Austin Mobile .NET Develops Group

Thank Sponsors

- Microsoft - Ryan Joy
- Xamarin - James Montemagno
- Netrix - Rabi Satter

Xamarin

Enabling Developers to Create Native
iOS, Android, Mac, & Windows apps in C#

Rabi Satter

Rabi@Satter.org

www.satter.org

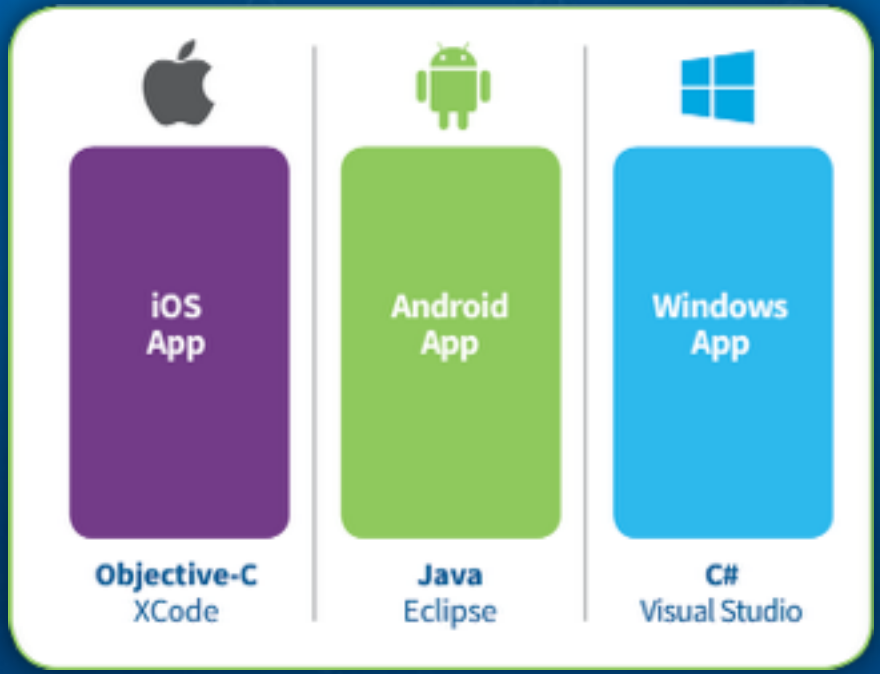
@rsatter42

What is Xamarin?

- Xamarin enables developers to reach all major mobile platforms!
  - Native User Interface
  - Native Performance
  - Shared Code Across Platforms
  - C# & .NET Framework

Mobile Development Approaches

Silo’d Approach



Build Apps Multiple Times

- Multiple Teams
- Multiple Code Bases
- Different toolsets

Write Once, Run Anywhere Approach

- Lowest common denominator
- Browser fragmentation
- Developing & designing for 1 platform, happen to get other platforms

Xamarin's Unique Approach

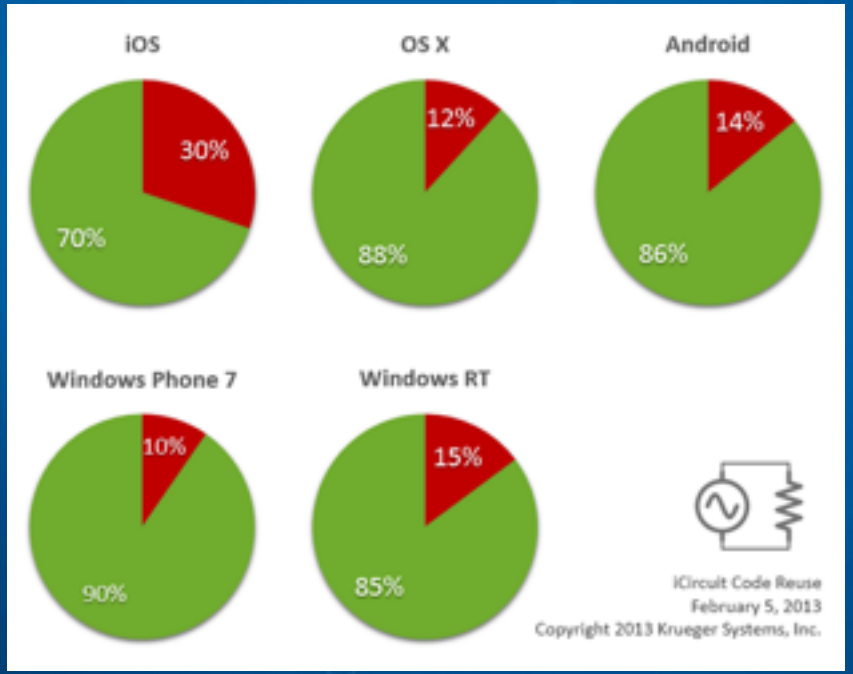
- Native User Interface
- Native Performance
- Shared code across platforms
- C# & .NET Framework
- Full API Coverage

Native Performance

Xamarin.iOS does full Ahead Of Time (AOT) compilation to produce an ARM binary for Apple's App Store.

