

A professional portrait of Adrian Stevens, a man with short brown hair, wearing a dark grey suit jacket, a white collared shirt, and a white tie. He is smiling and looking slightly to his right. The background is a blurred outdoor scene.

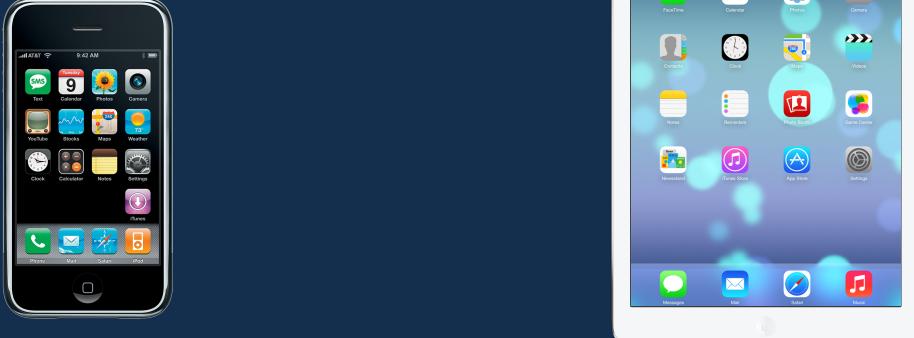
Xamarin Evolve 2014

# Dynamic Layouts in iOS 8: Making your apps look great everywhere

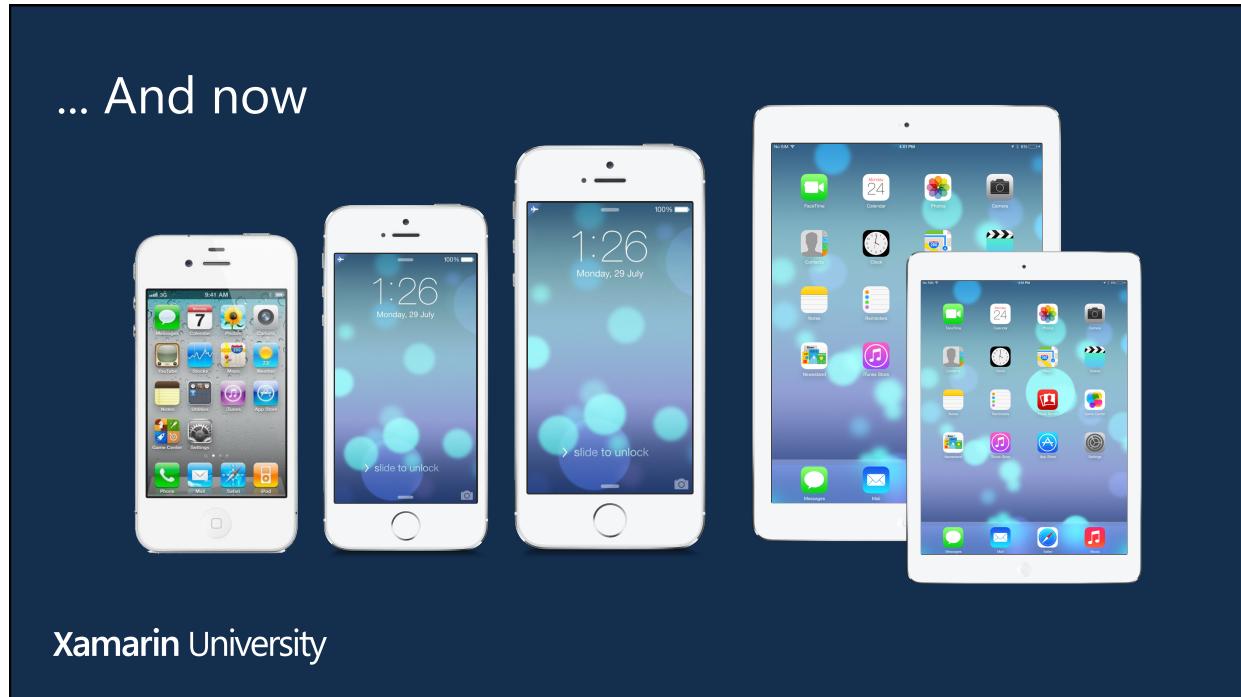
Adrian Stevens  
[adrian.stevens@xamarin.com](mailto:adrian.stevens@xamarin.com)

 **Xamarin**  
University

## In the beginning .....

Two devices are shown side-by-side against a dark blue background. On the left is a black iPhone displaying its home screen with various app icons. On the right is a white iPad displaying its home screen, which includes a larger central icon for the Mail app. Both screens show a similar set of standard iOS apps like Calendar, Photos, and Settings.

