XAM101

Getting Started with Xamarin

Download class materials from
university.xamarin.com

Microsoft

Xamarin University

Getting Started at Xamarin University

❖ To get the most out of your Xamarin University experience, be sure to watch the short welcome video which outlines the features and benefits

Getting Started at Xamarin University
Welcome to Xamarin University
MORE VIDEOS
0:06 / 15:52

university.xamarin.com/getting-started

The Xamarin Approach

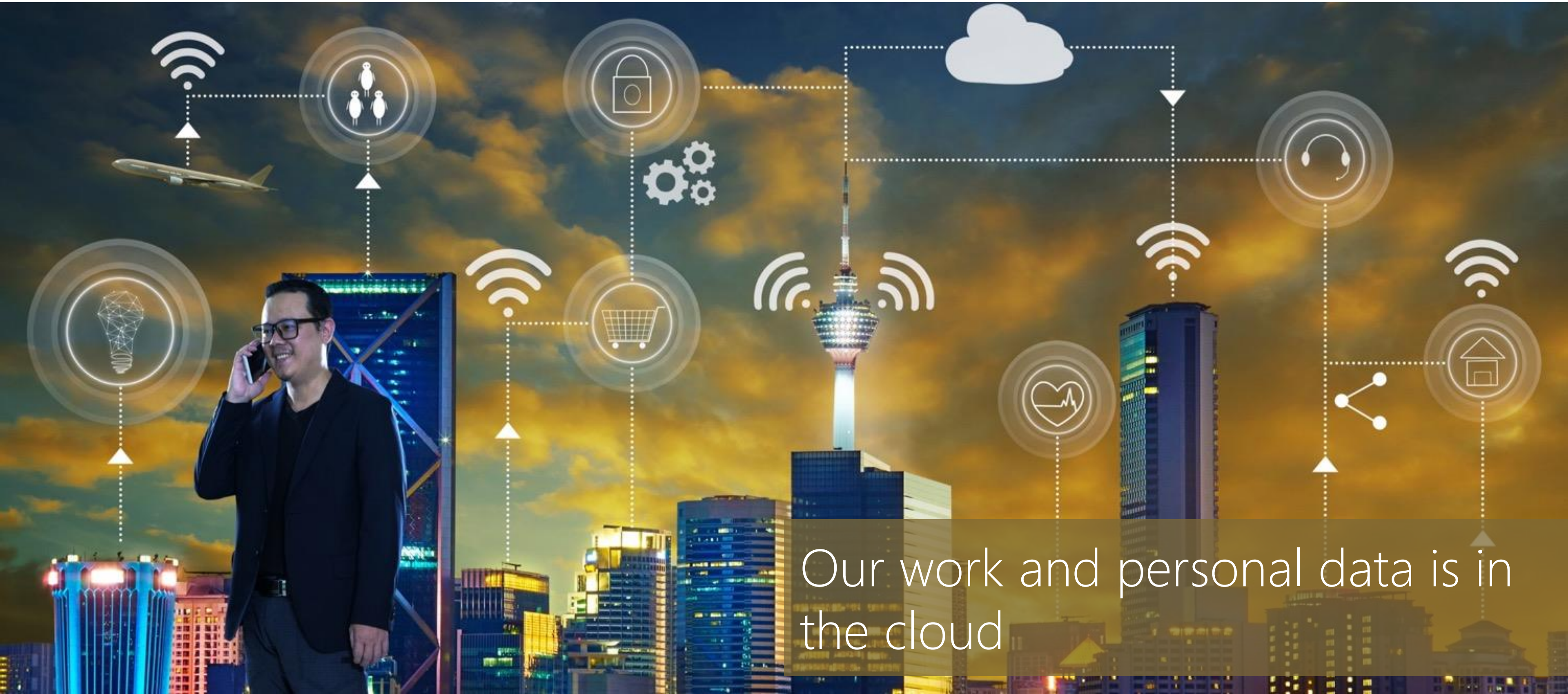
Xamarin University

Tasks

- The state of mobile development today
- Discuss mobile app trends
- Identify approaches to development
- Discover the Xamarin Approach

