Research on Xamarin

By Rong Zhuang
August 19, 2015

1. Introduction

1.1 Main Features

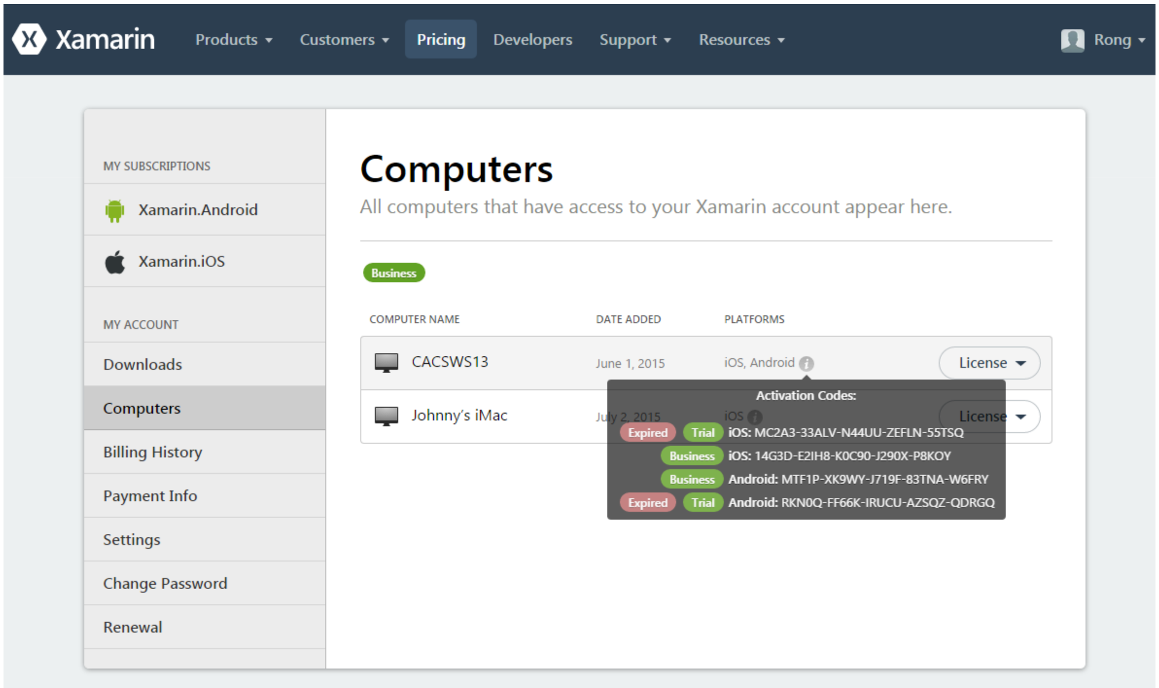
- Cross-Platform(C#)
  - iOS(Xamarin.iOS)
  - Android(Xamarin.Android)
  - Mac(Xamarin.Mac)
- Xamarin Insights
- Xamarin Test Cloud(Xamarin.UITest)

1.2 IDE

- Mac: Xamarin Studio, Xcode
- Windows: Xamarin Studio or Visual Studio+Xamarin's plug-in for Visual Studio, Mac, Xcode, Xamarin.iOS Build Host

1.3 Xamarin Account(per developer, per device platform)

- Trail(30days)-free
- Indie-$25/Month($300/Year)
- Business-$83/Month($999/Year)
- Enterprise-$158/Month($1899/Year)
- For Student-Free(Need to apply)



Computers

All computers that have access to your Xamarin account appear here.

Business

COMPUTER NAME	DATE ADDED	PLATFORMS	
CACSWS13	June 1, 2015	iOS, Android	License
Johnny's iMac	July 2, 2015	iOS	License

Activation Codes:

Expired Trial iOS: MC2A3-33ALV-N44UU-ZEFLN-55TSQ
Business iOS: 14G3D-E2IH8-K0C90-J290X-P8KOY
Business Android: MTF1P-XK9WY-J719F-83TNA-W6FRY
Expired Trial Android: RKN0Q-FF66K-IRUCU-AZSQZ-QDRGQ

2. iOS Development

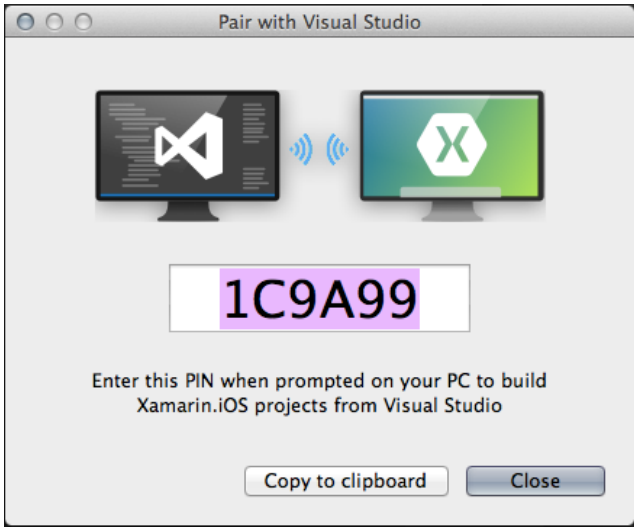
There are two options to setup the development environment for iOS:

- Xamarin Studio on Mac
- Visual Studio on PC(with another mac connected to the same network of PC), see below.

Note: Xamarin Studio on PC cannot be used for iOS development.

2.1 Installation(Windows)

- Xamarin Platform for Windows
- Xamarin.iOS Build Host on Mac

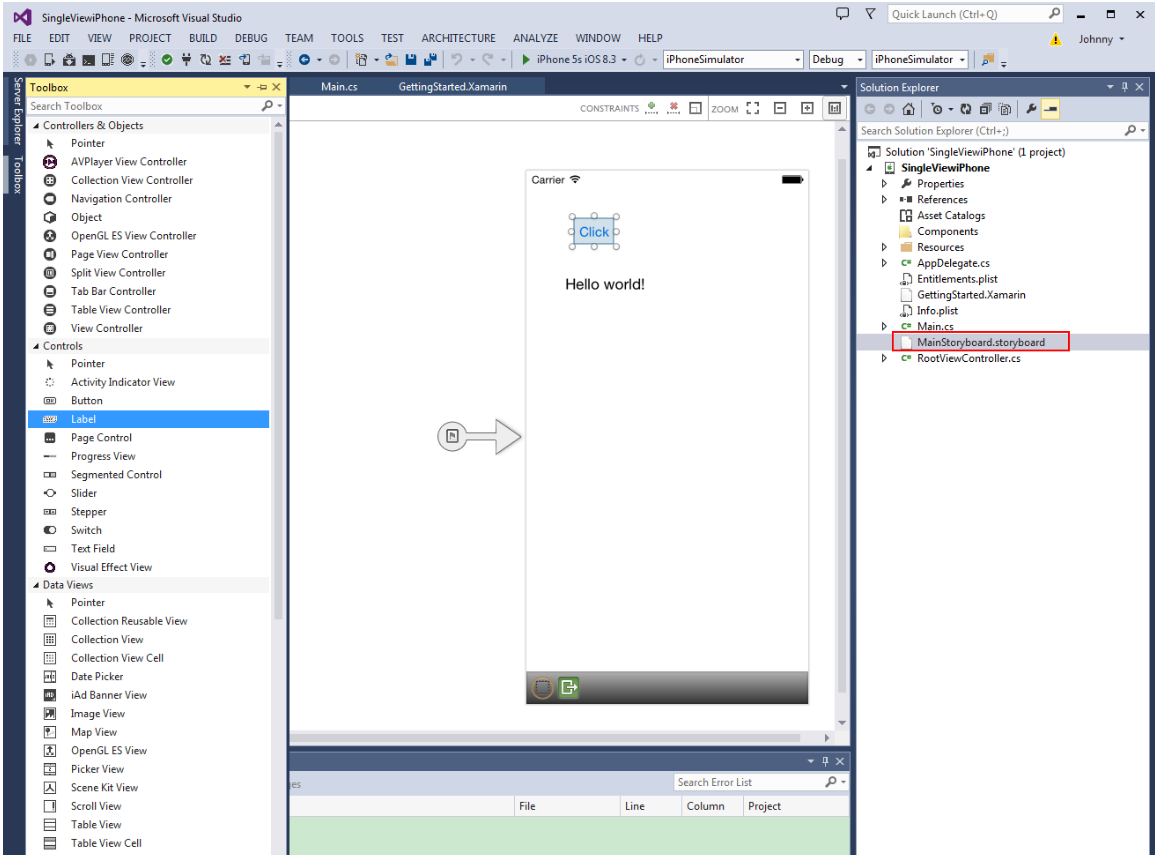


Pair with Visual Studio

1C9A99

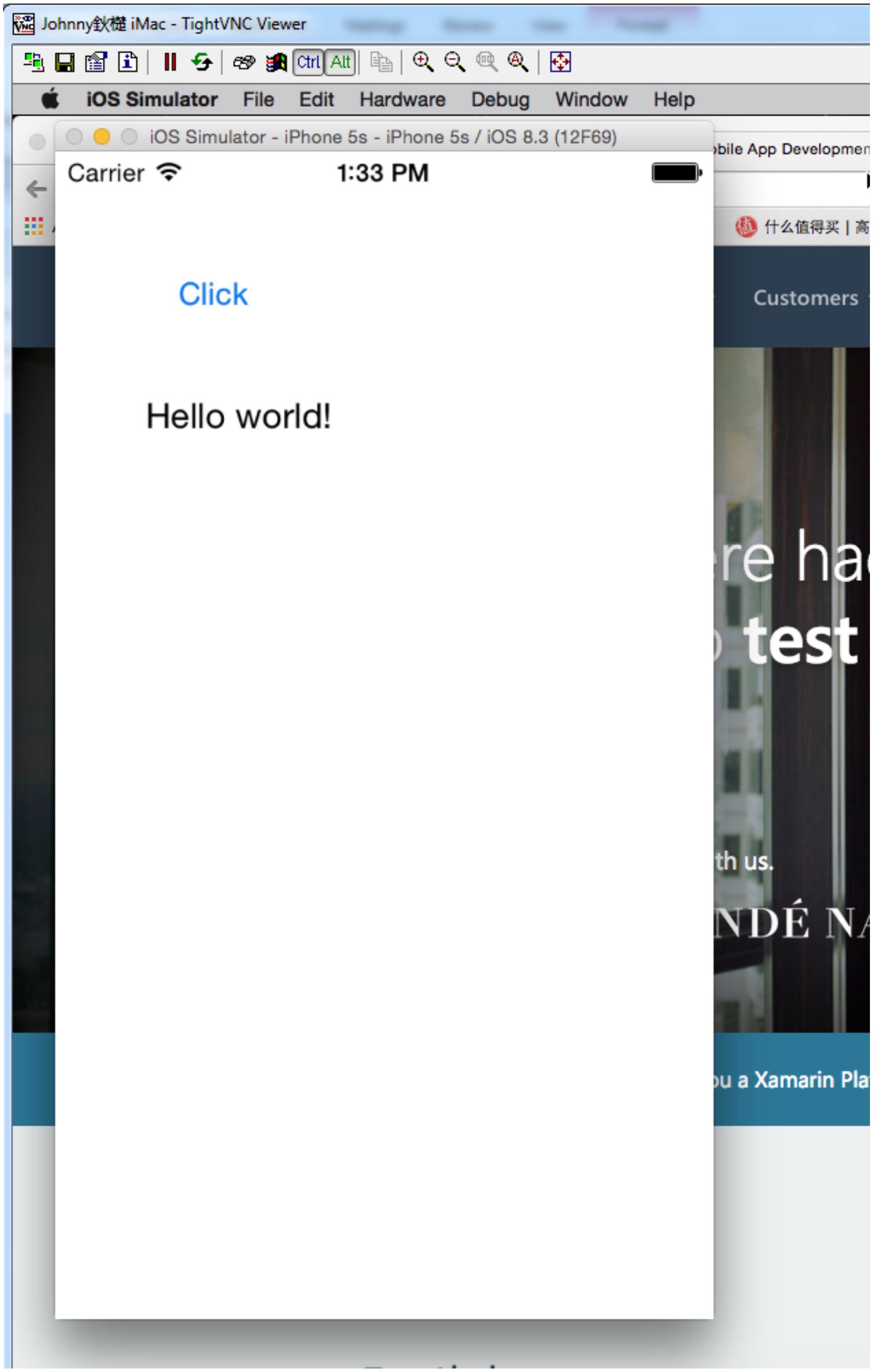
Enter this PIN when prompted on your PC to build Xamarin.iOS projects from Visual Studio

2.2 Developing in Visual Studio 2013

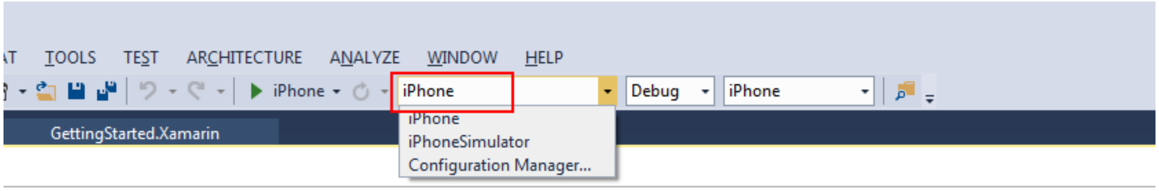


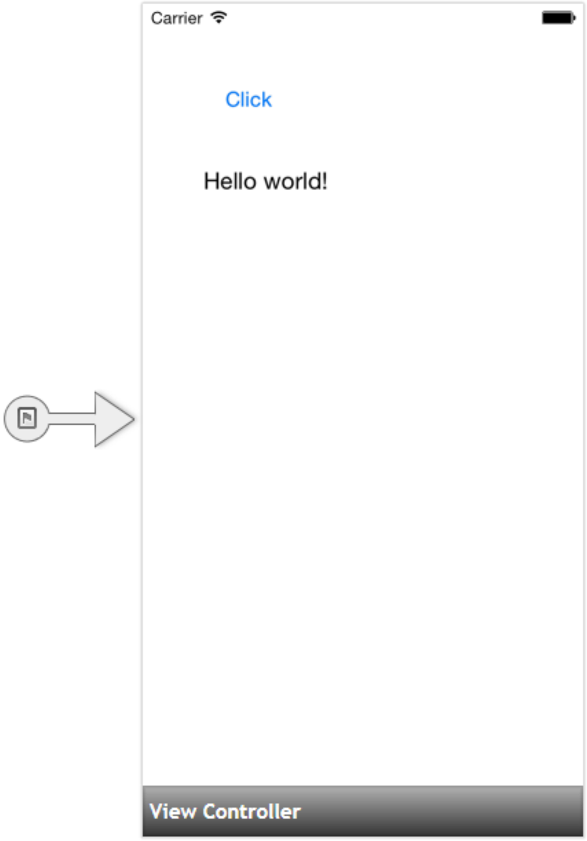
Note: Must be connecting to Build Host of Mac when editing the layout.

