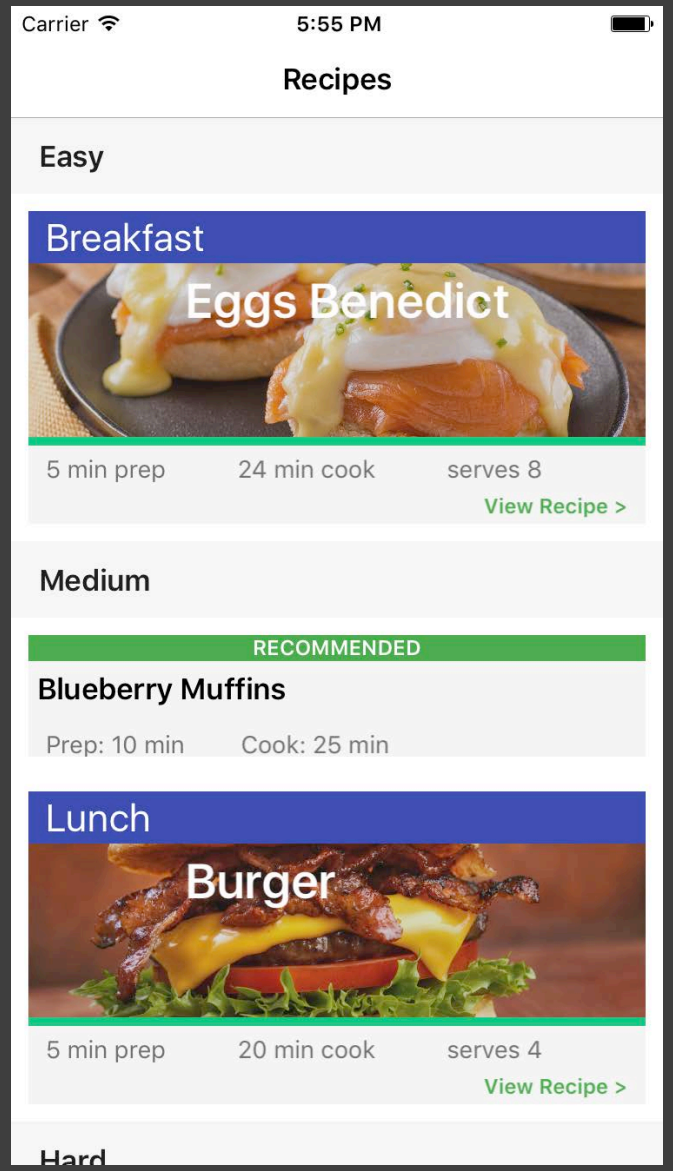
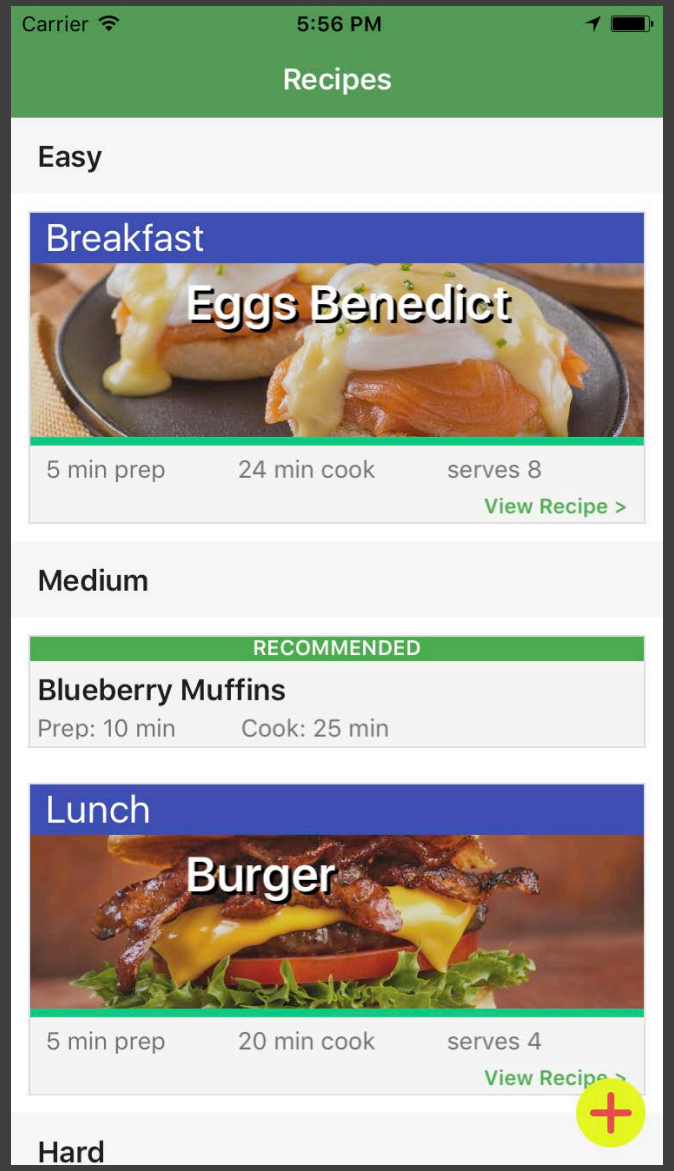
Accessing Native UI Elements