Xamarin University

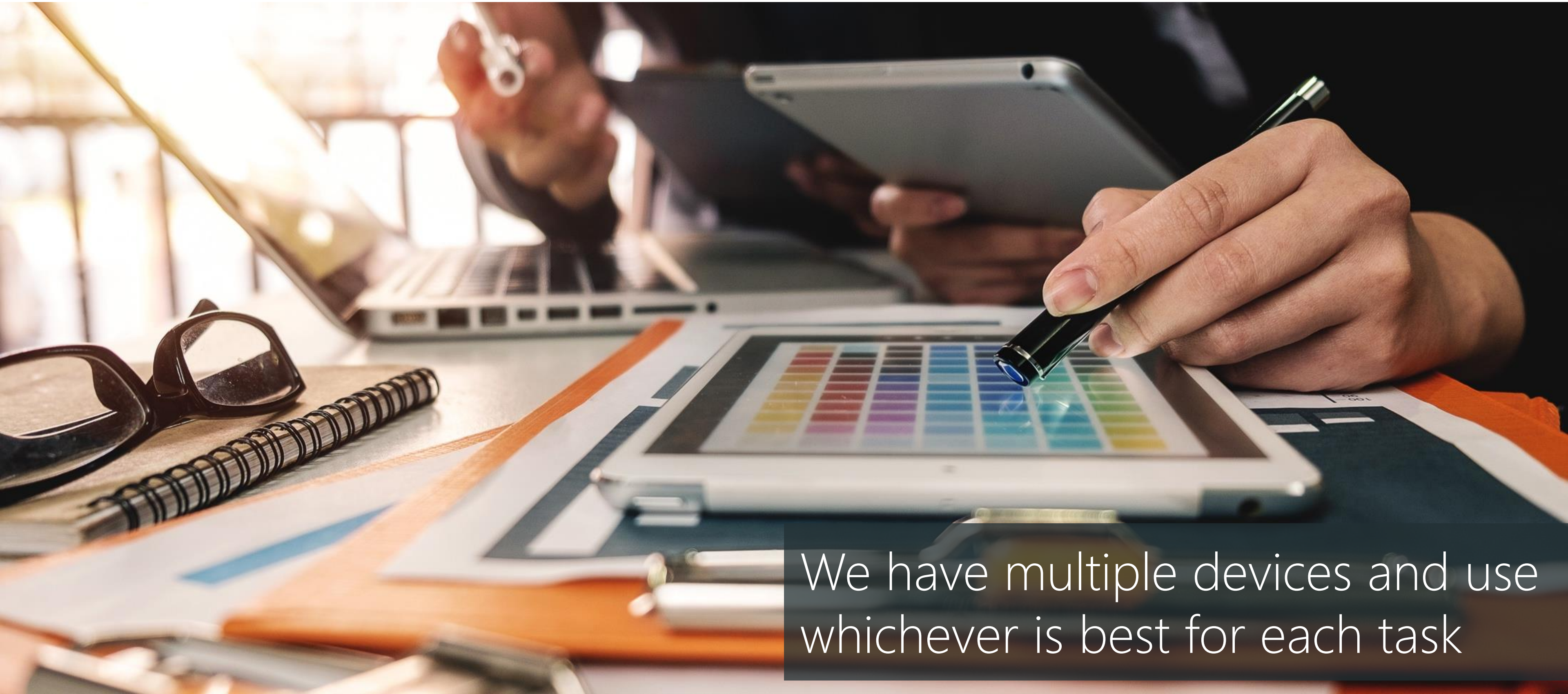
It's a mobile-first, cloud-first world



Our work and personal data is in the cloud

Xamarin University

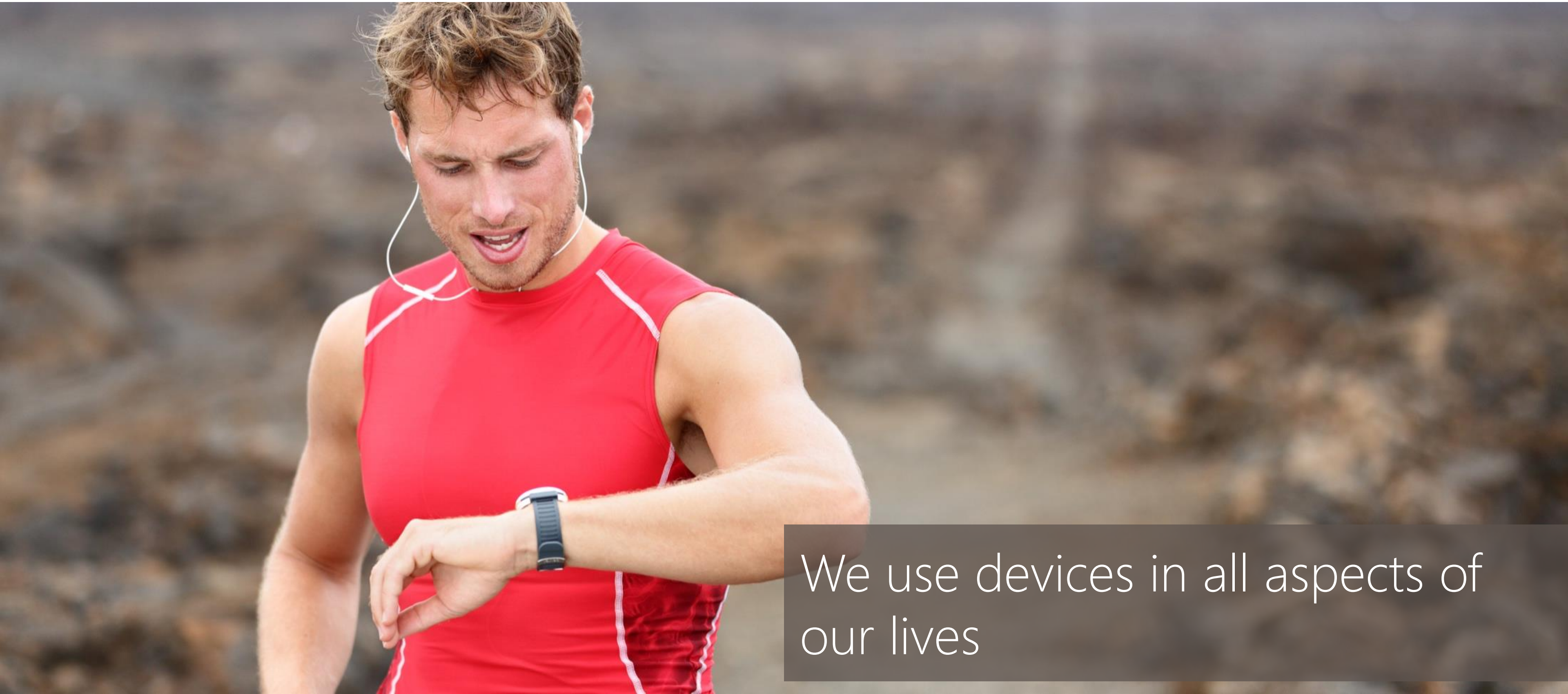
It's a mobile-first, cloud-first world



We have multiple devices and use whichever is best for each task

Xamarin University

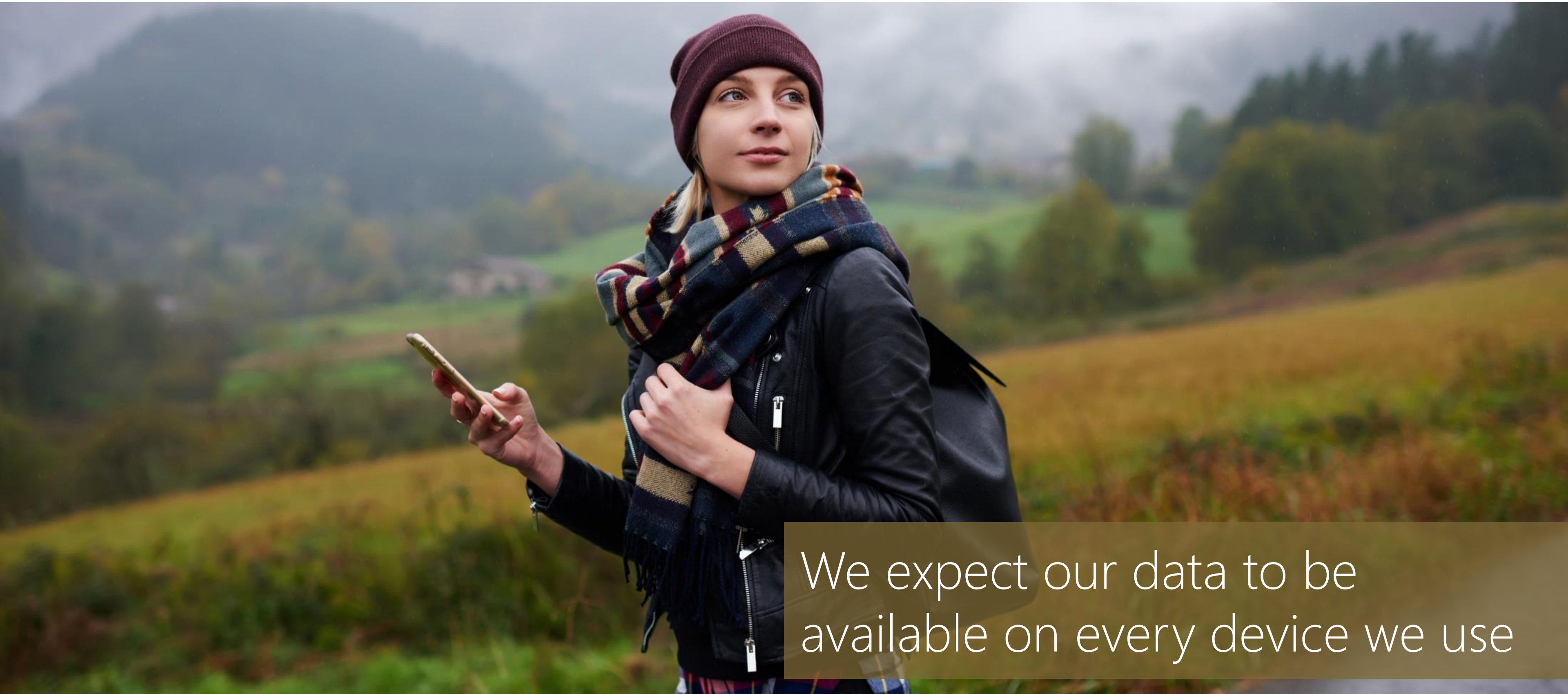
It's a mobile-first, cloud-first world



We use devices in all aspects of our lives

Xamarin University

It's a mobile-first, cloud-first world

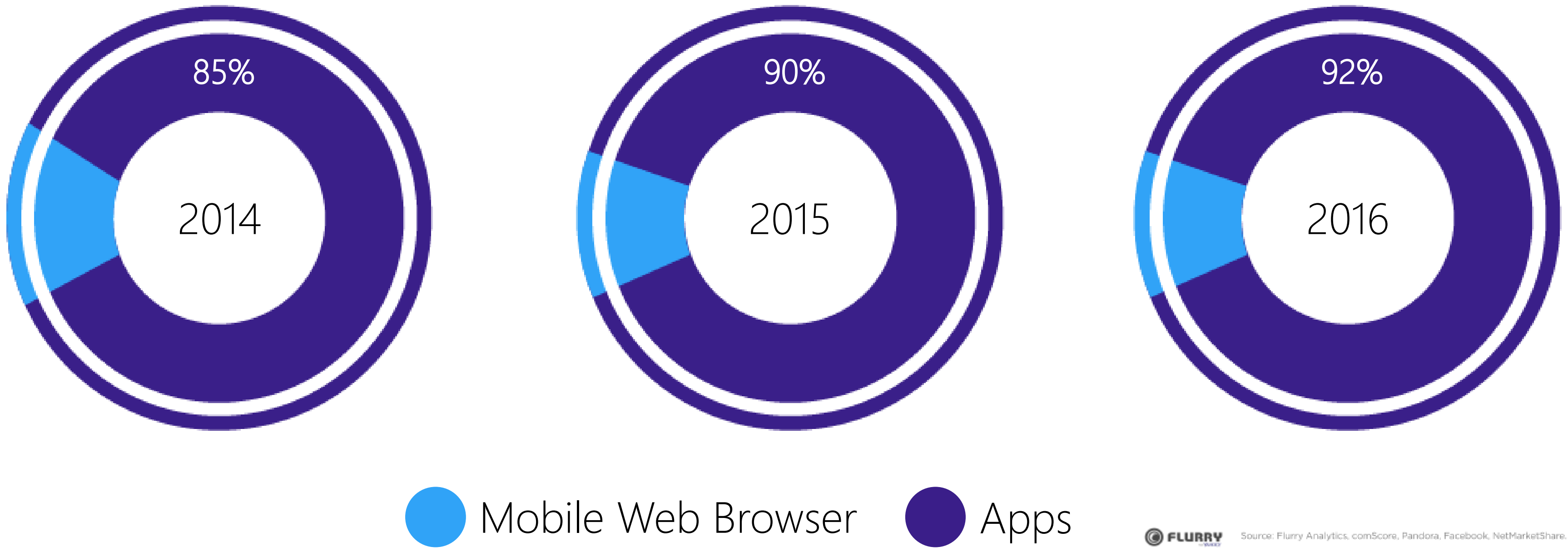


We expect our data to be available on every device we use

Xamarin University

Mobile app trends

- Users prefer apps over browsers on their mobile devices

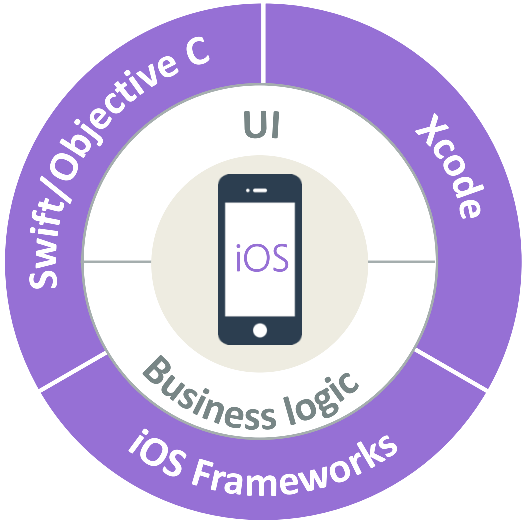


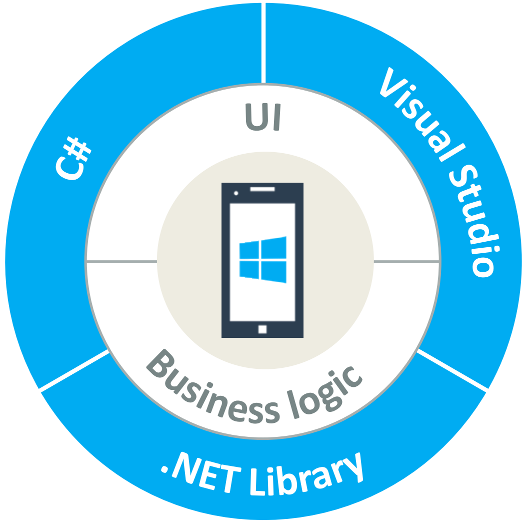
FLURRY Source: Flurry Analytics, comScore, Pandora, Facebook, NetMarketShare. Note: US Jun 2015

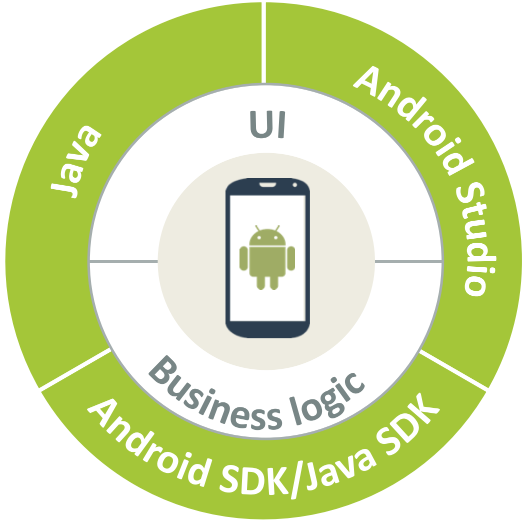
Xamarin University

Traditional approach [definition]

❖ Traditionally, apps have separate code bases written in their native language, are built using native tools, and utilize platform-specific features







Xamarin University

Traditional approach [cons]

- Traditional app development takes longer, requires multiple teams, multiple IDEs, and **cannot share code**