Xamarin.Android takes advantage of Just In Time (JIT) compilation on the Android device.

Code Sharing: Accelerate Development



- Up to 90% Code Reuse
- Leverage existing libraries

Code-sharing stats from iCircuit, built with Xamarin and featured on the App Store.

Source: http://praeclarum.org/post/42378027611/icircuit-code-reuse-part-cing

Why C#?

C# Is Awesome

```
from p in Table<Person> ()
    where p.ID == id
    select p;
```

- LINQ Support

```
var doc = XDocument.Load(url);
foreach(var item in doc.Root.Elements()) {
    var text = item.Value;
}
```

- Work With XML Easily XDocument

```
button.TouchUpInside += (s, o) => {
    message.Text = "Hello!";
};
```

- Event Handling & Delegates

C# Is Awesome – JSON Made Easy

Json.NET offers simple conversions to and from JSON strings and .NET objects, with SerializeObject and DeserializeObject methods.

```
public class Person
{
    public string Name { get; set; }
    public DateTime Birthday { get; set; }
}
var person = new Person { Name = "Bob", Birthday = new DateTime (1987, 2, 2) };
var output = Newtonsoft.Json.JsonConvert.SerializeObject (person);

person = Newtonsoft.Json.JsonConvert.DeserializeObject<Person> (output);
Console.WriteLine ("{0} - {1}", person.Name, person.Birthday);
```

See the Difference – Attributed Strings

Objective-C

```
CFStringRef keys[] = {
    kCTFontAttributeName,
    kCTForegroundColorAttributeName
};

CFTypeRef bval[] = {
    cfListLineCTFontRef,
    CGColorGetConstantColor(kCGColorBlack)
};

attr = CFDictionaryCreate (kCFAllocatorDefault,
    (const void **) &keys, (const void **) &bval,
    sizeof(keys) / sizeof(keys[0]), &kCFTypeDictionaryKeyCallBacks,
    &kCFTypeDictionaryValueCallBacks);

astr = CFAttributedStringCreate(kCFAllocatorDefault, CFSTR("Hello World"),
```

C# with Xamarin

```
var attrs = new CFStringAttributes {
    Font = listLineCTFont,
    ForegroundColor = UIColor.Black.CGColor
};

var astr = new NSAttributedString ("Hello World", attrs);
```

Probing for properties on an AudioFile

Objective-C

```
UInt32 maxPacketSize;
UInt32 PropertySize = sizeof(maxPacketSize);
AudioFileGetProperty (
    audioFileID,
    kAudioFilePropertyPacketSizeUpperBound,
    &PropertySize,
    &maxPacketSize
    );
```

C# with Xamarin

```
var maxPacketSize = audioFile.PacketSizeUpperBound;
```

See the Different – Android ItemClick

Java

```
listView.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

    // Value of item
    String  itemValue    = (String) listView.getItemAtPosition(position);

    // Show Toats
    Toast.makeText(getActivity(),"Position :"+ position + "  ListItem : "
                            + itemValue , Toast.LENGTH_LONG).show();
    }

});
```

C# with Xamarin

```
listView.ItemClick += (sender, args) => {

    // Value of item
    var itemValue = (string)listView.GetItemAtPosition(args.Position);

    //Show Toast
    Toast.MakeText(this, string.Format("Postition: {0} ListItem: {1}",
                        args.Position, itemValue), ToastLength.Long).Show();
};
```

C# & Async with Xamarin

```
listView.ItemClick += async (sender, args) => {

    // Value of item
    var itemValue = (string)listView.GetItemAtPosition(args.Position);

    //Show Toast
    Toast.MakeText(this, string.Format("Postition: {0} ListItem: {1}",
                        args.Position, itemValue), ToastLength.Long).Show();
};
```

Async/Await

```
public async Task ExecuteGetPodcastsCommand()
{
  if (IsBusy)
    return;

  try
  {
    IsBusy = true;
    var client = new HttpClient();
    // Request from server podcast xml
    var podcastString = await client.GetStringAsync(PodcastUrl);

    // Parse Xml into data model and load into list
    var casts = await ParseXml(podcastString);

    foreach (var cast in casts)
    {
      Podcasts.Add(cast);
      FilteredPodcasts.Add(cast);
    }

  }
```

Write Beautiful & Maintainable Code

Why Xamarin?

Write Everything in C#

iOS, Android, Windows, Windows Phone, Mac

2.5+ Billion Devices!

Xamarin History

Over a Decade of Enterprise Production Use

2000

Ximian Founded

2001

Mono Started

2003

Ximian Acquired by Novell

2009

First iOS product (now Xamarin.iOS) launches

2011

Xamarin Founded

First Release of Xamarin.Android

2012

Reach 200,000 Developer Mark

First release of Xamarin.Mac

Launch Partner Program

2013

500,000+ Developers
100+ Partners
100+ Components

Xamarin 2.0

Component Store

Evolve 2013

Xamarin Test Cloud

Microsoft Partnership

100% API Coverage

Anything you can do in Objective-C or Java can be done in C# and Visual Studio with Xamarin!

