview version, since we want you to be successful developing the applications and not running into unforeseen issues with a preview version of the software.

If you don’t have Hyper-V capabilities on your machine, you can use an emulator you can download from Genymotion. (https://www.genymotion.com/) Here you select the option for a *Trial Version*.



□ I have read and agree to the terms and conditions.

Ubuntu – 44MB

☑ I have read and agree to the terms and conditions.

With virtualbox – 153MB

Without virtualbox – 47M

□ I have read and agree to the terms and conditions.

Mac OS – 83MB

Now select the version with *Virtual Box*!

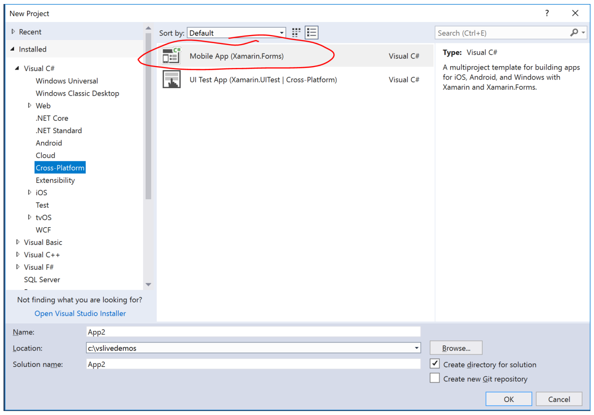
Verification process

Please make sure you can run the following steps **before** you come to our workshop. We have seen many times now that people show up without a working copy of Visual Studio and the emulators and this could ruin your whole day. You are best off spending time at home on better WiFi than in a hotel, downloading all these tools and making sure everything works as expected.

Here are the steps to verify that your laptop works and to ensure you can get most out of the day:

Start Visual Studio and create a new project:

File -> New -> Project ->



Type: Visual C#

A multiproject template for building apps for iOS, Android, and Windows with Xamarin and Xamarin.Forms.

Name: App2

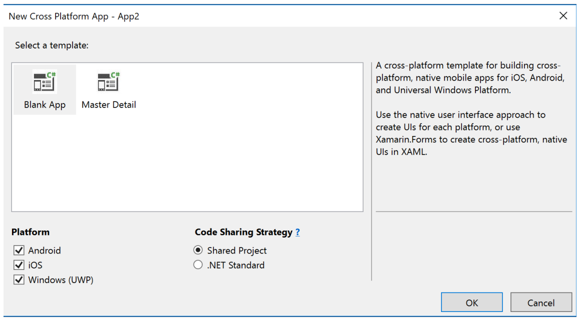
Location: c:\vslivedemos

Solution name: App2

☑ Create directory for solution

□ Create new Git repository

Keep the default settings.



A cross-platform template for building cross-platform, native mobile apps for iOS, Android, and Universal Windows Platform.

Use the native user interface approach to create UIs for each platform, or use Xamarin.Forms to create cross-platform, native UIs in XAML.

Platform

☑ Android

☑ iOS

☑ Windows (UWP)

Code Sharing Strategy ?