2.3 Test in simulator on Mac.



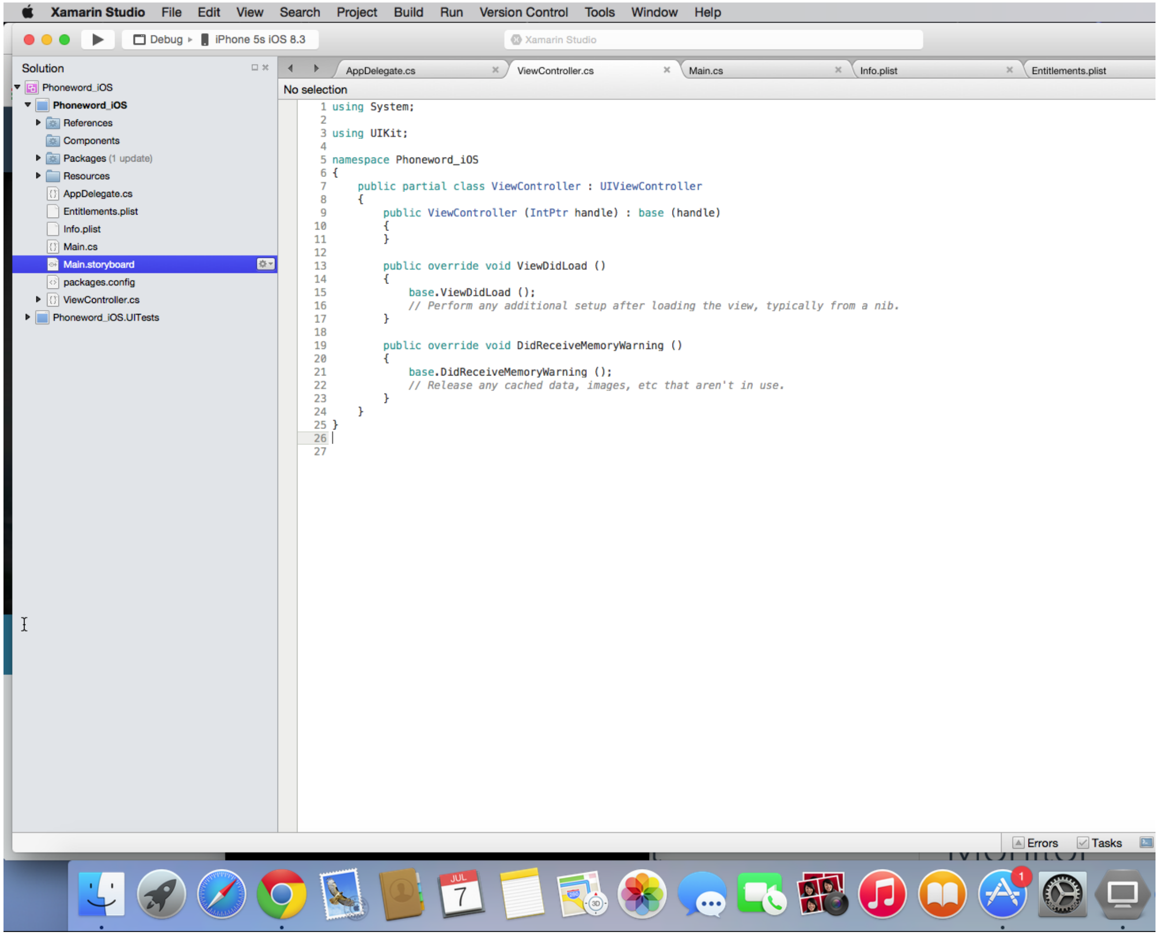
2.4 Test on real device.





Note: Plug in your iphone or ipad to MAC before running the test.

2.5 Developing in Xamarin Studio(Mac)



```
using System;

using UIKit;

namespace Phoneword_iOS
{
    public partial class ViewController : UIViewController
    {
        public ViewController (IntPtr handle) : base (handle)
        {
        }

        public override void ViewDidLoad ()
        {
            base.ViewDidLoad ();
            // Perform any additional setup after loading the view, typically from a nib.
        }

        public override void DidReceiveMemoryWarning ()
        {
            base.DidReceiveMemoryWarning ();
            // Release any cached data, images, etc that aren't in use.
        }
    }
}
```

2.6 Deployment

Publishing to the App Store

http://developer.xamarin.com/guides/ios/deployment,_testing,_and_metrics/app_distribution_overview/publishing_to_the_app_store/

3. Android Development

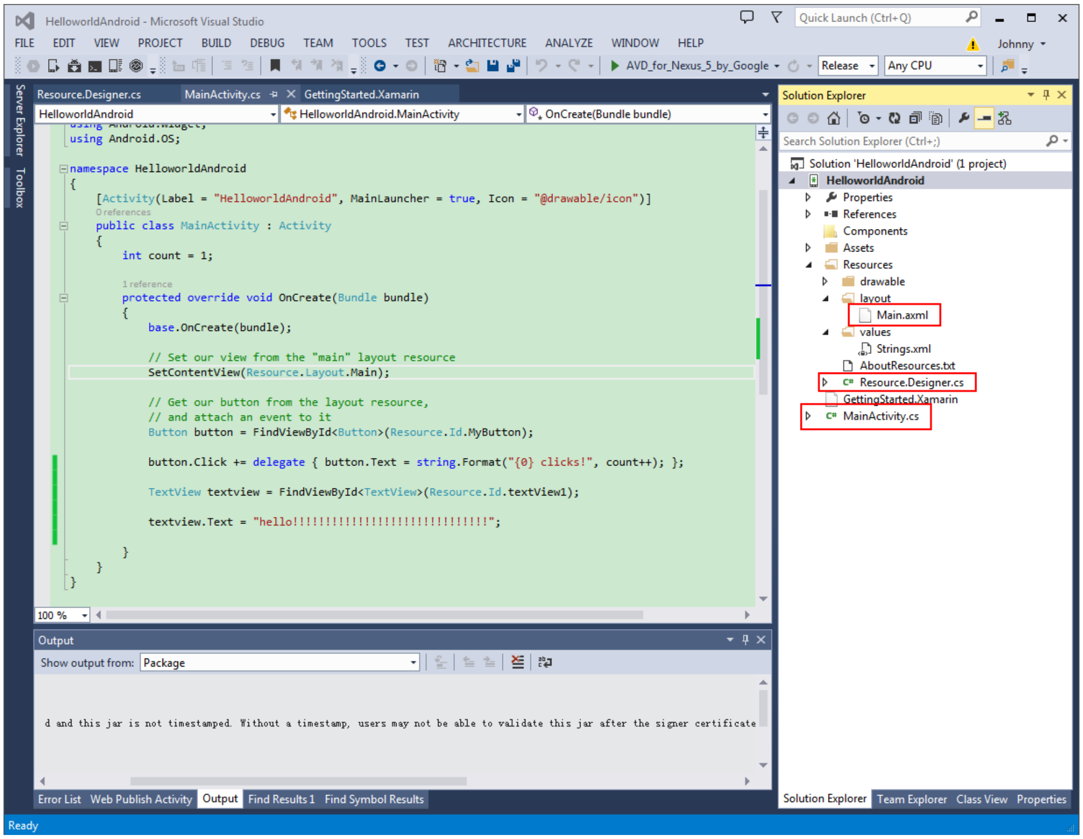
There are three options to setup the development environment for Android:

- Xamarin Studio on Mac
- Xamarin Studio on PC
- Visual Studio on PC

3.1 Installation(Windows)

- Xamarin Platform for Windows
- Android SDK and Emulator

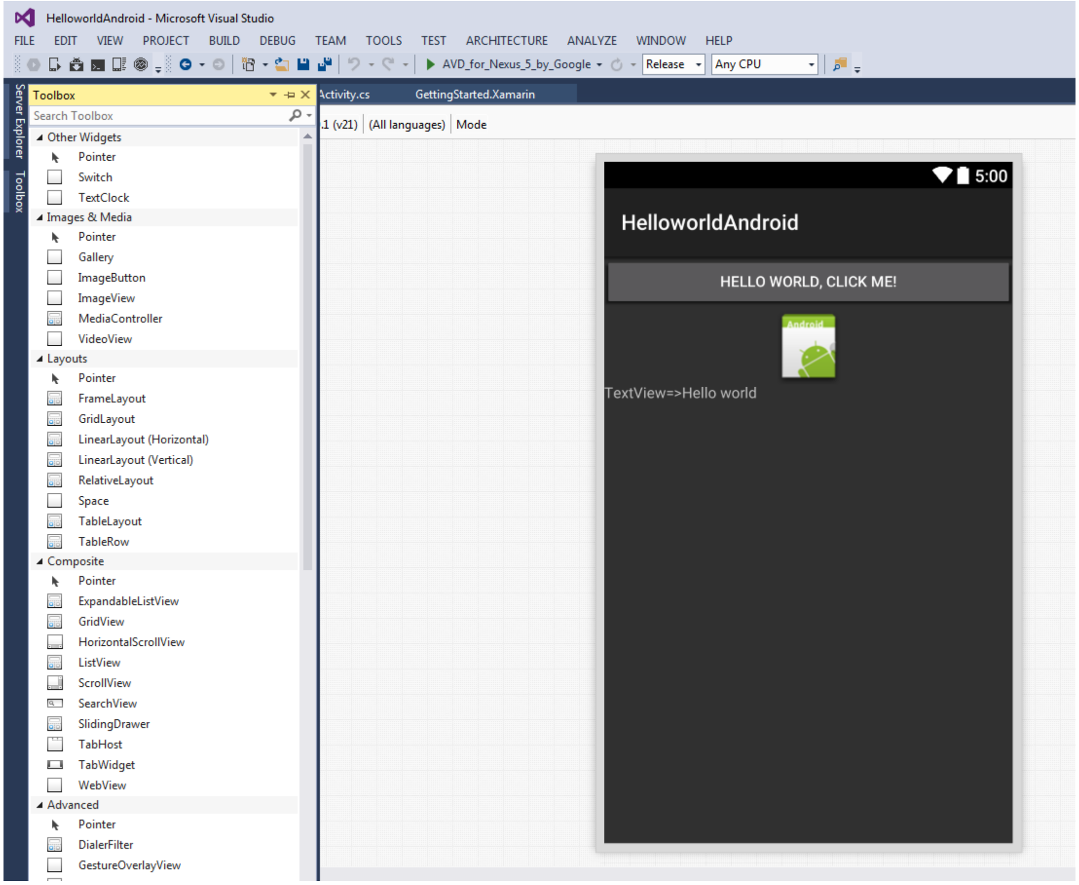
3.2 Developing in Visual Studio 2013.



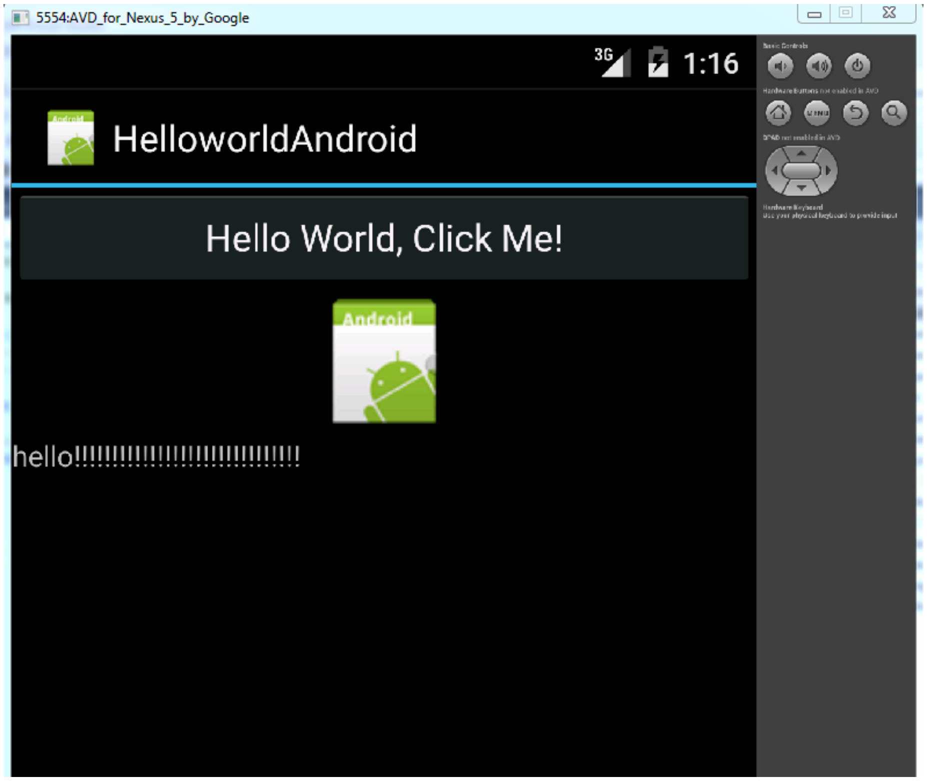
```
using Android.Widget;
using Android.OS;

namespace HelloworldAndroid
{
    [Activity(Label = "HelloworldAndroid", MainLauncher = true, Icon = "@drawable/icon")]
    public class MainActivity : Activity
    {
        int count = 1;

        protected override void OnCreate(Bundle bundle)
        {
            base.OnCreate(bundle);

            // Set our view from the "main" layout resource
            SetContentView(Resource.Layout.Main);

            // Get our button from the layout resource,
            // and attach an event to it
            Button button = FindViewById<Button>(Resource.Id.MyButton);

            button.Click += delegate { button.Text = string.Format("{0} clicks!", count++); };

            TextView textview = FindViewById<TextView>(Resource.Id.textView1);

            textview.Text = "hello!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!";

        }
    }
}
```