Always Up to Date

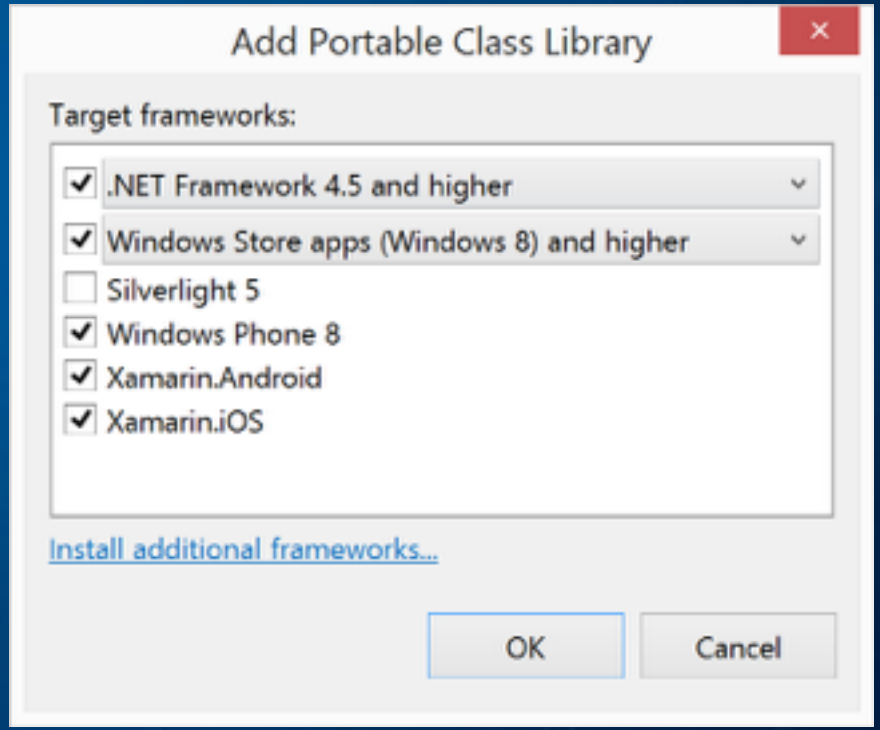
Always up-to-date with the latest APIs.

Great track record.

Shipped same-day support for iOS 5, iOS 6, iOS 6.1, **iOS 7**

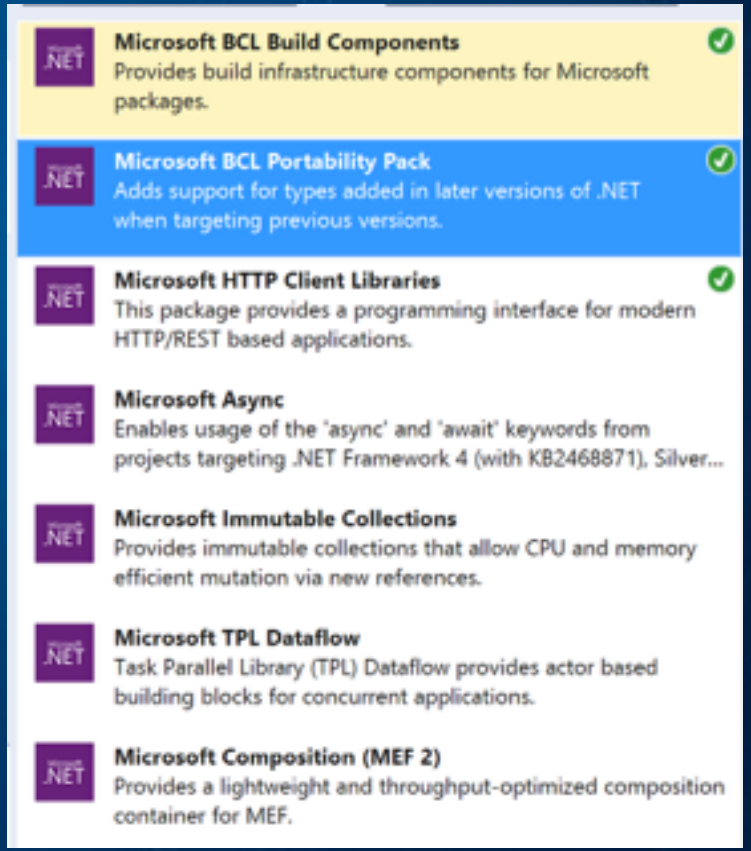
Portable Class Libraries

- 1 Assembly
- Multiple Platforms
- **Including:**
  - **Xamarin.Android**
  - **Xamarin.iOS**



Portable Class Libraries Features

- Centralized Code Sharing
  - How you expect it to work
  - Debug seamlessly into and out of PCL
- Project/Assembly Sharing
  - NuGet



PCLs – Well Documented

System.Collections.Generic Namespace

.NET Framework 4.5 | Other Versions | 41 out of 50 rated this helpful - Rate this topic

The System.Collections.Generic namespace contains interfaces and classes that define generic collections, which allow users to create strongly typed collections that provide better type safety and performance than non-generic strongly typed collections.

