

Android on Windows 11

A developers Perspective

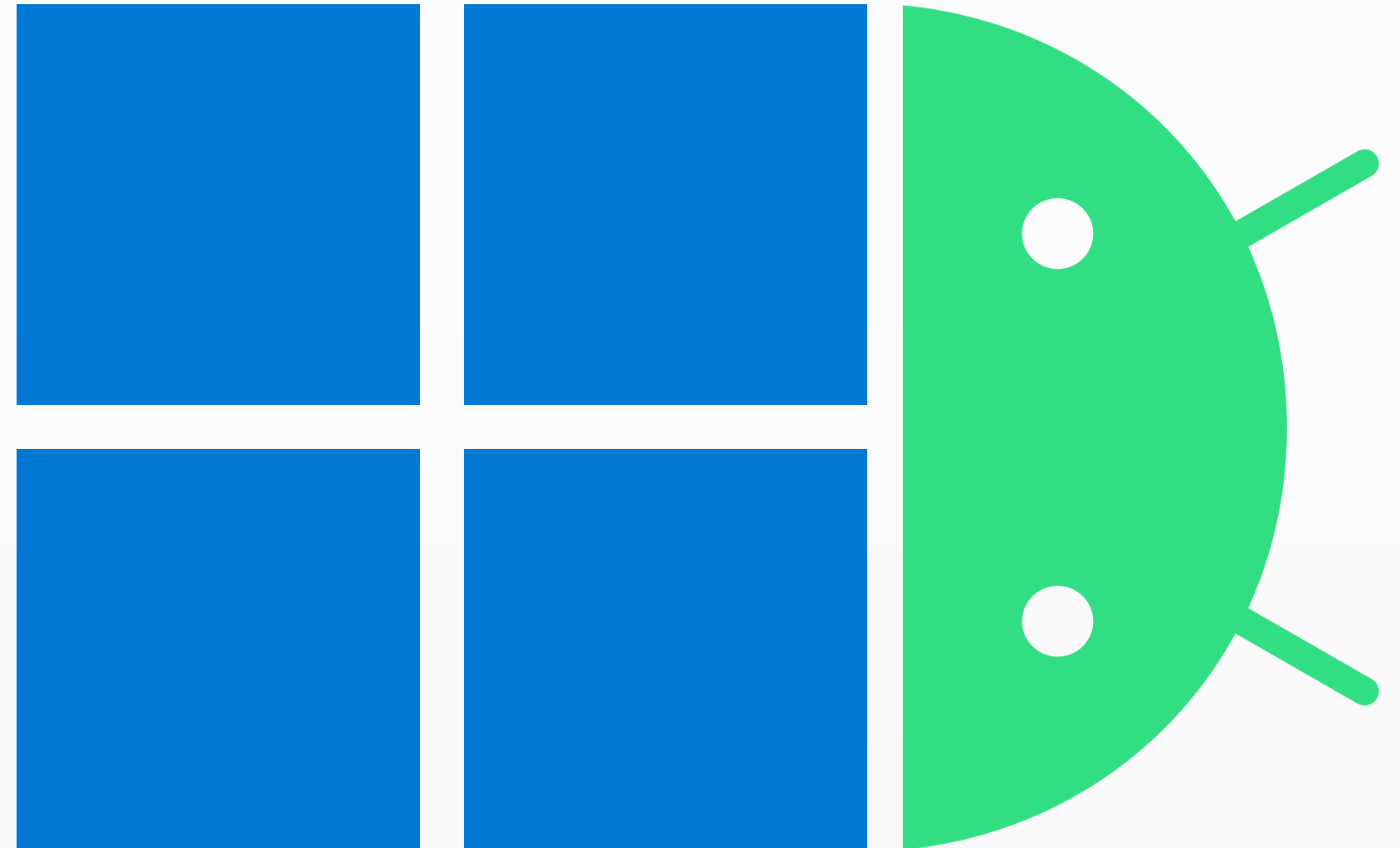


Scott Hanselman
Partner Program Manager
Microsoft
@shanselman



Jim McKeeth
Developer Advocate
Embarcadero
jim.mckeeth@embarcadero.com
@JimMcKeeth

Windows Subsystem for Android (WSA)



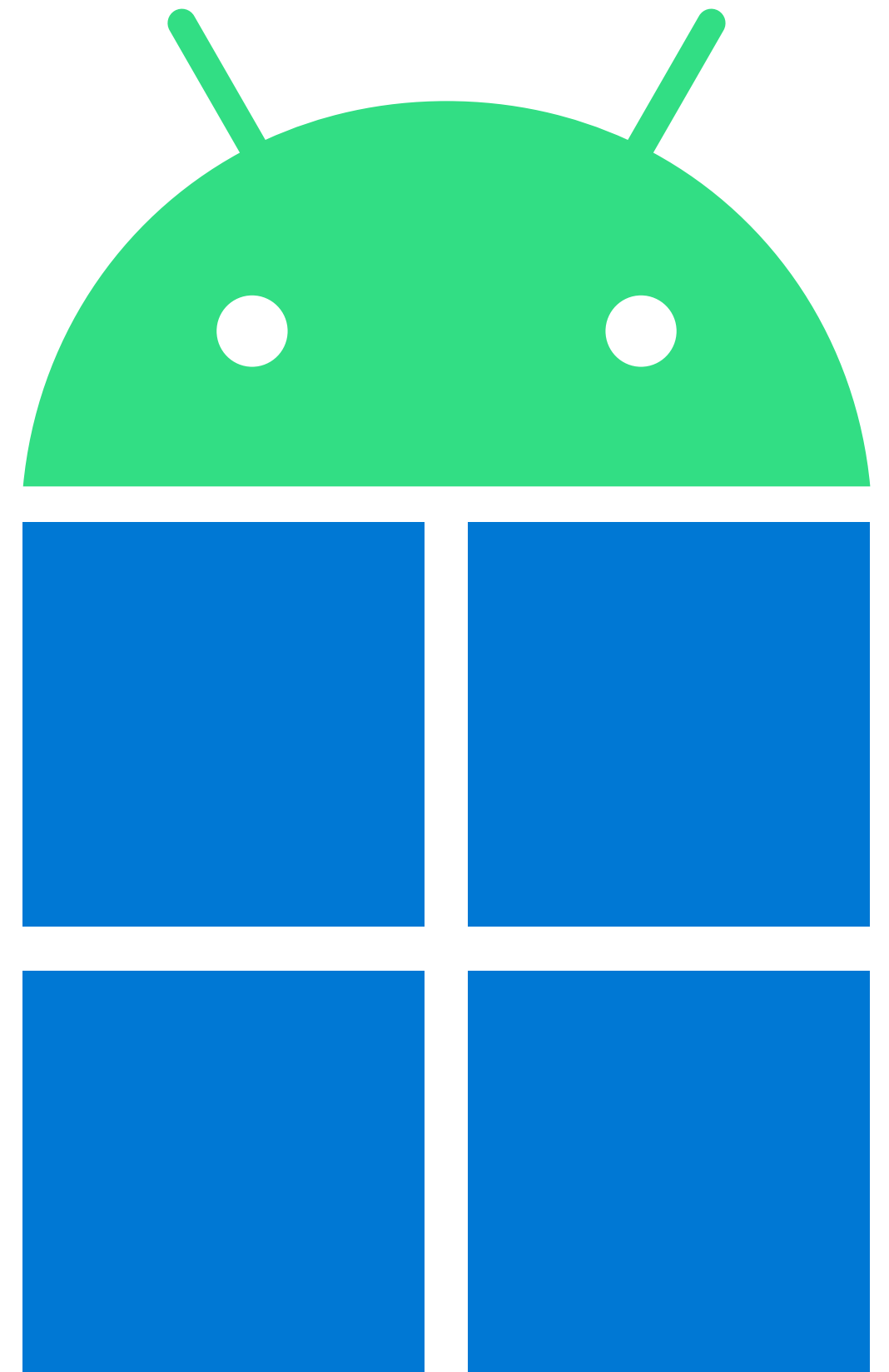
Agenda

- Introduction
- Installation & Configuration
- Performance & Hardware
- Developer Mode & Sideload
- Targeting from Delphi
- More information
- Q&A



Android on Windows?

