

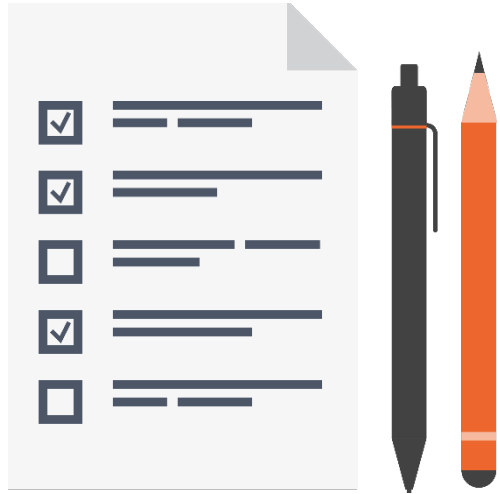
Understanding Xamarin Forms



Jim Wilson

@hedgehogjim | blog.jwhh.com

What to Expect in This Module



What is Xamarin Forms

What's good and what to be aware of

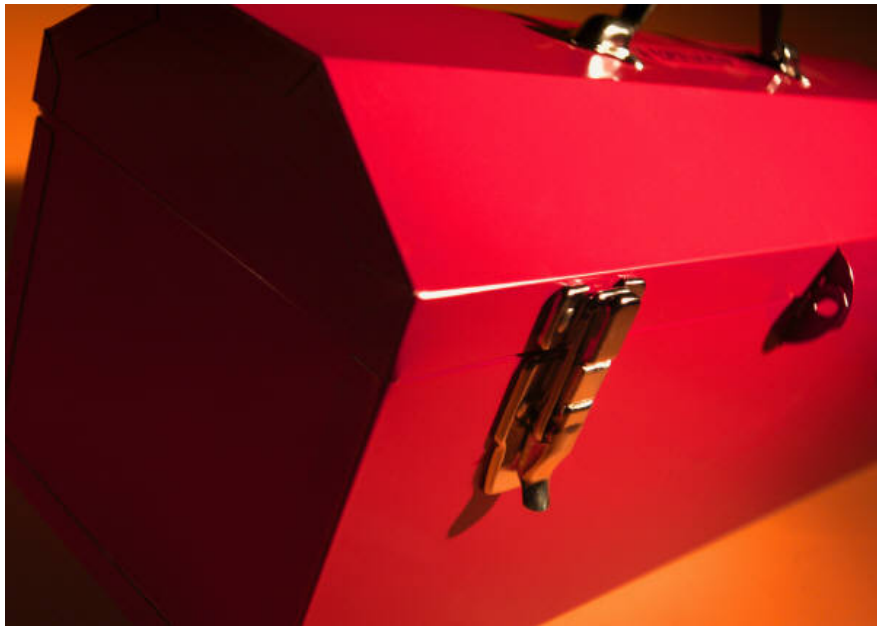
Xamarin vs. Xamarin Forms

Xamarin Forms app startup

Xamarin Forms component types

Mixing platform and shared behavior

Xamarin.Forms is mainly a UI Toolkit



Provides native UI abstraction

Common model across platforms

Uses native components at runtime

Maintains native look of the platform

Underlying platform remains accessible

Xamarin.Forms can interact with platform

Helps with cross-platform code sharing

Simplifies mixing platform and shared behavior

Xamarin.Forms Apps Are Native Apps



You build and deploy a native app to each platform

Requires a specific solution structure

Each platform has it's own project
Cross-platform shared code is in a separate project

What's Good and What to Be Aware of

What's good...

- UI code can be shared across platforms
- Simplifies mixing platform and shared behavior
- Leverage C# knowledge and skill
- Can get started with limited knowledge of underlying platforms

What to be aware of...