Classes

	Class	Description
	Comparer<T>	Provides a base class for implementations of the IComparer<T> generic interface.
	Dictionary<TKey, TValue>	Represents a collection of keys and values.
	Dictionary<TKey, TValue>.KeyCollection	Represents the collection of keys in a Dictionary<TKey, TValue>. This class cannot be inherited.
	Dictionary<TKey, TValue>.ValueCollection	Represents the collection of values in a Dictionary<TKey, TValue>. This class cannot be inherited.
	EqualityComparer<T>	Provides a base class for implementations of the IEqualityComparer<T> generic interface.
	HashSet<T>	Represents a set of values.
	KeyedByTypeCollection<TItem>	Provides a collection whose items are types that serve as keys.
	KeyNotFoundException	The exception that is thrown when the key specified for accessing an element in a collection does not match any key in the collection.
	LinkedList<T>	Represents a doubly linked list.
	LinkedListNode<T>	Represents a node in a LinkedList<T>. This class cannot be inherited.
	List<T>	Represents a strongly typed list of objects that can be accessed by index. Provides methods to search, sort, and manipulate lists.

Microsoft and Xamarin Partner Globally

With Xamarin, developers combine all of the productivity benefits of C#, Visual Studio 2013 and Windows Azure with the flexibility to quickly build for multiple device targets.”