Xamarin University



Agenda

- Unified Storyboards
- Size Classes
- Trait Collections

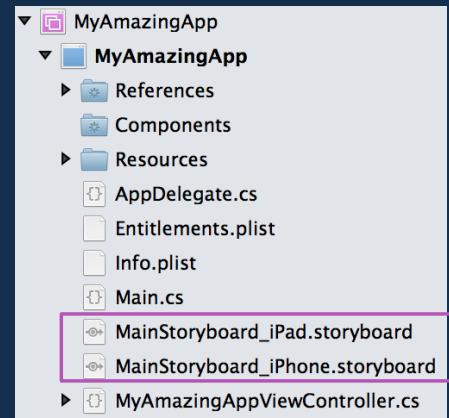
Xamarin University

Create your UI once  
and support all current  
and future iOS devices

Xamarin University

## < iOS8 Solutions

- Prior to iOS8, "**Universal**" apps would have a unique storyboard for **iPhone** and **iPad** to allow for differences in the form factor
- Monitor **orientation** change notifications for dynamic layout
- Use Layout **Constraints** to manage the different sizes within a single device class (3.5", 4", etc.)



Xamarin University

Xamarin University

## Introducing Unified Storyboards

- With iOS8, Apple is reversing course and instead using a **single storyboard** for all devices, they call this *unified storyboards*



The screenshot shows the Xamarin Studio environment. On the left is the Project Explorer with files like UVApp.csproj, UVApp, AppDelegate.cs, Info.plist, Main.cs, and MainStoryboard.storyboard. The MainStoryboard.storyboard file is highlighted with a red box. On the right is the Properties window with the 'Build' tab selected. Under 'iOS Application Target', the 'Devices' dropdown is set to 'Universal'. A pink arrow points from the highlighted storyboard file in the Project Explorer to this 'Devices' dropdown.

## Dealing with orientation changes

- When designing a layout, we have always been interested in two things:
  - What *type* of device are we dealing with (phone vs. tablet)?
  - What *orientation* is the device in (portrait vs. landscape)?
- But the *real* question we need the answer to is:

How big is my drawing surface?

Xamarin University

## Introducing "Size Classes"

Xamarin Evolve 2014

"Size classes for iOS 8 enable designing a single universal storyboard with customized layouts for both iPhone and iPad. With size classes you can **define** common views and constraints **once**, and then **add variations for each supported form factor**. iOS Simulator and asset catalogs fully support size classes as well."

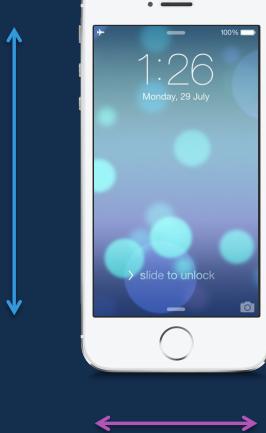
[What's New in Xcode 6](#)  
[Apple Documentation](#)



## Pixels vs Size Class

Pixels: 1136 high

Size Class: Regular Vertical



Pixels: 640 wide

Size Class: Compact Horizontal

Xamarin University

iPhone 5s

## What is it good for?

- Size Classes allow you to define your UI in a *single storyboard* where you have variations of the same UI defined for each supported size class
- If you want to have a *completely* different look for your iPhone 4 vs. iPhone 5 app, or you want to support iOS 7 or below, then size classes are not the solution



Xamarin University

## Size Class Definitions

- Content area is determined by how much space is available horizontally and vertically; each dimension can be one of two different values

Indicates that the specified dimension has more space available, either because the device can "scroll" or because it has more size

Regular

Compact

Compact indicates that the dimension is constrained and has limited viewing capability