- You will need to eventually understand the underlying platforms
- Effective cross-platform development is challenging
- No Xamarin.Forms designer is available (*as of January 2015*)



Xamarin vs. Xamarin.Forms

Xamarin

.NET environment for
iOS, Android, and Mac

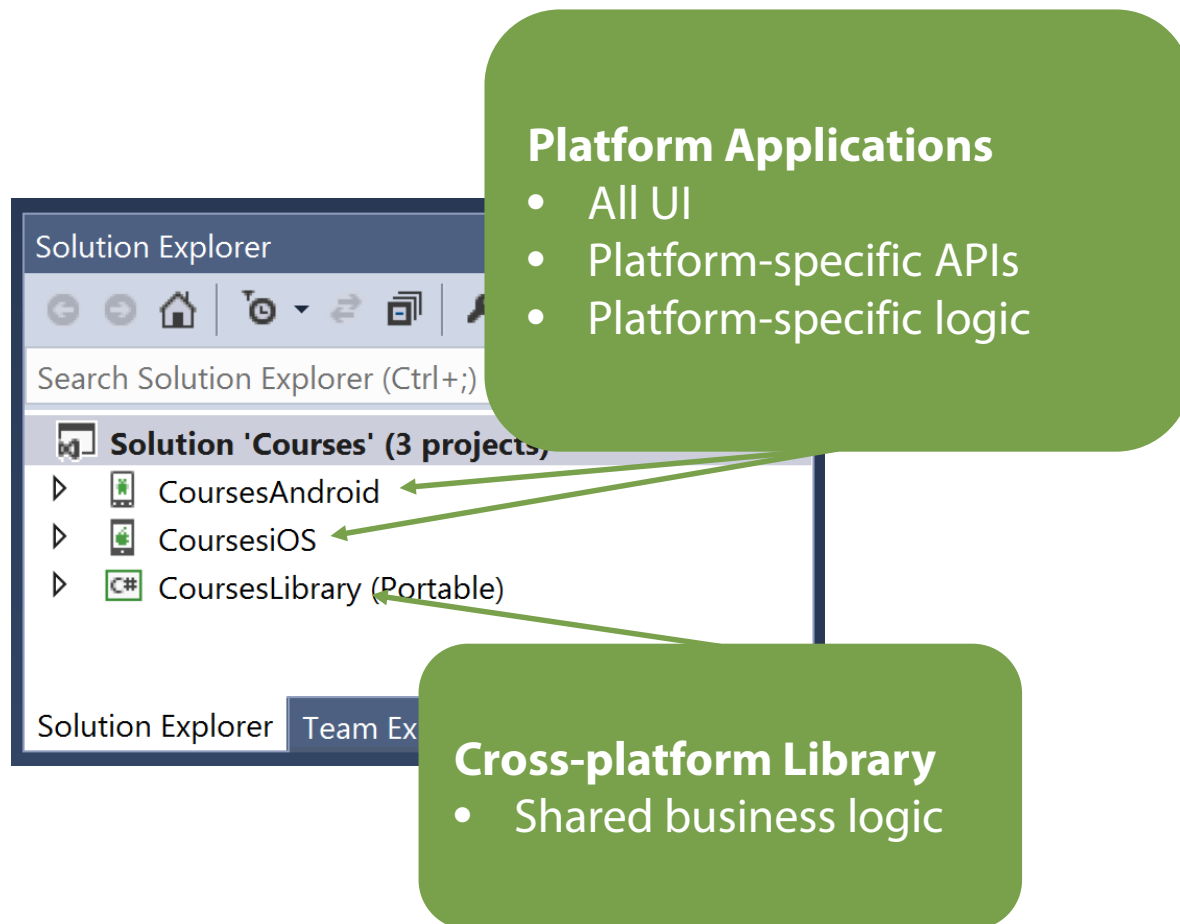


Xamarin Forms

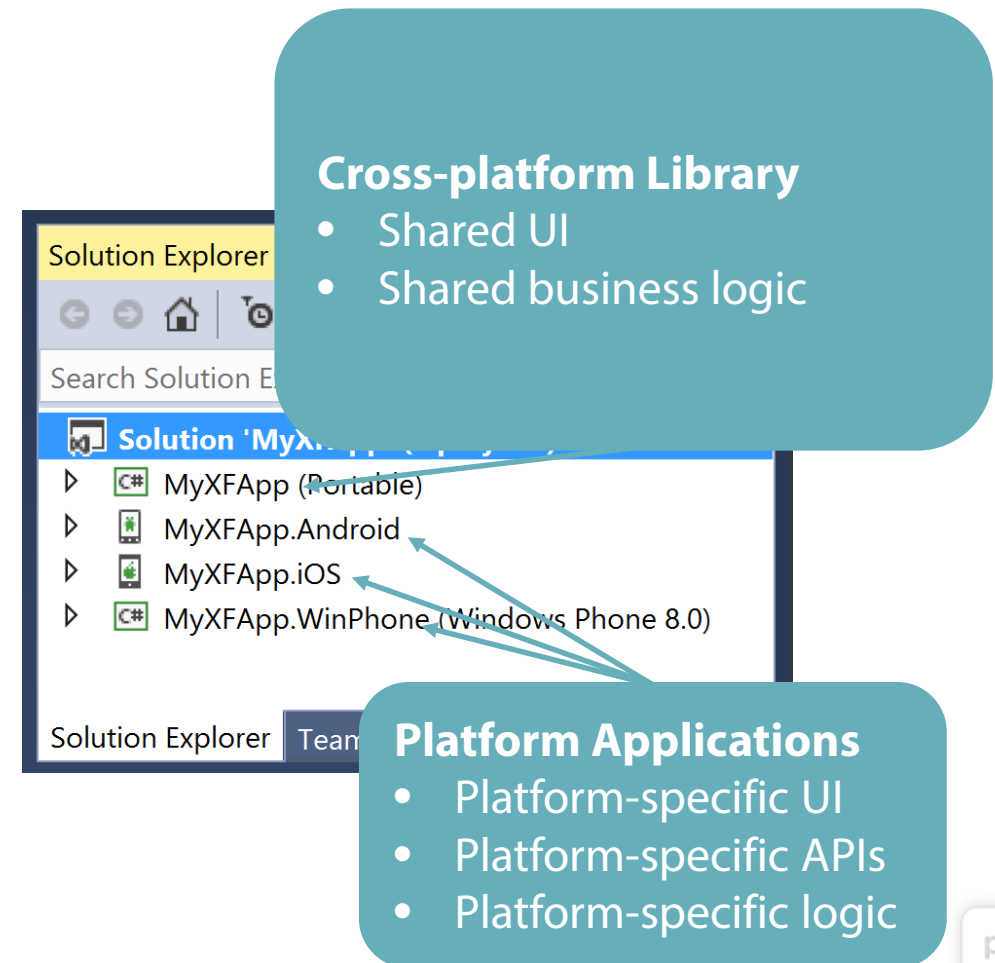
.NET based UI toolkit
for iOS, Android, and
Windows Phone

Xamarin vs. Xamarin.Forms Solution Structure

Xamarin



Xamarin.Forms





Xamarin Forms App Startup

- Each Platform app must provide a platform-specific class
 - Inherits from a Xamarin Forms type specific to that platform
 - Initializes Xamarin.Forms
 - Displays the Form definition from the shared project

iOS App Startup

```
public partial class AppDelegate : UIApplicationDelegate
{
    UIWindow window;

    public override bool FinishedLaunching(UIApplication app, NSDictionary options)
    {
        Forms.Init();
        window = new UIWindow(UIScreen.MainScreen.Bounds);
        window.RootViewController = App.GetMainPage().CreateViewController();
        window.MakeKeyAndVisible();
        return true;
    }
}
```

Platform-specific base class

Initialize Xamarin Forms

Display form definition

Android App Startup

```
public class MainActivity : AndroidActivity
{
    Platform-specific
    base class
    protected override void OnCreate(Bundle bundle)
    {
        base.OnCreate(bundle);
        Initialize
        Xamarin Forms
        Xamarin.Forms.Forms.Init(this, bundle);
        SetPage(App.GetMainPage());
        Display form
        definition
    }
}
```