C#

```
double ComputeTax(Item[] items)
{ ...
   foreach (var item in items)
   ...
}
```

Swift

```
func computeTax(items: [Item]) -> Double
{ ...
  for item in items
  ...
}
```

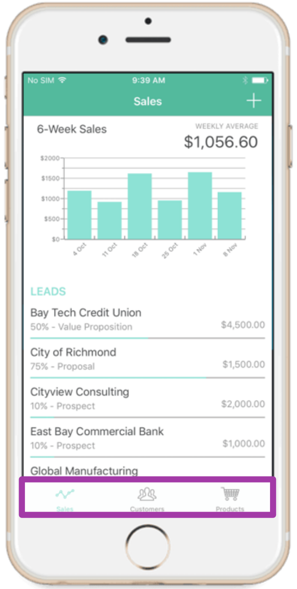
Java

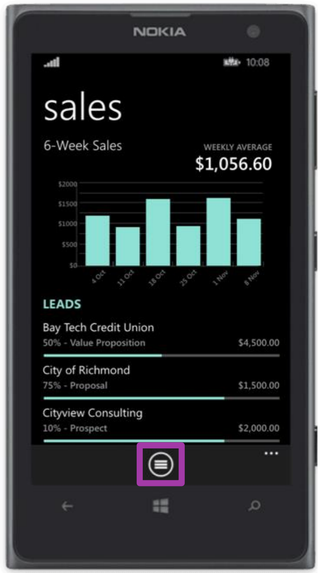
```
double computeTax(Item[] items)
{ ...
  for (Item item : items)
  ...
}
```

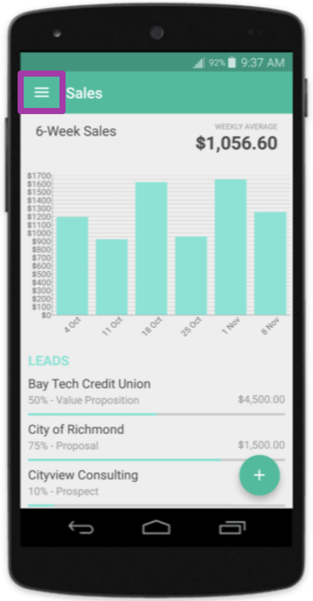
Traditional approach [pros]

❖ Traditional apps typically follow each platform's user-experience guidelines for things like navigation style, page layout, settings, etc.

E.g. implement the navigation style that users of each platform expect







Xamarin University

What is Xamarin?

❖ Xamarin is an app-development platform that lets you build apps for many operating systems from a single, shared code base

Xamarin University

Xamarin tools

- You use Visual Studio, C#, and the .NET Libraries to build Xamarin apps

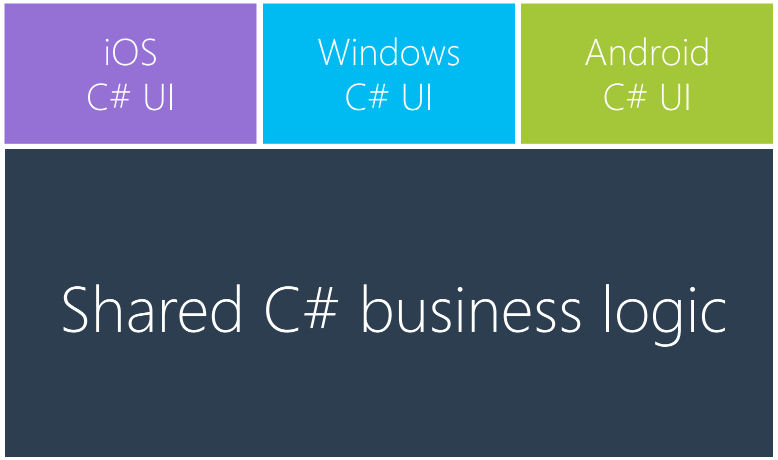
Visual Studio
for Windows or Mac

C#

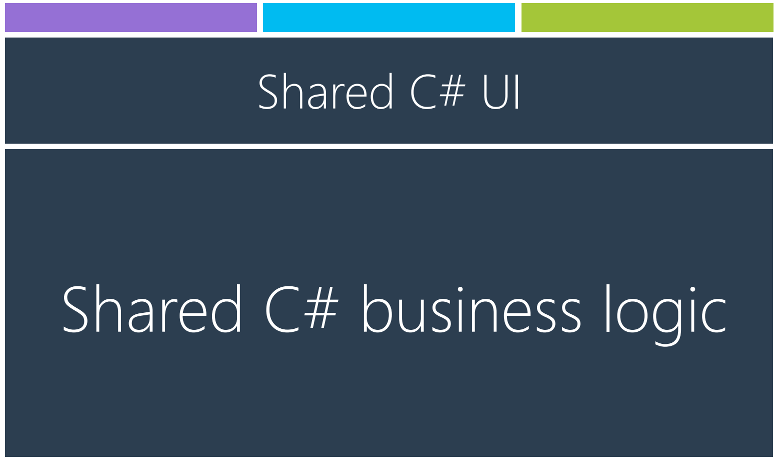
System.Net	System
System.Data	System.Windows
System.IO	System.Linq
System.Numerics	System.Core
System.Xml	System.ServiceModel

Xamarin development approaches

- Xamarin offers you two strategies: separate UI or shared UI



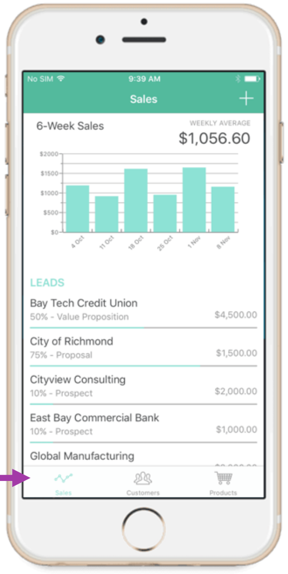
Xamarin.iOS and Xamarin.Android



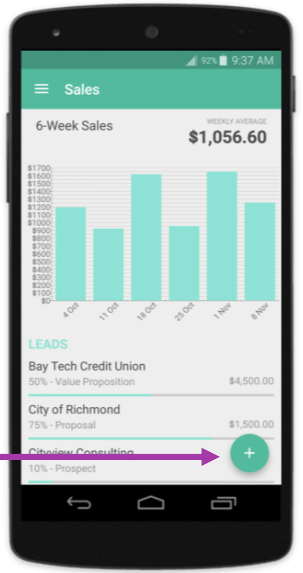
Xamarin.Forms

Xamarin.iOS and Xamarin.Android

❖ Create your business logic once and share it across platforms, while leveraging all of the native controls/features your users expect



Familiar navigation



Familiar controls

Shared code base

```
double ComputeTax(Item[] items)
{ ...
  foreach (var item in items)
  ...
}
```

Xamarin.iOS – 100% API Coverage

- Anything you can do in Swift or Objective C for iOS you can do with Xamarin using C#

MapKit	UIKit	iBeacon	CoreGraphics	CoreMotion
System.Net	System	System.IO	System.Linq	System.Xml
System.Data	System.Windows	System.Numerics	System.Core	System.ServiceModel

100% API coverage with the added benefit of the .NET APIs

Xamarin.Android – 100% API Coverage

- Anything you can do in Java for Android you can do with Xamarin using C#

Text-to-speech	Toolbar	Printing Framework	Renderscript	NFC
System.Net	System	System.IO	System.Linq	System.Xml
System.Data	System.Windows	System.Numerics	System.Core	System.ServiceModel

100% API coverage with the added benefit of the .NET APIs

Windows

- Windows apps are built in C# with all of the Native APIs

Microsoft.Phone	Microsoft.Networking	Windows.Storage	Windows.Foundation	Microsoft.Devices
System.Net	System	System.IO	System.Linq	System.Xml
System.Data	System.Windows	System.Numerics	System.Core	System.ServiceModel

Windows apps support C# natively

Platform libraries

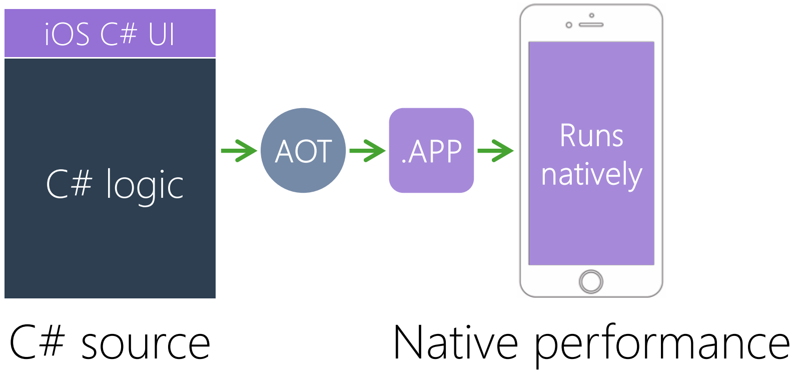
❖ Xamarin provides a C# version of every native library type

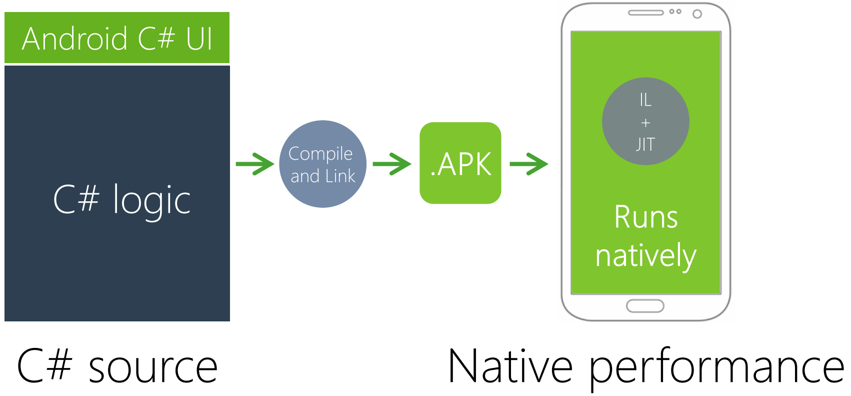
```
public class TextView : View ...
{ ...
   public string Text { get; set; }
   public event EventHandler<TextChangedEventArgs> TextChanged;
}
```

```
public class TextView extends View ...
{  ...
   public CharSequence getText() { return null; }
   public final void setText(CharSequence text) { }

   public void addTextChangedListener(TextWatcher watcher) { }
   public void removeTextChangedListener(TextWatcher watcher) { }
}
```