Matthew Soucoup
PRINCIPAL

@codemillmatt codemilltech.com

Native UI Enhancements





Native UI

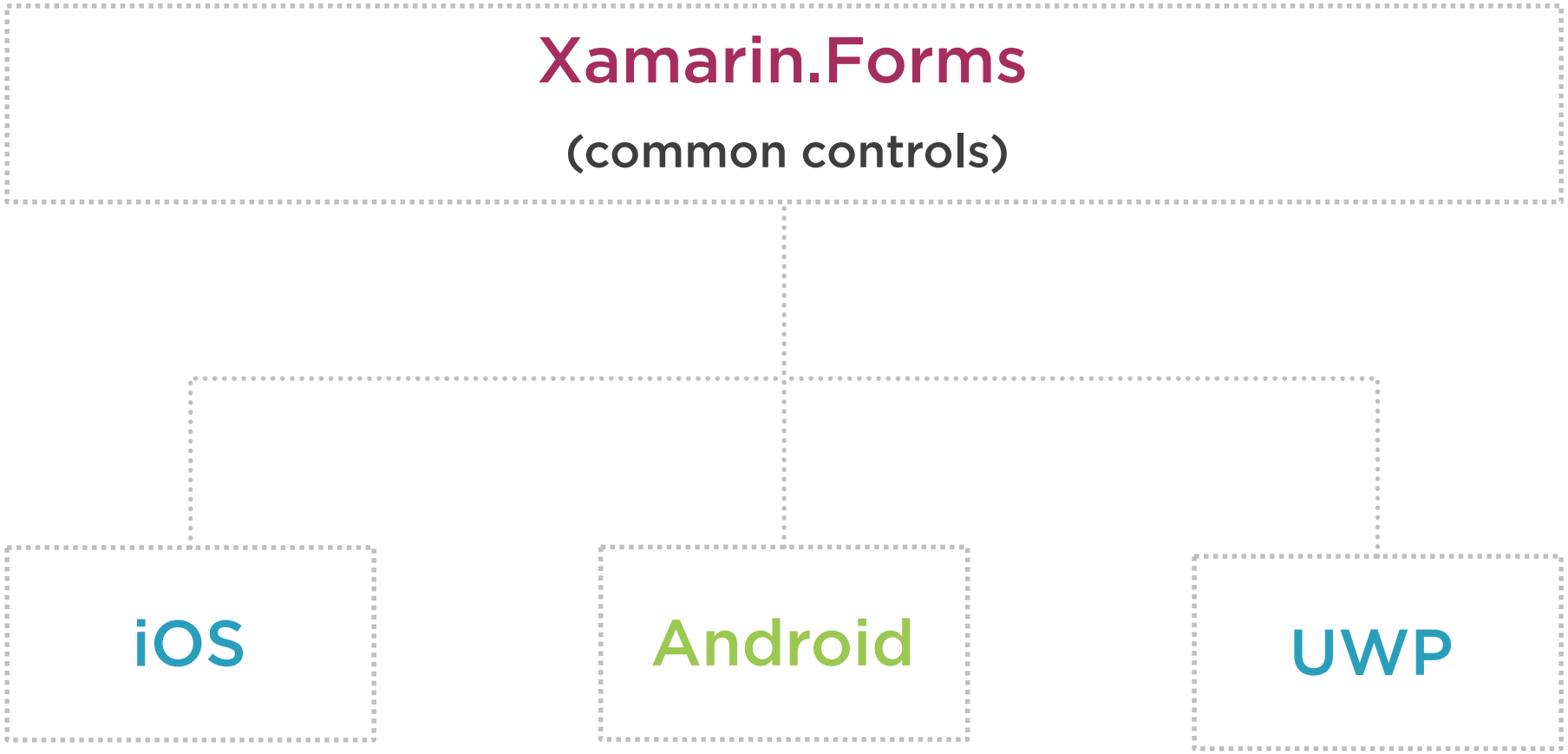
Modifying app appearance on platform

Working with platform specifics

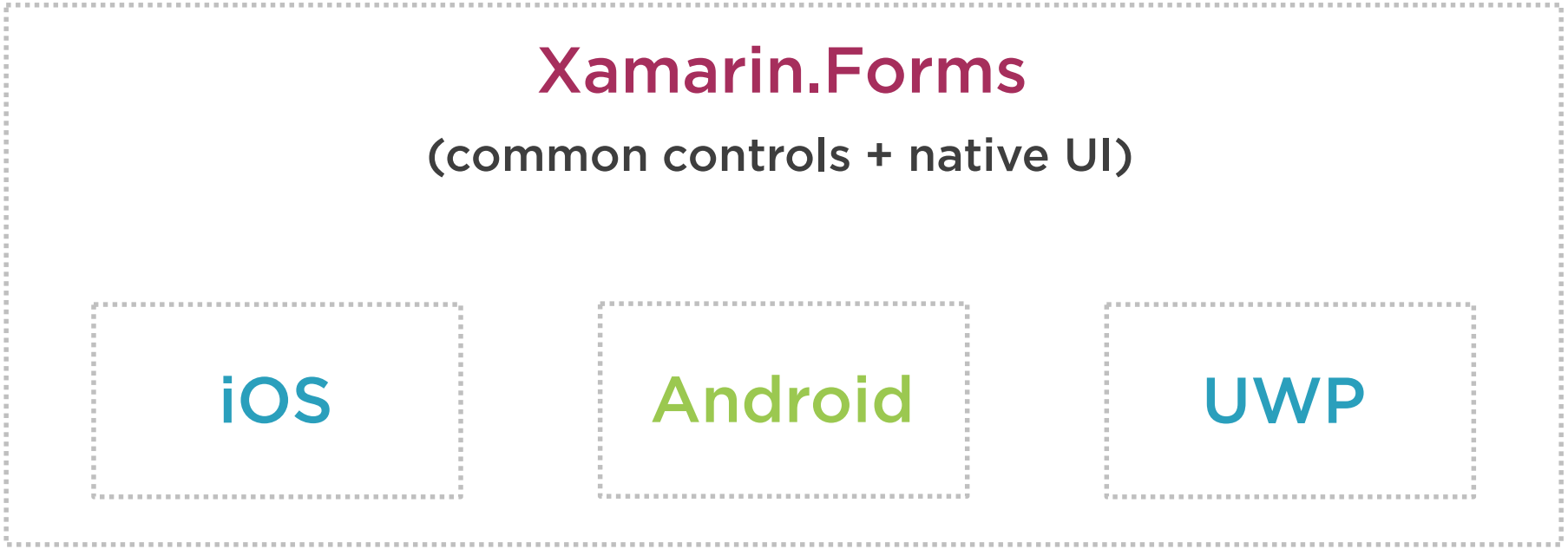
Creating effects

Embedding native views

Xamarin.Forms UI Abstraction



Xamarin.Forms with Native UI



Modifying App Appearance on Platform

```
public override bool FinishedLaunching(UIApplication app, NSDictionary options)
{
    UINavigationBar.Appearance.BarTintColor = UIColor.FromRGB(57, 141, 60);
    UINavigationBar.Appearance.TintColor = UIColor.White;
    UINavigationBar.Appearance.SetTitleTextAttributes(
                new UITextAttributes() { TextColor = UIColor.White });

    UISwitch.Appearance.OnTintColor = UIColor.FromRGB(57, 141, 60);
    UISwitch.Appearance.ThumbTintColor = UIColor.FromRGB(214, 216, 224);

    global::Xamarin.Forms.Forms.Init();

    LoadApplication(new App());
```