- Windows Subsystem for Android (WSA) runs Android apps on Windows 11
- Runs Android 11 behind the scenes
- Similar to Windows Subsystem for Linux (WSL)
- Comes with Amazon Appstore (*US Only for now*)
 - An Amazon account is required to use the Amazon Appstore
- Allows for developer mode support and sideloading
- Android apps are on your start menu
- Only in Windows 11
- WSA utilizes a virtual machine (VM)



Hardware Requirements

- Requires an SSD
- I wasn't able to run it in a VM
- On Windows x64 (Intel or AMD) it uses an interoperability layer
- On Windows ARM64 it runs natively
- Windows 11 enforces a minimum screen requirement of 720p resolution (1280x720) with a >9" screen.

RAM	8 GB (minimum) 16 GB (recommended)
Storage type	Solid State Drive (SSD)
Processor	Intel Core i3 8th Gen (minimum) or above AMD Ryzen 3000 (minimum) or above Qualcomm Snapdragon 8c (minimum) or above
Processor architecture	x64 or ARM64
Virtual Machine Platform	This setting needs to be enabled in BIOS.

System Requirements Documentation

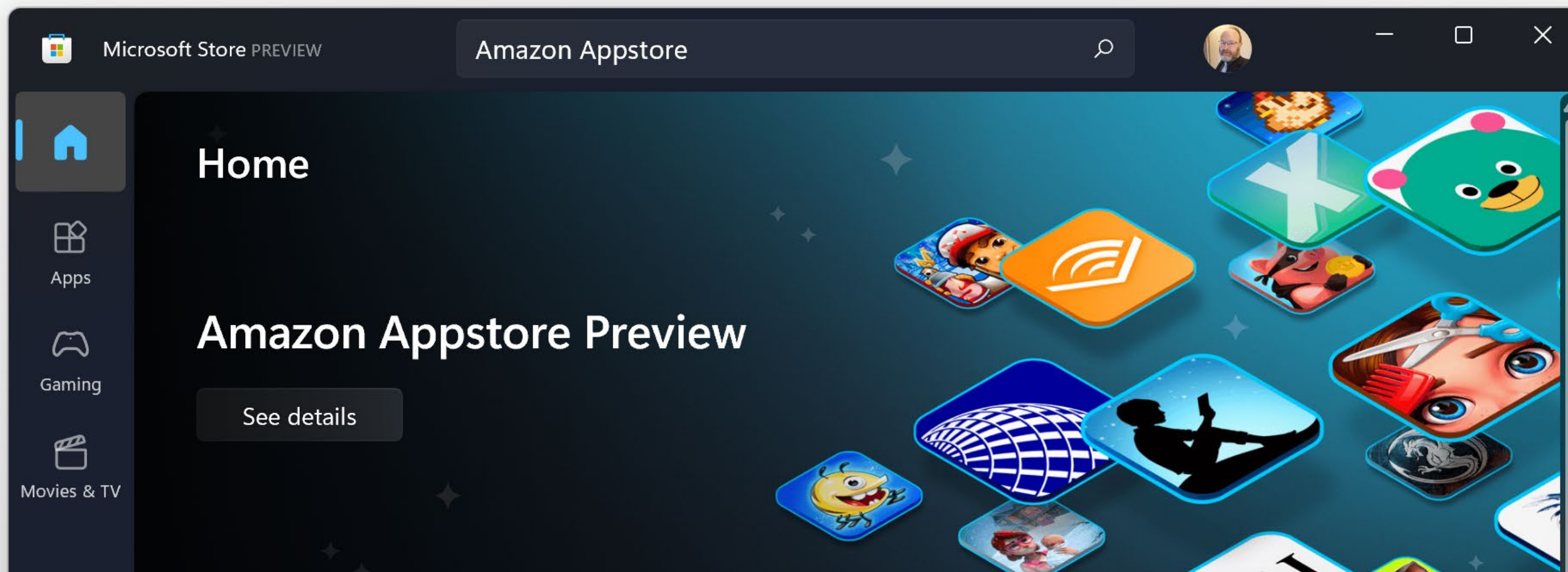
WSA support.microsoft.com/windows/f8d0abb5-44ad-47d8-b9fb-ad6b1459ff6c

Windows 11 support.microsoft.com/windows/86c11283-ea52-4782-9efd-7674389a7ba3



WSA Installation

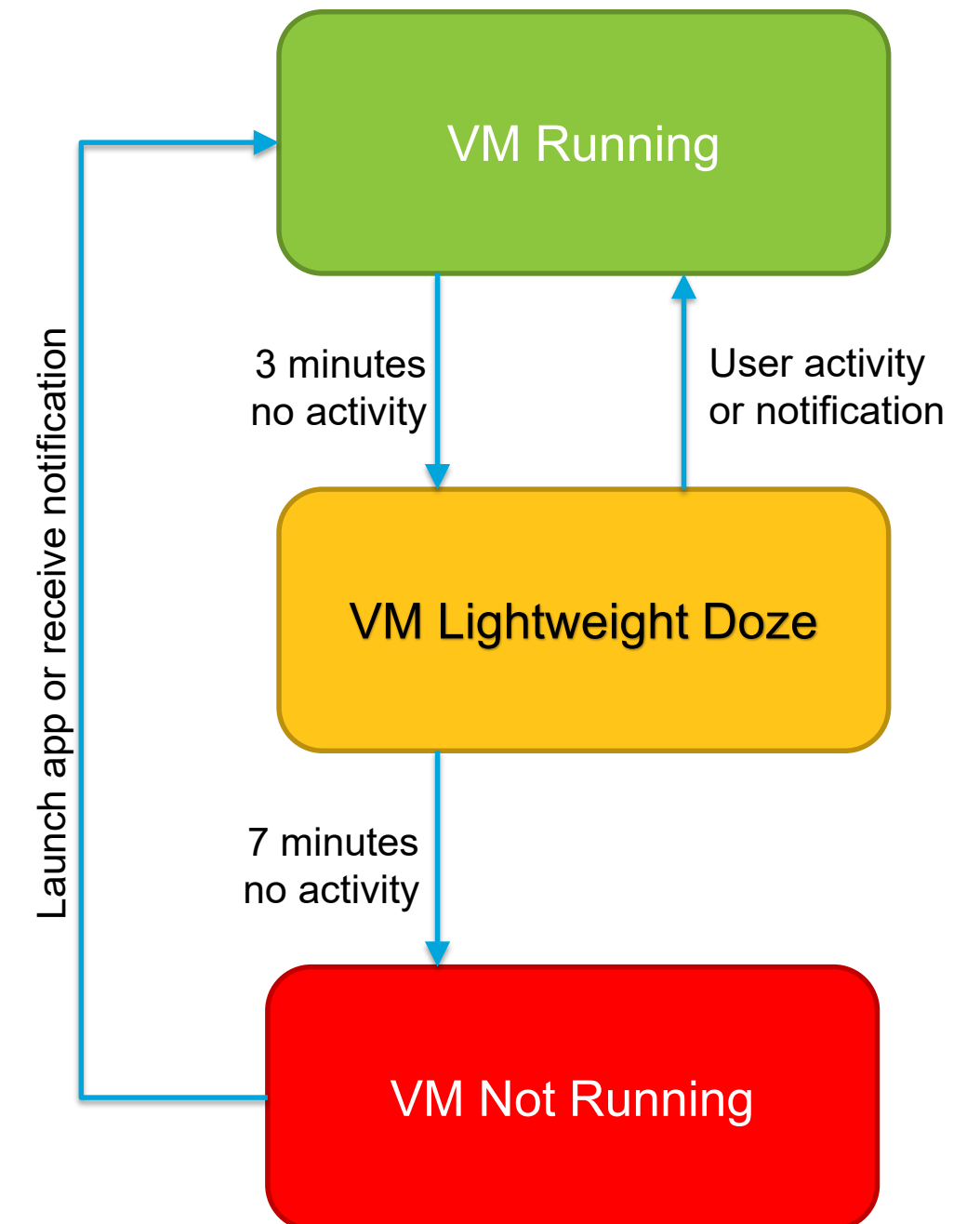
- Windows Subsystem for Android is available for **public preview** on Windows 11.
- Install the **Amazon Appstore** from the **Microsoft Store**.
 - Alternatively just install an **Android app** from the Microsoft Store.



