

Cross-Platform Mobile

The Xamarin.Forms Way

Alien Arc Technologies, LLC

www.alienarc.com
@AlienArcLLC

Duane Newman

About.me/DuaneNewman
@DuaneNewman



Co-Founder
Alien Arc Technologies, LLC

A Modern App Development Company

What will be covered

- Quick look at...
 - The Mobile Landscape
 - Traditional app development
 - Xamarin
- Xamarin.Forms
 - Getting started, licensing
 - Concepts, Benefits & Drawbacks
 - Examples

The Mobile Landscape

- Android (Phone & Tablet)
 - Leading the market
 - 2x the user base of iOS
- iOS (Phone & Tablet)
 - First to market
 - Brought us the first wildly successful app store
 - Users spend \$4 on apps to every \$1 Android users spend (Fortune '14)
- Windows Phone
 - Slow adoption in US, but higher rates worldwide
 - Continues to grow and has lots of resources behind it (MS)
 - Developers earn avg. \$0.23/download compared to \$0.04/download on Android
- Windows Store (Metro)
 - Windows 8/8.1 adoption rates increasing
 - Surface Pro 3 has had strong sales

Traditional App Development

- Write an app on Platform X using Language and Tools X
- Eventually completely re-write the app on Platform Y using Language and Tools Y
- Maybe completely re-write the app again on Platform Z using Language and Tools Z, maybe
- Is there a better way?

Hybrid Apps - Cordova (PhoneGap)

- Lots of support and implementations
 - Adobe
 - Telerik
- Allows you to write your app once
 - Uses HTML & JS
- Not Native
 - Runs in embedded browser
 - Only has access to some components

Xamarin

- Founded in 2011 by Miguel de Icaza and Nat Friedman
- Original founders of Ximian and creators of Mono
- Maintains Mono, the open source implementation of .NET
- Uses the .NET specifications and Mono to facilitate writing C# code that is compiled to native platform specific binaries

I'm often asked "What should I be learning?" I'm convinced
C#/XAML Programmers should be looking at Xamarin.
-Jesse Liberty

Recent News

- Xamarin
 - Support for Visual Studio 2015 Preview
 - Buy Xamarin with Visual Studio
 - 20% discount on Xamarin
 - Xamarin Starter Edition
 - Support for Visual Studio (including Community Edition)
 - Doubling application size limit
- Microsoft
 - Announced Android Emulator for Visual Studio 2015
 - Runs on Hyper-V, so you can do Android and Windows Phone without rebooting.

Native apps, written in .NET & C#

- Xamarin Apps look and feel native because they are native
 - You have access to the Native User Interfaces
 - You have access to Native APIs and libraries
 - You have the performance of a Native app
- Your code is compiled to native binaries for the target platform
 - “Traditional” Xamarin development allows sharing of roughly 75% of your codebase with the other 25% being UI and platform specific services.

Xamarin Pros and Cons

- Advantages
 - C# - because it's awesome
 - Truly common code
 - Native app, full access to anything each platform offers
 - Android Player – fast Android emulator
- Disadvantages
 - Third Party
 - Cost, best option is the business license = \$\$\$\$
 - Environment choices
 - Xamarin Studio (can't do WP at all, can't do iOS on Windows)
 - Visual Studio (requires business)

Xamarin Forms

- Builds on top of existing Xamarin Framework
- Adds a common UI model
- Express your UI in C# or XAML
- Renders the defined UI using native controls for the target platform
- Supports MVVM pattern
- Includes Binding and INotifyPropertyChanged support

So you can concentrate doing the 60-70% but you always have to spend some time writing the UI for that platform and Xamarin.Forms will help you [do] the extra 30% in a breeze
-Lino Tardos, .NET Rocks #1042

