Mike Berlin
@mikeberlin
http://mike-berlin.com

Kansas City Mobile .NET Developers Group

Xamarin

Intro to WatchKit using Xamarin

Yay advertising!

Xamarin

The Xamarin devs r0x0r!

Xamarin

March 6, 2015

Like Clockwork, Xamarin Releases Watch Kit Support

By Miguel de Icaza

We're very excited to announce official support for Apple's just-released Watch Kit.

To start building Watch apps with Xamarin immediately:

- Download the official Xcode 6.2 release from Apple.
- Update **Xamarin.iOS** to the latest version.

What you can do

Xamarin

- Three primary Apple Watch interactions
  - WatchKit Application
  - Glances
  - Actionable Notifications
- Digital crown (scrolling only at the moment)
- Tap gestures
- Annnnnnd…

Force ~~Choke~~ Touch

* http://developer.xamarin.com/guides/ios/watch/controls/menu/

WatchKit Apps

Xamarin

- Full screen user interface
- Able to launch and fully interact with UI
- One or more screens

Glances

- Timely read-only information
- Quick and lightweight view of your app
- Does not scroll
- Tapping a glance launches your WatchKit app
- Think of it as a widget

Actionable Notifications

- Quick view information of your app
- Take action right from the notification
- Short-Look interface
  - Does not scroll
  - Can not be customized, system template
- Long-Look interface
  - Scrollable
  - Can use customized interface, but not required

What you can't do

Xamarin

- The App/Extension will not run without an active connection to an iPhone
- You cannot route notifications to the Watch only, or the iPhone only
- No access to the sensors and/or hardware on the Apple Watch
  - Taptic/Vibration
  - Sound
  - Digital Crown (scrolling only)
- The APIs that allow the Watch to control iTunes on the connected phone are private

What you can't do…continued

Xamarin

- There is no access to the display graphics context on the Watch
  - You can use Core Graphics to render a bitmap, but it is expensive
  - The Watch App has a 20MB image cache (this limit cannot be changed, and must be managed by the developer)
- Apple will not allow 3rd-party custom watch faces.
- Watch Kit Extensions cannot perform any processing after being deactivated
  - Any pending asynchronous functions (e.g., data loading) will not execute.
  - Watch Kit Extensions cannot use background execution

Really???