docs.microsoft.com/en-us/windows/android/wsa/ & support.microsoft.com/windows/f8d0abb5-44ad-47d8-b9fb-ad6b1459ff6c

WSA VM Lifecycle

- There are three possible states for the VM running apps with Windows Subsystem for Android:
 - Running
 - Lightweight Doze: After no app activity for **3 minutes**. Deactivated by user activity or an app notification.
 - Not Running: After activity for **7 minutes**.
- Transitions between these states are triggered by user activity, Android app or an app notification.
- Android apps are paused and then stopped when their window is minimized.



Performance

The data on this chart is gathered from user-submitted Geekbench 5 CPU results from the Geekbench Browser: browser.geekbench.com

CPU/System	Single-Core		Multi-Core	
	Android	Windows	Android	Windows
AMD Ryzen 9 5900X	1404	1671	9592	14,026
Intel Core i9-10900	1230	1315	7202	9260
Intel Core i7-11700K	1464	1692	7980	9826
My Core i7-11700K	1350	n/a	6886	n/a
Samsung S21 Ultra	1083	n/a	3101	n/a
Samsung S22 Ultra	1243	n/a	3301	n/a
Google Pixel 6 Pro	1035	n/a	2831	n/a

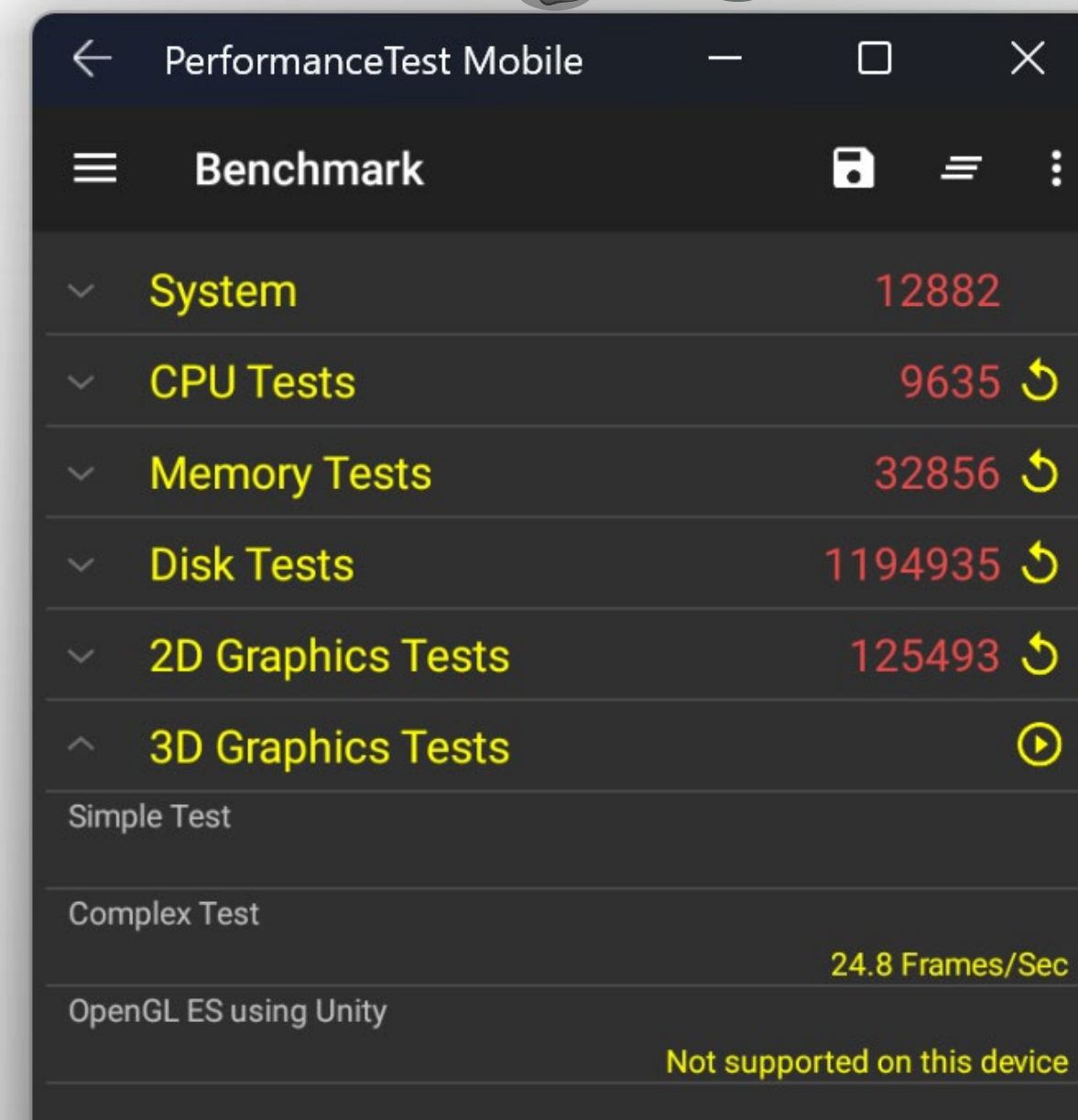


Sources: browser.geekbench.com/v5/cpu/13076342, browser.geekbench.com/processors/amd-ryzen-9-5900x, browser.geekbench.com/v5/cpu/13065478, browser.geekbench.com/processors/intel-core-i9-10900, browser.geekbench.com/v5/cpu/13080168, browser.geekbench.com/v5/cpu/13080587, browser.geekbench.com/v5/cpu/13080585

Passmark Performance Test



	i7-11700K WSA	Samsung S22 Ultra SM-S908U1	OnePlus MT2111 9RT
System	12,993	15,686	14,825
Mem Mark	32,856	31,499	27,525
2D Mark	125,493	35,161	51,452
CPU Mark	9,635	7,341	6,988
Disk Mark	1,194,935	n/a	n/a
3D Mark	n/a	57,805	29,616



Sources: PassMark PerformanceTest Mobile
www.passmark.com/products/pt_mobile/index.php
androidbenchmark.net/phone.php?phone=Samsung+SM-S908U1
androidbenchmark.net/phone.php?phone=OnePlus+MT2111

Hardware Report

Windows Subsystem for Android

Model	Microsoft Corporation Subsystem for Android(TM)
Platform	Android x86 (64-bit)
OS	Android 11
CPU	Intel Core i7-11700K
Memory	5.85 GB
GPU	Android Emulator OpenGL ES Translator
Resolution	992 x 748
Display PPI	320