● Shared Project

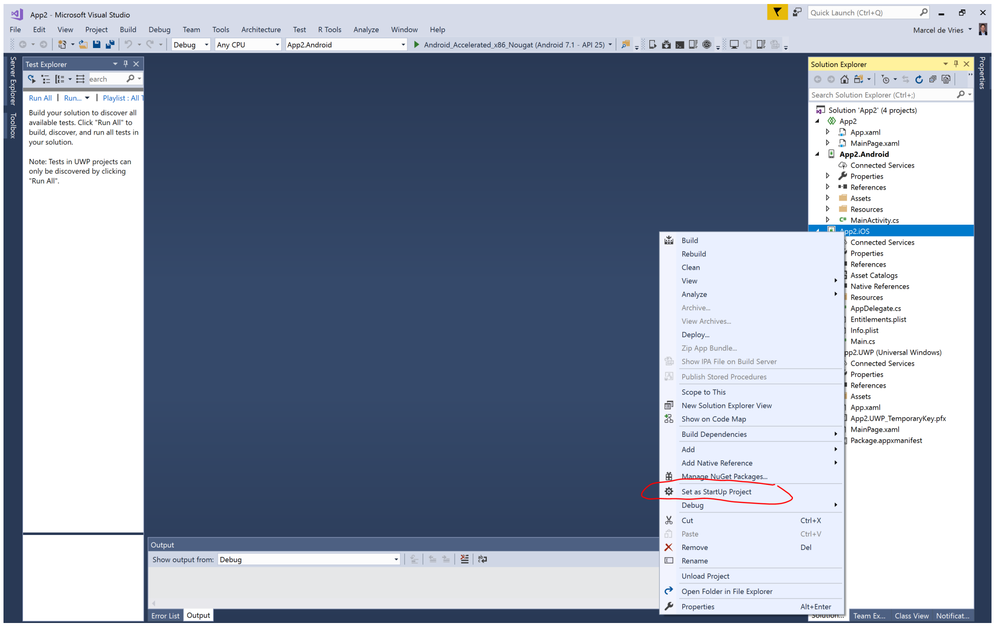
○ .NET Standard

During the workshop we will tell you all about what you are doing, but in order to test things, please just accept the defaults.

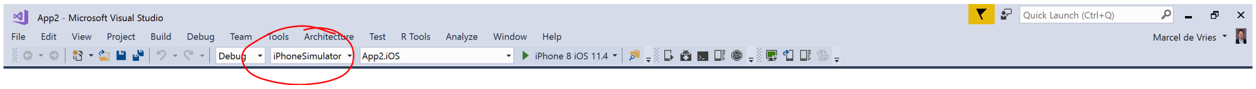
A new project is created.

Validation if you can compile and run for iOS (skip if you don't have a Mac)

Right Click on the iOS project and select: *Set as StartUp Project*.

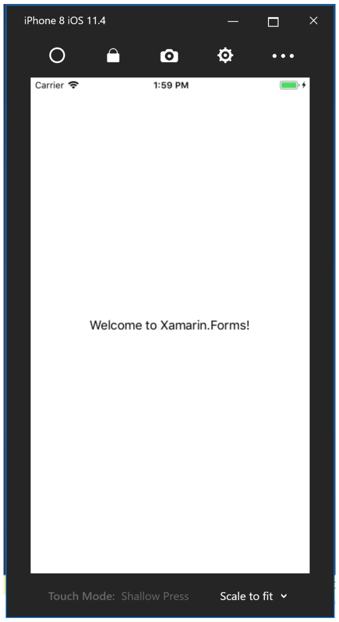


This will initiate the connection to the Mac Build Host and enables the ability to select a device. Ensure you select the *iPhone Simulator* here. Otherwise you will get a compilation error.


App2 - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Architecture Test R Tools Analyze Window Help
Debug
iPhoneSimulator
App2.iOS
iPhone 8 iOS 11.4
Quick Launch (Ctrl+Q)
Marcel de Vries