Xamarin performance

- Xamarin apps are fully native, you get fully native performance with the benefits of shared code





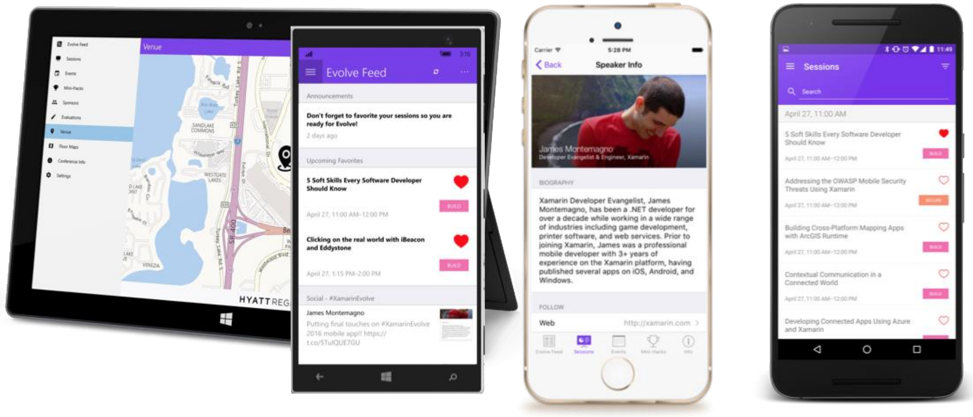
Demonstration

Cross-platform applications with Xamarin

Xamarin University

Xamarin.Forms

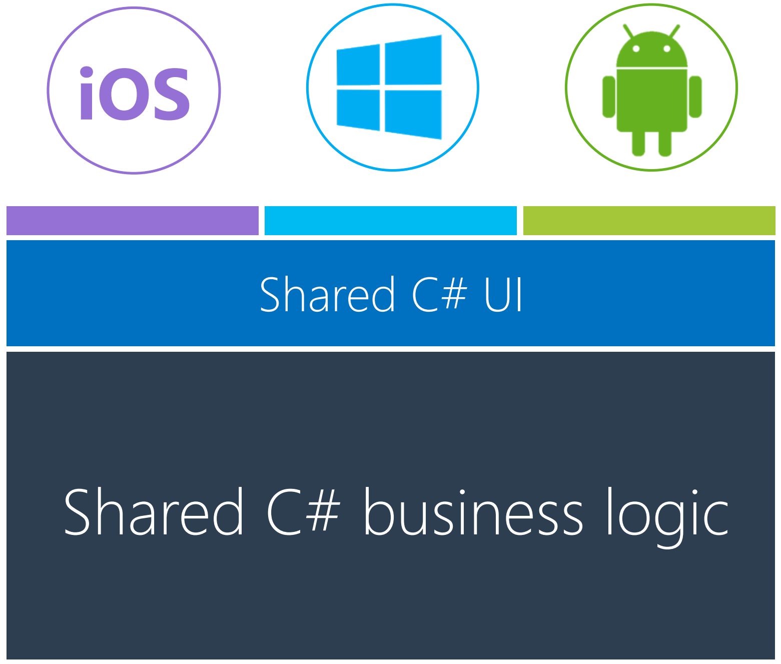
- Xamarin.Forms enables even more code-sharing through a shared UI definition when deep platform integration is unnecessary



Build native UIs for Android, iOS, and Windows from a single, shared C# codebase

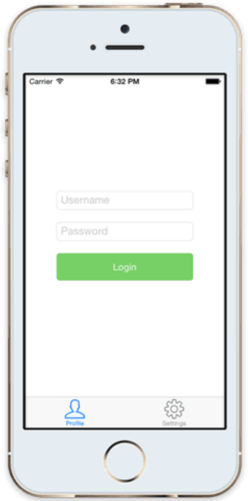
Included in Xamarin.Forms

- ✓ UI building blocks like pages, layouts, and controls
- ✓ XAML-defined UI
- ✓ Data binding
- ✓ Navigation
- ✓ Animation API
- ✓ Dependency Service
- ✓ Messaging Center

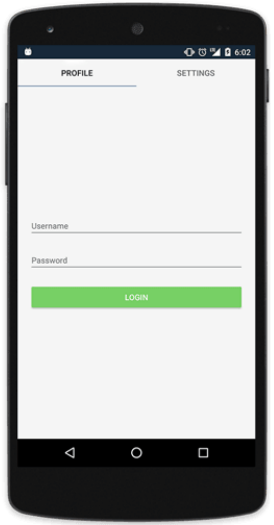


Native controls are used at runtime

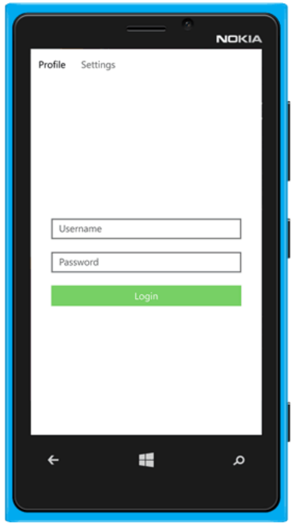
```
<?xml version="1.0" encoding="UTF-8"?>
<TabbedPage xmlns="http://xamarin.com/schemas/2014/forms"
          xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
          x:Class="MyApp.MainPage">

   <ContentPage Title="Profile" Icon="Profile.png">
      <StackLayout Spacing="20" Padding="20"
                   VerticalOptions="Center">
         <Entry Placeholder="Username" Text="{Binding Username}"/>
         <Entry Placeholder="Password" Text="{Binding Password}"
                IsPassword="true"/>
         <Button Text="Login" TextColor="White"
                 BackgroundColor="#77D065"
                 Command="{Binding LoginCommand}"/>
      </StackLayout>
   </ContentPage>

   <ContentPage Title="Settings" Icon="Settings.png">
   </ContentPage>

</TabbedPage>
```



UITextField



EditText



TextBox

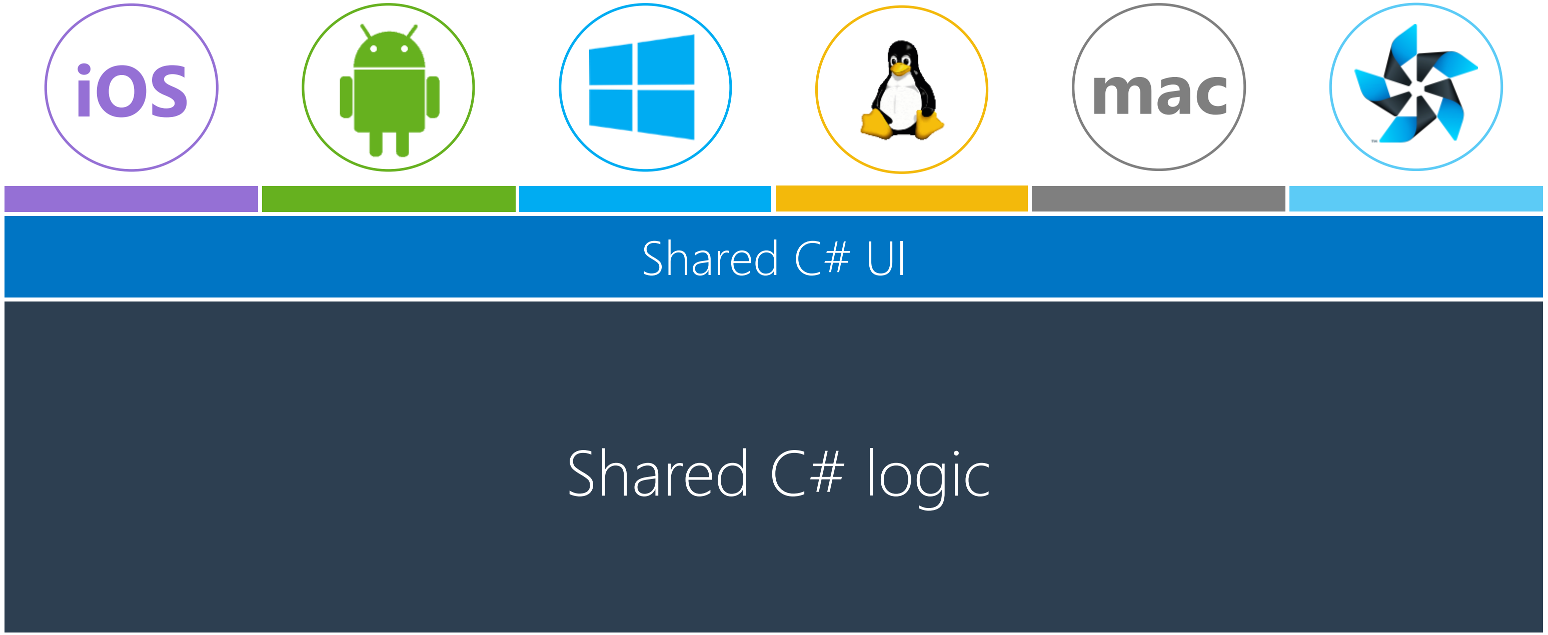
Demonstration

Cross-platform applications with Xamarin

Xamarin University

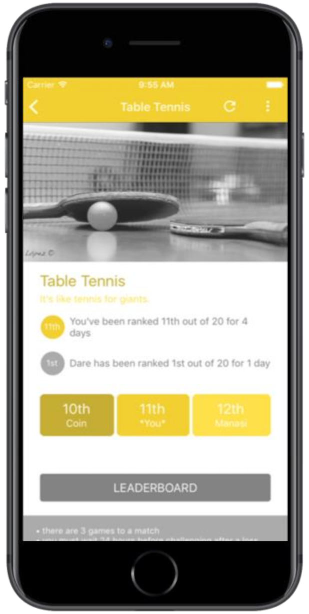
Xamarin.Forms platform support

❖ Xamarin.Forms supports a broad selection of mobile and desktop platforms and UI frameworks



Beautiful apps in less time

- Create great looking apps that have feature parity with native performance and enjoy the benefit of shared UI and business logic with Xamarin.Forms



Carrier 9:55 AM

Table Tennis

Table Tennis

It's like tennis for giants.