Show output from: Package

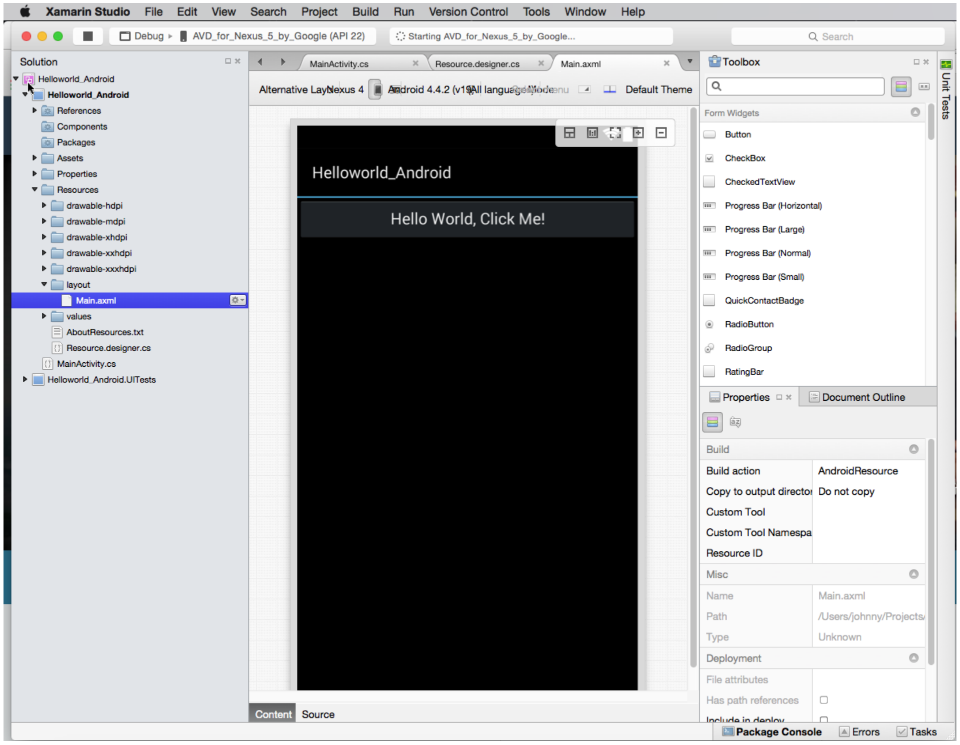
```
d and this jar is not timestamped. Without a timestamp, users may not be able to validate this jar after the signer certificate
```



3.3 Test in emulator, AVD(Android Virtual Device).


5554:AVD_for_Nexus_5_by_Google
3G
1:16
HelloworldAndroid
Hello World, Click Me!
Android
hello!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Basic Controls
Hardware Buttons not enabled in AVD
DPAD not enabled in AVD
Hardware Keyboard
Use your physical keyboard to provide input

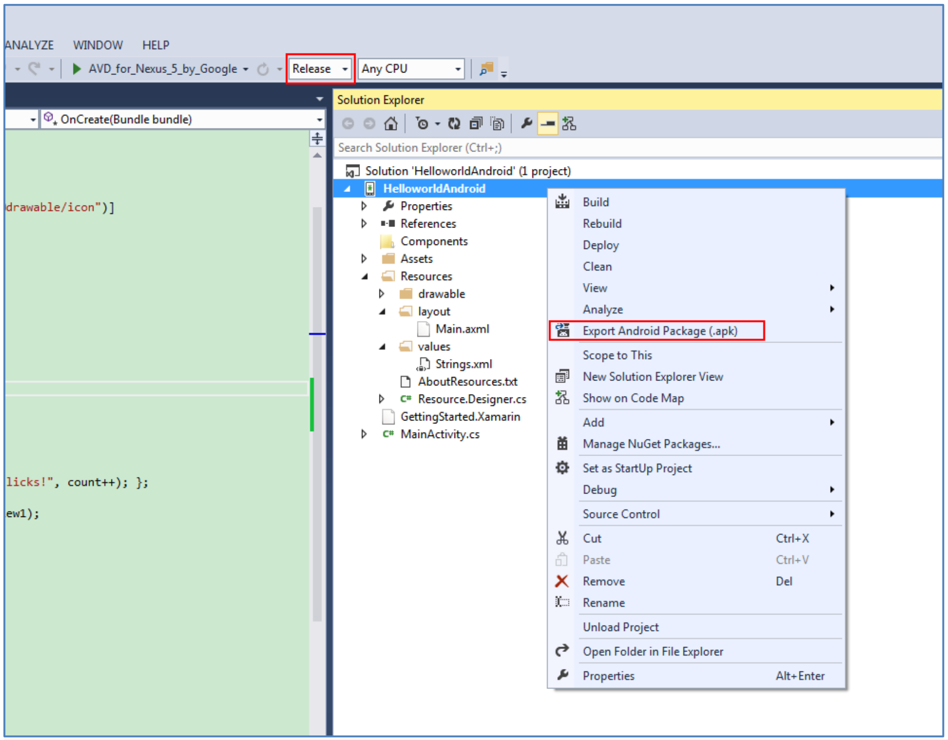
3.4 Developing in Xamarin Studio(Mac)

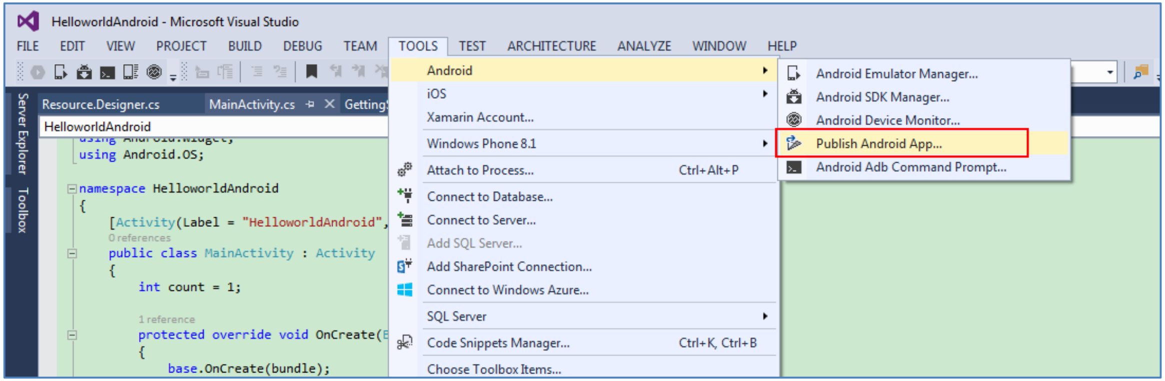


Starting AVD_for_Nexus_5_by_Google...

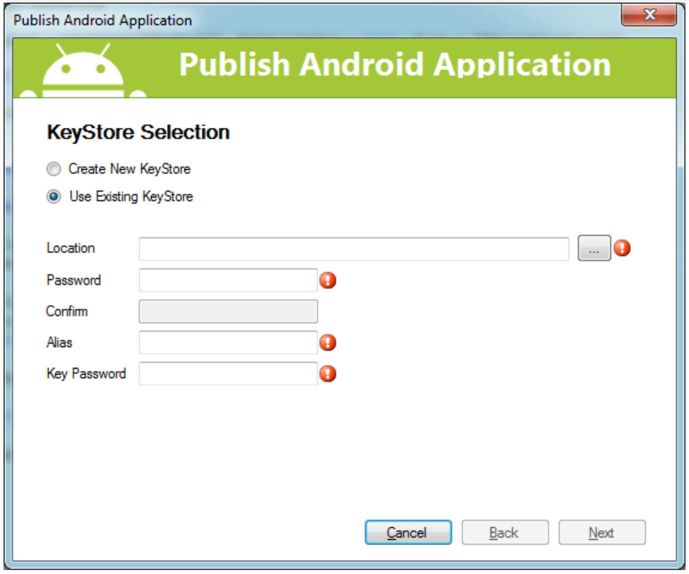
Build	
Build action	AndroidResource
Copy to output director	Do not copy
Custom Tool	
Custom Tool Namespa	
Resource ID	
Misc	
Name	Main.axml
Path	/Users/johnny/Projects/
Type	Unknown
Deployment	
File attributes	
Has path references	□
Include in deploy	□

3.5 Publish and Deployment in Visual Studio.





```
using Android.Widget;
using Android.OS;

namespace HelloworldAndroid
{
    [Activity(Label = "HelloworldAndroid",
    0 references
    public class MainActivity : Activity
    {
        int count = 1;

        1 reference
        protected override void OnCreate(B
        {
            base.OnCreate(bundle);
```

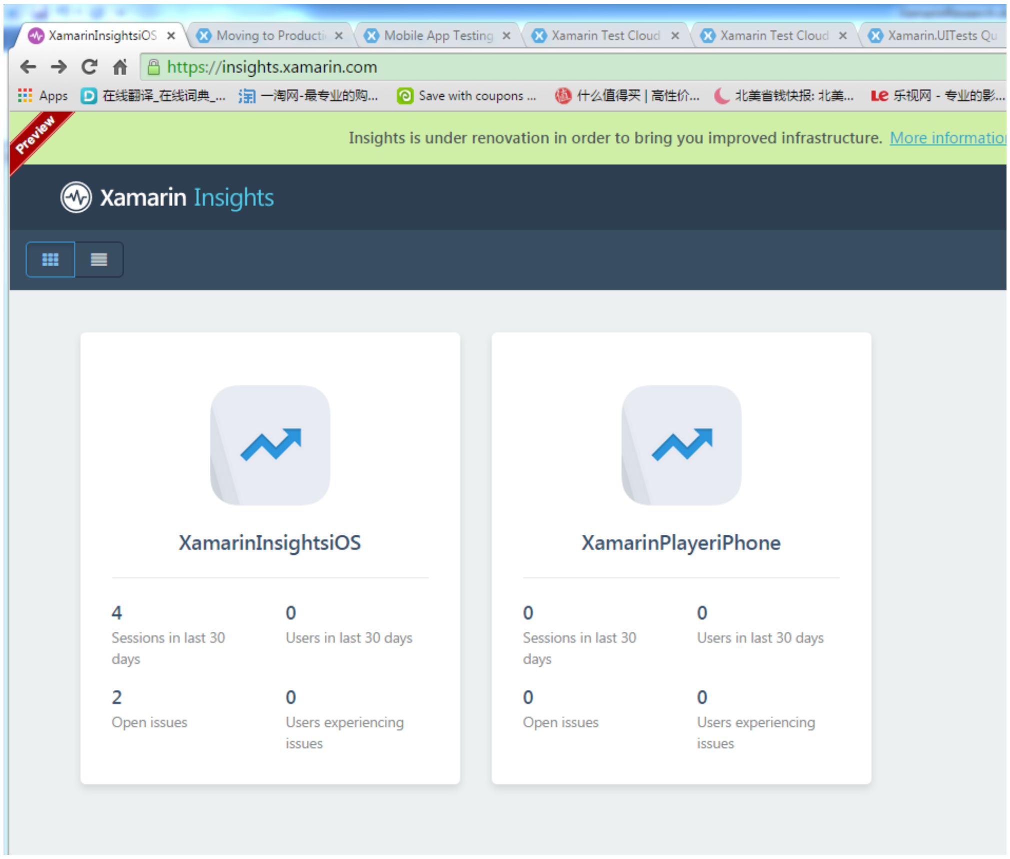


KeyStore Selection

○ Create New KeyStore
● Use Existing KeyStore

Location	
Password	
Confirm	
Alias	
Key Password	

4. Xamarin Insights



https://insights.xamarin.com

Insights is under renovation in order to bring you improved infrastructure. More informatio

XamarinInsightsiOS

4 Sessions in last 30 days

0 Users in last 30 days

2 Open issues

0 Users experiencing issues

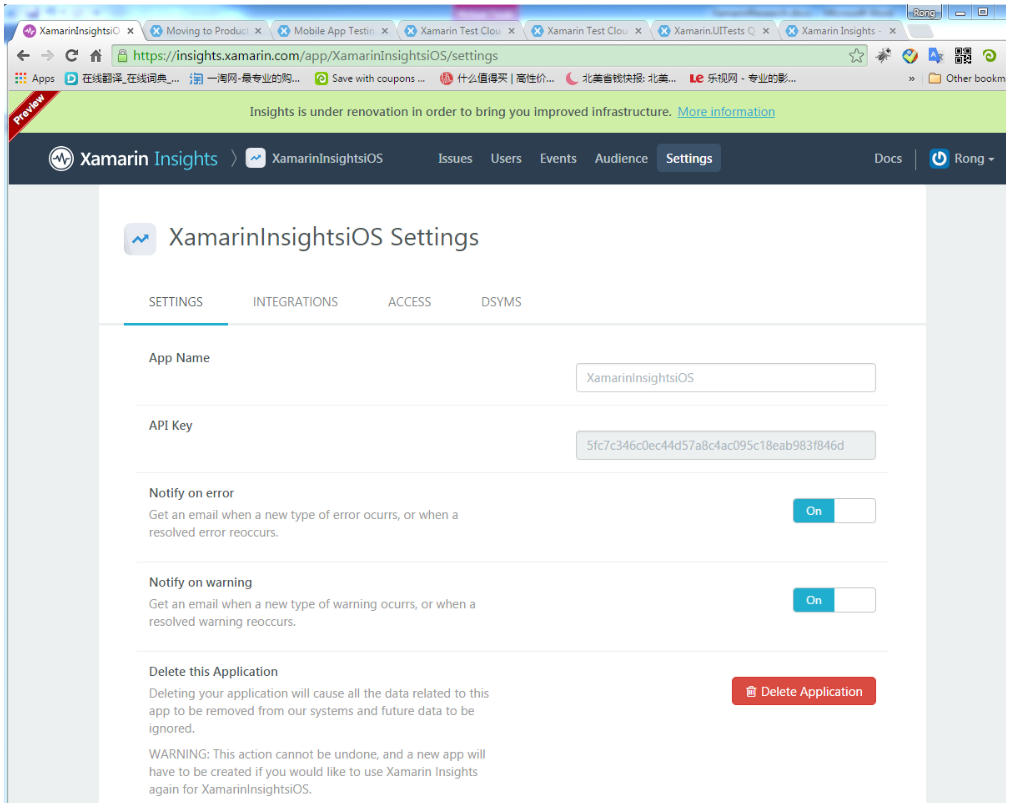
XamarinPlayeriPhone

0 Sessions in last 30 days

0 Users in last 30 days

0 Open issues

0 Users experiencing issues



https://insights.xamarin.com/app/XamarinInsightsiOS/settings

Insights is under renovation in order to bring you improved infrastructure. More information