Notes: From Geekbench 5

Resolution is based on Android app window size

320 PPI when running at 200% scale

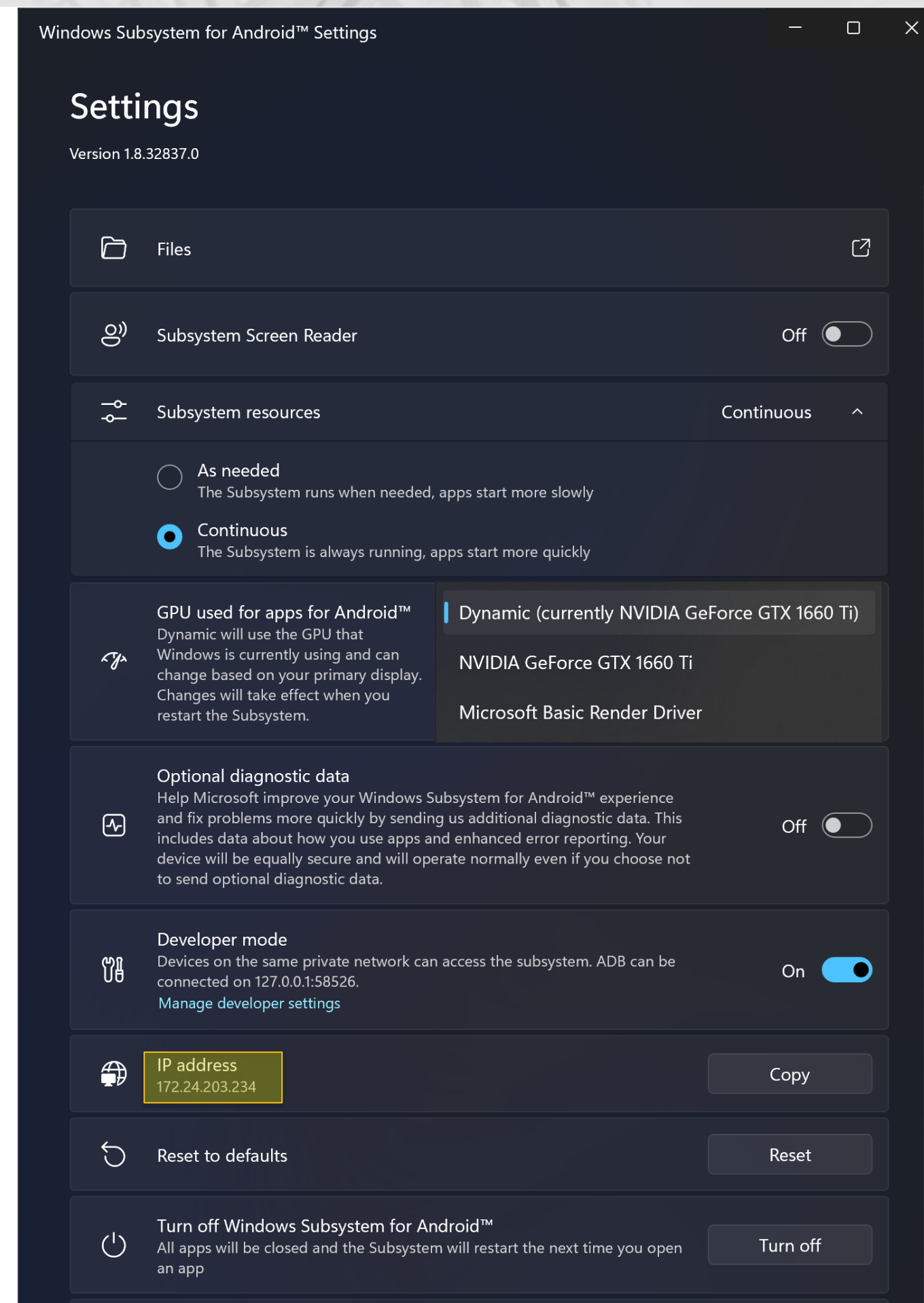
160 PPI when running at 100% scale

Windows 11 Desktop

System Information	
Operating System:	Microsoft Windows 11 Pro (64-bit)
Model:	Gigabyte Technology Co., Ltd. Z590 AORUS ELITE AX
Motherboard:	Gigabyte Technology Co., Ltd. Z590 AORUS ELITE AX
Memory:	32.0 GB DDR4 SDRAM
Processor Information	
Name:	Intel Core i7-11700K
	1 Processor, 8 Cores, 16 Threads
Codename:	Rocket Lake
Package:	Socket 1200 LGA
Base Frequency:	3.60 GHz
Maximum Frequency:	4900 MHz

Settings & Developer Mode

- Launch Windows Subsystem for Android
- Scroll down & enable Developer mode
- Copy the IP address
- Use ADB to connect
`adb connect 172.24.203.234`
- Optional:
 - Connect from other local computers via your IP address and port specified
 - Manage Android Settings via link
- Note:
 - Subsystem can run “As needed” or “Continuous”
 - Files let’s you browse but not transfer files



Digging into Details

adb shell getprop

ro.build.version.sdk	30
ro.board.platform	windows
ro.boot.hardware	windows_x86_64
ro.build.version.release	11
ro.build.version.security_patch	10/1/2021
ro.dalvik.vm.isa.arm	x86
ro.dalvik.vm.isa.arm64	x86_64
ro.dalvik.vm.native.bridge	libhoudini.so
ro.product.cpu.abi	x86_64
ro.product.cpu.abi.list	x86_64, x86, arm64-v8a, armeabi-v7a, armeabi
ro.product.cpu.abi.list.32	x86, armeabi-v7a, armeabi
ro.product.cpu.abi.list.64	x86_64, arm64-v8a
ro.product.manufacturer	Microsoft Corporation

● SDK version: **30** (Android 11)

● Platform: **windows**

● Release version: **Android 11**

● Native bridge: **libhoudini.so**

● Architecture **x86_64** with support for **arm64-v8a, armeabi-v7a, armeabi**

Who is Houdini?

- ARM to x86 native bridge by Intel.
- Not a lot of official public information.
- Also used in Intel Chromebooks, and desktop emulators like BlueStacks & NOX.
- Reads ARM opcodes and produces corresponding behavior in x86.
- **binfmt_misc** (Miscellaneous Binary Format) Linux kernel feature passes ARM to Houdini.

Related files on WSA

- /system/lib64/libhoudini.so
- /system/lib/libhoudini.so
- /vendor/lib/libhoudini.so
- /vendor/lib64/libhoudini.so
- /proc/sys/fs/binfmt_misc
- /vendor/etc/binfmt_misc/arm64_dyn
- /vendor/etc/binfmt_misc/arm64_exe
- /vendor/etc/binfmt_misc/arm_dyn
- /vendor/etc/binfmt_misc/arm_exe