11th You've been ranked 11th out of 20 for 4 days

1st Dare has been ranked 1st out of 20 for 1 day

10th Coin	11th *You*	12th Manasi

LEADERBOARD

• there are 3 games to a match

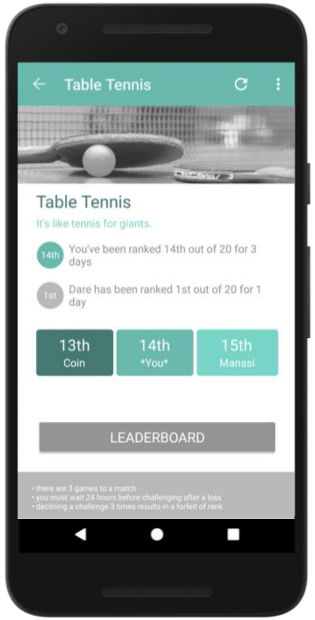


Table Tennis

Table Tennis

It's like tennis for giants.

14th You've been ranked 14th out of 20 for 3 days

1st Dare has been ranked 1st out of 20 for 1 day

13th Coin	14th *You*	15th Manasi

LEADERBOARD

• there are 3 games to a match
• you must wait 24 hours before challenging after a loss
• declining a challenge 3 times results in a forfeit of rank

Open Source – open.xamarin.com

Build the future of apps with Xamarin.

Xamarin SDK is now fully available under the MIT license.

Xamarin University

One code base, unlimited possibilities

- With one code base and native performance you can meet your customers where they need to be

Install Xamarin

Xamarin University

Before we start…

- Download and launch the Xamarin Installer *now* on your development machine to begin the automated setup so you are ready when we hit the installation section

Visual Studio Installer
https://www.visualstudio.com

Note: for iOS development with Visual Studio on Windows, you also need to set up a Mac with the Xamarin tools.

Supported operating systems

❖ Xamarin tools can be installed on macOS and Windows

iOS, Android, and macOS
Visual Studio for Mac

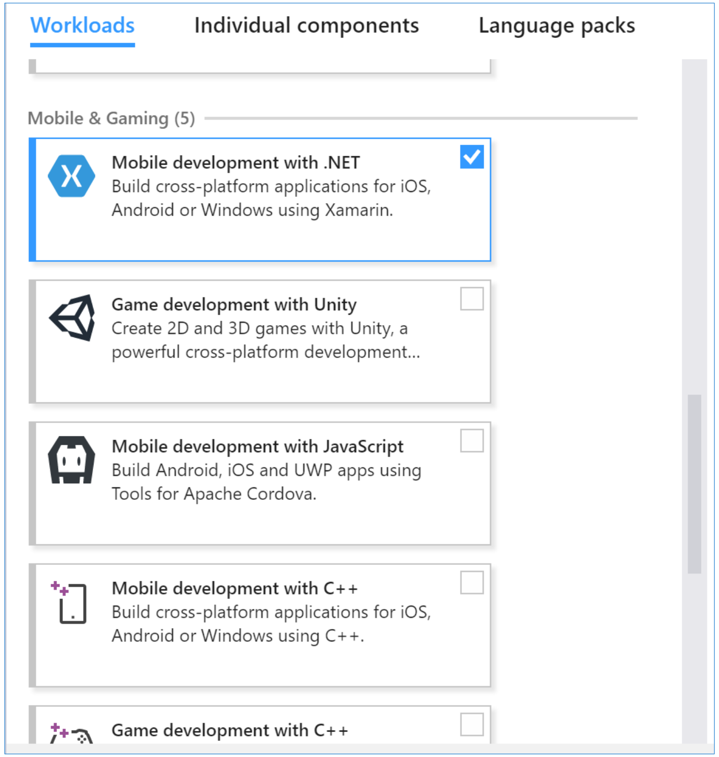
iOS, Android, and Windows
Visual Studio IDE

Xamarin University

Install on Windows

- On Windows, Xamarin installs directly from the Visual Studio Installer

Make sure the Mobile development with .NET Workload is selected



[✓] **Mobile development with .NET**
Build cross-platform applications for iOS, Android or Windows using Xamarin.

[ ] **Game development with Unity**
Create 2D and 3D games with Unity, a powerful cross-platform development...

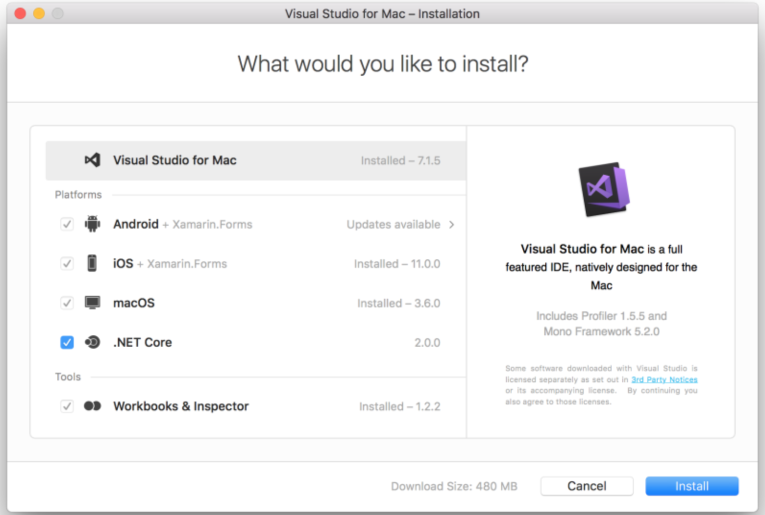
[ ] **Mobile development with JavaScript**
Build Android, iOS and UWP apps using Tools for Apache Cordova.

[ ] **Mobile development with C++**
Build cross-platform applications for iOS, Android or Windows using C++.

[ ] **Game development with C++**

Install on a Mac

- When developing on a Mac, the first thing you should do is install Xcode and use the Xamarin Unified Installer to download and install required components



Visual Studio for Mac – Installation

What would you like to install?

	Visual Studio for Mac	Installed – 7.1.5
Platforms		
✓	Android + Xamarin.Forms	Updates available ›
✓	iOS + Xamarin.Forms	Installed – 11.0.0
✓	macOS	Installed – 3.6.0
✓	.NET Core	2.0.0
Tools		
✓	Workbooks & Inspector	Installed – 1.2.2

Visual Studio for Mac is a full featured IDE, natively designed for the Mac

Includes Profiler 1.5.5 and Mono Framework 5.2.0

Some software downloaded with Visual Studio is licensed separately as set out in 3rd Party Notices or its accompanying license. By continuing you also agree to those licenses.

Download Size: 480 MB

Visual Studio Enterprise benefits

❖ There are additional benefits included with a Visual Studio Enterprise license

Bytecode hiding for Android APKs

Live app inspector

Profiler

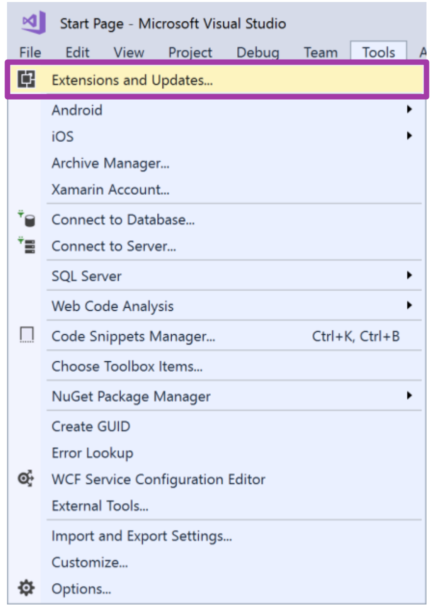
Test Recorder

Important: Make sure to use the Enterprise installer from the Visual Studio download page to ensure you get the correct edition of the development environment installed!

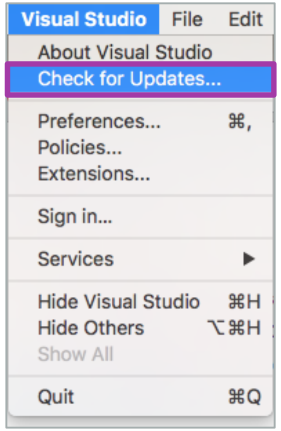
Xamarin University

Keep Xamarin up to date

- Xamarin releases updates to add new APIs, match vendor releases, and fix issues



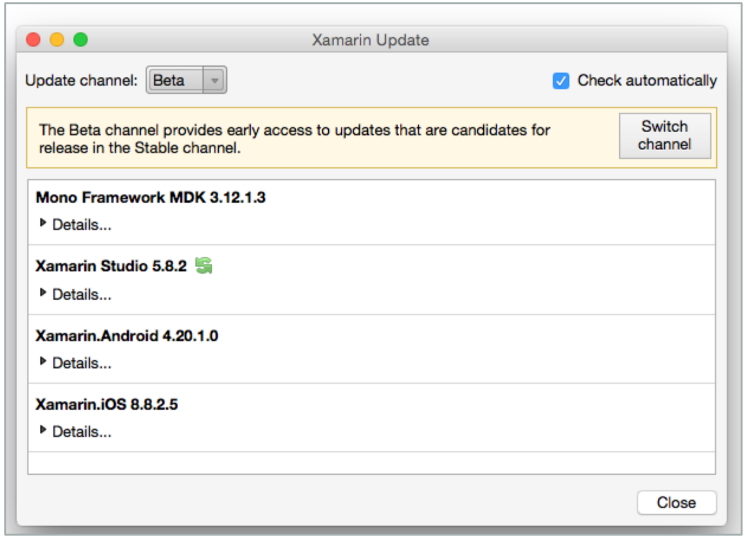
Visual Studio on Windows



Visual Studio for Mac

Xamarin macOS release channels

- Xamarin updates are deployed in stages for macOS, and exposed through release channels (Alpha > Beta > Stable)



Update channel: Beta

Check automatically: ✓

The Beta channel provides early access to updates that are candidates for release in the Stable channel.