XamarinInsightsiOS Settings

SETTINGS INTEGRATIONS ACCESS DSYMS

App Name: XamarinInsightsiOS

API Key: 5fc7c346c0ec44d57a8c4ac095c18eab983f846d

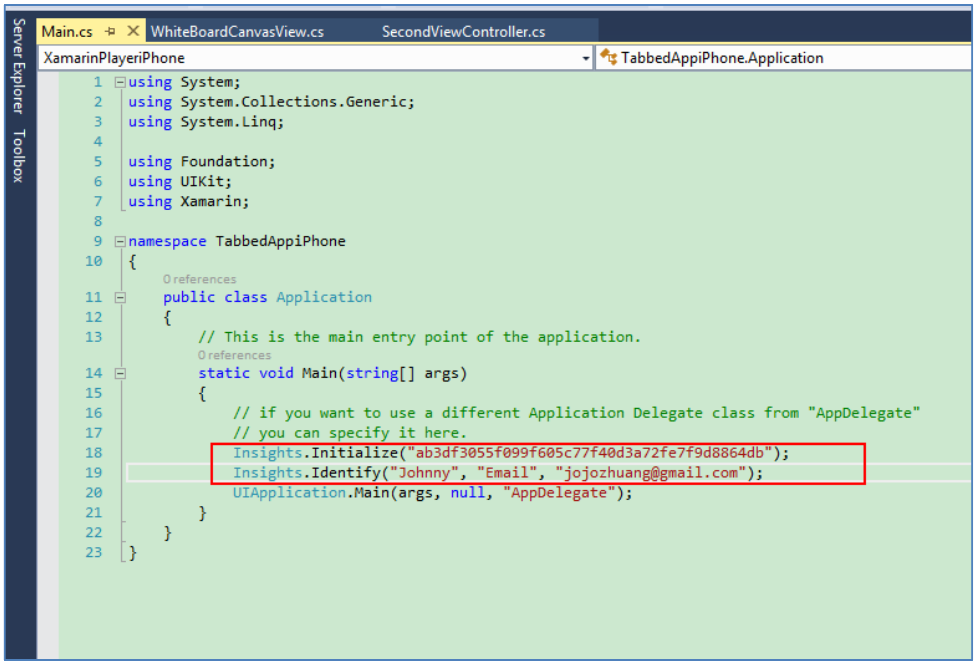
Notify on error: On
Get an email when a new type of error ocurrs, or when a resolved error reoccurs.

Notify on warning: On
Get an email when a new type of warning ocurrs, or when a resolved warning reoccurs.

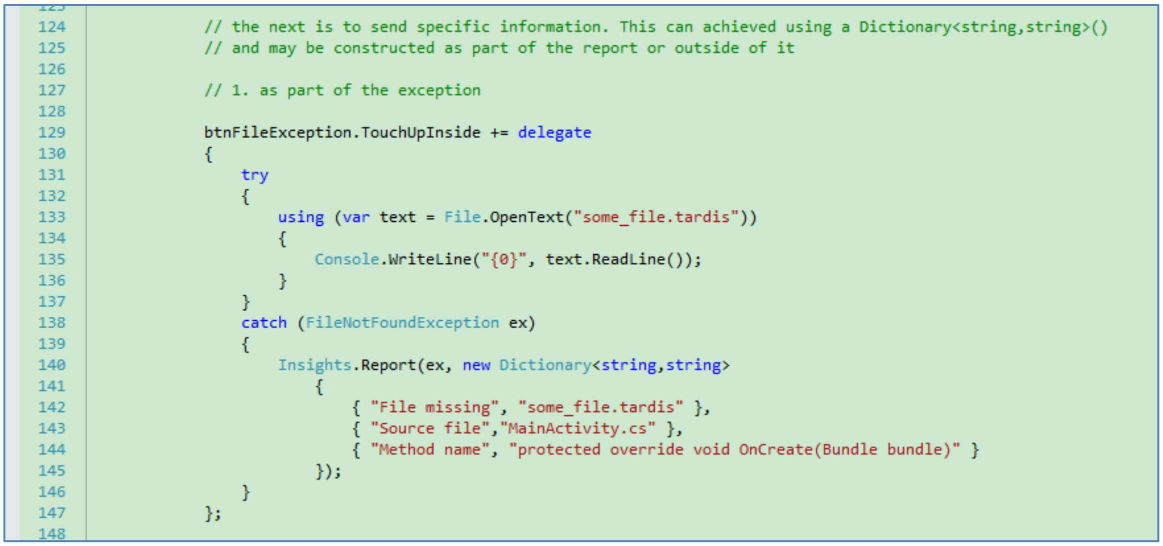
Delete this Application
Deleting your application will cause all the data related to this app to be removed from our systems and future data to be ignored.

WARNING: This action cannot be undone, and a new app will have to be created if you would like to use Xamarin Insights again for XamarinInsightsiOS.

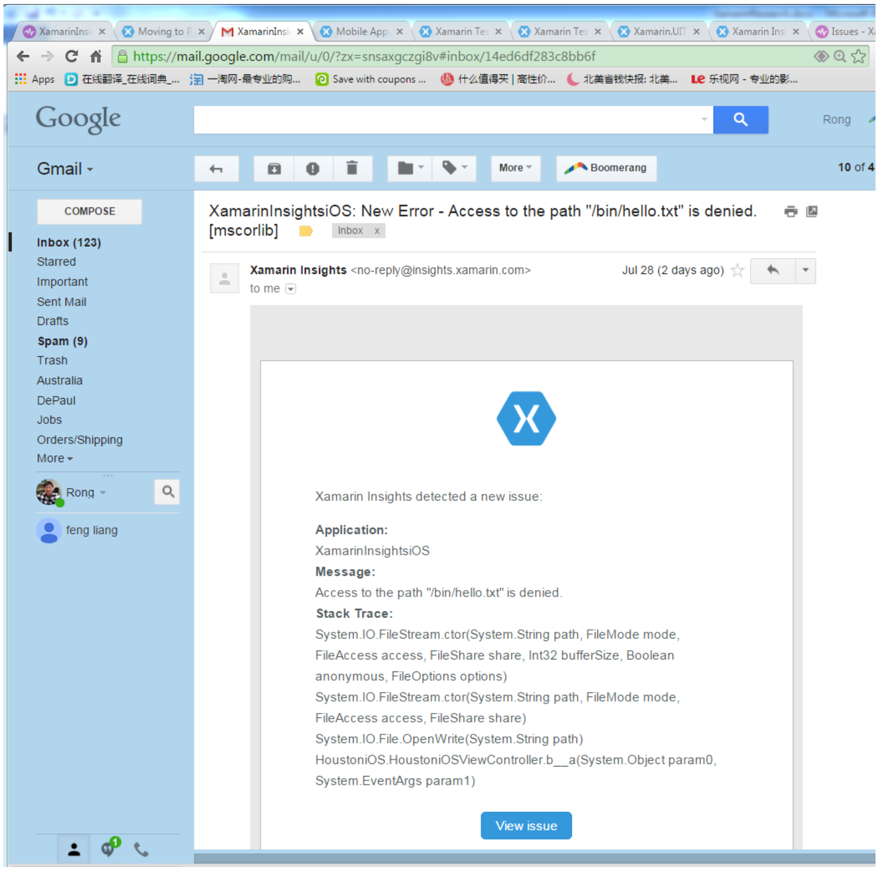
Delete Application



```
using System;
using System.Collections.Generic;
using System.Linq;

using Foundation;
using UIKit;
using Xamarin;

namespace TabbedAppiPhone
{
    public class Application
    {
        // This is the main entry point of the application.
        static void Main(string[] args)
        {
            // if you want to use a different Application Delegate class from "AppDelegate"
            // you can specify it here.
            Insights.Initialize("ab3df3055f099f605c77f40d3a72fe7f9d8864db");
            Insights.Identify("Johnny", "Email", "jojozhuang@gmail.com");
            UIApplication.Main(args, null, "AppDelegate");
        }
    }
}
```



```
            // the next is to send specific information. This can achieved using a Dictionary<string,string>()
            // and may be constructed as part of the report or outside of it

            // 1. as part of the exception

            btnFileException.TouchUpInside += delegate
            {
                try
                {
                    using (var text = File.OpenText("some_file.tardis"))
                    {
                        Console.WriteLine("{0}", text.ReadLine());
                    }
                }
                catch (FileNotFoundException ex)
                {
                    Insights.Report(ex, new Dictionary<string,string>
                        {
                            { "File missing", "some_file.tardis" },
                            { "Source file","MainActivity.cs" },
                            { "Method name", "protected override void OnCreate(Bundle bundle)" }
                        });
                }
            };

```



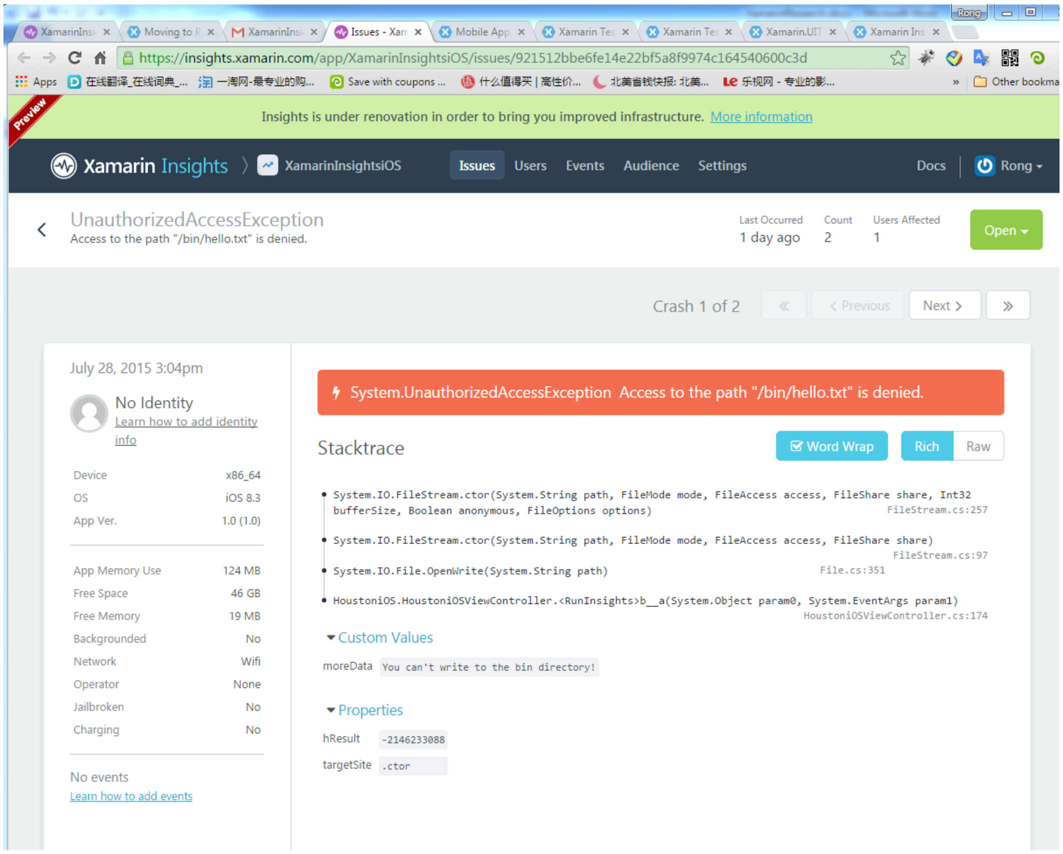
XamarinInsightsiOS: New Error - Access to the path "/bin/hello.txt" is denied. [mscorlib]

Xamarin Insights <no-reply@insights.xamarin.com> Jul 28 (2 days ago)
to me

Xamarin Insights detected a new issue:

Application:
XamarinInsightsiOS
Message:
Access to the path "/bin/hello.txt" is denied.
Stack Trace:
System.IO.FileStream.ctor(System.String path, FileMode mode, FileAccess access, FileShare share, Int32 bufferSize, Boolean anonymous, FileOptions options)
System.IO.FileStream.ctor(System.String path, FileMode mode, FileAccess access, FileShare share)
System.IO.File.OpenWrite(System.String path)
HoustoniOS.HoustoniOSViewController.b__a(System.Object param0, System.EventArgs param1)

View issue



Insights is under renovation in order to bring you improved infrastructure. More information

UnauthorizedAccessException
Access to the path "/bin/hello.txt" is denied.