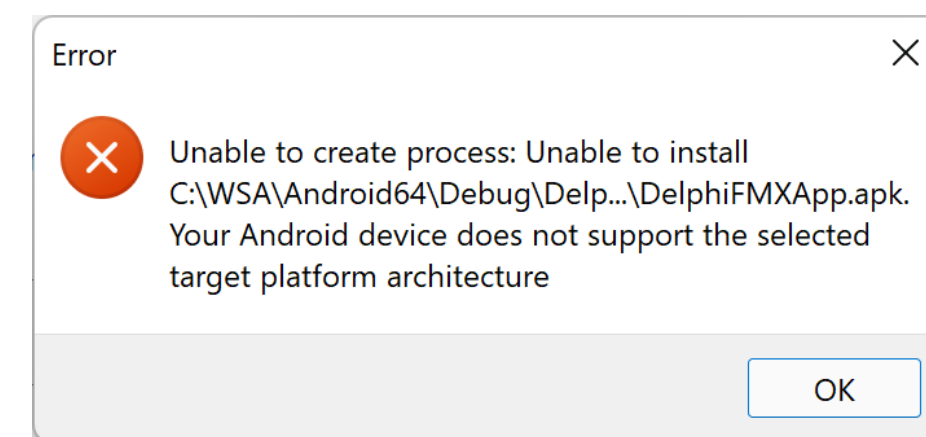
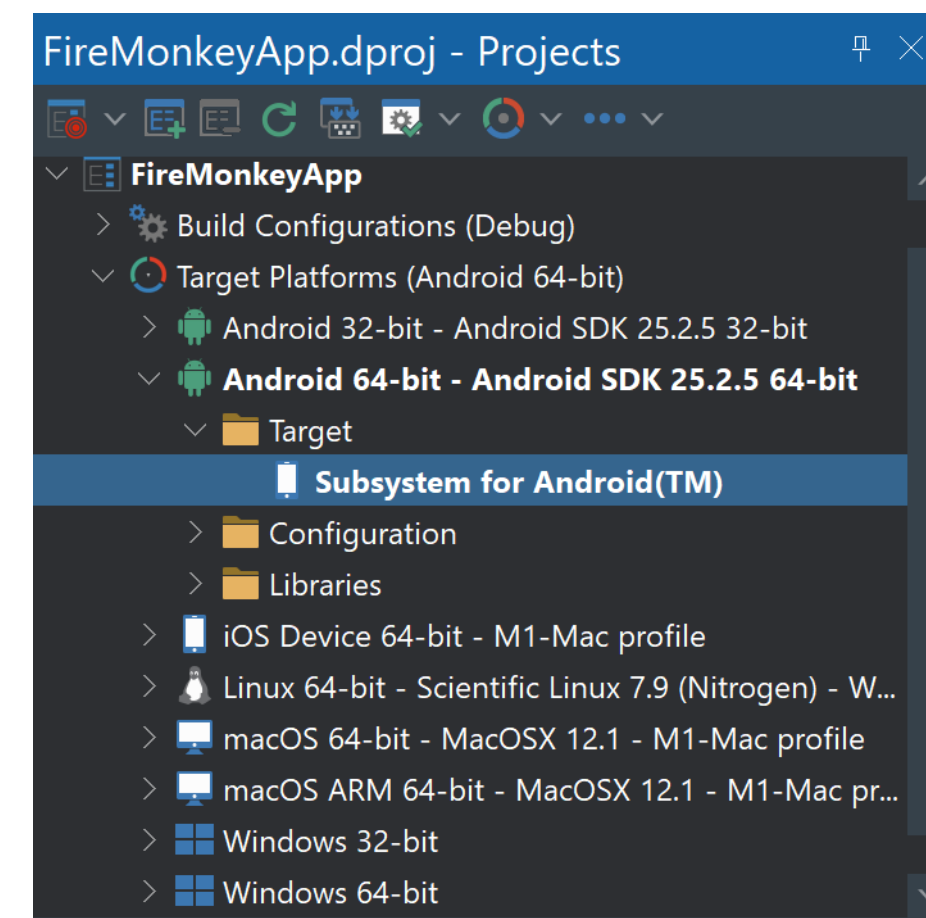


Sideload

- Connect: `adb connect <ip>`
- Check connection: `adb devices`
- Install: `adb install -r --no-streaming <apk file>`
- Uninstall: `adb shell pm uninstall -k com.embarcadero.ProjectName`
- Run app:
`adb shell am start -n com.embarcadero.ProjectName/com.embarcadero.firemonkey.FMXNativeActivity`
- Kill/Stop app: `adb shell am force-stop com.embarcadero.ProjectName`
- Open shell: `adb -d shell`
- Push file: `adb push <local file name> /storage/emulated/0/Download`
- Pull file: `adb pull /storage/emulated/0/Download <local file name>`

Targeting from Delphi & RAD Studio IDE

- A single FMX project can target Windows or Android already.
- WSA is in Preview and not yet fully support by Delphi, but that won't stop us....
- Connect with ADB and then it shows up in the IDE.
- FireMonkey apps either don't run or report error "Your Android device does not support selected platform architecture."
- Sideloaded!
 - Compile for Android 64
 - `adb install -r --no-streaming FireMonkeyApp.apk`
 - Also make resizable



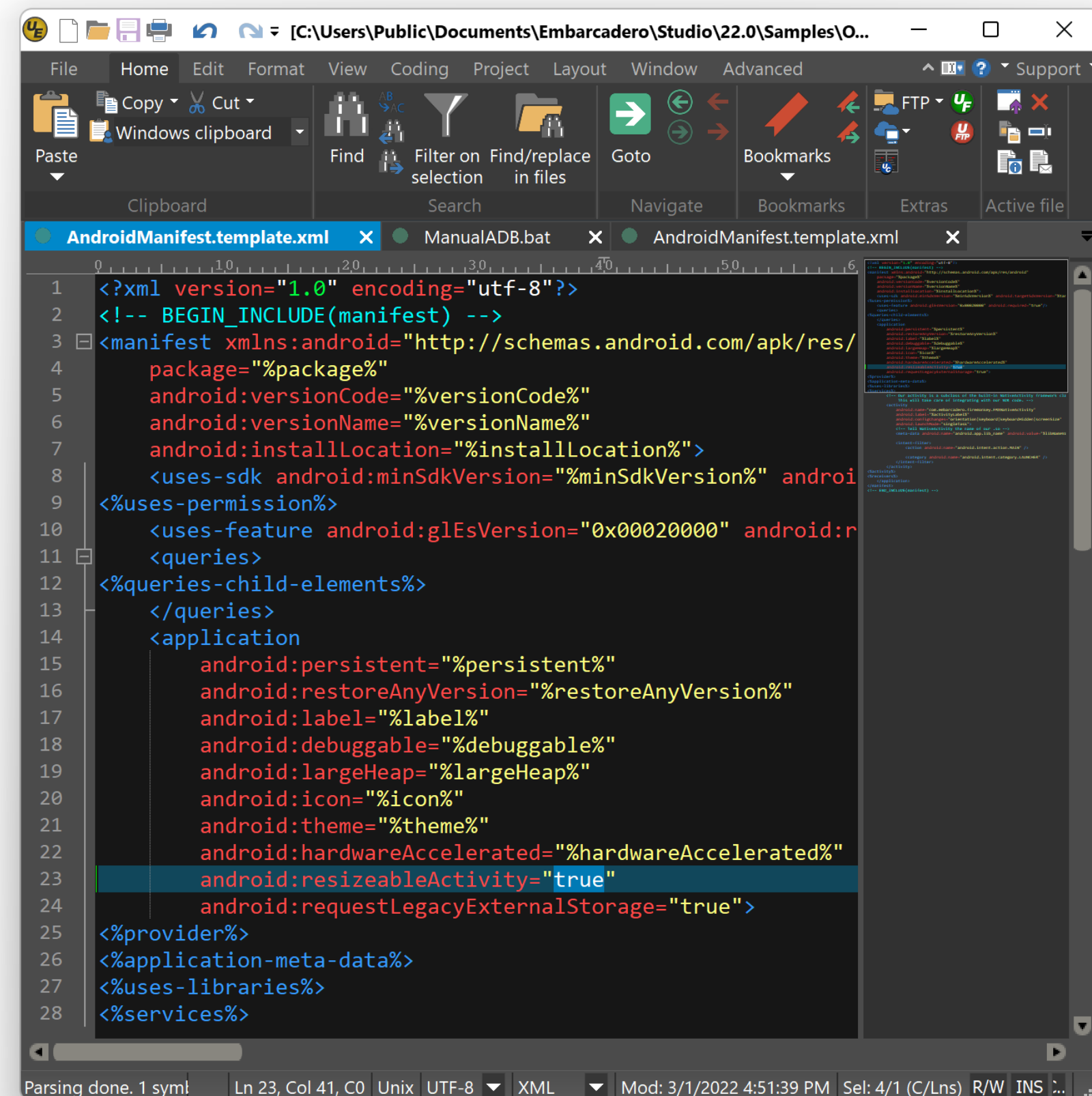
Making Your Android App Resizable

- Edit `AndroidManifest.Template.xml`

```
<application
```

```
    android:resizeableActivity="true">
```