Mono Framework MDK 3.12.1.3

Xamarin Studio 5.8.2

Xamarin.Android 4.20.1.0

Xamarin.iOS 8.8.2.5

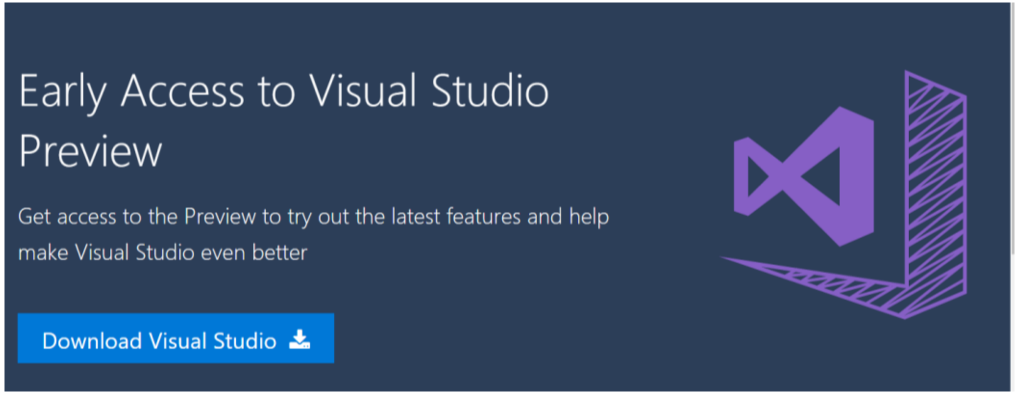
Alpha:	most current, least tested
Beta:	what's next
Stable:	released code, most tested

Important: We recommend the stable channel for most classes

Xamarin University

Xamarin pre-release on Windows

❖ Xamarin early access releases are deployed with Visual Studio Preview, available for download


Early Access to Visual Studio Preview

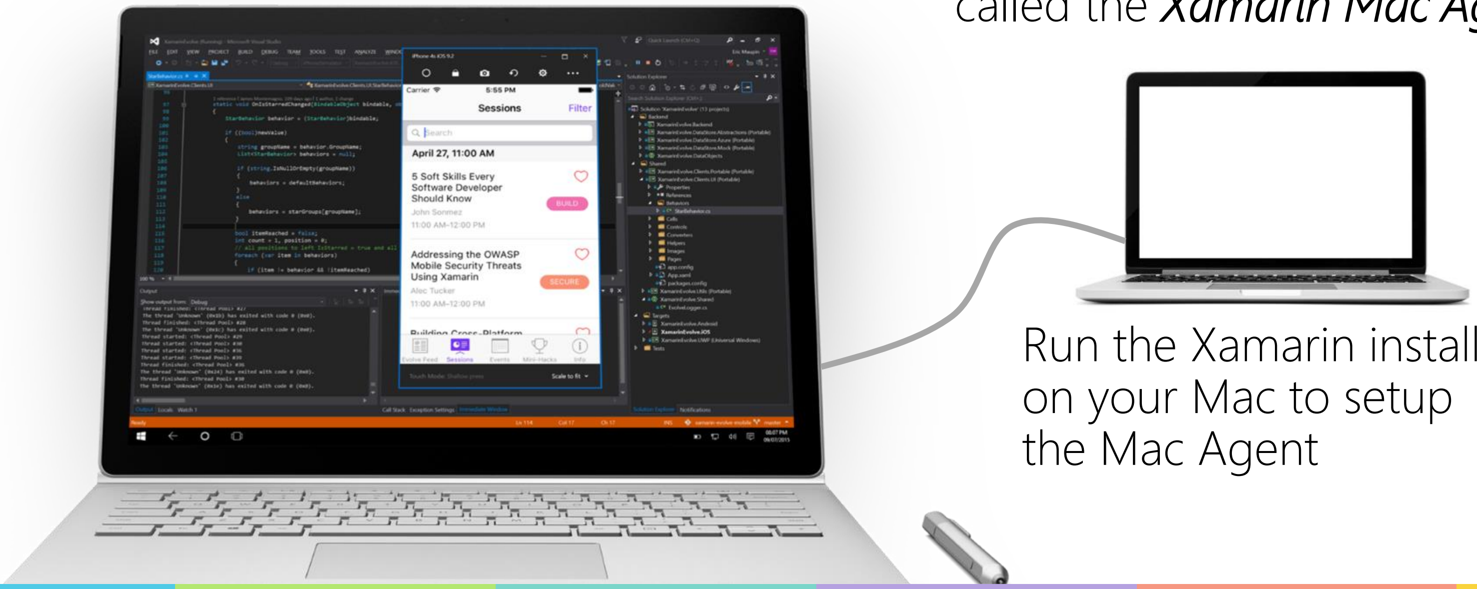
Get access to the Preview to try out the latest features and help make Visual Studio even better

Important: We recommend the standard Visual Studio release for most classes

iOS Development Requires a Mac

You work in Visual Studio on Windows

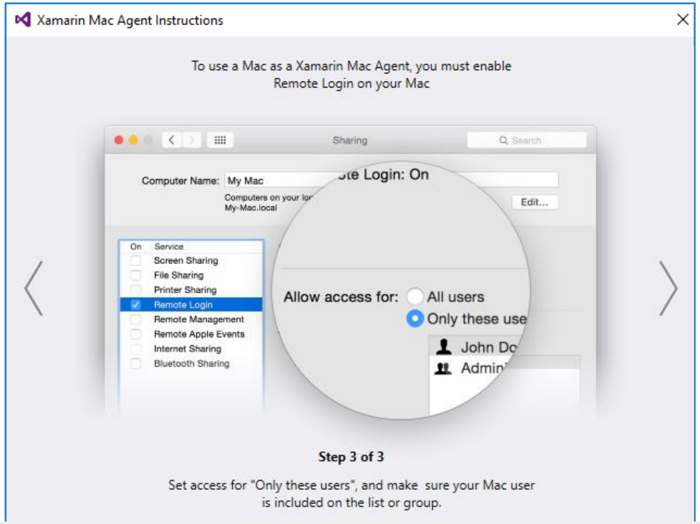
Delegates parts of the build to a Mac using a server process called the *Xamarin Mac Agent*



Run the Xamarin installer on your Mac to setup the Mac Agent

Connecting to Mac Agent

❖ Creating or opening an iOS project in VS will login to the associated Mac host, if no host is available, it will launch the connection wizard



To use a Mac as a Xamarin Mac Agent, you must enable Remote Login on your Mac

Step 3 of 3

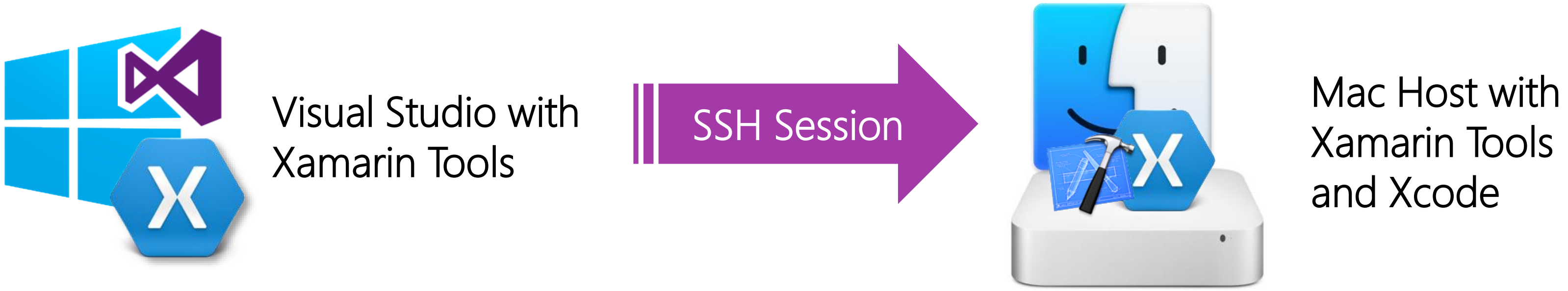
Set access for "Only these users", and make sure your Mac user is included on the list or group.

You can also use **Tools > iOS > Xamarin Mac Agent** to launch connection wizard manually to connect to a different host

Xamarin University

Connecting to the Mac

- Building an iOS application will automatically connect to the build agent

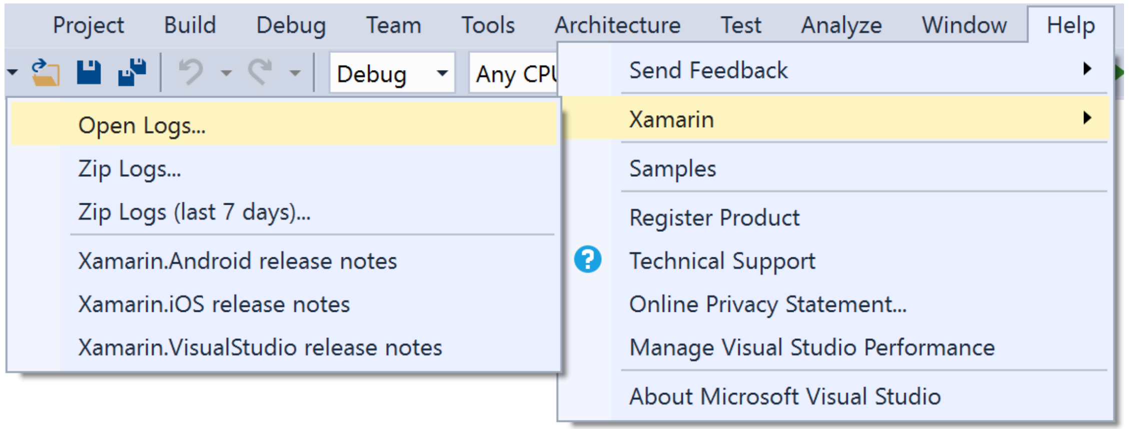

Visual Studio with Xamarin Tools
SSH Session
Mac Host with Xamarin Tools and Xcode

```
Starting connection to Mac 192.168.0.193...
Starting Broker in port 54837...
Connection successfully established with the Mac 192.168.0.193:54837
Starting agents on Mac 192.168.0.193 (192.168.0.193)
Starting Agent IDB...
Starting Agent Build...
Starting Agent Designer...
Agent Build is running
Agent IDB is running
Agent Designer is running
Connected to the Mac 192.168.0.193 (192.168.0.193) with Full support.
```

Xamarin University

Troubleshooting Xamarin Mac Agent

- The Xamarin Mac Agent will generally diagnose and help correct connection issues; use **Help > Xamarin** for more detailed log information if necessary