Xamarin University

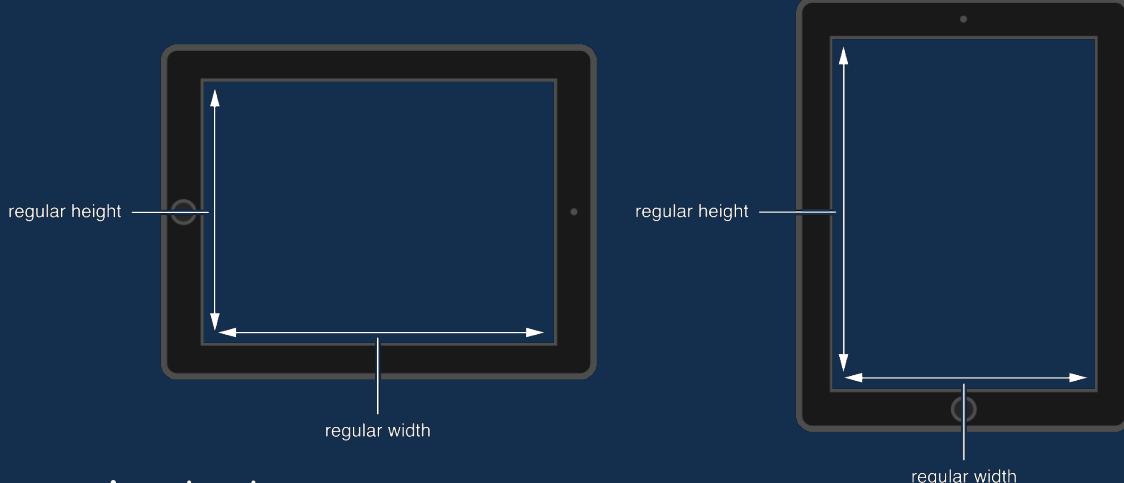
## Size Classes

Horizontal Size Classes

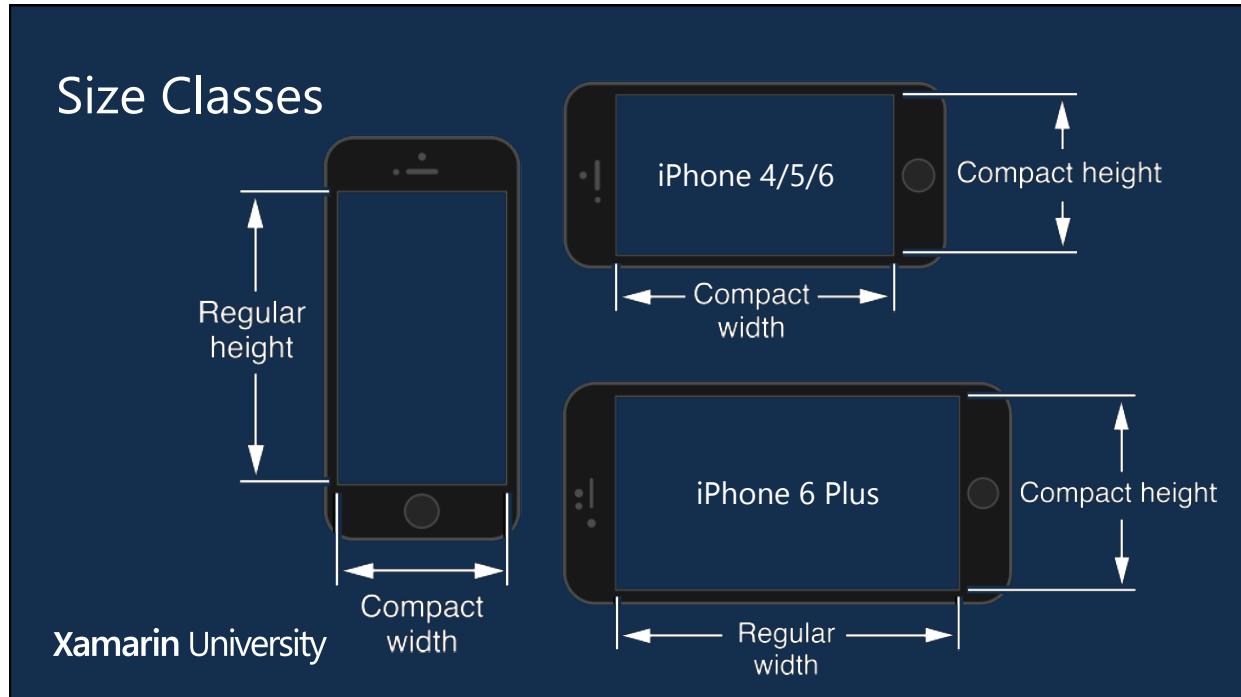
Vertical Size Classes	Regular	Compact
Regular	iPad Portrait & Landscape	iPhone Portrait
Compact	??	iPhone Landscape