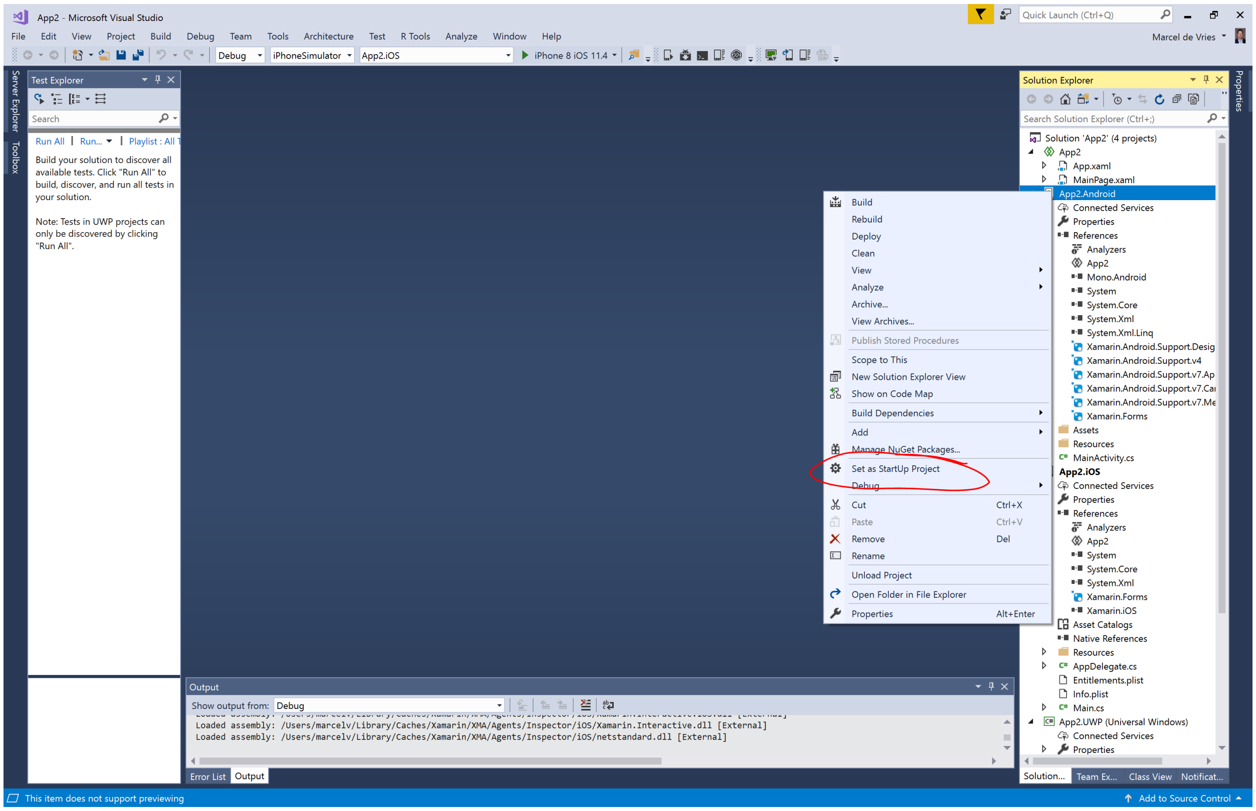
Now select the *Run* button in the menu or press *F5* and run the application.

You should see the following application on your screen:



Validation if you can compile and run for Android

Right click on the Android Project and choose *Set as StartUp Project*. If it is already a startup project but your run toolbar shows the Play button grayed-out, then right click the Windows project and *Set as StartUp Project* and then switch back to Android again. This should enable the toolbar to run your Android application.

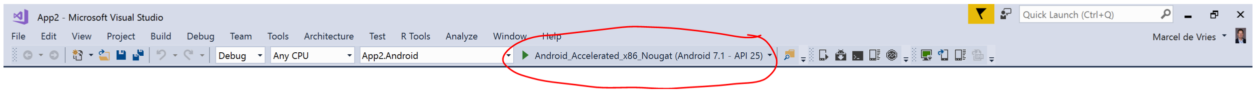


Build your solution to discover all available tests. Click "Run All" to build, discover, and run all tests in your solution.

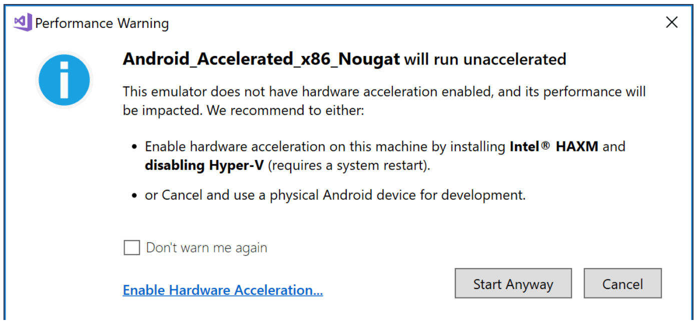
Note: Tests in UWP projects can only be discovered by clicking "Run All".

```
Loaded assembly: /Users/marcelv/Library/Caches/Xamarin/XMA/Agents/Inspector/iOS/Xamarin.Interactive.dll [External]
Loaded assembly: /Users/marcelv/Library/Caches/Xamarin/XMA/Agents/Inspector/iOS/netstandard.dll [External]
```

After you have enabled this project as startup, you will now see the following options in your *Run* toolbar:


App2 - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Architecture Test R Tools Analyze Window Help
Debug Any CPU App2.Android
Android_Accelerated_x86_Nougat (Android 7.1 - API 25)
Quick Launch (Ctrl+Q)
Marcel de Vries

Now when you run the application you will probably see the following message on the screen:



Performance Warning

Android_Accelerated_x86_Nougat will run unaccelerated

This emulator does not have hardware acceleration enabled, and its performance will be impacted. We recommend to either:

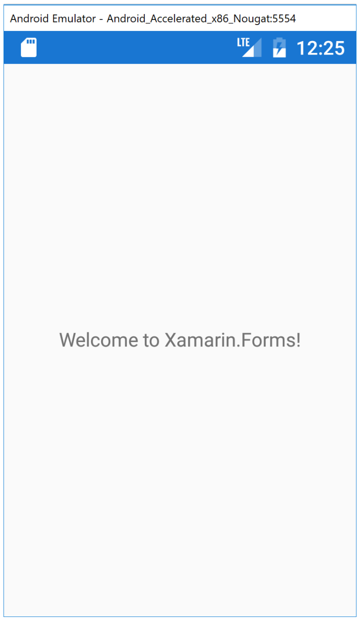
- Enable hardware acceleration on this machine by installing **Intel® HAXM** and **disabling Hyper-V** (requires a system restart).
- or Cancel and use a physical Android device for development.

□ Don't warn me again

Enable Hardware Acceleration...

Start Anyway | Cancel

If you don't have Hyper-V and did not install any other emulator, then just please click on *Start Anyway*. This will take about 10-15 minutes for the emulator to start. When it has started you should see the following: (it took 17,5 minutes on a new Surface Book 2 i7 to show up)



Android Emulator - Android_Accelerated_x86_Nougat:5554

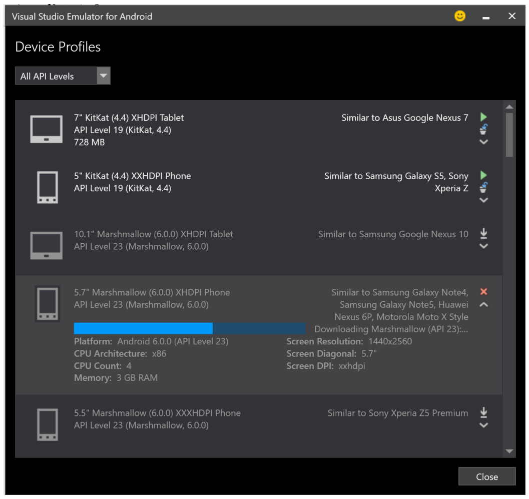
Welcome to Xamarin.Forms!

So while this works, it will slow you down a lot.

Faster solution

If you downloaded either the Microsoft Hyper-V emulator or the Genymotion emulator, then you can take the following steps to connect Visual Studio to your emulator:

First download an emulator with a higher API level, so we support the latest features. The *Nougat* or *Oreo* emulators are not available yet, so we choose the *Marshmallow* for now.



Visual Studio Emulator for Android

Device Profiles

All API Levels

7" KitKat (4.4) XHDPI Tablet
API Level 19 (KitKat, 4.4)
728 MB
Similar to Asus Google Nexus 7

5" KitKat (4.4) XXHDPI Phone
API Level 19 (KitKat, 4.4)
Similar to Samsung Galaxy S5, Sony Xperia Z