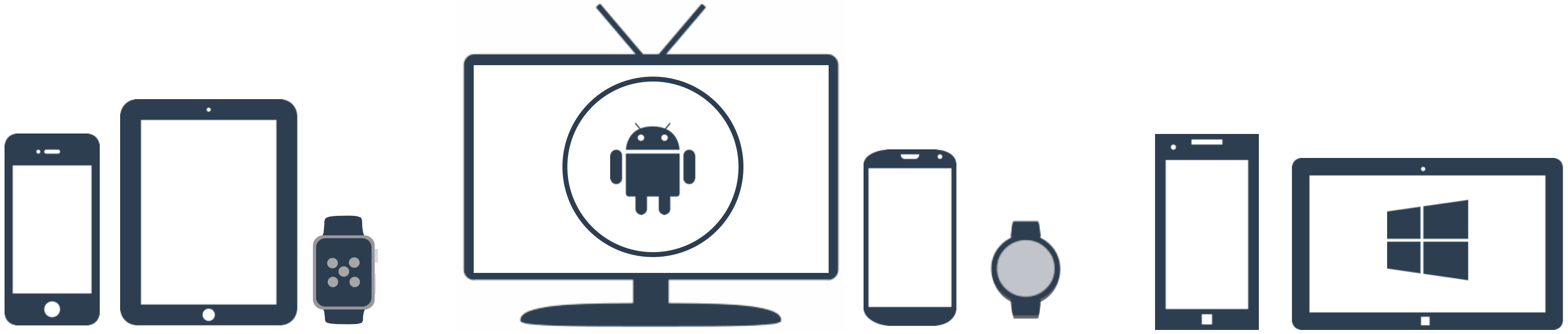


Project
Build
Debug
Team
Tools
Architecture
Test
Analyze
Window
Help
Debug
Any CPU
Send Feedback
Xamarin
Samples
Register Product
Technical Support
Online Privacy Statement...
Manage Visual Studio Performance
About Microsoft Visual Studio
Open Logs...
Zip Logs...
Zip Logs (last 7 days)...
Xamarin.Android release notes
Xamarin.iOS release notes
Xamarin.VisualStudio release notes

Xamarin University

Running your applications

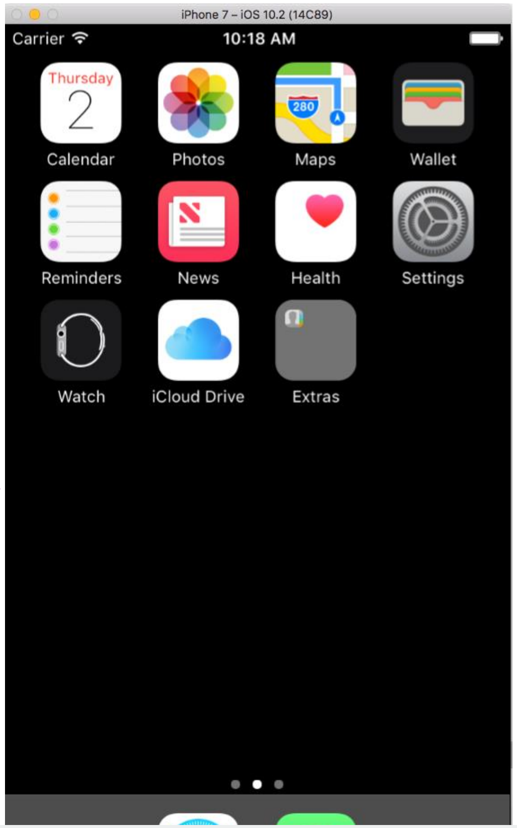
- You need to run applications to test them – can run on devices, or use emulators and simulators which simulate a real device in software



iOS, Android, and Windows all have emulators or simulators

Running iOS apps

- Apple supplies an iOS simulator with Xcode which can be launched on the Mac host

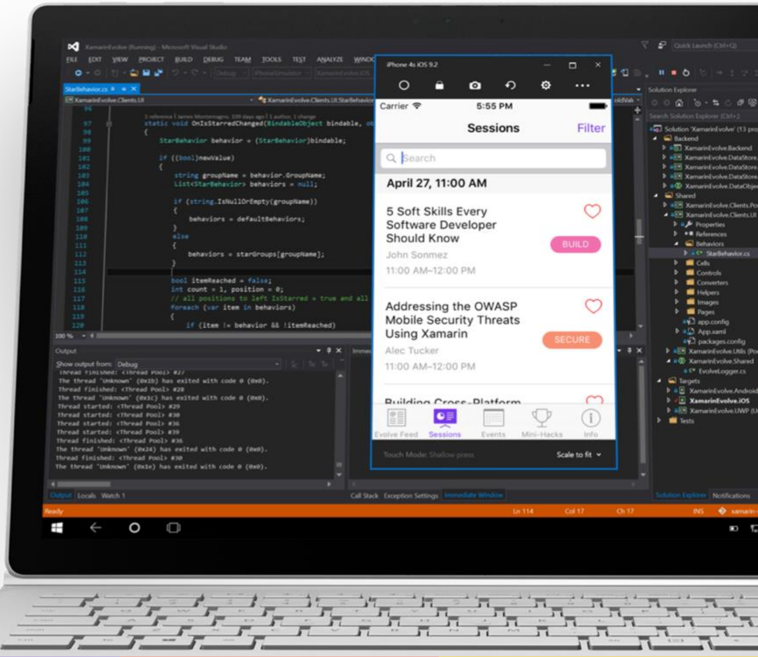


The simulator supports different devices, resolutions

Only the latest iOS release is installed by default, but you can use Xcode to install older iOS versions

Remoted iOS Simulator (for Windows)

- The Remoted iOS Simulator for Windows makes testing and debugging iOS apps is entirely possible within Visual Studio Enterprise on Windows
  - Supports rotation, screenshots, and location changes
  - Multi-touch and pressure-sensitive interaction
  - Performant



```
static void OnIsStarredChanged(BindableObject bindable, ob
{
    StarBehavior behavior = (StarBehavior)bindable;

    if ((bool)newValue)
    {
        string groupName = behavior.GroupName;
        List<StarBehavior> behaviors = null;

        if (string.IsNullOrEmpty(groupName))
        {
            behaviors = defaultBehaviors;
        }
        else
        {
            behaviors = starGroups[groupName];
        }

        bool itemReached = false;
        int count = 1, position = 0;
        // all positions to left IsStarred = true and all
        foreach (var item in behaviors)
        {
            if (item != behavior && !itemReached)
```

```
Thread finished: <Thread Pool> #27
The thread 'Unknown' (0x1b) has exited with code 0 (0x0).
Thread finished: <Thread Pool> #28
The thread 'Unknown' (0x1c) has exited with code 0 (0x0).
Thread started: <Thread Pool> #29
Thread started: <Thread Pool> #30
Thread started: <Thread Pool> #36
Thread started: <Thread Pool> #39
Thread finished: <Thread Pool> #36
The thread 'Unknown' (0x24) has exited with code 0 (0x0).
Thread finished: <Thread Pool> #30
The thread 'Unknown' (0x1e) has exited with code 0 (0x0).
```

April 27, 11:00 AM

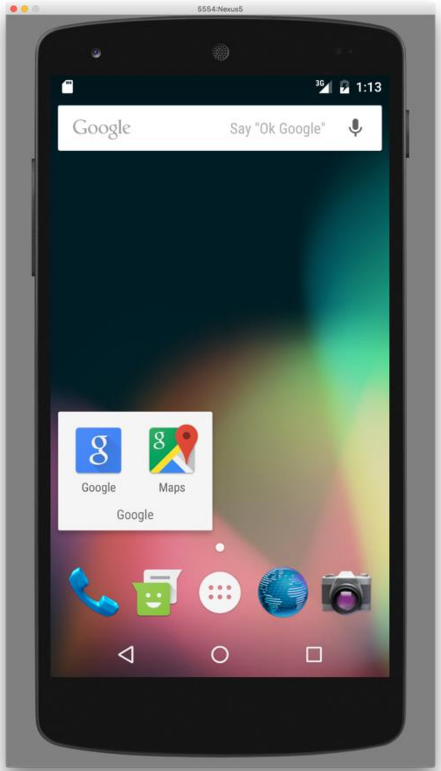
5 Soft Skills Every Software Developer Should Know
BUILD
John Sonmez
11:00 AM–12:00 PM

Addressing the OWASP Mobile Security Threats Using Xamarin
SECURE
Alec Tucker
11:00 AM–12:00 PM

Building Cross-Platform

Running Android apps

- Google provides the standard Android emulator and includes it with the Android SDK and often include Google apps support automatically

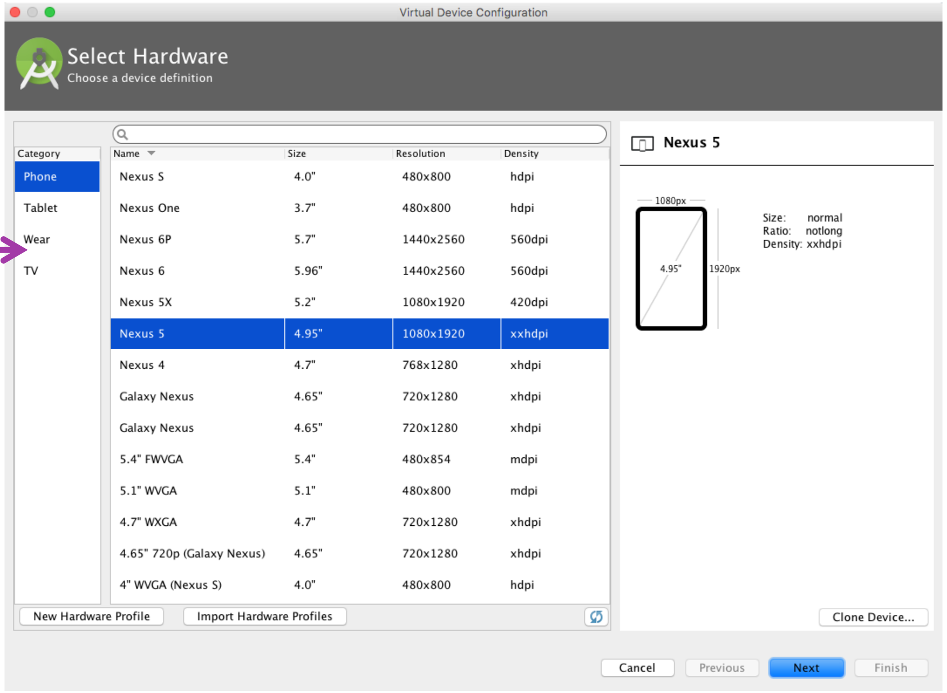


Improved emulator engine and configuration support is available if you install the **Android Studio** IDE from Google