Last Occurred	Count	Users Affected
1 day ago	2	1

July 28, 2015 3:04pm

No Identity
Learn how to add identity info

Device	x86_64
OS	iOS 8.3
App Ver.	1.0 (1.0)
App Memory Use	124 MB
Free Space	46 GB
Free Memory	19 MB
Backgrounded	No
Network	Wifi
Operator	None
Jailbroken	No
Charging	No

No events
Learn how to add events

System.UnauthorizedAccessException Access to the path "/bin/hello.txt" is denied.

Stacktrace

```
• System.IO.FileStream.ctor(System.String path, FileMode mode, FileAccess access, FileShare share, Int32
  bufferSize, Boolean anonymous, FileOptions options)                                  FileStream.cs:257

• System.IO.FileStream.ctor(System.String path, FileMode mode, FileAccess access, FileShare share)
                                                                                        FileStream.cs:97
• System.IO.File.OpenWrite(System.String path)                              File.cs:351

• HoustoniOS.HoustoniOSViewController.<RunInsights>b__a(System.Object param0, System.EventArgs param1)
                                                                          HoustoniOSViewController.cs:174
```

Custom Values

moreData `You can't write to the bin directory!`

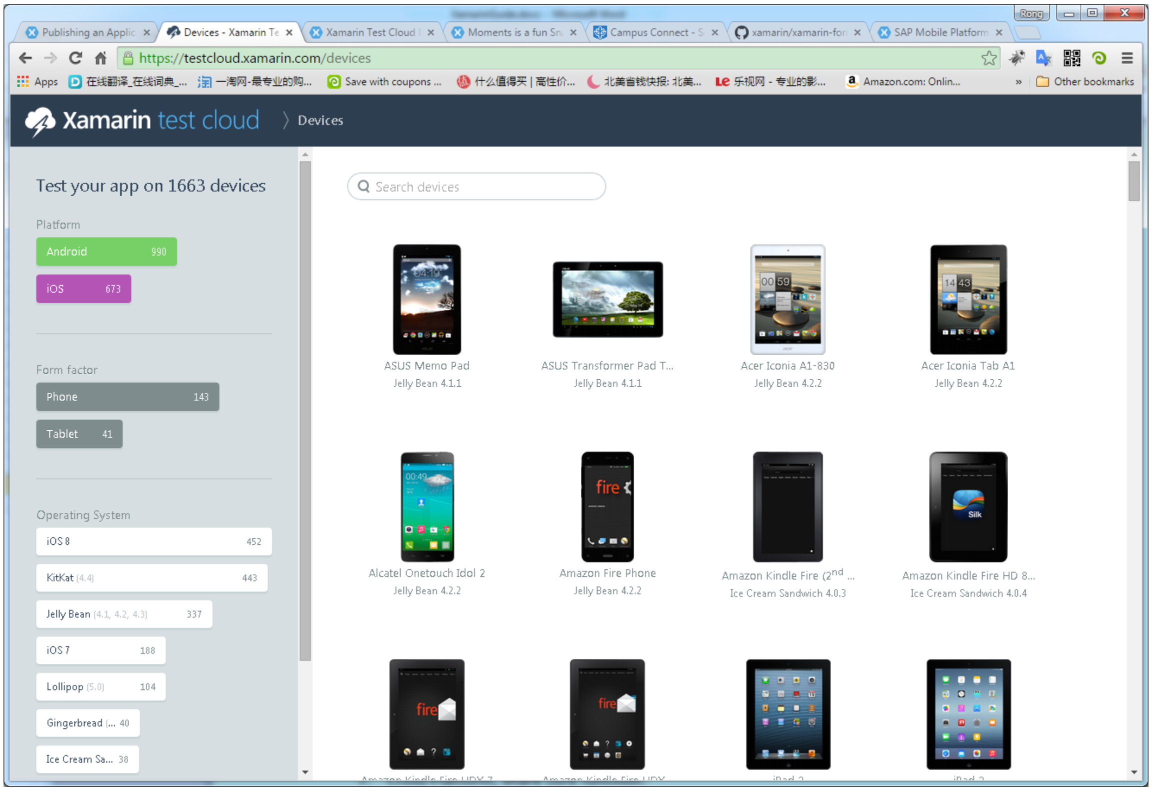
Properties

hResult `-2146233088`

targetSite `.ctor`

5. Xamarin Test Cloud

- Xamarin.UITest(C#)
- Calabash(Ruby)



Test your app on 1663 devices

Platform

Android	990
iOS	673

Form factor

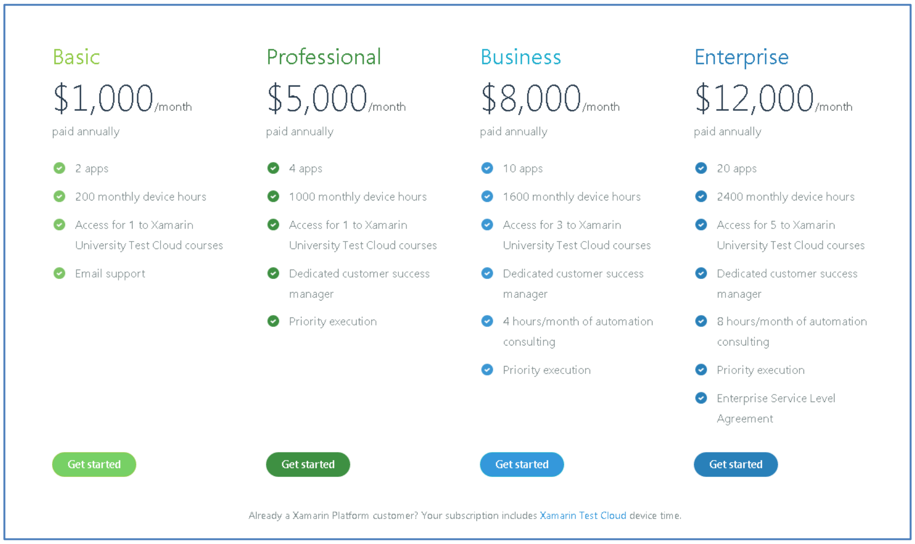
Phone	143
Tablet	41

Operating System

iOS 8	452
KitKat (4.4)	443
Jelly Bean (4.1, 4.2, 4.3)	337
iOS 7	188
Lollipop (5.0)	104
Gingerbread (...	40
Ice Cream Sa...	38

ASUS Memo Pad	Jelly Bean 4.1.1
ASUS Transformer Pad T...	Jelly Bean 4.1.1
Acer Iconia A1-830	Jelly Bean 4.2.2
Acer Iconia Tab A1	Jelly Bean 4.2.2
Alcatel Onetouch Idol 2	Jelly Bean 4.2.2
Amazon Fire Phone	Jelly Bean 4.2.2
Amazon Kindle Fire (2nd ...	Ice Cream Sandwich 4.0.3
Amazon Kindle Fire HD 8...	Ice Cream Sandwich 4.0.4

Costs:



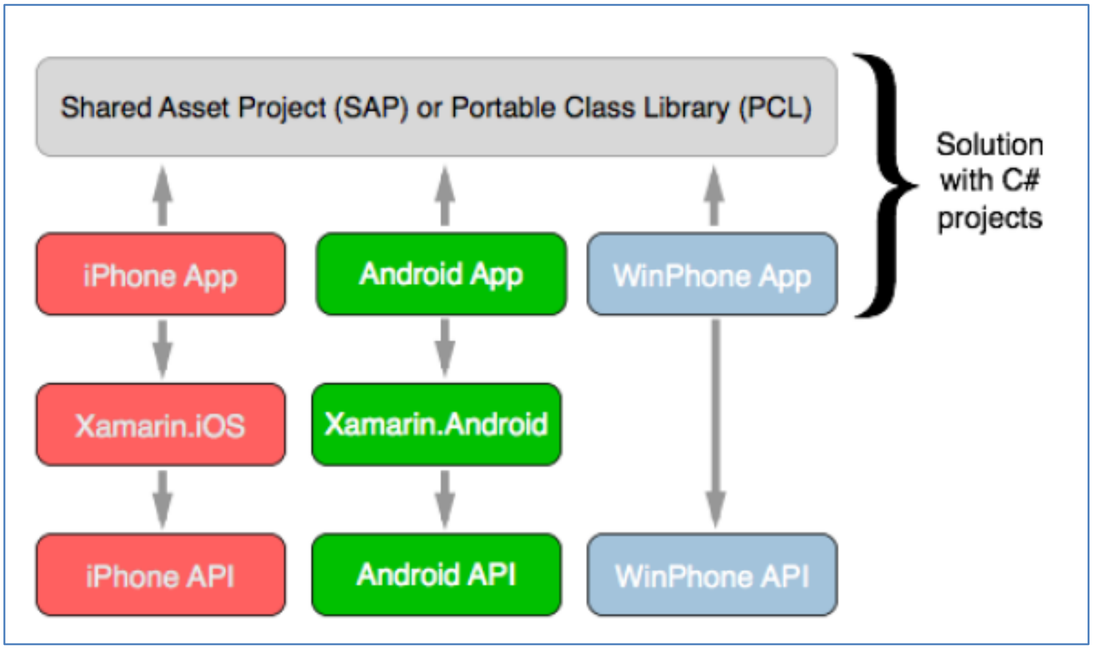
Basic	Professional	Business	Enterprise
$1,000/month	$5,000/month	$8,000/month	$12,000/month
paid annually	paid annually	paid annually	paid annually
2 apps	4 apps	10 apps	20 apps
200 monthly device hours	1000 monthly device hours	1600 monthly device hours	2400 monthly device hours
Access for 1 to Xamarin University Test Cloud courses	Access for 1 to Xamarin University Test Cloud courses	Access for 3 to Xamarin University Test Cloud courses	Access for 5 to Xamarin University Test Cloud courses
Email support	Dedicated customer success manager	Dedicated customer success manager	Dedicated customer success manager
	Priority execution	4 hours/month of automation consulting	8 hours/month of automation consulting
		Priority execution	Priority execution
			Enterprise Service Level Agreement
Get started	Get started	Get started	Get started

Already a Xamarin Platform customer? Your subscription includes Xamarin Test Cloud device time.

6. Cross-Platform

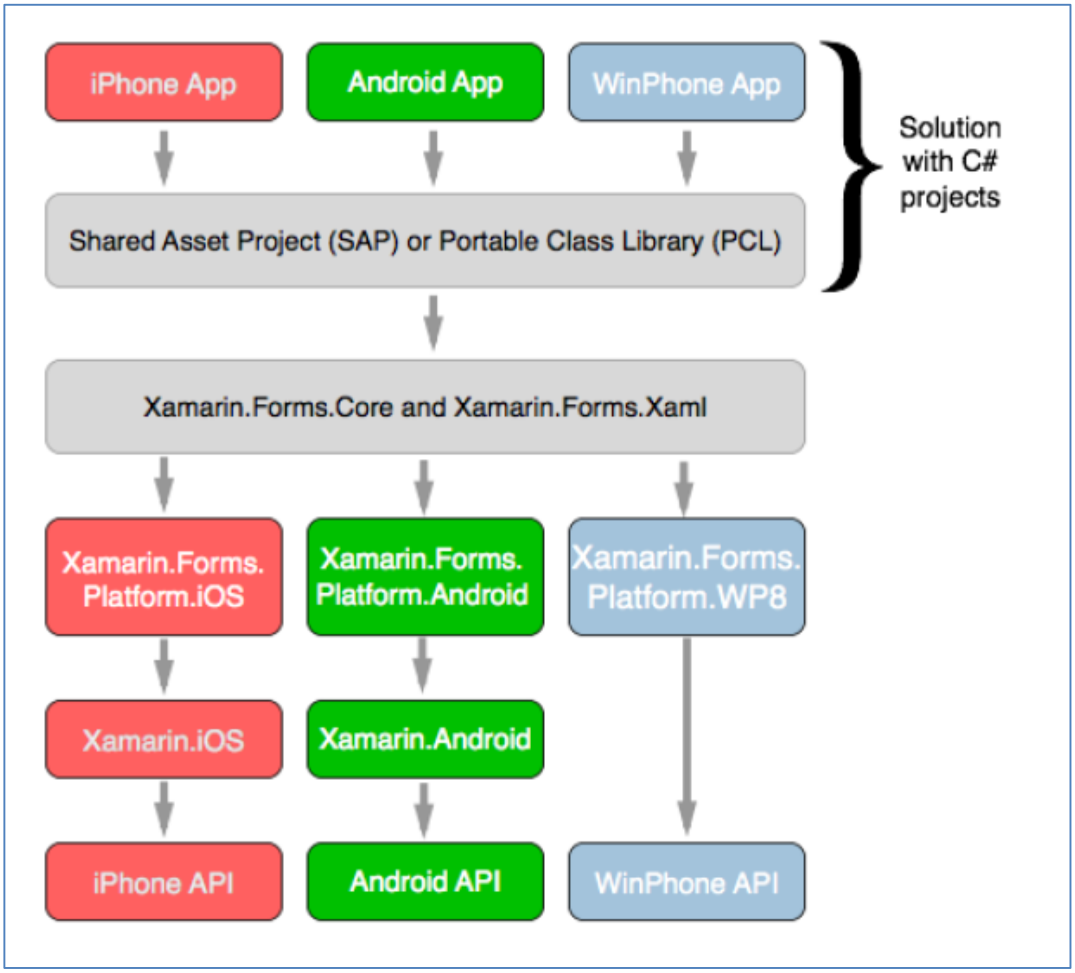
6.1 Separated components

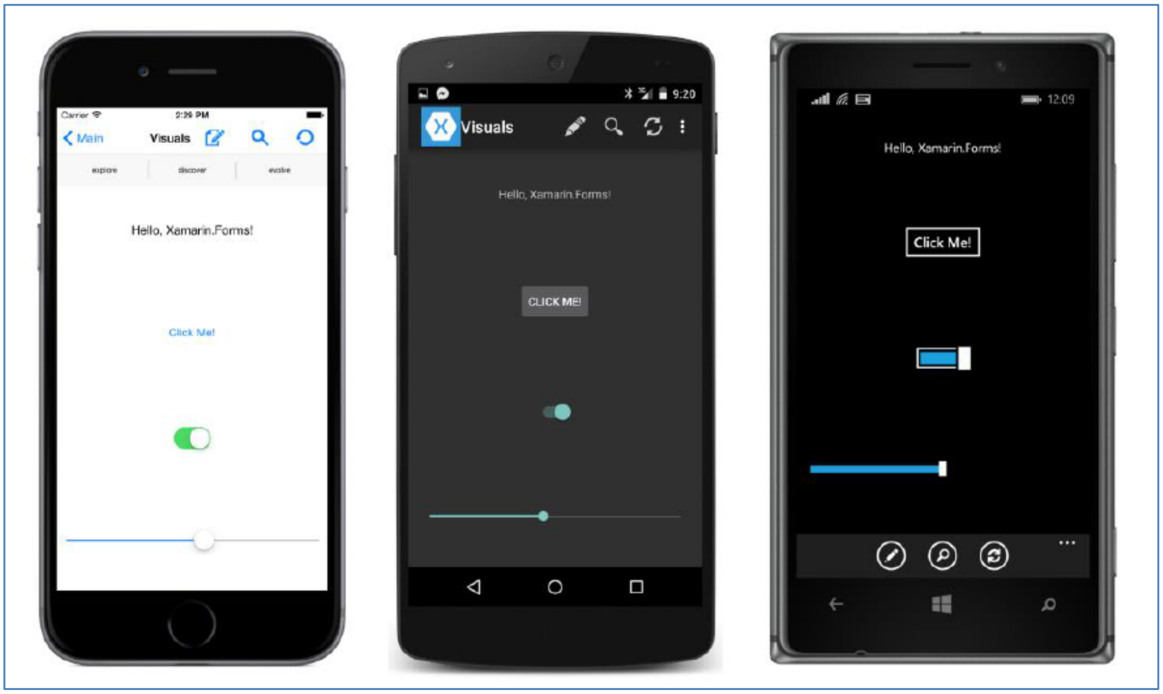
- Xamarin.Android
- Xamarin.iOS
- Xamarin.WP8