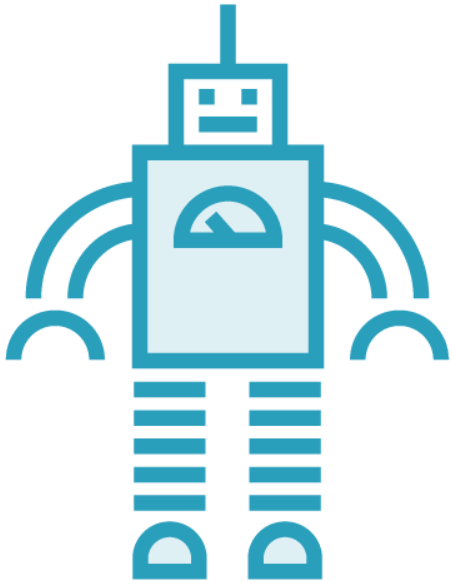
iOS App-wide Appearance

AppDelegate

FinishedLaunching

UIAppearance

Android App-wide Appearance



FormsAppCompatActivity

- Specify a theme

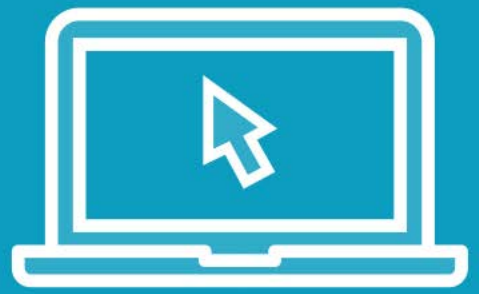
Implement in Styles.xml

- Resources/values directory

Styles.xml

```xml
<resources>
    <style name="MyTheme" parent="MyTheme.Base">
    </style>
    <style name="MyTheme.Base" parent="Theme.AppCompat.Light.DarkActionBar">
        <item name="windowNoTitle">true</item>
        <item name="windowActionBar">false</item>
        <item name="colorPrimary">#398D3C</item>
        <item name="colorPrimaryDark">#006400</item>
        <item name="colorAccent">#3F51B5</item>
        <item name="android:textColorPrimary">#212121</item>
        <item name="android:textColorSecondary">#757575</item>
        <item name="colorControlNormal">#BDBDBD</item>
        <item name="colorButtonNormal">#BDBDBD</item>
        <item name="windowActionModeOverlay">true</item>
    </style>
    <style name="AppCompatDialogStyle" parent="Theme.AppCompat.Light.Dialog">
        <item name="colorAccent">#3F51B5</item>
    </style>
</resources>
```

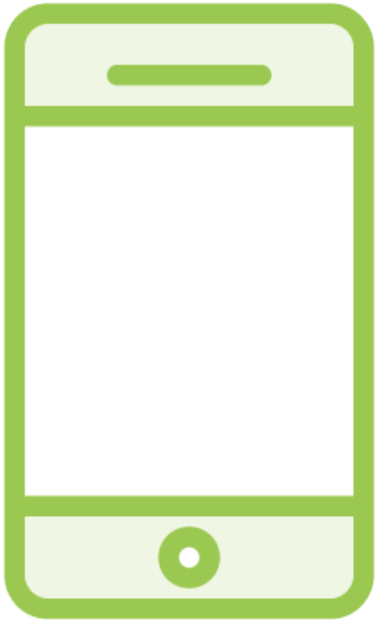
Demo



Add app-wide colors in iOS

Add app-wide colors in Android

Platform App Appearance Summary



Can change overall app appearance in platform projects

iOS

- ApplicationDelegate
- UIAppearance

Android

- FormsAppCompatActivity
- Styles.xml

Platform Specifics

Platform Specifics

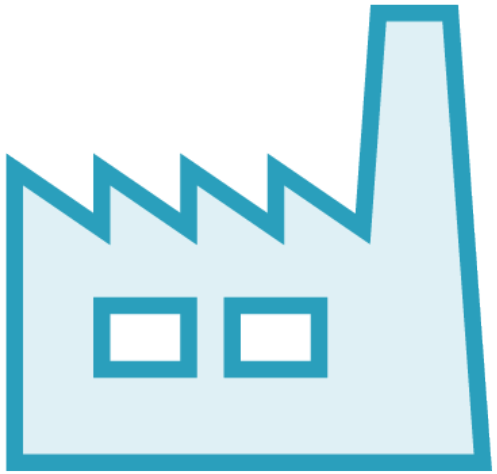
Access platform-specific code in core project

No need for developer to do extra work to access platform

Several built-in to Xamarin.Forms

XAML or C#

Xamarin.Forms Built-in Platform Specifics



Android

- AdjustResize and AdjustPan for soft keyboard layout

iOS

- Blur
- Translucent navigation bar

Windows

- Change toolbar placement
- Collapse master/detail page

```
<ContentPage
    xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:local="clr-namespace:Foodie;assembly=Foodie"
    xmlns:iOS="clr-namespace:Xamarin.Forms.PlatformConfiguration.iOSSpecific;
        assembly=Xamarin.Forms.Core"

    <Image Source="{Binding ImageName, Converter={StaticResource imgConvert}"
        iOS:VisualElement.BlurEffect="Dark"
        Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="3" Grid.RowSpan="3" />
```

Consuming an iOS Platform Specific

XML namespace

Attached property

Carrier 4:43 AM

Recipes **Details** Edit

5 min prep | 24 min cook | serves 8

Breakfast

** Easy difficulty **

Ingredients

1 tablespoon white vinegar
8 large eggs
Hollandaise Sauce
1/2 pound (16 slices) Canadian bacon
4 English muffins, split in half, toasted