S. Somasegar, Corporate Vice President, Microsoft

Distribute Everywhere

A Xamarin app is an app, distribute it everywhere

Products

Xamarin for
Visual Studio

Xamarin.iOS

Xamarin Studio

Xamarin.Android

Xamarin Test Cloud

Xamarin.Mac

Component Store

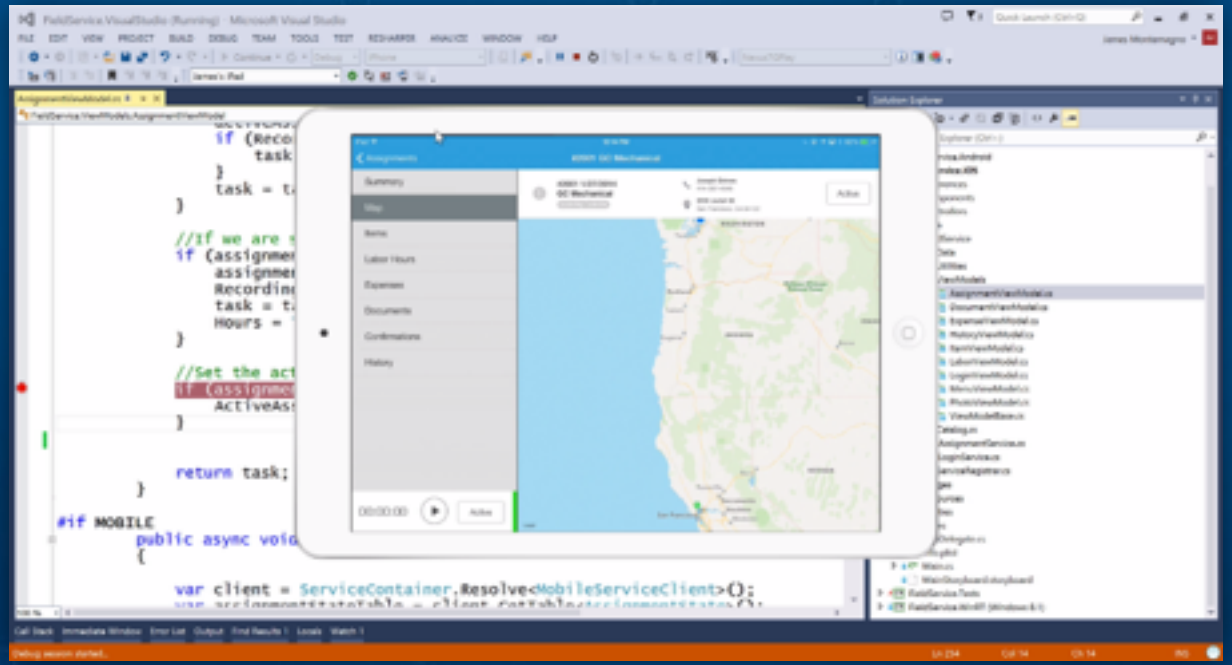
.NET Mobility Scanner

Development Environment

Xamarin Studio
PC or Mac

Visual Studio
Plugin
VS
2010/2012/2013

Visual Studio Integration



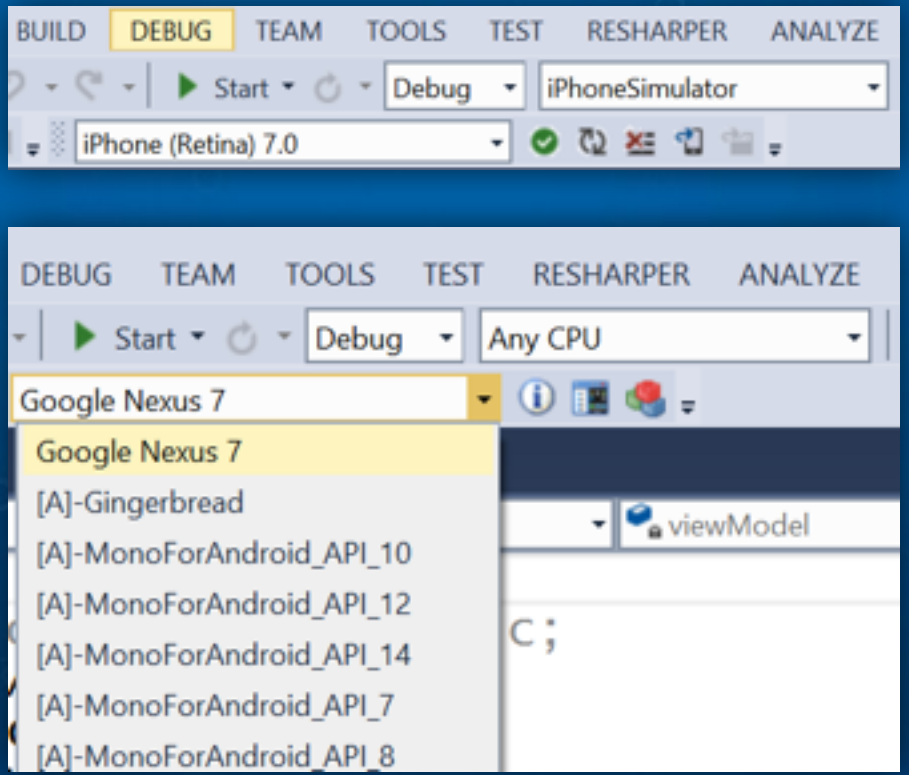
A single solution:

- iOS
- Android
- Windows Phone
- Windows Store

Leverage the entire Microsoft ecosystem:

- ReSharper
- Team Foundation Server
- Your favorite code coverage and profiling tools

Visual Studio Integration



Debug to:

- Emulators
- Devices

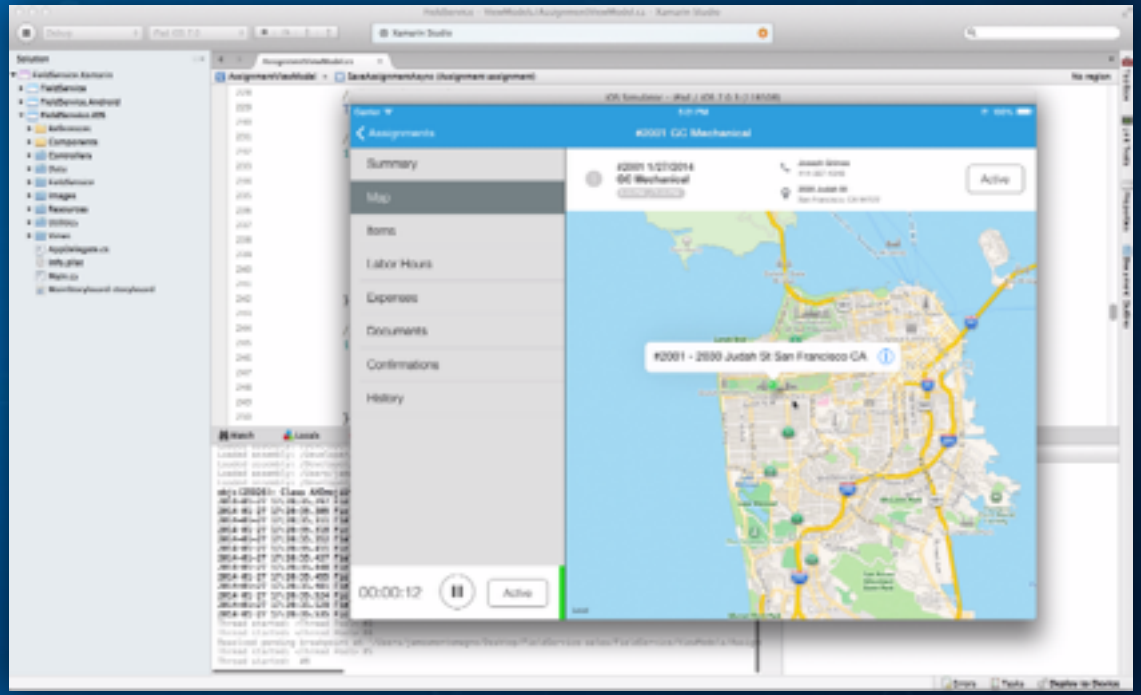
Integrated into toolbar

- Status
- Logs
- List of devices

Just Click Start Debugging!