6.2 Xamarin.Form(Unified UI development process)

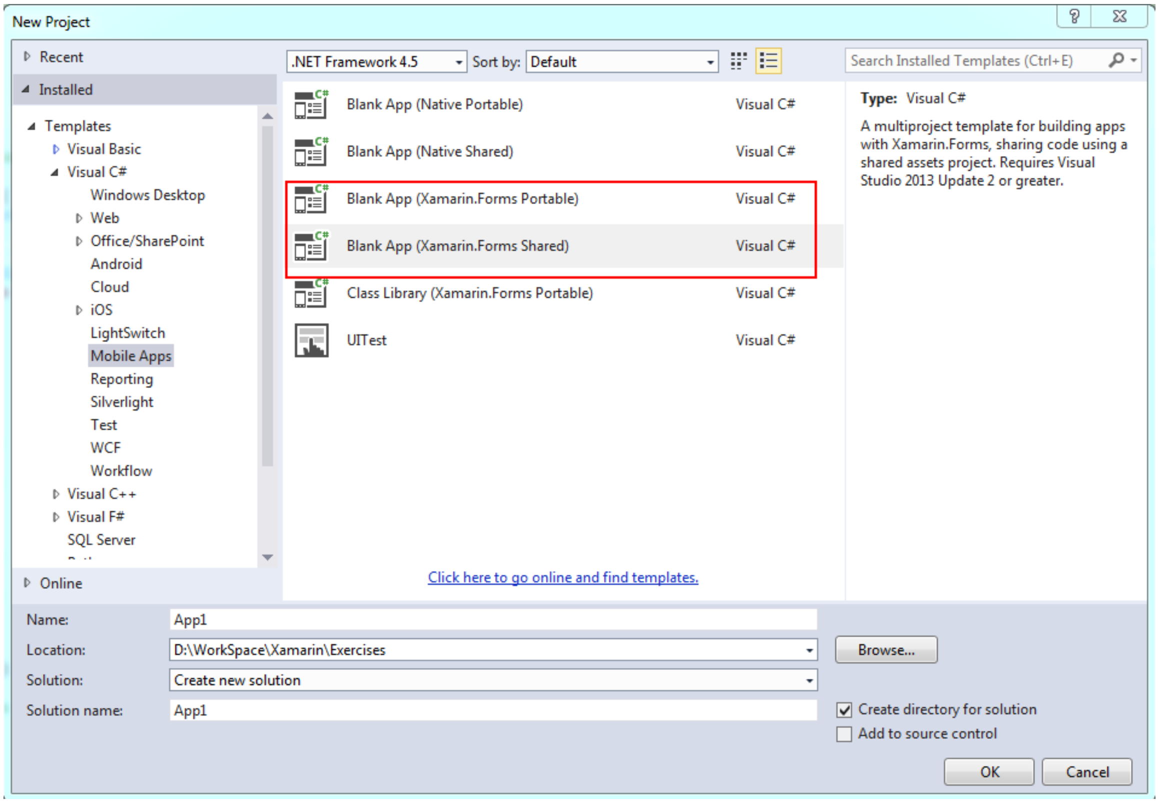
Xamarin.Forms integrates with Xamarin.iOS and Xamarin.Android.


iPhone App
Android App
WinPhone App
Solution with C# projects
Shared Asset Project (SAP) or Portable Class Library (PCL)
Xamarin.Forms.Core and Xamarin.Forms.Xaml
Xamarin.Forms. Platform.iOS
Xamarin.Forms. Platform.Android
Xamarin.Forms. Platform.WP8
Xamarin.iOS
Xamarin.Android
iPhone API
Android API
WinPhone API



6.3 Sharing Codes

- Portable Class Library (PCL): dlls
- Shared Asset Project (SAP): code and assets files



New Project

- Recent
- Installed
  - Templates
    - Visual Basic
    - Visual C#
      - Windows Desktop
      - Web
      - Office/SharePoint
      - Android
      - Cloud
      - iOS
      - LightSwitch
      - Mobile Apps
      - Reporting
      - Silverlight
      - Test
      - WCF
      - Workflow
    - Visual C++
    - Visual F#
    - SQL Server
- Online

.NET Framework 4.5 Sort by: Default

Search Installed Templates (Ctrl+E)

Blank App (Native Portable)	Visual C#
Blank App (Native Shared)	Visual C#
Blank App (Xamarin.Forms Portable)	Visual C#
Blank App (Xamarin.Forms Shared)	Visual C#
Class Library (Xamarin.Forms Portable)	Visual C#
UITest	Visual C#

Type: Visual C#

A multiproject template for building apps with Xamarin.Forms, sharing code using a shared assets project. Requires Visual Studio 2013 Update 2 or greater.

Click here to go online and find templates.

Name: App1

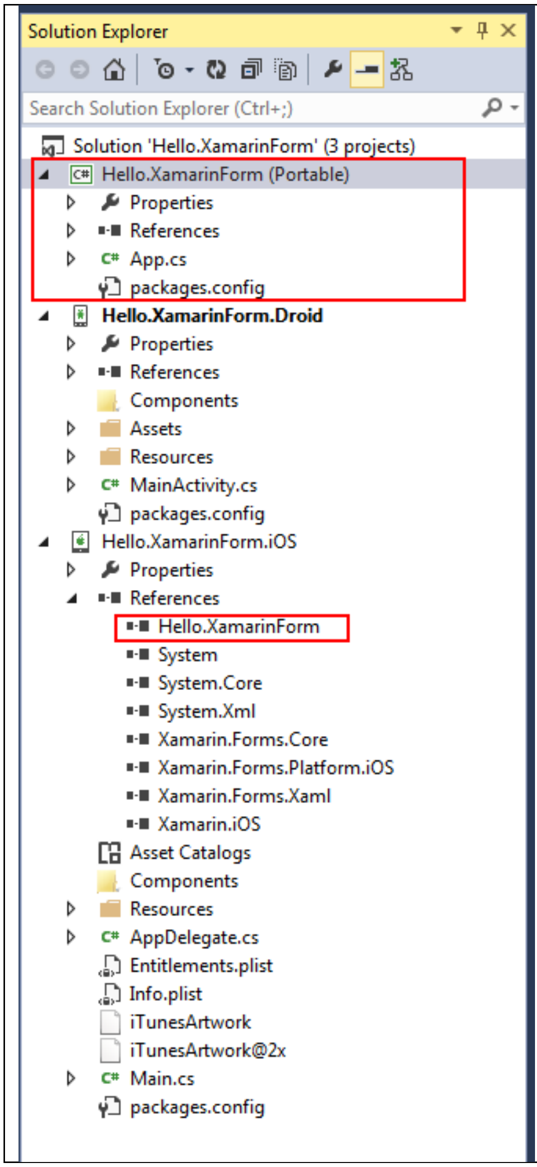
Location: D:\WorkSpace\Xamarin\Exercises

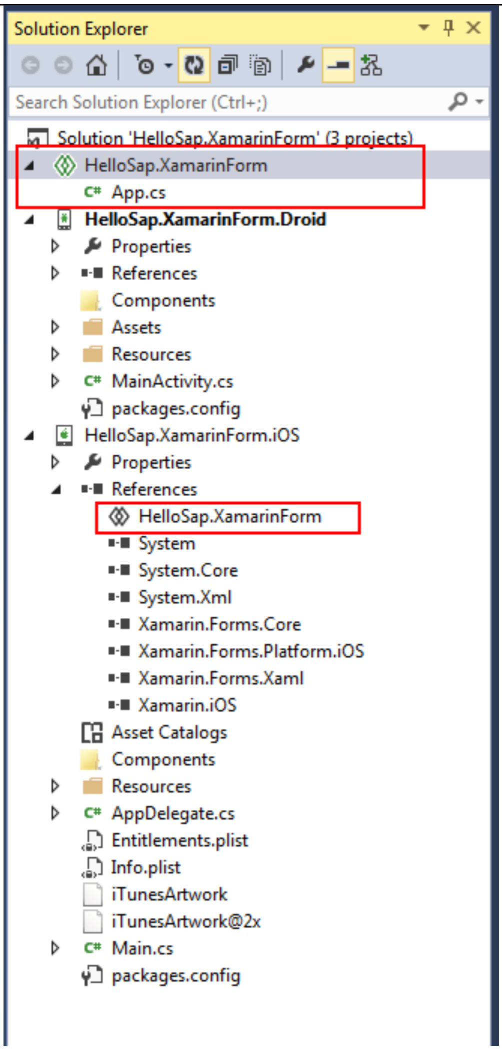
Solution: Create new solution

Solution name: App1

☑ Create directory for solution

□ Add to source control



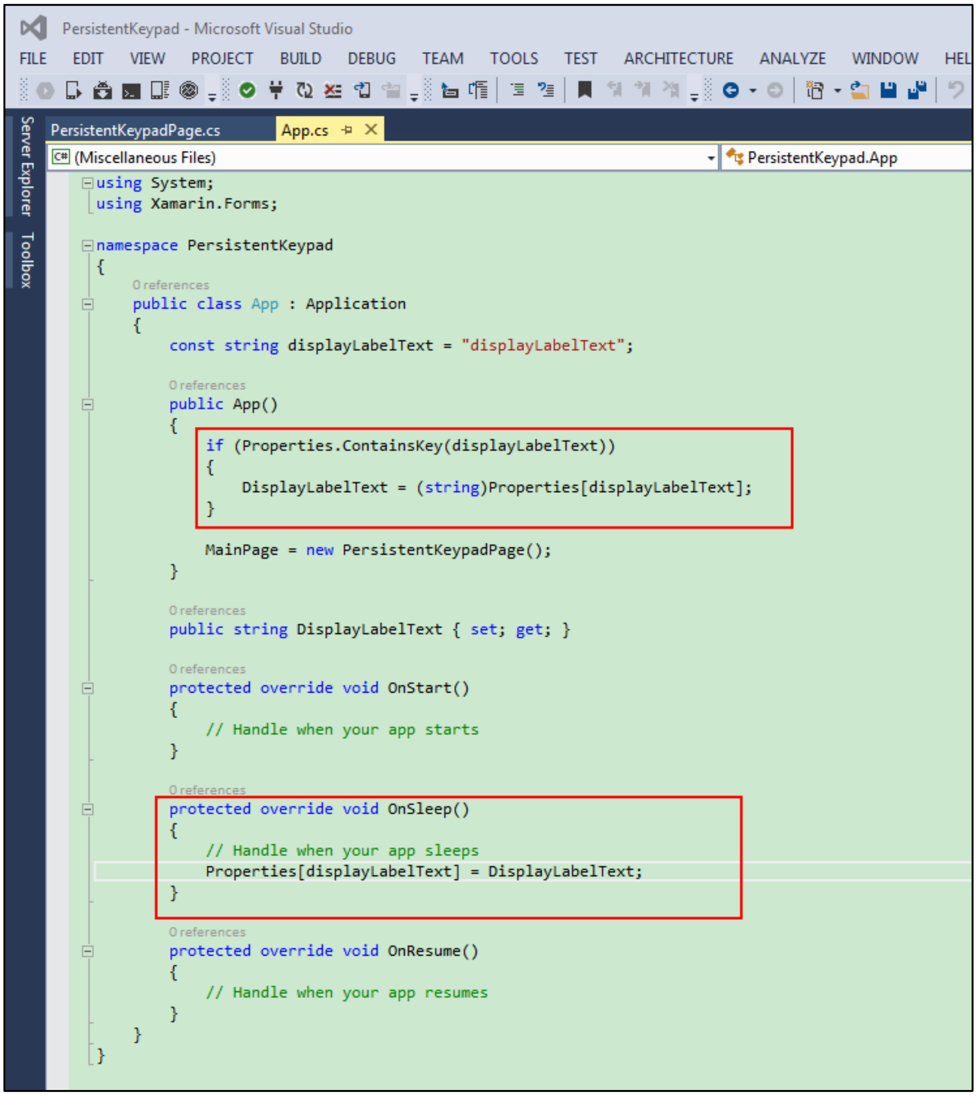


The iOS and Android projects have access to pretty much the same version of .NET, but it is **not** the same version of .NET that a Windows Phone project uses. This means that any .NET classes accessed by the shared code might be somewhat different depending on the platform. As you'll discover later in this book, this is the case for some **file I/O classes in the System.IO** namespace.