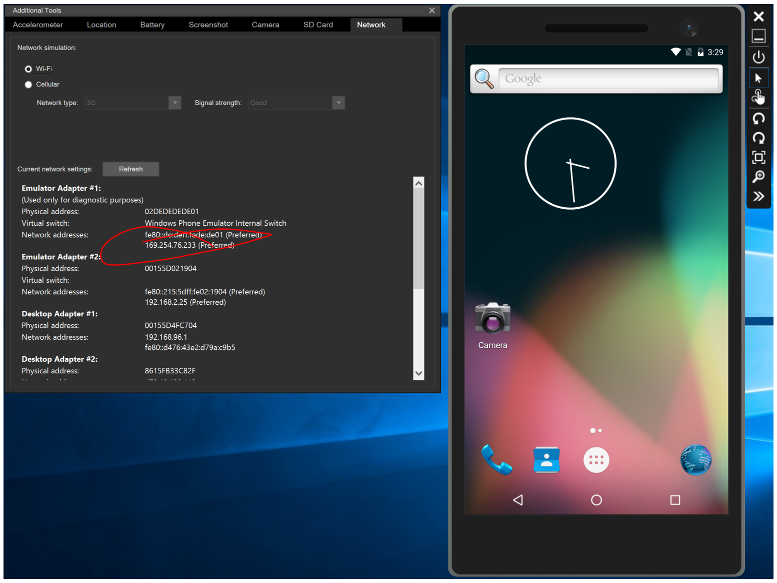
10.1" Marshmallow (6.0.0) XHDPI Tablet
API Level 23 (Marshmallow, 6.0.0)
Similar to Samsung Google Nexus 10

5.7" Marshmallow (6.0.0) XHDPI Phone
API Level 23 (Marshmallow, 6.0.0)
Similar to Samsung Galaxy Note4, Samsung Galaxy Note5, Huawei Nexus 6P, Motorola Moto X Style
Downloading Marshmallow (API 23):...
Platform: Android 6.0.0 (API Level 23)
CPU Architecture: x86
CPU Count: 4
Memory: 3 GB RAM
Screen Resolution: 1440x2560
Screen Diagonal: 5.7"
Screen DPI: xxhdpi

5.5" Marshmallow (6.0.0) XXXHDPI Phone
API Level 23 (Marshmallow, 6.0.0)
Similar to Sony Xperia Z5 Premium

After it is downloaded we run the emulator.

When it is up and running you need to know the IP address the emulator is running on. For the Microsoft Emulator you can see that as follows:



Network simulation:

○ Wi-Fi
● Cellular

Network type: 3G   Signal strength: Good

Current network settings:

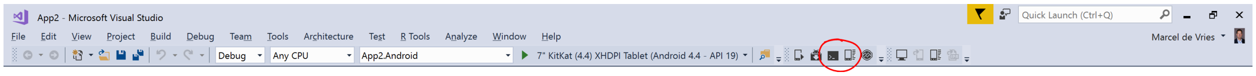
Emulator Adapter #1:
(Used only for diagnostic purposes)
Physical address: 02DEDEDEDE01
Virtual switch: Windows Phone Emulator Internal Switch
Network addresses: fe80::de:deff:fede:de01 (Preferred)
169.254.76.233 (Preferred)

Emulator Adapter #2:
Physical address: 00155D021904
Virtual switch:
Network addresses: fe80::215:5dff:fe02:1904 (Preferred)
192.168.2.25 (Preferred)

Desktop Adapter #1:
Physical address: 00155D4FC704
Network addresses: 192.168.96.1
fe80::d476:43e2:d79a:c9b5

Desktop Adapter #2:
Physical address: 8615FB33C82F

Now in Visual Studio go to the toolbar as shown below and click on the icon: *Open Android ADB Command Prompt*

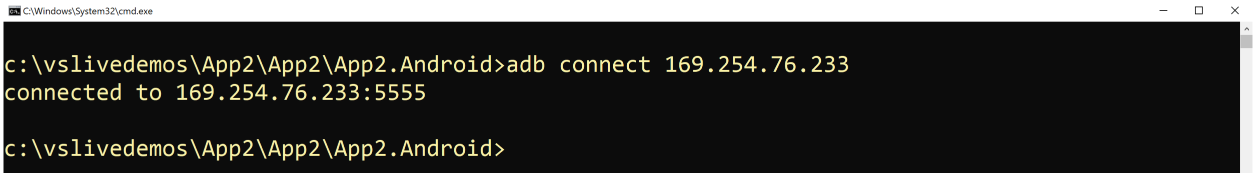


Debug | Any CPU | App2.Android | 7" KitKat (4.4) XHDPI Tablet (Android 4.4 - API 19)

At the command prompt, type the following command:

```
adb connect <ipadress of your emulator>
```

See the expected results below:

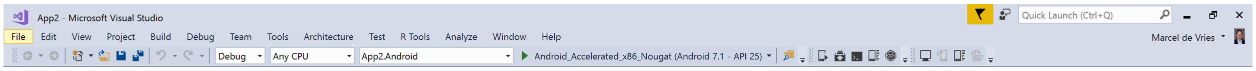


```
c:\vslivedemos\App2\App2\App2.Android>adb connect 169.254.76.233
connected to 169.254.76.233:5555

c:\vslivedemos\App2\App2\App2.Android>
```

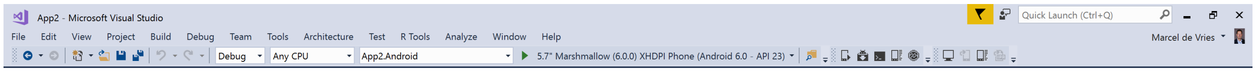
After *adb* confirms the connection, you can select this emulator in Visual Studio:

Before:



Debug | Any CPU | App2.Android | Android_Accelerated_x86_Nougat (Android 7.1 - API 25)

After:



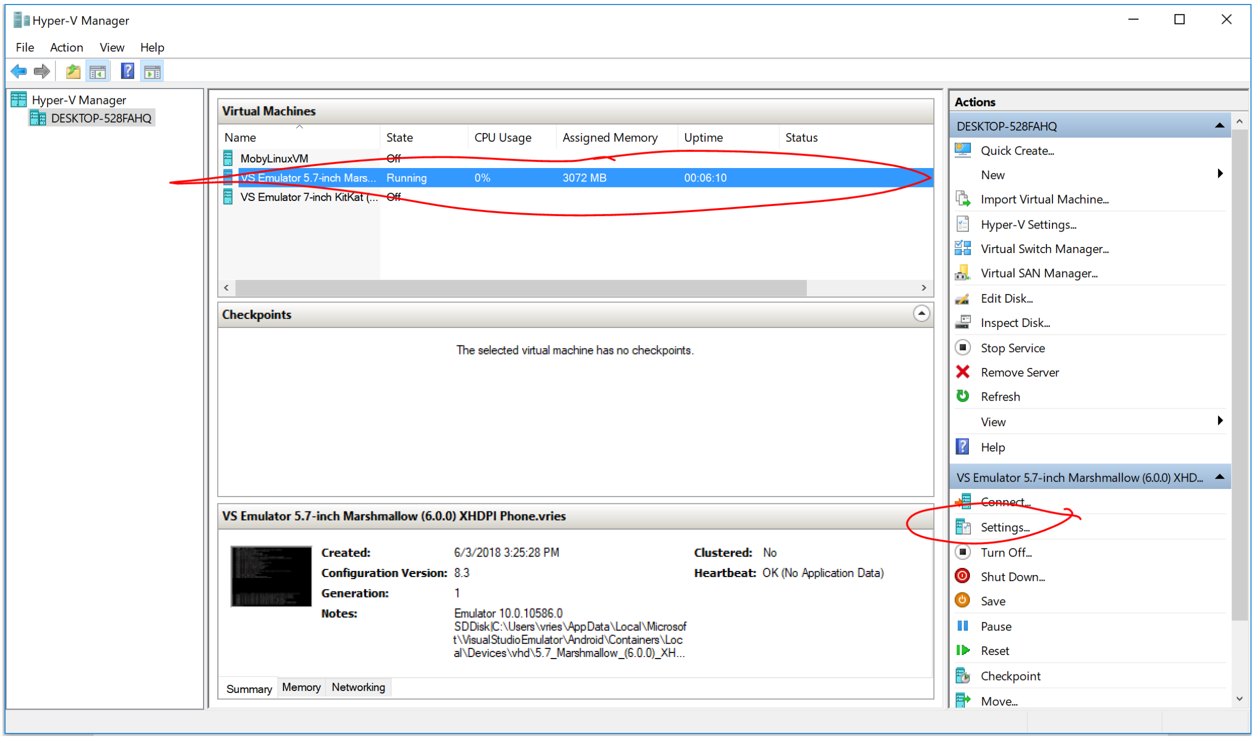
Debug | Any CPU | App2.Android | 5.7" Marshmallow (6.0.0) XHDPI Phone (Android 6.0 - API 23)

Now try to run the application by clicking the *Play* button or pressing F5

Help, my application does not start!