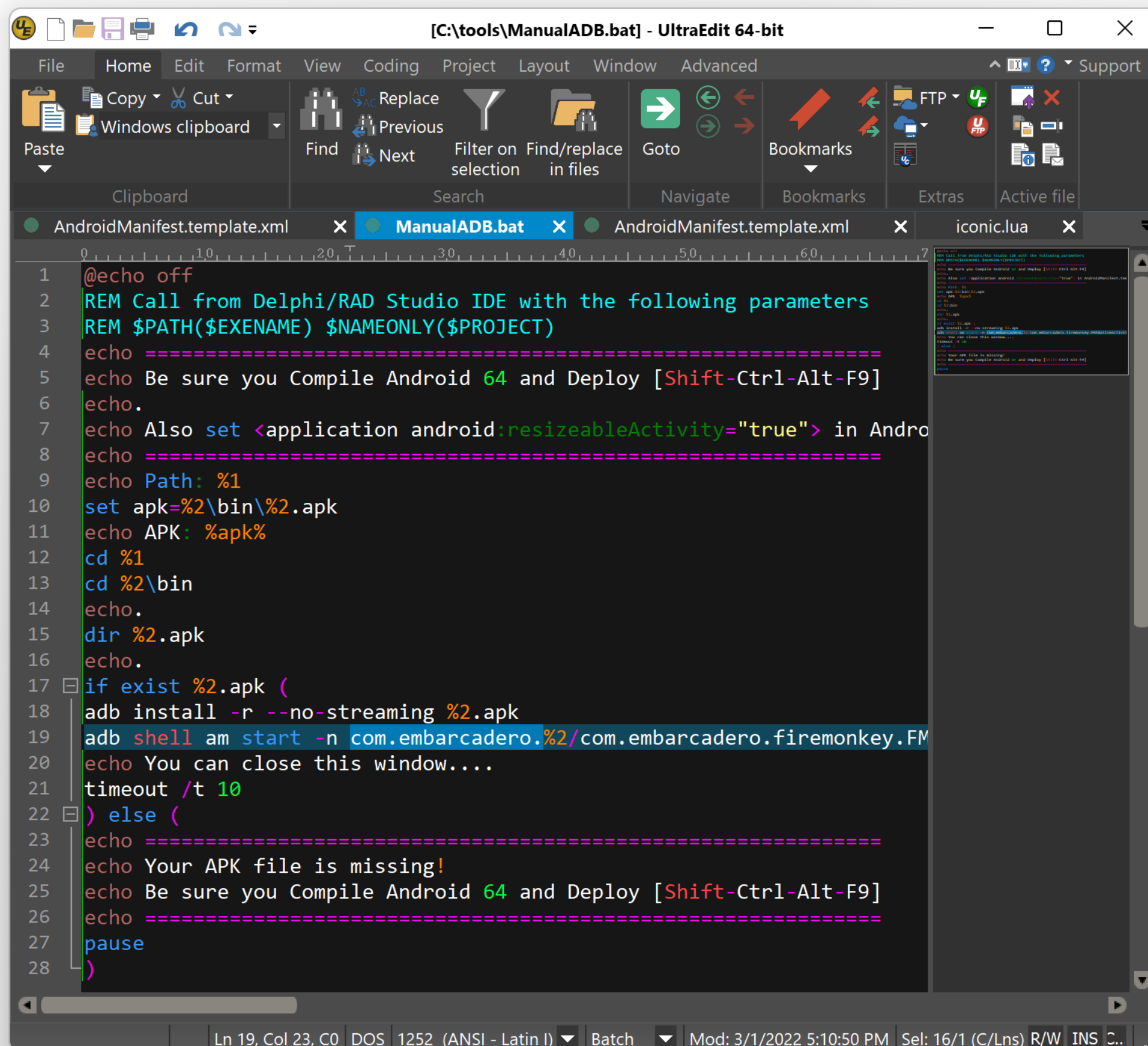
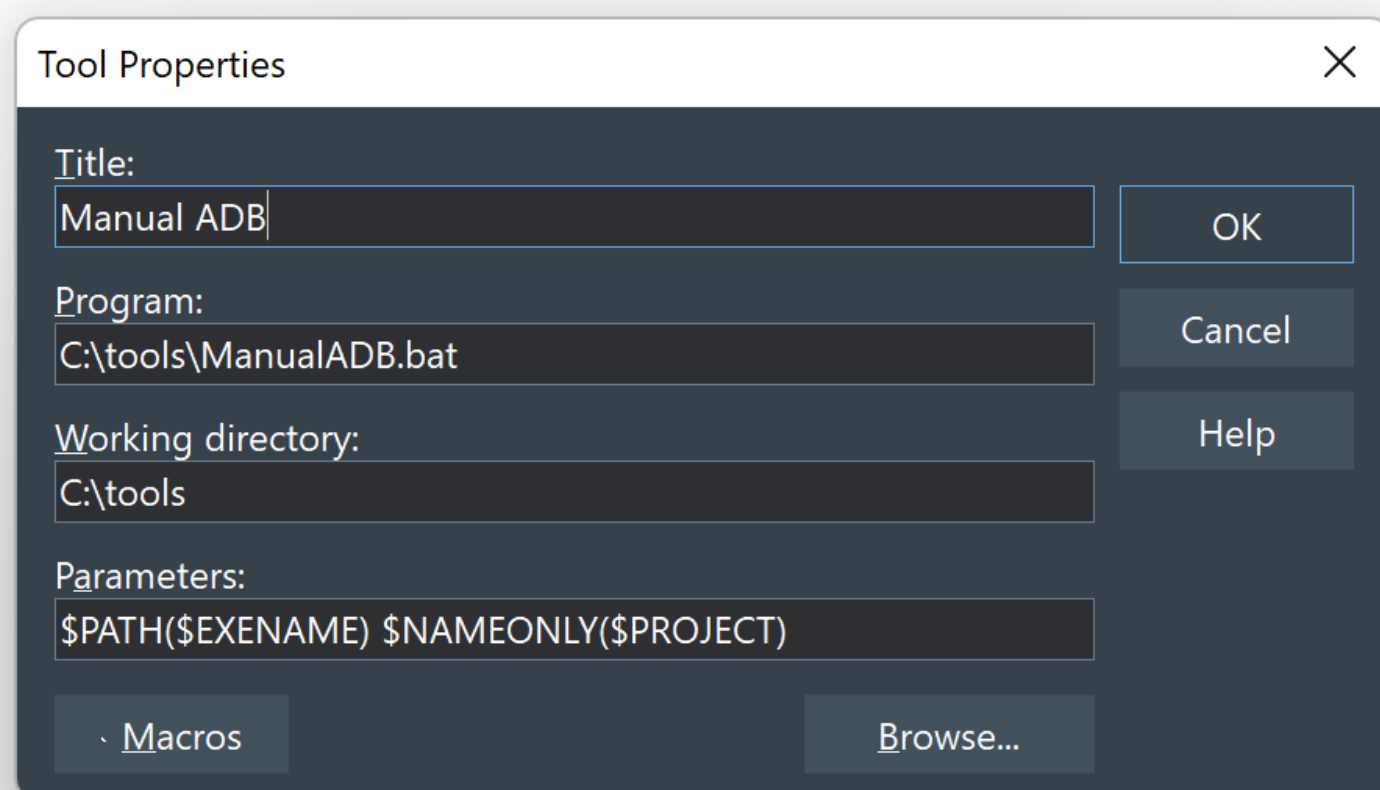
- Note that if you have multiple Android apps in the same folder they share the same `AndroidManifest.Template.xml`
- Allows FMX Android app to resize as expected.



```
<?xml version="1.0" encoding="utf-8"?>
<!-- BEGIN_INCLUDE(manifest) -->
<manifest xmlns:android="http://schemas.android.com/apk/res/
package="%package%"
android:versionCode="%versionCode%"
android:versionName="%versionName%"
android:installLocation="%installLocation%"
<uses-sdk android:minSdkVersion="%minSdkVersion%" android:
<%uses-permission%>
<uses-feature android:glEsVersion="0x00020000" android:r
<queries>
<%queries-child-elements%>
</queries>
<application
    android:persistent="%persistent%"
    android:restoreAnyVersion="%restoreAnyVersion%"
    android:label="%label%"
    android:debuggable="%debuggable%"
    android:largeHeap="%largeHeap%"
    android:icon="%icon%"
    android:theme="%theme%"
    android:hardwareAccelerated="%hardwareAccelerated%"
    android:resizeableActivity="true"
    android:requestLegacyExternalStorage="true">
<%provider%>
<%application-meta-data%>
<%uses-libraries%>
<%services%>
```