7. Store Data

7.1 Transient Data

- Application.Properties(string key, object item)
- Application Events(OnStart, OnSleep, OnResume)


PersistentKeypad - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ARCHITECTURE ANALYZE WINDOW HEL
Server Explorer
Toolbox
PersistentKeypadPage.cs
App.cs
(Miscellaneous Files)
PersistentKeypad.App

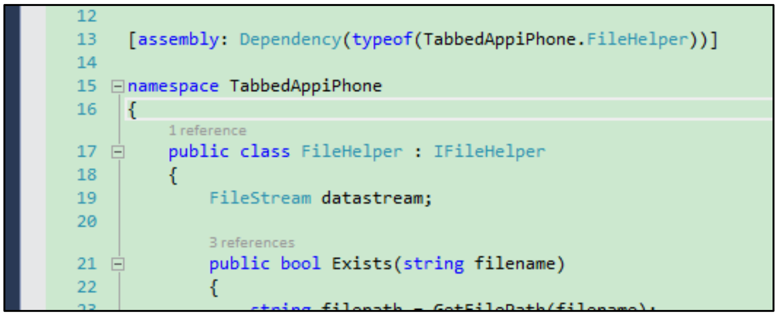
```
using System;
using Xamarin.Forms;

namespace PersistentKeypad
{
    0 references
    public class App : Application
    {
        const string displayLabelText = "displayLabelText";

        0 references
        public App()
        {
            if (Properties.ContainsKey(displayLabelText))
            {
                DisplayLabelText = (string)Properties[displayLabelText];
            }

            MainPage = new PersistentKeypadPage();
        }

        0 references
        public string DisplayLabelText { set; get; }

        0 references
        protected override void OnStart()
        {
            // Handle when your app starts
        }

        0 references
        protected override void OnSleep()
        {
            // Handle when your app sleeps
            Properties[displayLabelText] = DisplayLabelText;
        }

        0 references
        protected override void OnResume()
        {
            // Handle when your app resumes
        }
    }
}
```

Note: If each item of this data is an entry in the Properties dictionary, each item needs a dictionary key. However, if a program needs to save a large file such as a word-processing document, it shouldn't use the Properties dictionary, but in-stead should access the platform's file system directly.

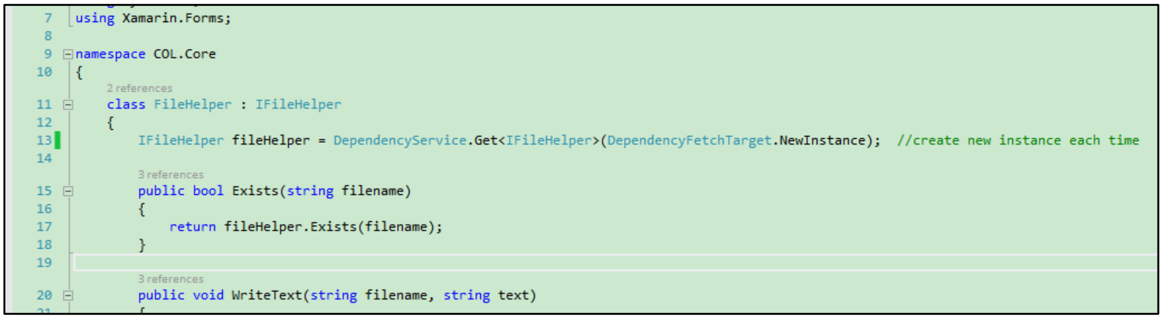
7.2 Large File

Platform's file system:

```
Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);
Xamarin.Forms.DependencyService
```



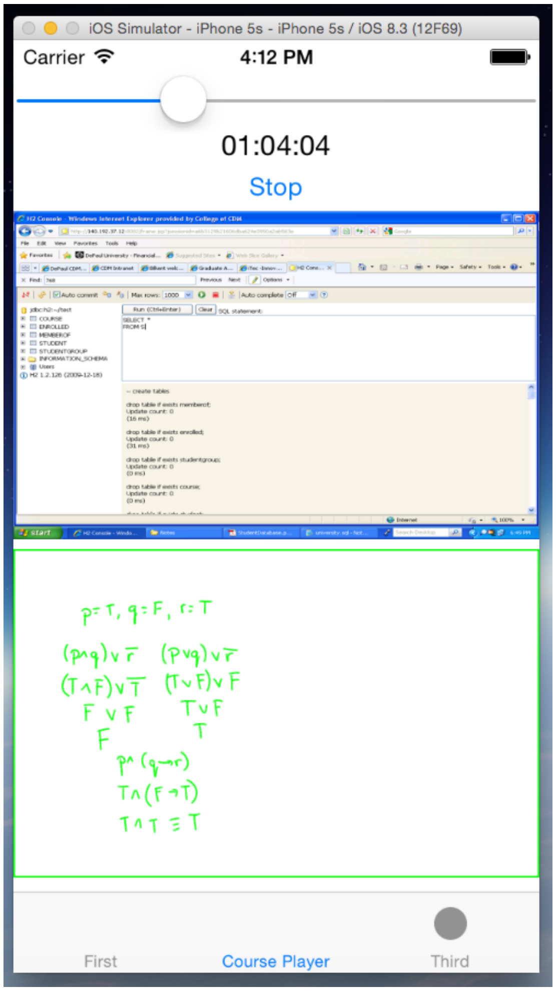
```

[assembly: Dependency(typeof(TabbedAppiPhone.FileHelper))]

namespace TabbedAppiPhone
{
    1 reference
    public class FileHelper : IFileHelper
    {
        FileStream datastream;

        3 references
        public bool Exists(string filename)
        {
            string filepath = GetFilePath(filename);
```



```
using Xamarin.Forms;

namespace COL.Core
{
    2 references
    class FileHelper : IFileHelper
    {
        IFileHelper fileHelper = DependencyService.Get<IFileHelper>(DependencyFetchTarget.NewInstance);  //create new instance each time

        3 references
        public bool Exists(string filename)
        {
            return fileHelper.Exists(filename);
        }

        3 references
        public void WriteText(string filename, string text)
```

8. Prototype of Course Player

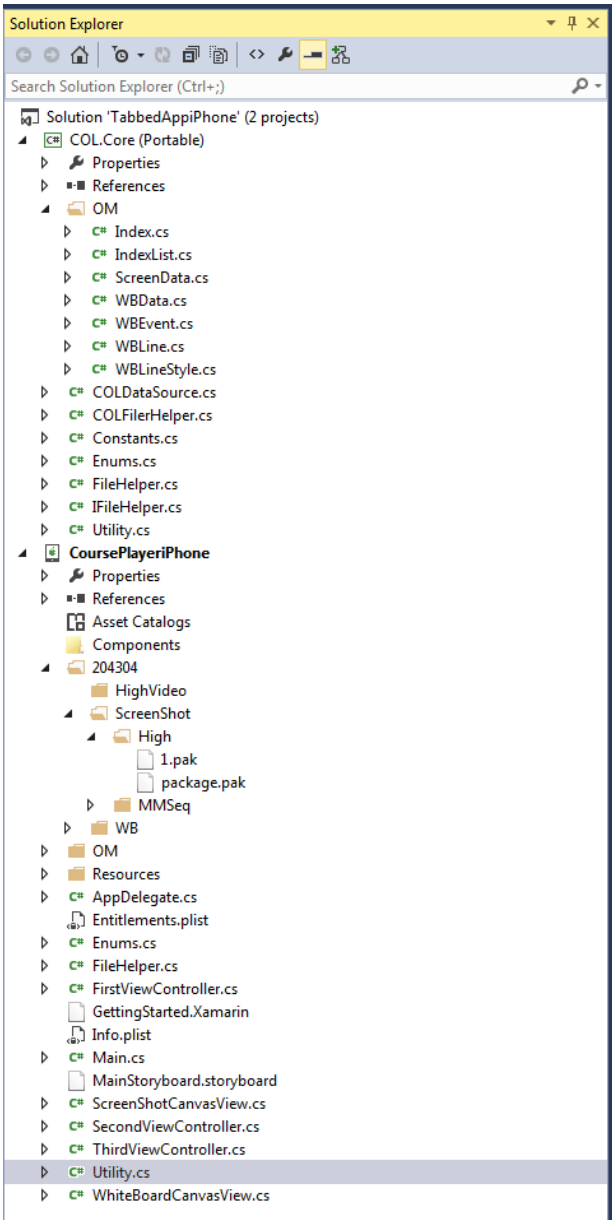
8.1 Layout



Use a slider, label and button to simulate video playing.

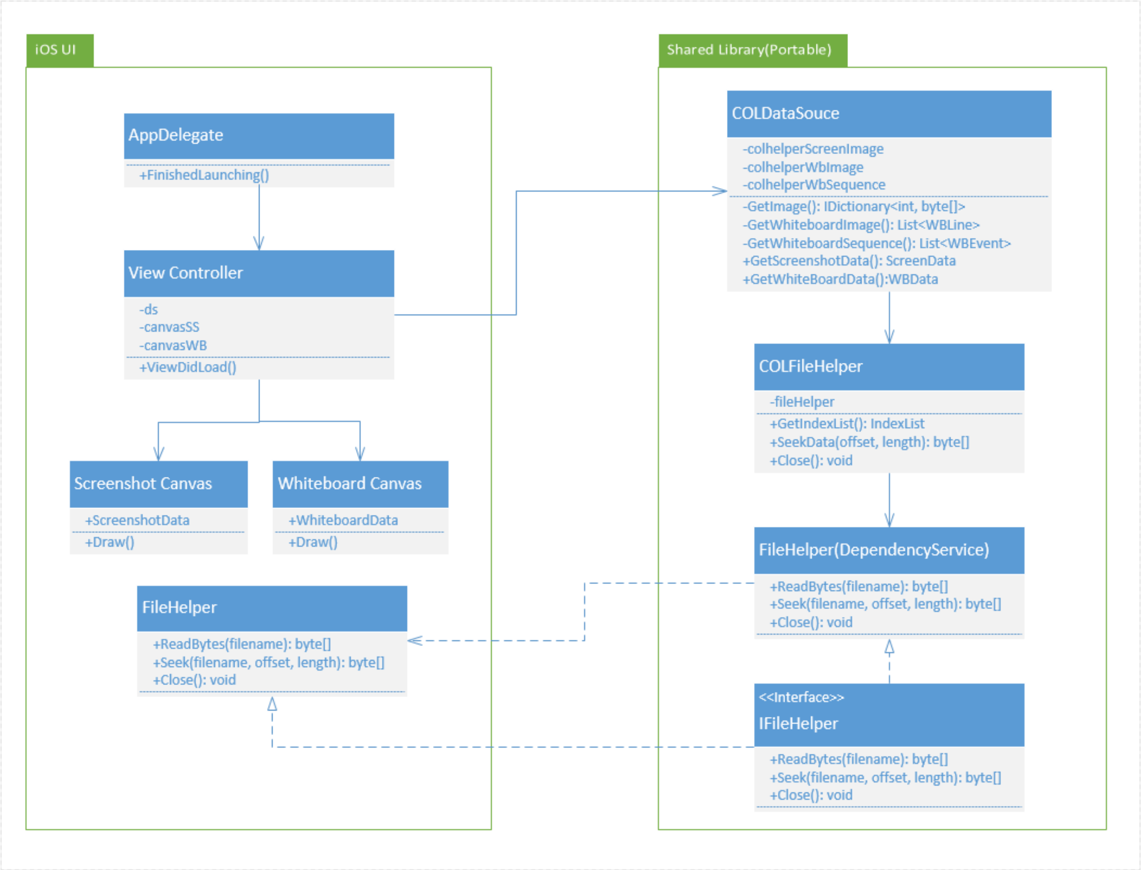
8.2 Components

- COL.Core contains the core function to get and convert data.
- CoursePlayeriPhone contains the course related files and render the UI.



```
Solution 'TabbedAppiPhone' (2 projects)
  COL.Core (Portable)
    Properties
    References
    OM
      Index.cs
      IndexList.cs
      ScreenData.cs
      WBData.cs
      WBEvent.cs
      WBLine.cs
      WBLineStyle.cs
    COLDataSource.cs
    COLFilerHelper.cs
    Constants.cs
    Enums.cs
    FileHelper.cs
    IFileHelper.cs
    Utility.cs
  CoursePlayeriPhone
    Properties
    References
    Asset Catalogs
    Components
    204304
      HighVideo
      ScreenShot
        High
          1.pak
          package.pak
        MMSeq
      WB
    OM
    Resources
    AppDelegate.cs
    Entitlements.plist
    Enums.cs
    FileHelper.cs
    FirstViewController.cs
    GettingStarted.Xamarin
    Info.plist
    Main.cs
    MainStoryboard.storyboard
    ScreenShotCanvasView.cs
    SecondViewController.cs
    ThirdViewController.cs
    Utility.cs
    WhiteBoardCanvasView.cs
```

8.3 UML class diagram


iOS UI
AppDelegate
+FinishedLaunching()
View Controller
-ds
-canvasSS
-canvasWB
+ViewDidLoad()
Screenshot Canvas
+ScreenshotData
+Draw()
Whiteboard Canvas
+WhiteboardData
+Draw()
FileHelper
+ReadBytes(filename): byte[]
+Seek(filename, offset, length): byte[]
+Close(): void
Shared Library(Portable)
COLDataSouce
-colhelperScreenImage
-colhelperWbImage
-colhelperWbSequence
-GetImage(): IDictionary<int, byte[]>
-GetWhiteboardImage(): List<WBLine>
-GetWhiteboardSequence(): List<WBEvent>
+GetScreenshotData(): ScreenData
+GetWhiteBoardData():WBData
COLFileHelper
-fileHelper
+GetIndexList(): IndexList
+SeekData(offset, length): byte[]
+Close(): void
FileHelper(DependencyService)
+ReadBytes(filename): byte[]
+Seek(filename, offset, length): byte[]
+Close(): void
<<Interface>>
IFileHelper
+ReadBytes(filename): byte[]
+Seek(filename, offset, length): byte[]
+Close(): void