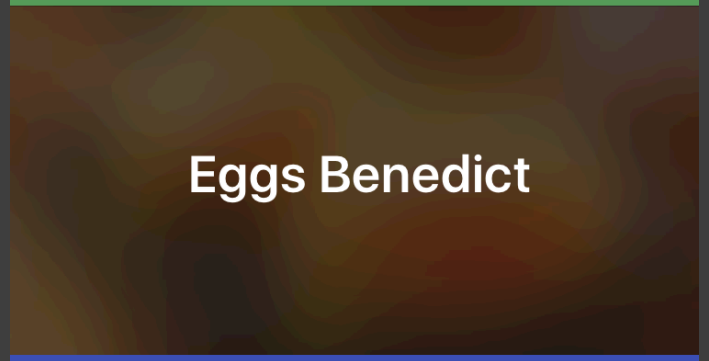
Directions

1. Fill a large saucepan with about 4 inches of water, add vinegar, and bring to a boil. Fill a shallow dish or pie plate with warm water. Reduce heat under saucepan to medium, so water is just barely simmering. Break 1 egg at a

Make Again!

Carrier 4:43 AM

Recipes **Details** Edit



5 min prep | 24 min cook | serves 8

Breakfast

** Easy difficulty **

Ingredients

1 tablespoon white vinegar
8 large eggs
Hollandaise Sauce
1/2 pound (16 slices) Canadian bacon
4 English muffins, split in half, toasted

Directions

1. Fill a large saucepan with about 4 inches of water, add vinegar, and bring to a boil. Fill a shallow dish or pie plate with warm water. Reduce heat under saucepan to medium, so water is just barely simmering. Break 1 egg at a

Make Again!

Platform Specifics Summary

Means to tweak targeted platform specific UI in core project

To use

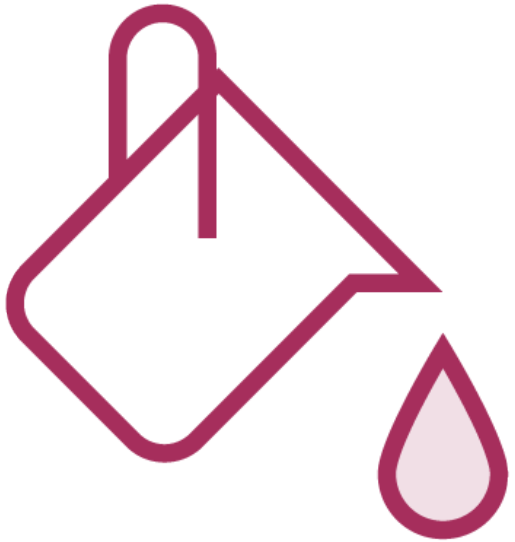
- Import namespace
- Reference attached property

Several built-in to Xamarin.Forms

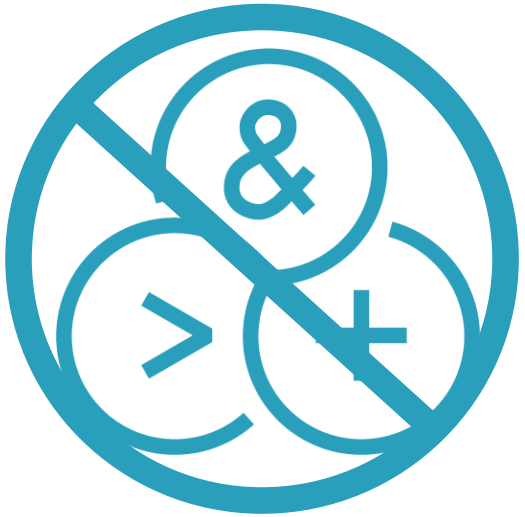
Usually provided by community or vendors