Xamarin Studio

- Optimized for cross-platform mobile development
- Explore native APIs with code completion
- World class Android and iOS designers
- Powerful debugging on simulator or device



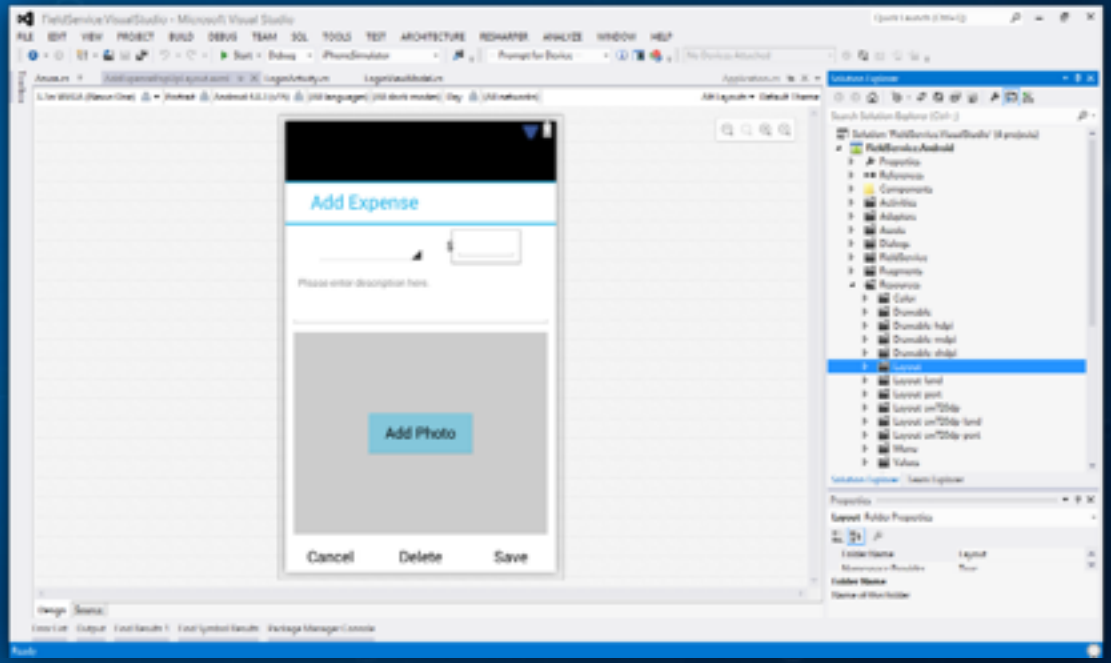
Android Designer

- Worlds best Android designer
- Available in:
  - Xamarin Studio
  - Visual Studio