Automating the Manual

- Add manualadb.bat to IDE Tools menu
 - Tools -> Configure Tools -> Add
- With parameters:
`$PATH($EXENAME) $NAMEONLY($PROJECT)`
- Source:
gist.github.com/jimmckeeth/3a453ccdb1a387644ca137b499d01879

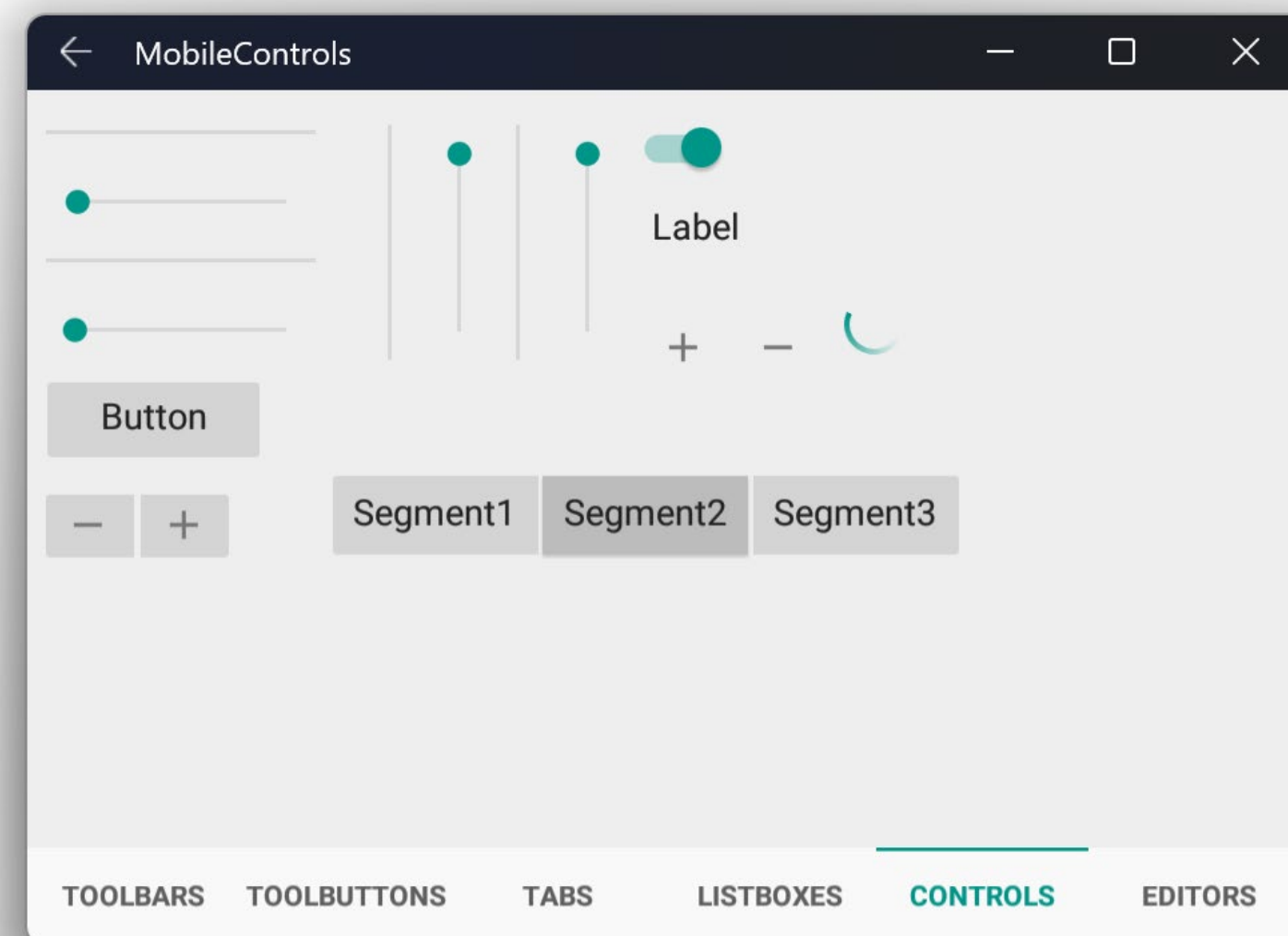
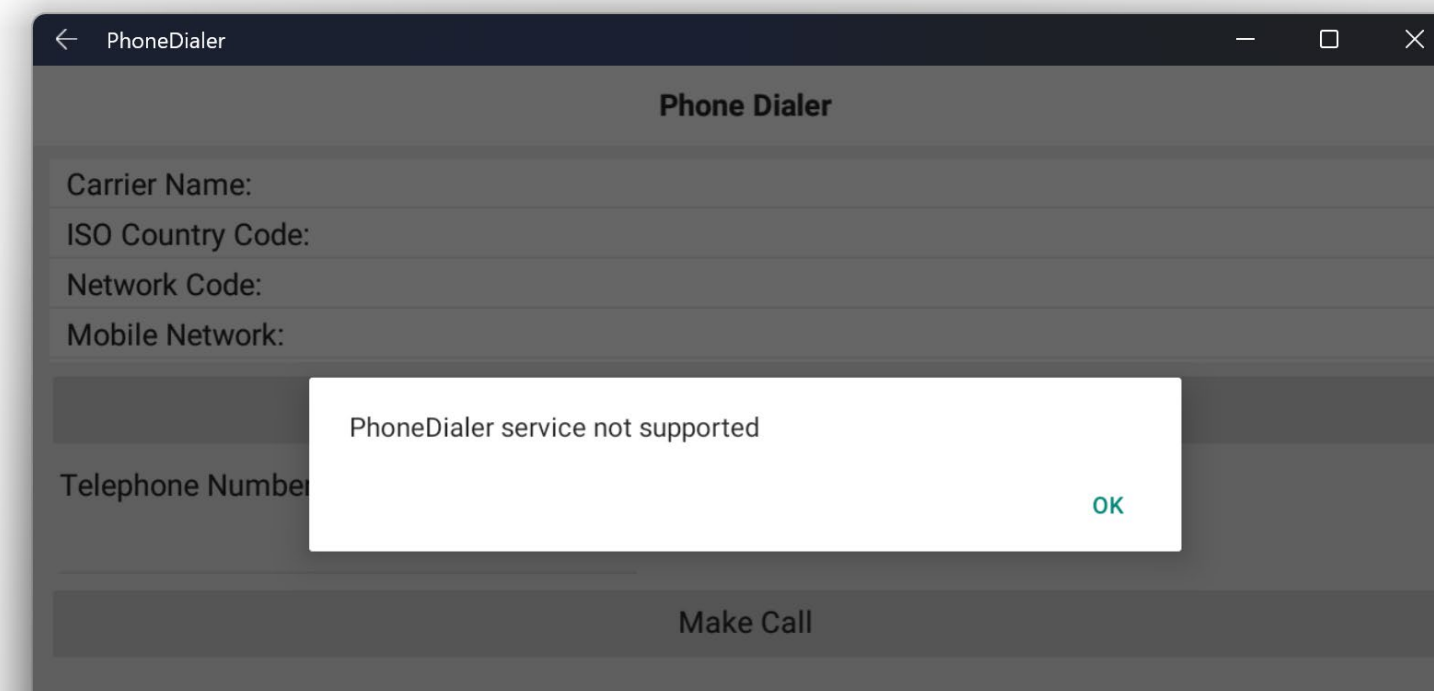


```
1 @echo off
2 REM Call from Delphi/RAD Studio IDE with the following parameters
3 REM $PATH($EXENAME) $NAMEONLY($PROJECT)
4 echo =====
5 echo Be sure you Compile Android 64 and Deploy [Shift-Ctrl-Alt-F9]
6 echo.
7 echo Also set <application android:resizeableActivity="true"> in Andro
8 echo =====
9 echo Path: %1
10 set apk=%2\bin\%2.apk
11 echo APK: %apk%
12 cd %1
13 cd %2\bin
14 echo.
15 dir %2.apk
16 echo.
17 if exist %2.apk (
18 adb install -r --no-streaming %2.apk
19 adb shell am start -n com.embarcadero.%2/com.embarcadero.firemonkey.FM
20 echo You can close this window....
21 timeout /t 10
22 ) else (
23 echo =====
24 echo Your APK file is missing!
25 echo Be sure you Compile Android 64 and Deploy [Shift-Ctrl-Alt-F9]
26 echo =====
27 pause
28 )
```