Effects

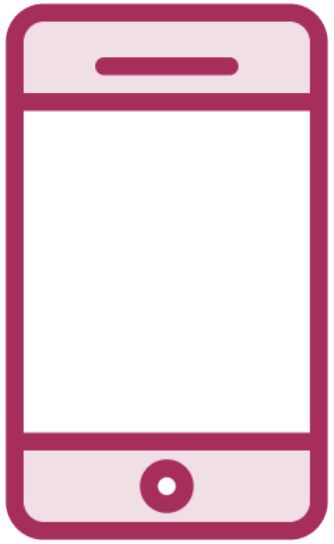
Effects



Small tweaks to the UI



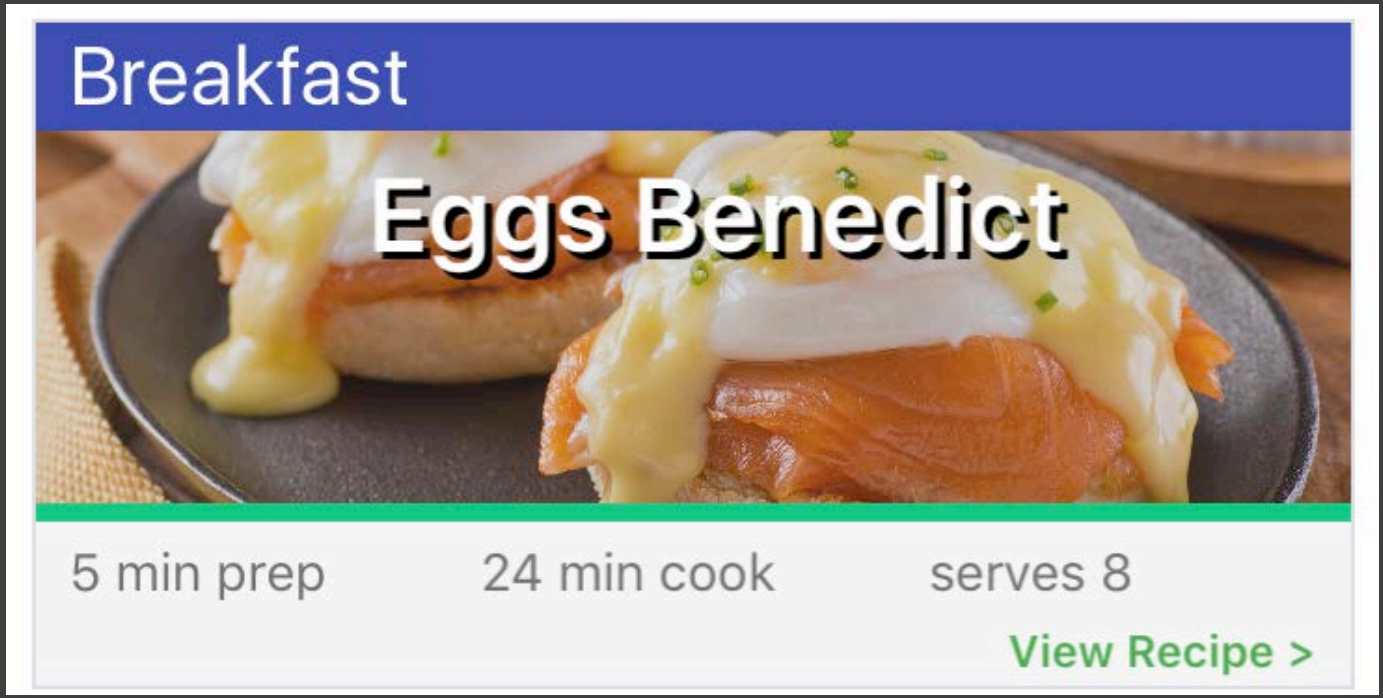
Not meant for behavior changes



Written in platform projects

Consumed in core project

Shadow Effect



Subclass PlatformEffect

Override

- OnAttached
- OnDetached
- OnElementPropertyChanged

Properties

- Element
- Control
- Container

Effect Lifecycle

OnAttached()

Perform any setup
Save pre-existing state
Added to any control

OnDetached()

Perform cleanup
Unsubscribe from events
Revert to initial state

OnElementPropertyChanged()

React to changes in Xam.Forms state
PropertyChangedEventArgs
Indicates property name

Properties of PlatformEffect

Container

Platform control that performs layout of rest

Control

Holds platform equivalent of Forms control

Element

Forms control

Register an Effect

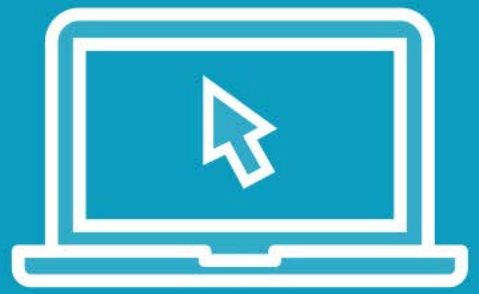
ResolutionGroupName attribute

- One per assembly
- Namespace to avoid collisions

ExportEffect attribute

- Registers a name for the effect

Demo



Create an effect

Register the effect

Consume the effect

```
public class ShadowEffect : RoutingEffect
{
    public ShadowEffect(): base("CodeMill.ShadowEffect")
    {
    }
}
```

Consuming effects

RoutingEffect

Constructor

- Concatenate ResolutionGroupName + effect name

Effect not needed on all platforms

```
xmlns:local="clr-namespace:Foodie; assembly=Foodie"
...
<Label Text="{Binding RecipeName}"
    Grid.Row="1" Grid.Column="1" Grid.ColumnSpan="2" >
    <Label.Effects>
        <local:ShadowEffect />
    </Label.Effects>
</Label>
```