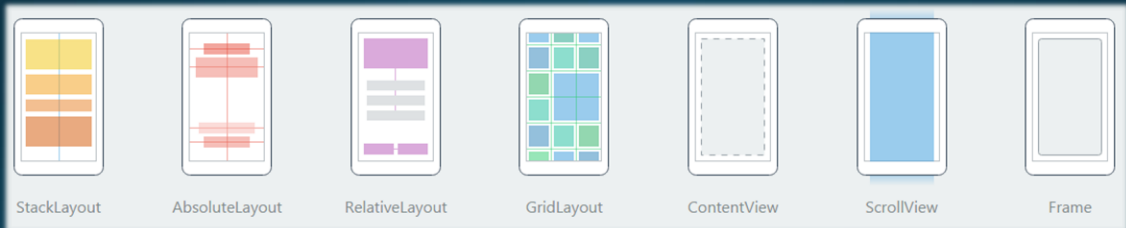
Anatomy of Xamarin Forms

- Pages



Anatomy of Xamarin Forms

- Pages
 - Layouts



Anatomy of Xamarin Forms

- Pages
 - Layouts
 - Controls

ActivityIndicator
BoxView
Button
DatePicker
Editor
Entry

Image
Label
ListView
Map
OpenGLView
Picker

ProgressBar
SearchBar
Slider
Stepper
TableView
TimePicker

WebView
EntryCell
ImageCell
SwitchCell
TextCell
ViewCell

Getting Started

- Environment
- Project type
- UI Approach

Environment

- Windows & Mac required to support all platforms
 - WP requires Windows and Visual Studio
 - iOS requires a Mac and OSX
 - Android can be done on either environment
 - Ideal setup for doing all OS development:
 - Either
 - Windows PC and Mac Mini (for builds)
 - Mac with Parallels and Windows
 - Visual Studio Professional+ (and now Community Edition)
 - Xamarin Business
 - Allows you to do all development in 1 IDE

Project Type

- Shared Project
 - Advanced for of file linking (Add File->Add as Link)
 - You code is compiled once for each platform
 - Allows you to do #if
 - Can't reference embedded resources (no assembly to reference)
- Portable Class Library
 - Creates a single assembly for your base library
 - Compiled once and referenced by each platform project
 - Preferred method if doing XAML

UI Approach

- Coded UI
- XAML

Coded UI

- UI is defined in C#
- Similar to looking at Windows Forms
- Full Intellisense support

DEMO 1

- C# Coded UI

XAML

- Xamarin Forms XAML
 - Not WPF/Silverlight/Win8 XAML
 - Different control names
 - Slightly different properties
- Same core concepts
 - Declarative Markup
 - Binding
 - Commands

So that's the big thing is that your expressing what your intent is with the XAML but its rendered with the native UI of the tool.
-Carl Franklin, .NET Rocks #1042

What is XAML?

```
<ContentPage>
  <ListView ItemsSource="{Binding Speakers}">
    <ListView.ItemTemplate>
      <DataTemplate>
        <ViewCell>
          <StackLayout>
            <Label Text="{Binding Name}" Font="Bold, Large" />
            <Label Text="{Binding Bio}" />
            <StackLayoutOrientation="Horizontal">
              <Label Text="City: " />
              <Label Text="{Binding City}" />
            </StackLayout>
          </StackLayout>
        </ViewCell>
      </DataTemplate>
    </ListView.ItemTemplate>
  </ListView>
</ContentPage>
```

DEMO 2

- Revise C# UI to XAML

High/Low Game Demo

- Binding
- MVVM
- Built-In Services
 - DependencyService
- OnPlatform
 - Custom UI based on the current platform.

Dates & Numbers Demo

- Data Entry/Binding
 - Dates
 - Numbers

Xamarin Forms Improvements

- Xamarin Forms is under constant development
- V1.3.0 technical preview
 - Styles! (Plus platform specific styles)
 - Dynamic resources
 - Behaviors
 - Triggers
 - Improved binding support
 - Sleep/Resume/Start events
 - Application level resource dictionary
 - Better font properties

Resources

- Video
 - Xamarin Evolve – evolve.xamarin.com
- Audio
 - .NET Rocks
 - Yet Another Podcast (Jesse Liberty)
- Blogs
 - Jesse Liberty – www.JesseLiberty.com
- Code
 - James Montemagno - github.com/JamesMontemagno
 - Xamarin Samples - github.com/xamarin/xamarin-forms-samples
- Community Extensions
 - Xamarin Forms Labs - github.com/XLabs/Xamarin-Forms-Labs

Thank you!

Duane Newman
Co-Founder, Solution Architect
Alien Arc Technologies, LLC

About.me/DuaneNewman
[@DuaneNewman](https://twitter.com/DuaneNewman)