Xamarin University

## Size Classes – iPad is always regular



Xamarin University



## Size Classes

Horizontal Size Classes

	Regular	Compact
Regular	iPad Portrait & Landscape	iPhone Portrait
Compact	iPhone 6+ Landscape	iPhone Landscape

Xamarin University

# Demo

Xamarin University

## User Interface Traits

- UI definition is based on *traits* – these define how content and layout change as the environment changes

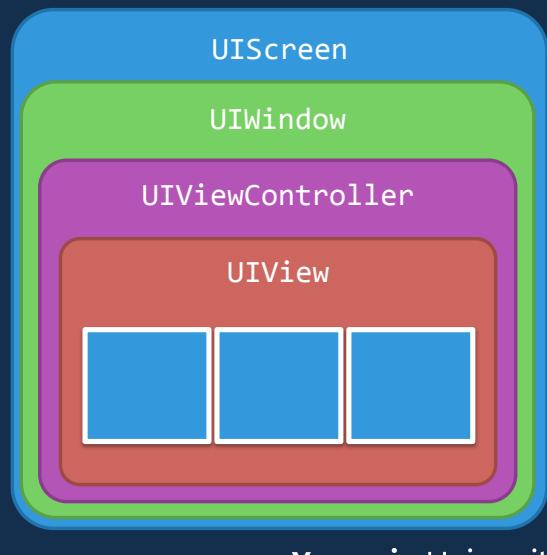
traits are contained in a UITraitCollection



Xamarin University

## Trait Collections

- Each UI component defines a trait collection, which together make up the *trait environment*
- Inherited from parent to child
- Can override `TraitCollectionDidChange` on view or view controller to handle changes in code, or use the designer



## Interacting with Traits

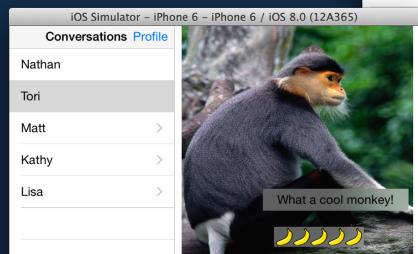
- **Trait Environments** have a `TraitCollection` property
- **TraitCollections** allow us to access the four individual traits in code

```
if(myViewController.TraitCollection.HorizontalSizeClass ==  
UIUserInterfaceSizeClass.Compact)  
{  
    ...  
}
```

Xamarin University

## Split View Controller Updated

- Master/Detail Interface
- Available on iPhone & iPad



Xamarin University

Demo

Xamarin University

## Designing UI with Traits

- Do not design to explicit screen sizes, instead UI should be designed for each supported device class