Mobile Snippets Tests

Working	Unsupported
Location*	Camera
Notifications	Accelerometer
Share Sheet	Phone Dialer
FireDAC SQLite	Orientation
FireDAC IB Lite	Gyroscope

- Location must be enabled in Windows to work in WSA
- Notification appear on desktop and can wake WSA



More information

- Webinar Slides, Links, Replay: blogs.embarcadero.com/?p=134192
- ADB Deployment GIST: gist.github.com/jimmckeeth/3a453ccdb1a387644ca137b499d01879
- Introduction blog: blogs.windows.com/windows-insider/2021/10/20/introducing-android-apps-on-windows-11-to-windows-insiders/
- Windows Subsystem for Android docs: docs.microsoft.com/en-us/windows/android/wsa/
- WSA Release Notes: docs.microsoft.com/en-us/windows/android/wsa/release-notes
- WSA Hardware Requirements: support.microsoft.com/windows/f8d0abb5-44ad-47d8-b9fb-ad6b1459ff6c
- Mobile apps on Windows: support.microsoft.com/windows/abed2335-81bf-490a-92e5-fe01b66e5c48
- Video from Scott on WSA: https://youtu.be/FN6zON_zKd8
- Scott Hanselman online: www.hanselman.com, www.hanselminutes.com and [@shanselman](https://twitter.com/shanselman)
- *Slight of ARM: Demystifying Intel Houdini* by Brian Hong from DefCon 29
 - Video: <https://youtu.be/9oQ5XjA1aq0> Slides: <https://bit.ly/3IzoMkM>

Next Steps

- Start a RAD Studio trial embarcadero.com/products/rad-studio/start-for-free
- Get Community Edition embarcadero.com/products/delphi/starter
- See special offers embarcadero.com/radoffer
- Subscribe on YouTube youtube.com/c/EmbarcaderoTechnologies
- Follow us on Twitter twitter.com/embarcaderotech
- Like us on Facebook facebook.com/embarcaderotech
- Follow us on LinkedIn linkedin.com/company/embarcadero-technologies
- Read our blog blogs.embarcadero.com

Android on Windows 11

A Developers Perspective

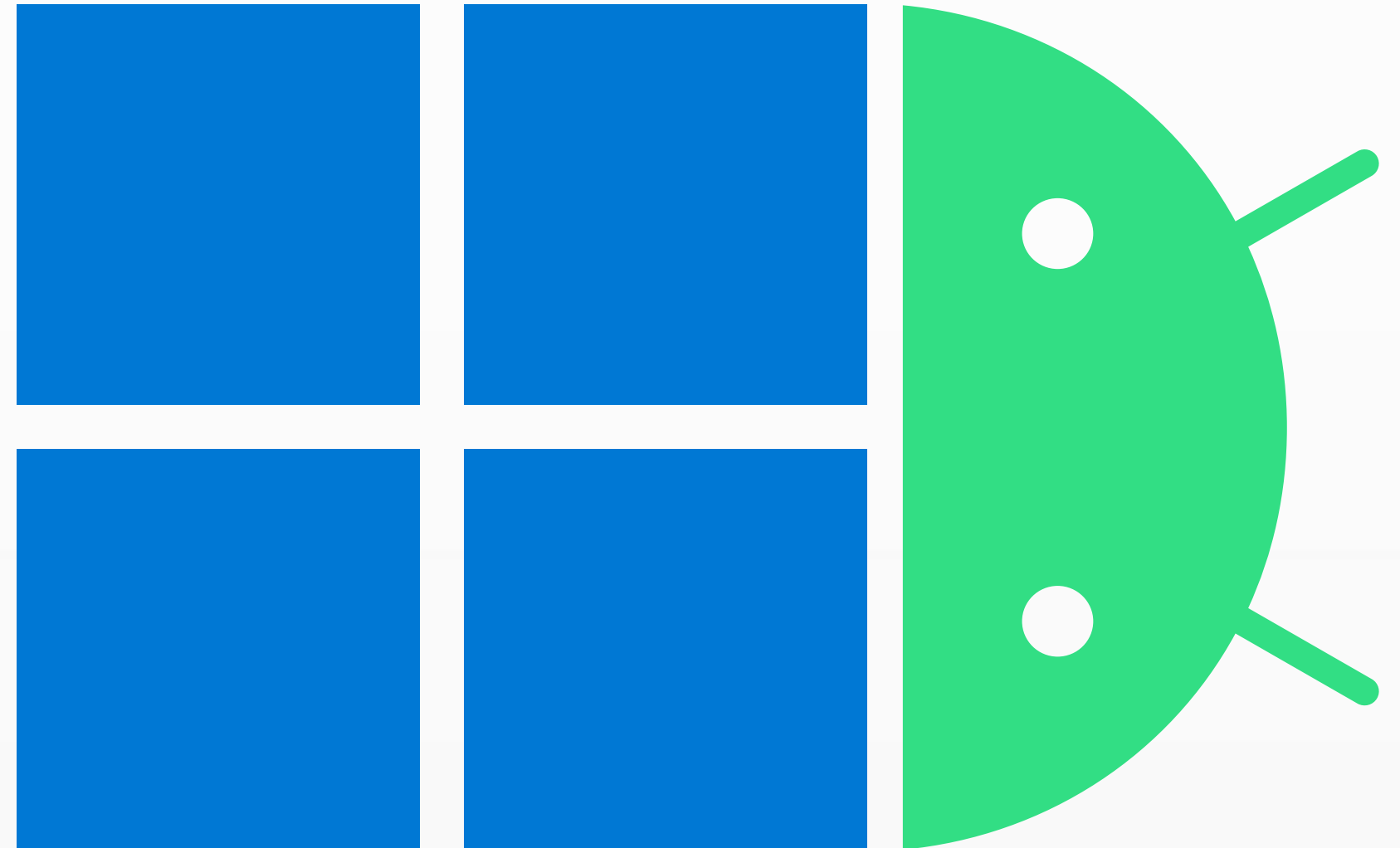


Scott Hanselman
Partner Program Manager
Microsoft
@shanselman



Jim McKeeth
Developer Advocate
Embarcadero
jim.mckeeth@embarcadero.com
@JimMcKeeth

Windows Subsystem for Android (WSA)



Q&A