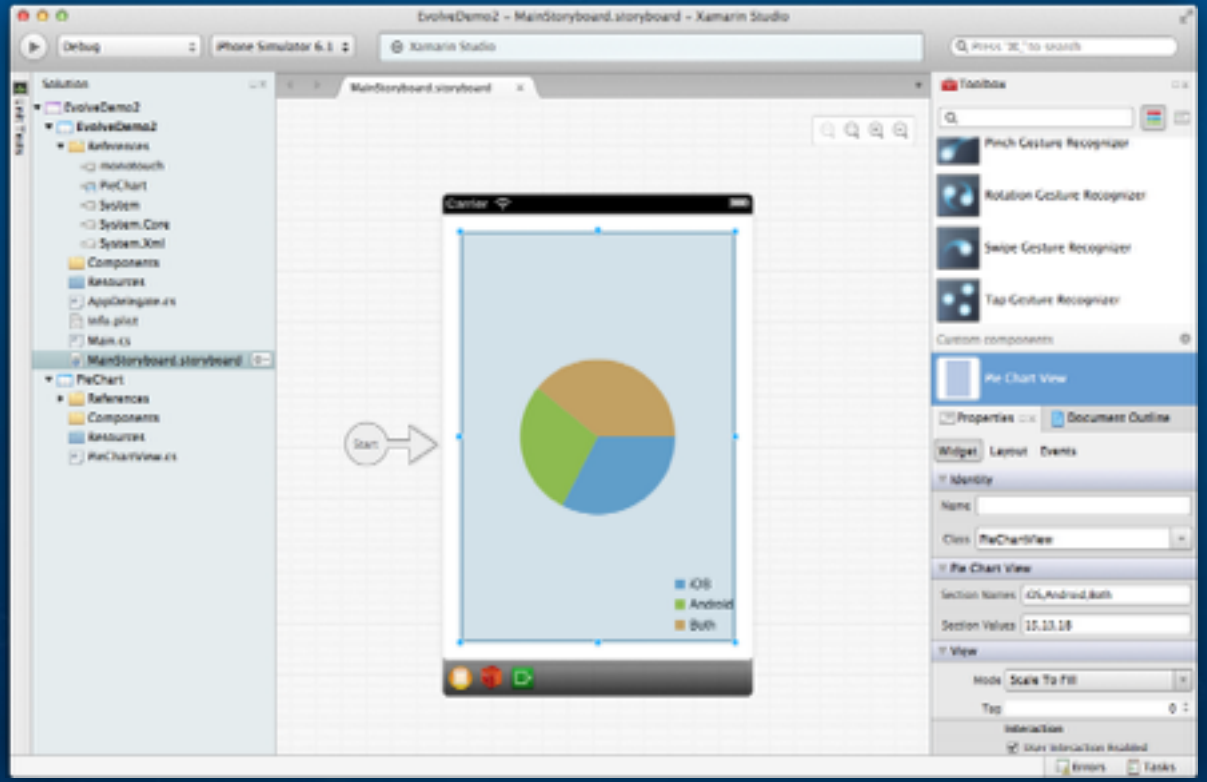
- Create UI with drag & drop simplicity

- Target multiple screen sizes, resolutions and Android versions

- Layouts saved in standard Android XML files

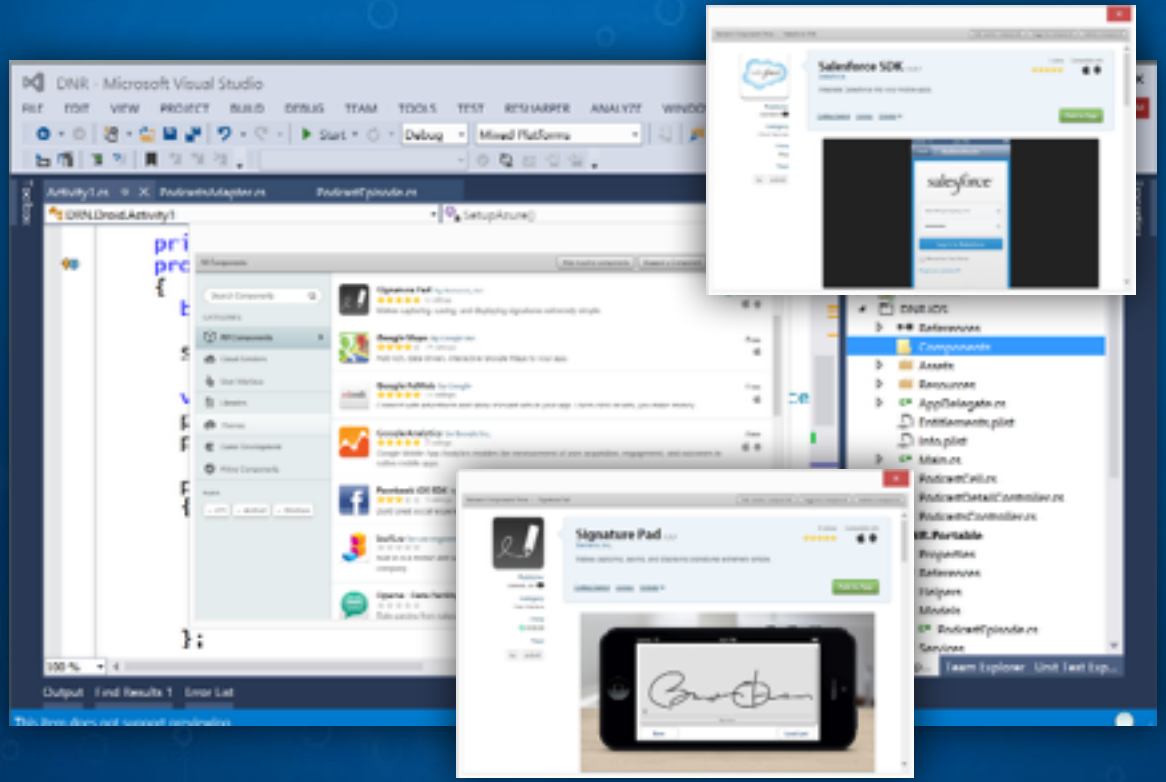


iOS Designer



- World's first iOS Designer available in Xamarin Studio (and soon Visual Studio)
- Follows familiar Visual Studio designer idioms
- Supports all UIKit elements
- Edit custom and 3rd party components
- Live preview of changes to properties

Xamarin Component Store



Build Apps Faster

- Add high quality pre-built app components directly from Visual Studio and Xamarin Studio
- Beautiful cross-platform UI controls, cloud services and enterprise backend integrations are just a

.NET Mobility Scanner

How mobile is your .NET?