Installing the Google emulators

- Google supports the widest variety of Android devices and versions

Includes TV and wearable definitions


Virtual Device Configuration

Select Hardware

Choose a device definition

Category: Phone, Tablet, Wear, TV

Name	Size	Resolution	Density
Nexus S	4.0"	480x800	hdpi
Nexus One	3.7"	480x800	hdpi
Nexus 6P	5.7"	1440x2560	560dpi
Nexus 6	5.96"	1440x2560	560dpi
Nexus 5X	5.2"	1080x1920	420dpi
Nexus 5	4.95"	1080x1920	xxhdpi
Nexus 4	4.7"	768x1280	xhdpi
Galaxy Nexus	4.65"	720x1280	xhdpi
Galaxy Nexus	4.65"	720x1280	xhdpi
5.4" FWVGA	5.4"	480x854	mdpi
5.1" WVGA	5.1"	480x800	mdpi
4.7" WXGA	4.7"	720x1280	xhdpi
4.65" 720p (Galaxy Nexus)	4.65"	720x1280	xhdpi
4" WVGA (Nexus S)	4.0"	480x800	hdpi

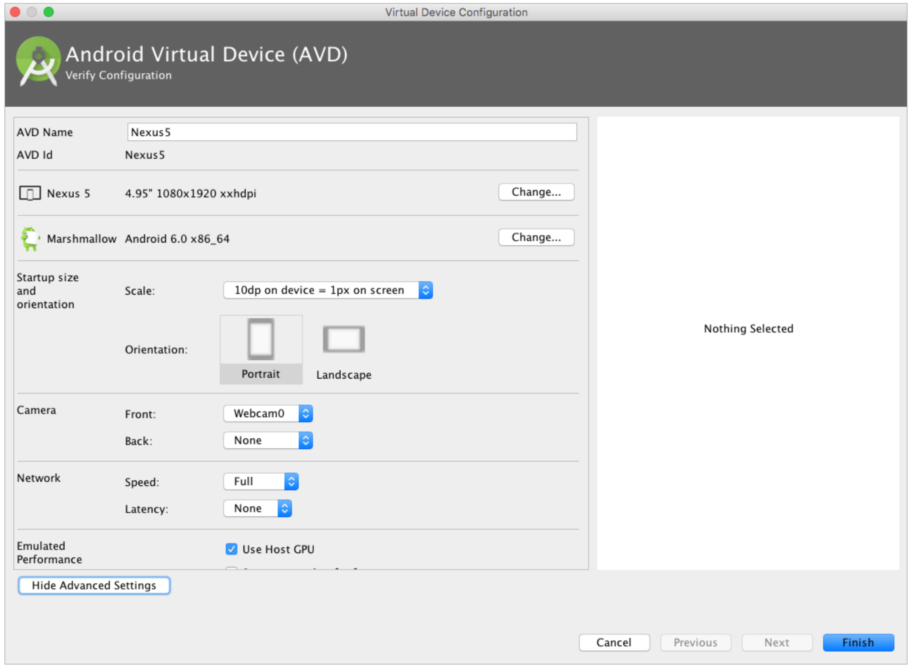
Nexus 5

Size: normal
Ratio: notlong
Density: xxhdpi

Xamarin University

Creating a new Google emulator

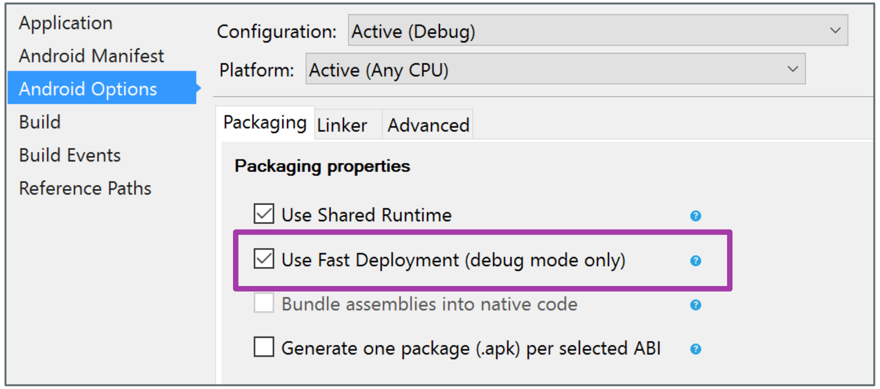
- Android Studio provides access to a much nicer configuration dialog



AVD Name: Nexus5
AVD Id: Nexus5
Nexus 5 4.95" 1080x1920 xxhdpi
Marshmallow Android 6.0 x86_64
Startup size and orientation
Scale: 10dp on device = 1px on screen
Orientation: Portrait
Camera
Front: Webcam0
Back: None
Network
Speed: Full
Latency: None
Emulated Performance
[✓] Use Host GPU

Running apps on Android emulators

- Some Android emulators do not support the "Fast Deployment" optimization which updates the app in-place on the device, if your application will not install, try turning this feature off in the project settings

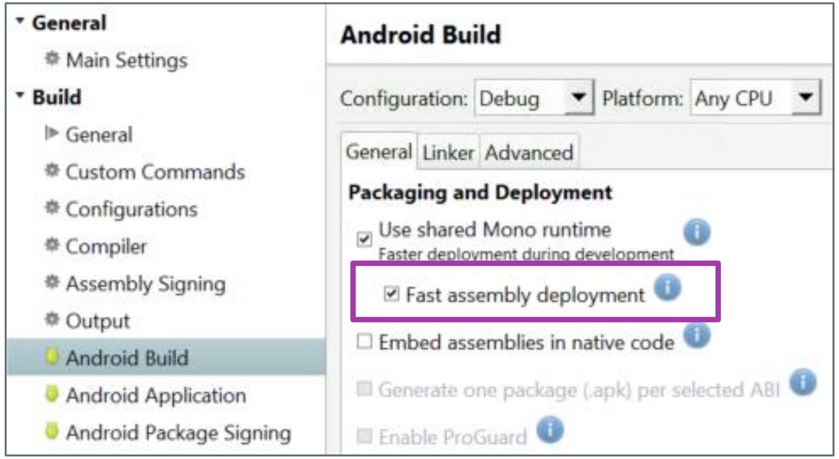

Application
Android Manifest
Android Options
Build
Build Events
Reference Paths

Configuration: Active (Debug)
Platform: Active (Any CPU)

Packaging | Linker | Advanced

Packaging properties

- [✓] Use Shared Runtime
- [✓] Use Fast Deployment (debug mode only)
- [ ] Bundle assemblies into native code
- [ ] Generate one package (.apk) per selected ABI


General
- Main Settings

Build
- General
- Custom Commands
- Configurations
- Compiler
- Assembly Signing
- Output
- Android Build
- Android Application
- Android Package Signing

Android Build

Configuration: Debug | Platform: Any CPU

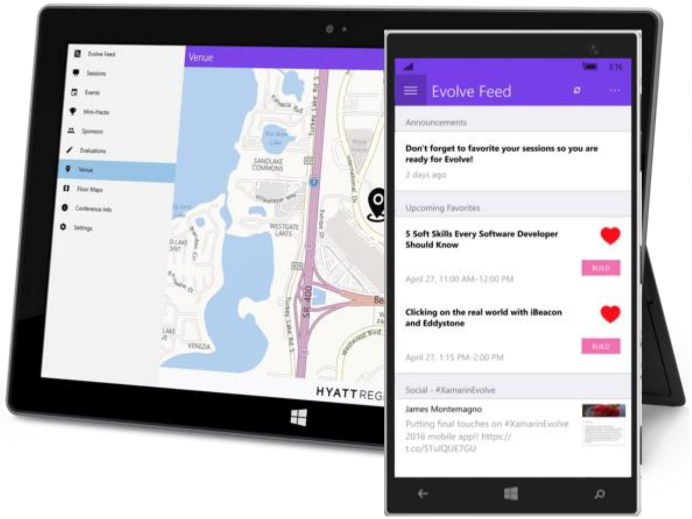
General | Linker | Advanced

Packaging and Deployment

- [✓] Use shared Mono runtime
  Faster deployment during development
- [✓] Fast assembly deployment
- [ ] Embed assemblies in native code
- [ ] Generate one package (.apk) per selected ABI
- [ ] Enable ProGuard

Running UWP apps (Windows)

❖ Visual Studio can deploy to local or remote Windows 10 devices as well as a optional Windows simulators



Venue
Evolve Feed
Announcements
Don't forget to favorite your sessions so you are ready for Evolve!
Upcoming Favorites
5 Soft Skills Every Software Developer Should Know
Clicking on the real world with iBeacon and Eddystone
Social - #XamarinEvolve
James Montemagno

Be aware that simulators require Hyper-V and can interfere with virtualization software like VMware and Virtual Box

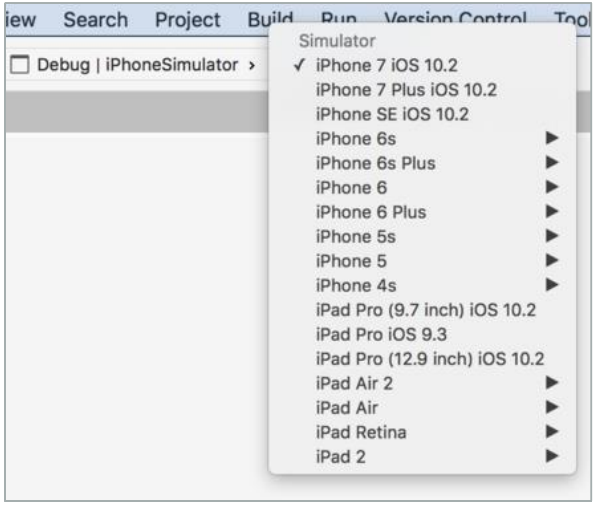
Using a real device

❖ Can use a physical device to run and debug your applications – requires some one-time platform-specific setup

- iOS: http://bit.ly/1R7YmH8
- Android: http://bit.ly/1PjDlFz
- Windows: http://bit.ly/2nsGf7i

Selecting a device or emulator

- Select the device (or emulator) to run your project using the drop-down on the Standard Toolbar



Setup Validation

Let's make sure your environment is ready to go!

Thank You!

Please complete the class survey in your profile:
university.xamarin.com/profile

Microsoft