Consuming Effects

Reference namespace

Add to Effects collection

New instance per control

Effects Summary

Use for small UI changes on platforms

Not meant for behavior changes

Subclass PlatformEffect in platform project

- OnAttached, OnDetached, OnElementPropertyChanged
- Element, Container, Control
- ResolutionGroupName and ExportEffect assembly attributes

Consume with RoutingEffect

- Effects do not need to be in all projects

Embedding Bindable Native Views

Bindable Native Views

Pure native view referenced in core project's XAML

Properties in native view bindable to view model

Core project can handle native view events

Invoke constructors and factory methods

```
<ContentPage

...

xmlns:foodieIOS="clr-
namespace:Foodie.iOS;assembly=Foodie.iOS;targetPlatform=iOS"

...

<foodieIOS:FoodieFab AbsoluteLayout.LayoutBounds="1,1,5,5"
    AbsoluteLayout.LayoutFlags="PositionProportional" />
```

Consuming on iOS

Native view already created

Reference native view namespace and assembly in xmlns

New targetPlatform attribute

```
<ContentPage
...

xmlns:foodieDroid="clr-
namespace:Foodie.Droid;assembly=Foodie.Droid;targetPlatform=Android"
xmlns:formsDroid="clr-
namespace:Android.Widget;assembly=Mono.Android;targetPlatform=Android”

...

<foodieDroid:FoodieFab x:Arguments="{x:Static formsDroid:Forms.Context}"
   UseCompatPadding="true" AbsoluteLayout.LayoutBounds="1,1,AutoSize,AutoSize"
   AbsoluteLayout.LayoutFlags="PositionProportional" />
```

Consuming on Android

Reference native view namespace and assembly in xmlns

Reference Mono.Android assembly in xmlns

Constructor requires Android Context class

```
<ios:UITextField Text="{Binding RecipeName}" />
```

Binding a Native View

Reference native property name

Same binding syntax as normal

```
<ios:UITextField Text="{Binding RecipeName, Mode=TwoWay,
    UpdateSourceEventName=Ended}" />
```

Two-way Binding with Events

Specify Mode=TwoWay

May need to specify UpdateSourceEventName

- Native event to update binding

Emedding Native View Gotchas

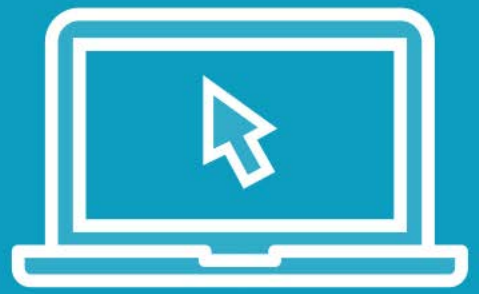
No XAML compiling

Cannot invoke functions on the native view

Cannot access in code behind

Cannot apply styles

Demo



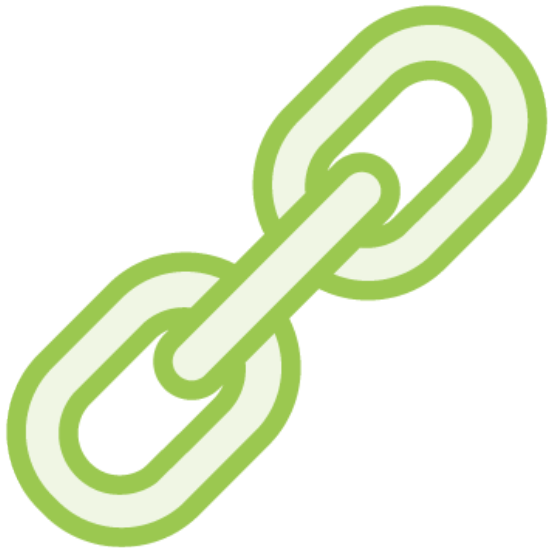
Load native view in XAML

Data bind

- One way
- Two way

Commanding

Bindable Native View Summary



- Can add native views to XAML
- Bindable
- Two-way binding with native events
- Commanding
- Gotchas to keep in mind