Landscape  
Phone



Portrait  
Phone



Landscape  
Tablet

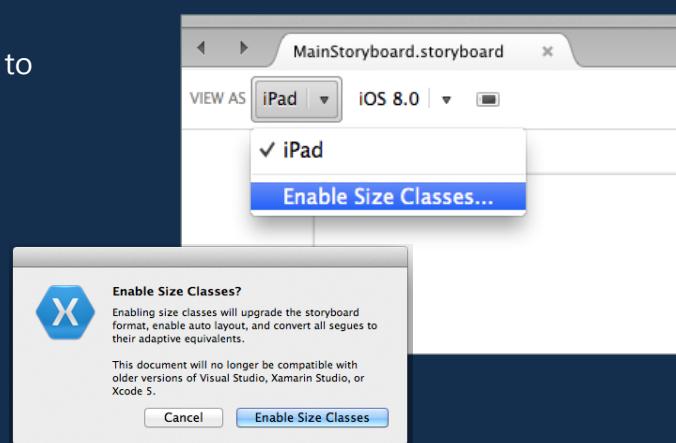


Portrait  
Tablet

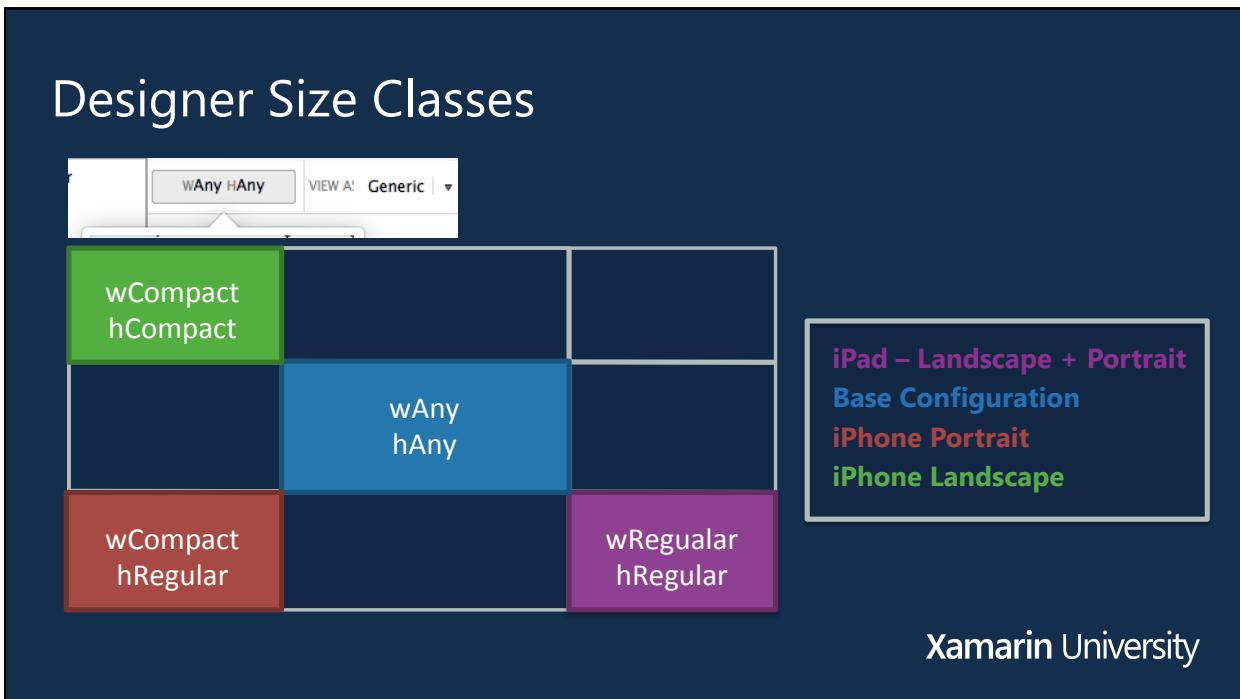
Xamarin University

## Xamarin Designer with Universal Storyboards

- Designer has been updated to support size classes
- Must target **iOS8**
- Set Storyboard to Universal
- Enable Size Classes in the Storyboard toolbar



Xamarin University



## Support for a New Screen Scale

- new **@3x** screen scale
- Currently for iPhone 6 Plus only
- 3x assets will be loaded automatically when using **UIImage.FromFile** & **UIImage.FromBundle**



```
UIImage.FromBundle("icon.png");
```

Xamarin University

## UIImage and Image Resources

@1x



@2x



@3x



Xamarin University

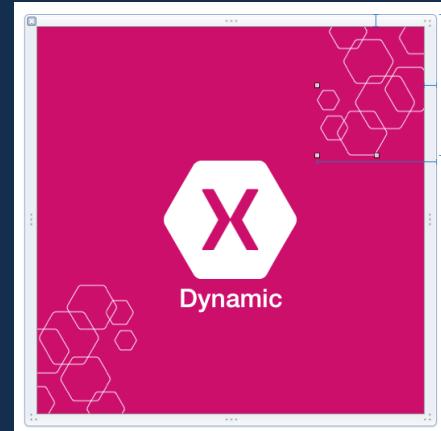
## UIImage and Image Resources



Xamarin University

## Dynamic Launch Screens

- Uses UIKIT classes
- Use a root view (UIView or UIViewController)
- Independent of app code (no Actions, Outlets, etc.)
- Uses **Auto Layout & Size Classes**
- Works with every device, resolution and orientation



Xamarin University

## Summary

- Unified Storyboards
- Size Classes
- Trait Collections
- Amazing Dynamic UIs

Xamarin University

A portrait of Adrian Stevens, a man with short brown hair, wearing a grey suit jacket, white shirt, and light blue tie, smiling at the camera. He is positioned on the left side of the slide.

Xamarin Evolve 2014

## Dynamic Layouts in iOS 8: Making your apps look great everywhere

Adrian Stevens  
[adrian.stevens@xamarin.com](mailto:adrian.stevens@xamarin.com)

 **Xamarin**  
University