If you have a newer type of machine with the latest Intel Processors, it can happen that your debugger starts and then immediately exits. If this is the case, you need to take the following step:

Open the Hyper-V manager on your machine:



Name	State	CPU Usage	Assigned Memory	Uptime	Status
MobyLinuxVM	Off				
VS Emulator 5.7-inch Mars...	Running	0%	3072 MB	00:06:10	
VS Emulator 7-inch KitKat (...	Off				

The selected virtual machine has no checkpoints.

VS Emulator 5.7-inch Marshmallow (6.0.0) XHDPI Phone.vries

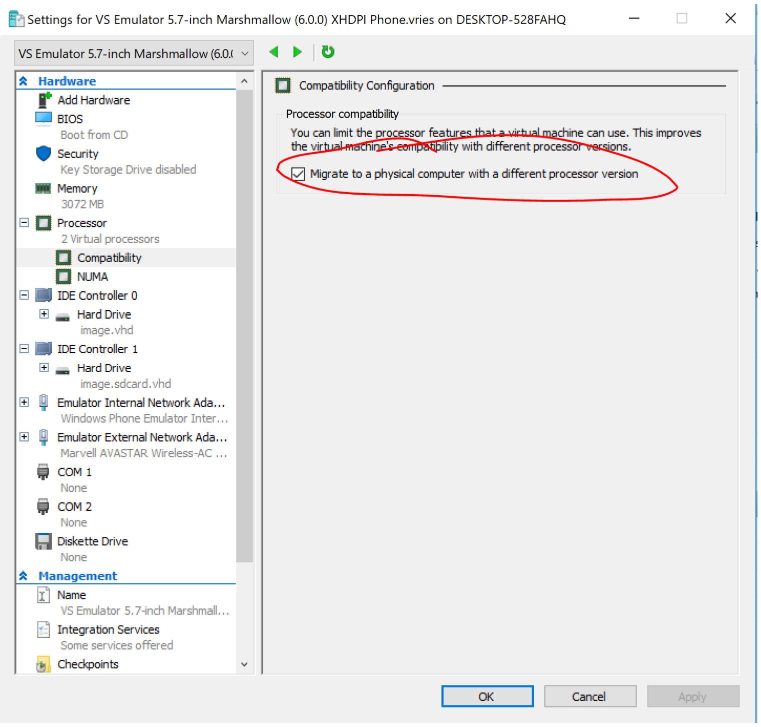
Created: 6/3/2018 3:25:28 PM
Configuration Version: 8.3
Generation: 1
Notes: Emulator 10.0.10586.0
SDDisk|C:\Users\vries\AppData\Local\Microsof
t\VisualStudioEmulator\Android\Containers\Loc
al\Devices\vhd\5.7_Marshmallow_(6.0.0)_XH...

Clustered: No
Heartbeat: OK (No Application Data)

Select the new emulator and shut it down, before you try to configure it.

Now in the configuration screen, select the two processors and there, select the first one.

On the compatibility tab, select the option: *"Migrate to physical computer with a different processor version"*


Settings for VS Emulator 5.7-inch Marshmallow (6.0.0) XHDPI Phone.vries on DESKTOP-528FAHQ

Compatibility Configuration

Processor compatibility

You can limit the processor features that a virtual machine can use. This improves the virtual machine's compatibility with different processor versions.

☑ Migrate to a physical computer with a different processor version

After changing this setting, you can start the emulator again, run the *adb* command and debug your application.

If all is well, you should now see your application running with the debugger attached.

Conclusion

You need your test device or emulator to work before you start with our workshop, since fixing the emulator will take a lot of time you otherwise cannot spend on Hands On Labs.

A comprehensive setup guide can also be found on the GitHub page for our Hands On Labs, including troubleshooting tips for most of the problems we have encountered so far. Go to: https://github.com/xpiritbv/xamarin-hands-on-labs#getting-started

Roy and Marcel will arrive in Boston on Saturday, so if you have issues getting it to run, please give us a shout out on twitter *(@marcelv* or *@roycornelissen*), or send us an email and we might be able to help you in the evening setting things up.

With kind regards,

Roy and Marcel