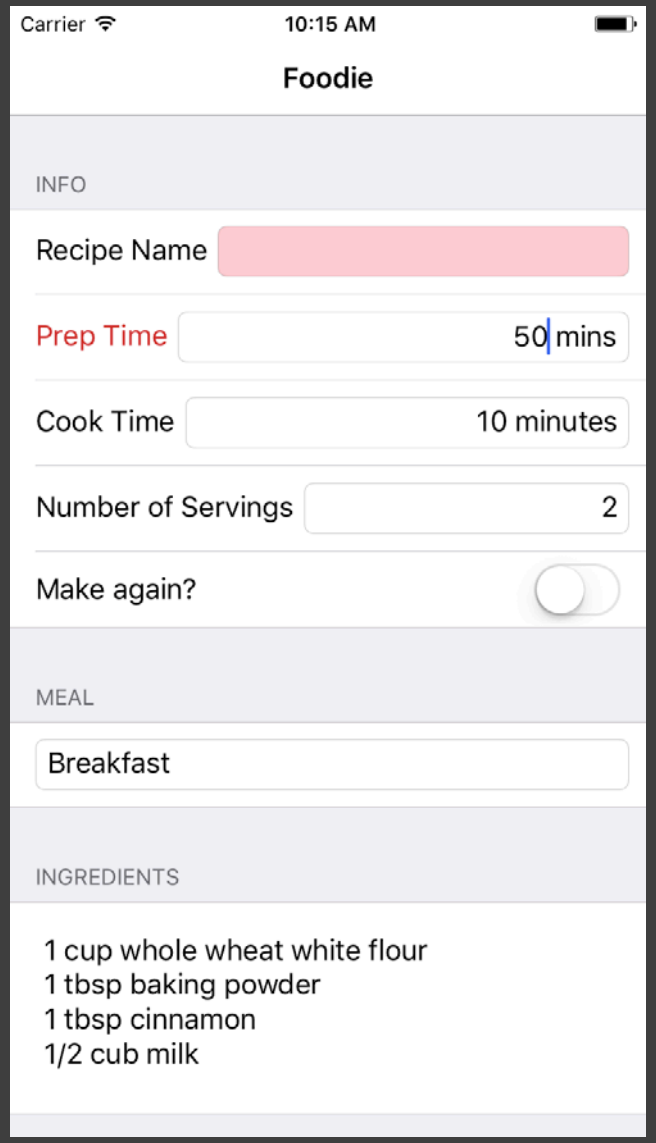
Tweak application-wide appearance in platform projects

Platform specifics easy way to change UI from pre-existing tools

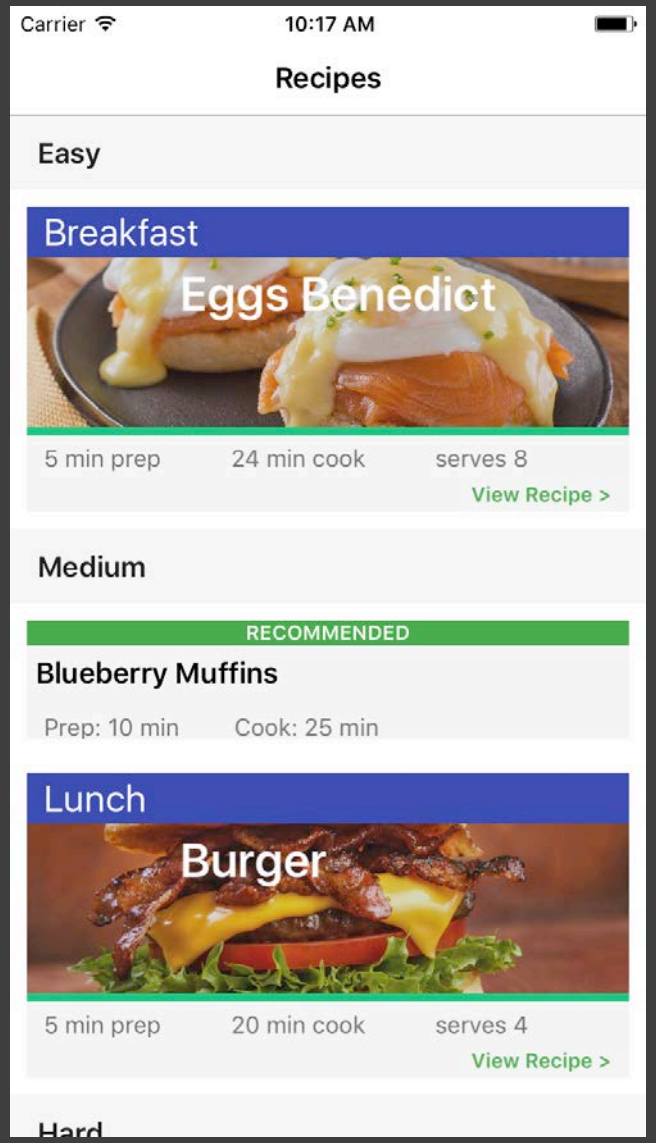
Effects change small UI properties

Embedding native views consume 100% platform specific bindable views in XAML

The Road Traveled – Entering Data



The Road Traveled – Displaying Data and Styles



The Road Traveled – Native UI