9. Important Concerns

9.1 Does Xamarin provide the support of the third-party libraries? How?

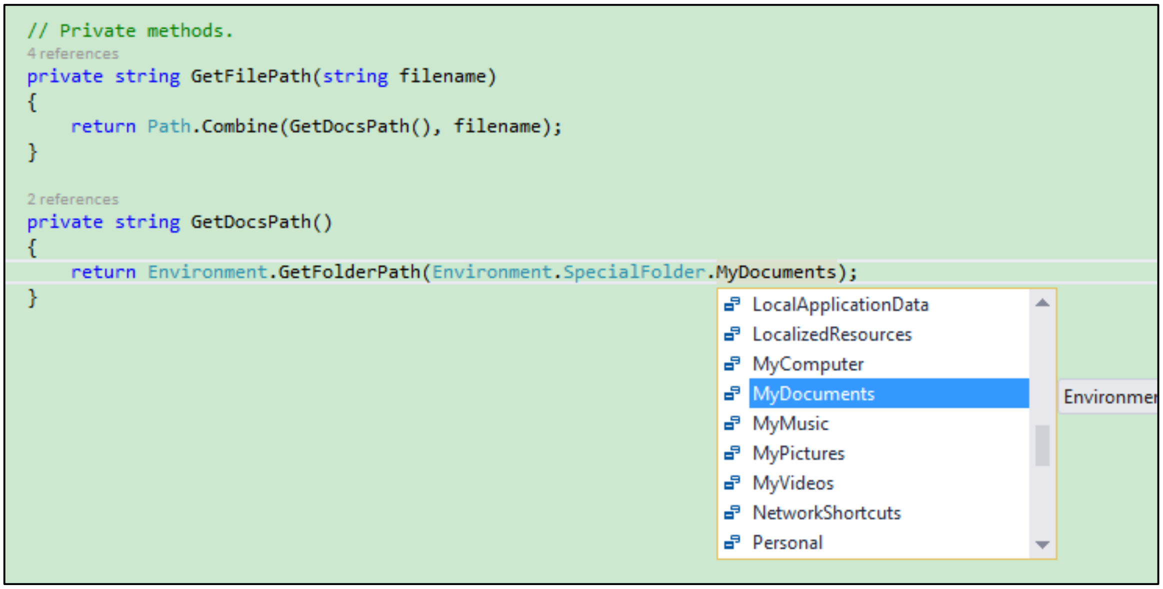
Yes, Xamrain supports not only .NET based class libraries, but also Objective-C, Java, HTML or Javascript to reuse in Xamarin.

Walkthrough: Binding an Objective-C Library

http://developer.xamarin.com/guides/ios/advanced_topics/binding_objective-c/Walkthrough_Binding_objective-c_library/

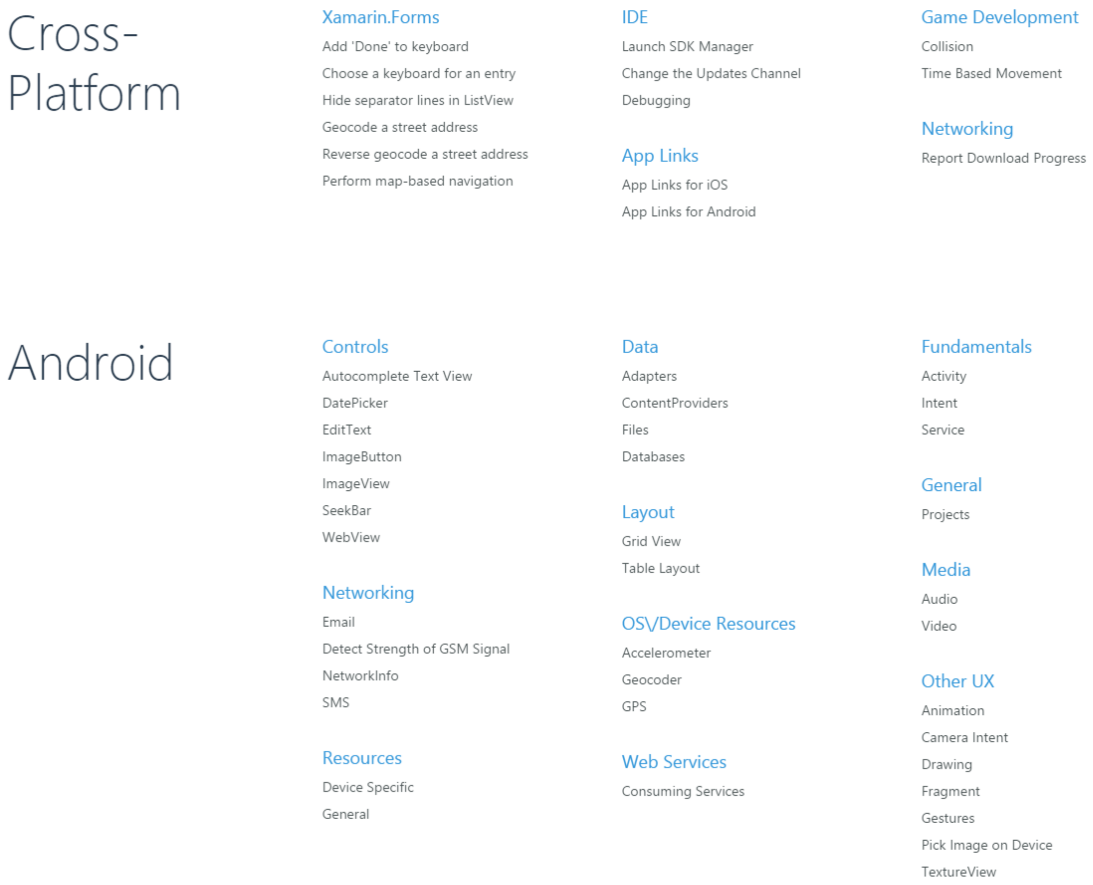
1) First, we'll create an Objective-C static library using Xcode.
2) Then we'll binding this static library with Xamarin.iOS.
3) Next, show how Objective Sharpie can reduce the workload by automatically generating some (but not all) of the necessary API definitions required by the Xamarin.iOS binding.
4) Finally, we'll create a Xamarin.iOS application that uses the binding.

9.2 IO interfaces, eg. file read/write



```
// Private methods.
4 references
private string GetFilePath(string filename)
{
    return Path.Combine(GetDocsPath(), filename);
}

2 references
private string GetDocsPath()
{
    return Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);
}
```

9.3 What features, especially the system relevant functionalities, does Xamarin provide? Briefly, Xamarin supports unified cross-platform features. Separately, it supports platform aimed features. http://developer.xamarin.com/recipes/



Cross-Platform

Xamarin.Forms
- Add 'Done' to keyboard
- Choose a keyboard for an entry
- Hide separator lines in ListView
- Geocode a street address
- Reverse geocode a street address
- Perform map-based navigation

IDE
- Launch SDK Manager
- Change the Updates Channel
- Debugging

App Links
- App Links for iOS
- App Links for Android

Game Development
- Collision
- Time Based Movement

Networking
- Report Download Progress

Android

Controls
- Autocomplete Text View
- DatePicker
- EditText
- ImageButton
- ImageView
- SeekBar
- WebView

Networking
- Email
- Detect Strength of GSM Signal
- NetworkInfo
- SMS

Resources
- Device Specific
- General

Data
- Adapters
- ContentProviders
- Files
- Databases

Layout
- Grid View
- Table Layout

OS\/Device Resources
- Accelerometer
- Geocoder
- GPS

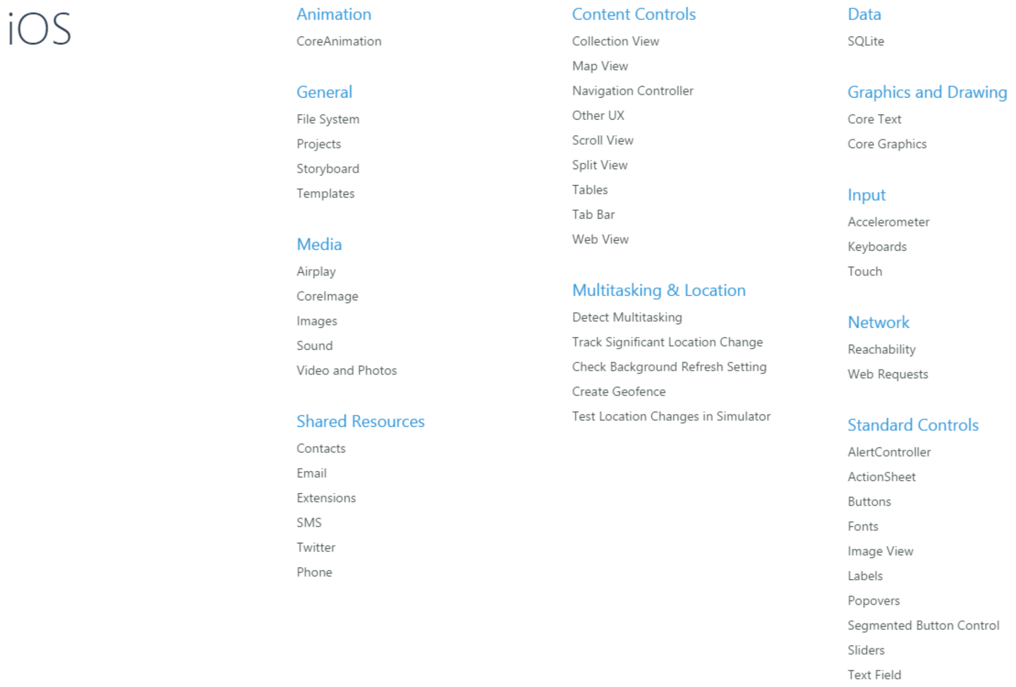
Web Services
- Consuming Services

Fundamentals
- Activity
- Intent
- Service

General
- Projects

Media
- Audio
- Video

Other UX
- Animation
- Camera Intent
- Drawing
- Fragment
- Gestures
- Pick Image on Device
- TextureView



iOS

Animation
CoreAnimation

General
File System
Projects
Storyboard
Templates

Media
Airplay
CoreImage
Images
Sound
Video and Photos

Shared Resources
Contacts
Email
Extensions
SMS
Twitter
Phone

Content Controls
Collection View
Map View
Navigation Controller
Other UX
Scroll View
Split View
Tables
Tab Bar
Web View

Multitasking & Location
Detect Multitasking
Track Significant Location Change
Check Background Refresh Setting
Create Geofence
Test Location Changes in Simulator

Data
SQLite

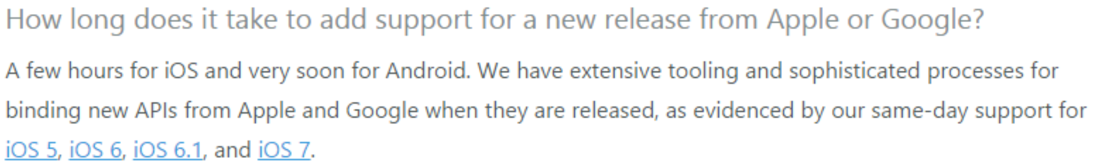
Graphics and Drawing
Core Text
Core Graphics

Input
Accelerometer
Keyboards
Touch

Network
Reachability
Web Requests

Standard Controls
AlertController
ActionSheet
Buttons
Fonts
Image View
Labels
Popovers
Segmented Button Control
Sliders
Text Field

9.4 How does Xamarin follow the new releases of native platforms, iOS and Android?



How long does it take to add support for a new release from Apple or Google?

A few hours for iOS and very soon for Android. We have extensive tooling and sophisticated processes for binding new APIs from Apple and Google when they are released, as evidenced by our same-day support for iOS 5, iOS 6, iOS 6.1, and iOS 7.

10. Overview Conclusion

1) Xamarin has great support for cross-platform mobile development.
2) The development cost is high by using Xamarin.
3) Developers are required to have the knowledge of C#(.Net), Xamarin(Mono), iOS development and Android Development.
4) For iOS, the layout design is not supported enough (eg. auto layout), maybe need to create separate projects for iPhone and iPad.

11. Feature Comparison from Portfolio Perspective

	Native	**Xamarin**
Development Community	Matured mobile platform with plenty of materials/documents.	Lack of resources, like, books, third-party libraries, etc.
Features/Functionalities	Can get the entire features/functionalities provided by the SDK.	Some of the native features are not supported very well(eg. Auto layout).
Third-Party Library	Fully supported	Needs some extra effort to achieve the same purpose.
Coding	Pure native codes	Hybrid of Mono, iOS and Android
Debugging	Comfortable with Xcode, Android Studio	Comfortable with Visual Studio
Deployment	Standard procedure	A little more extra work required.
Extendibility	Easy to add new features	In most cases, it is possible to add new feature, but need do some researching work first.
Reusable	Low	High, if more platforms are required. Besides, the shared library can be enhanced to support web application.
Effort in Development phase	High	Low, if more platforms are required
Effort in maintenance phase	High	Low, especially when the bug is not relevant with UI.
Required Developers	2 (1 for iOS, 1 for Android)	3 (1 for Xamarin shared library, 1 for iOS UI, 1 for Android UI)
Developer's Qualification	Platform specified knowledge	Cross-platform knowledge, Xamarin platform, plus C#, Mono
License Fee(Per Year)	iOS $99; Android free	iOS developer account $99 Xamarin iOS: $999 Xamarin Android: $999
Potential Risk	None	Uncertainty of Xamarin's future.

The comparison is based on the assumption that our APP only supports iOS and Android.

12. Reference

- http://xamarin.com/faq
- https://components.xamarin.com/
- http://developer.xamarin.com/recipes/ios/general/file_system/load_a_file/
- http://developer.xamarin.com/recipes/android/data/files/
- http://developer.xamarin.com/recipes/
- http://developer.xamarin.com/guides/cross-platform/insights/