Windows Phone App Startup

```
public partial class MainPage : PhoneApplicationPage
{
    public MainPage()
    {
        InitializeComponent();
        Forms.Init();
        Content = MyXFApp.App.GetMainPage().ConvertPageToUIElement(this);
    }
}
```

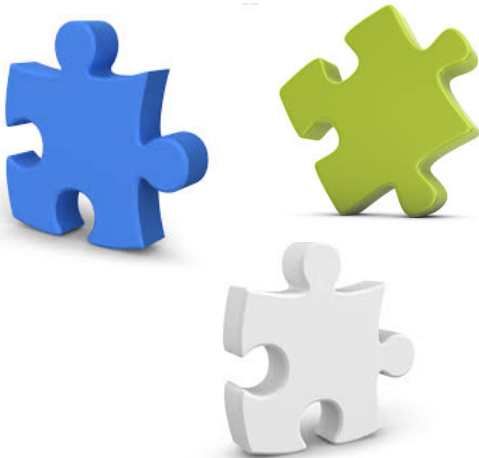
Platform-specific base class

Initialize Xamarin Forms

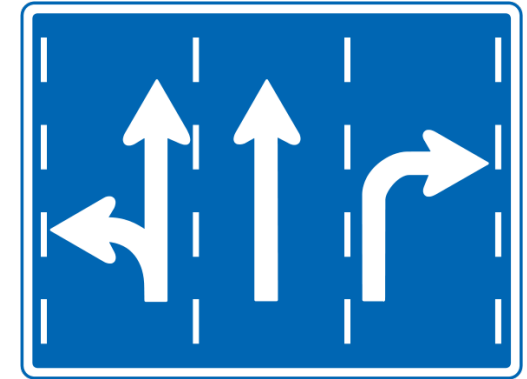
Display form definition

Xamarin.Forms Component Types

Views



Layouts

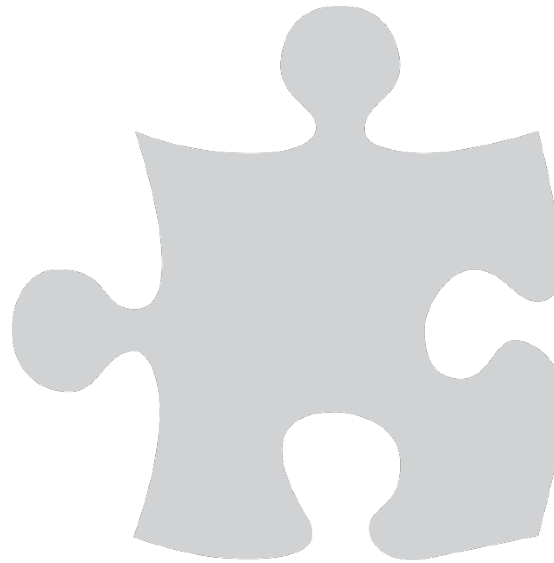


Pages



Using Components to Create the UI

C# Source Code



XAML

C# Source Code

```
var label1 = new Label
{
    Text = "I Live Xamarin.Forms",
    Font = Font.SystemFontOfSize(20)
};

Content = new StackLayout
{
    Spacing = 10,
    Orientation = StackOrientation.Horizontal,
    Children = { label1 }
};
```

XAML

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  x:Class="IntroXF.MyFirstApp">

  <StackLayout Spacing="10"
    Orientation="Horizontal">

    <Label Text="I Live Xamarin.Forms"
      Font="20"
      x:Name="label1" />

  </StackLayout>
</ContentPage>
```

Mixing Platform and Shared Behavior

- Device class
 - Provides methods for accessing platform specific methods and values
- Message Center
 - Publish/Subscribe feature to allow loose coupling
- Dependency Service
 - Registration service for accessing platform-specific interface implementations



Summary

- Xamarin Forms is mainly a UI Toolkit
- Utilizes platform UI components
- Creates a separate program for each platform
- Apps require special startup behavior
- UI can be described with code or XAML
- Provides facilities to simplify mixing platform and shared behavior