- Scan .exe or .dll to determine compatibility
- Generates full report:

	Calls	Platform-specific method name	Namespace	Suggestions	Compatibility
+	3	string Type.Name { get; }	System		
+	3	Assembly TypeInfo.Assembly { get; }	System.Reflection		
+	2	Headers.HttpRequestHeaders HttpRequestMessage.Headers { get; }	System.Net.Http		
+	2	HttpClient..ctor (HttpMessageHandler)	System.Net.Http		
+	2	HttpContent HttpRequestMessage.Content { set; }	System.Net.Http		
+	2	HttpContent HttpResponseMessage.Content { get; }	System.Net.Http		

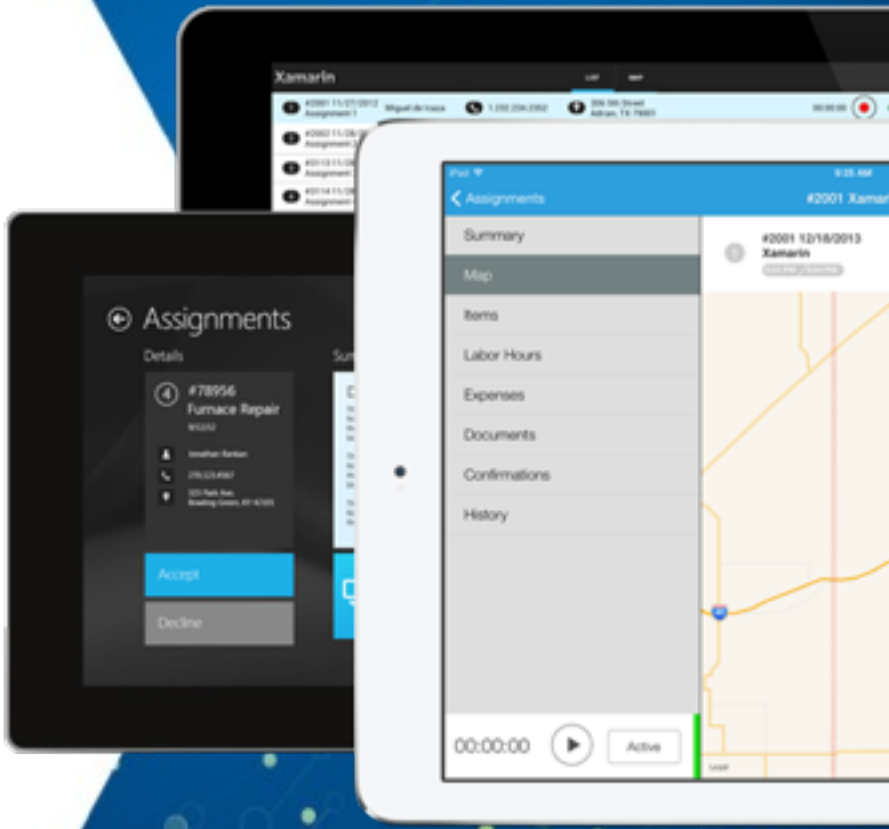
Prebuilt Field Service App

App Features:

- Location-based assignments / routing
- Time recording
- Camera capture of work
- Signature capture

Cross-Platform Techniques

- MVVM pattern
- Xamarin.Mobile
- Signature control from Component Store
- SQLite for storage on all 3 platforms



DEMO

Xamarin

GET YOUR C# SHIRT TODAY!

Xamarin Evolve 2013

#1
Trending on Twitter during keynote

Sold Out!!

650
Conference Attendees

8,400
Live stream hours viewed

Watch Sessions & Highlights: http://xamarin.com/evolve/2013

Xamarin
Evolve 2014
Atlanta
October 6 – 10
Evolve
2014
C#
www.xamarin.com/evolve

Rdio Case Study

About Rdio

- Rdio is the leading music streaming and sharing service started by the founders of Skype
- Rdio is already in 30 countries with over 20 million songs.
- Available in 31 countries

Wired Magazine

"Rdio's mobile app is also the best looking and best implemented of the three [music streaming apps reviewed]."

Sharing 50,000 lines of code

Challenge

- Unable to deliver features on all platforms simultaneously because of multiple, platform-specific code bases
- High costs to development and maintain apps

Results

- Using Xamarin, Rdio now **shares over 50,000 lines of C#** across iOS, Android, and Windows, enabling them to focus on feature-parity and user experience.

Sharing 50,000 lines of code

"Xamarin offers the best of all worlds. We deliver high performance, native apps that, until Xamarin, were only possible with Objective-C and Java. Sharing over 50,000 lines of code across platforms gives us more time to spend on great user experiences."

Matt Crocker
Director of Client Engineering

Xamarin University

- Go Mobile Program
  - Live online training from Xamarin experts
  - Lectures and labs
  - One-on-one with expert Xamarin mobile developers
- What You'll Learn

- iOS and Android Fundamentals
- Mobile App Lifecycle
- Native UI and Controls
- Cross-platform Architecture
- Mobile Best Practices
- Secure Backend Integration

Xamarin Developer Certifications

Xamarin skills are increasingly in demand – certifications let the market know you represent top talent