Xamarin

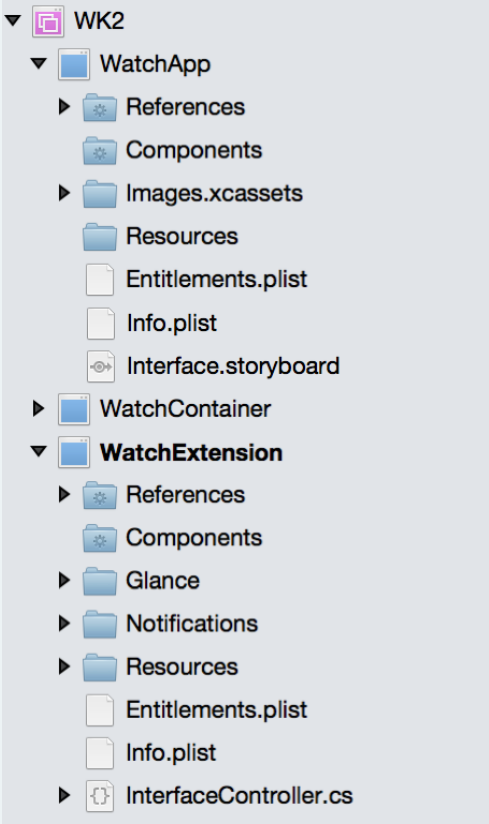
Trust Apple…more to come later in 2015

Xamarin

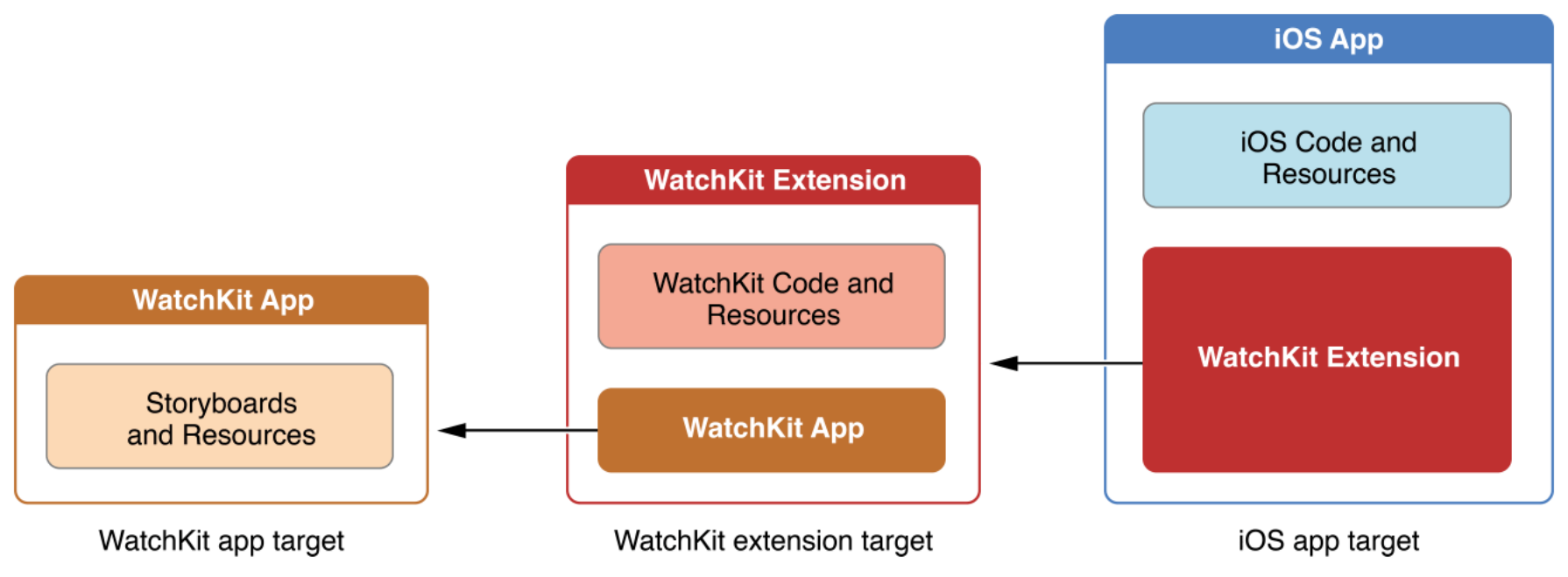
But really, come on now…

Solution Architecture

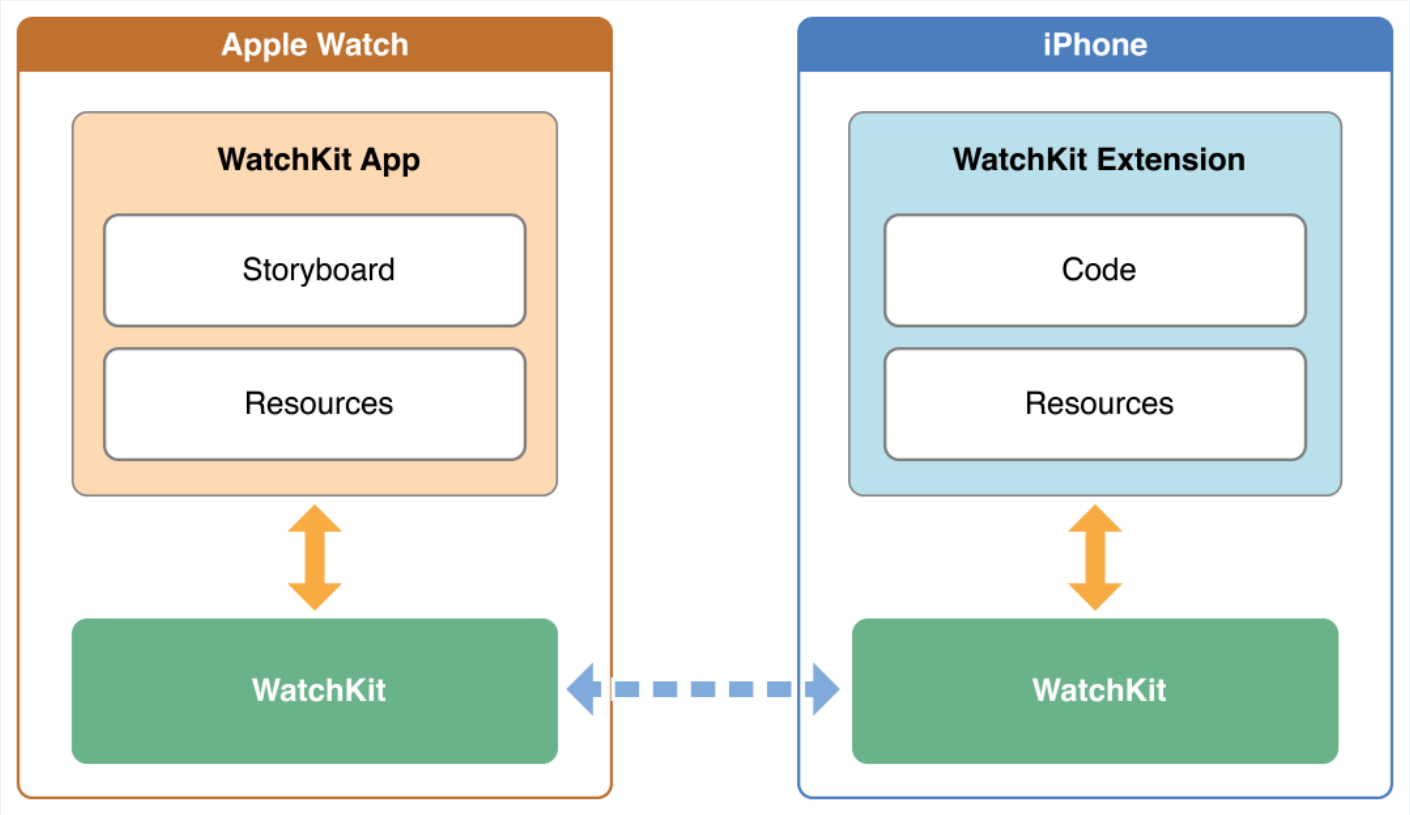
- Watch Container/Parent app
- Watch Extension
- Watch Application



Solution Architecture



Solution Architecture



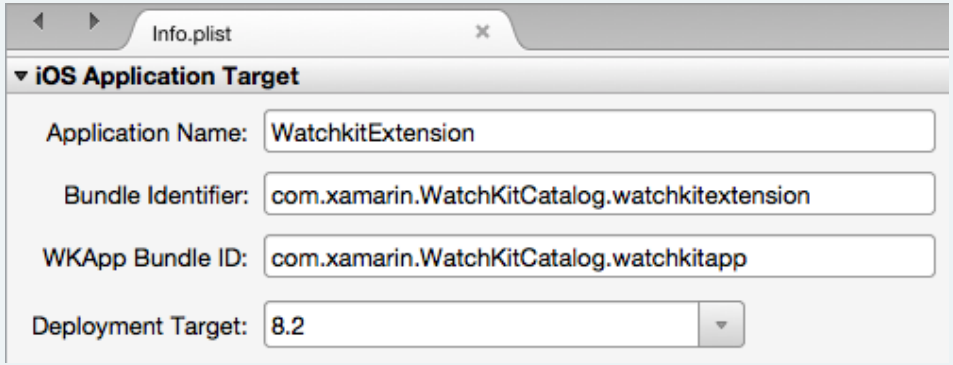
Solution Architecture

About Watch Kit

- iOS "Parent or Container" App
  - Watch Extension
  - Watch App (delivered to watch)
- Watch Extension and App communicate!
  - Code runs in Extension on iPhone
  - UI (storyboard, images) is loaded on the watch itself

Solution Architecture

- References
  - iPhone app references Watch **Extension**
  - Watch Extension references Watch **App**
- Bundle IDs



WKInterfaceController

Xamarin

- Serves the same purpose as a View Controller in iOS
  - Presents and manages content on the screen
  - Responds to user interaction
- Unlike a View Controller, it does not manage the View
  - Handled by WatchKit

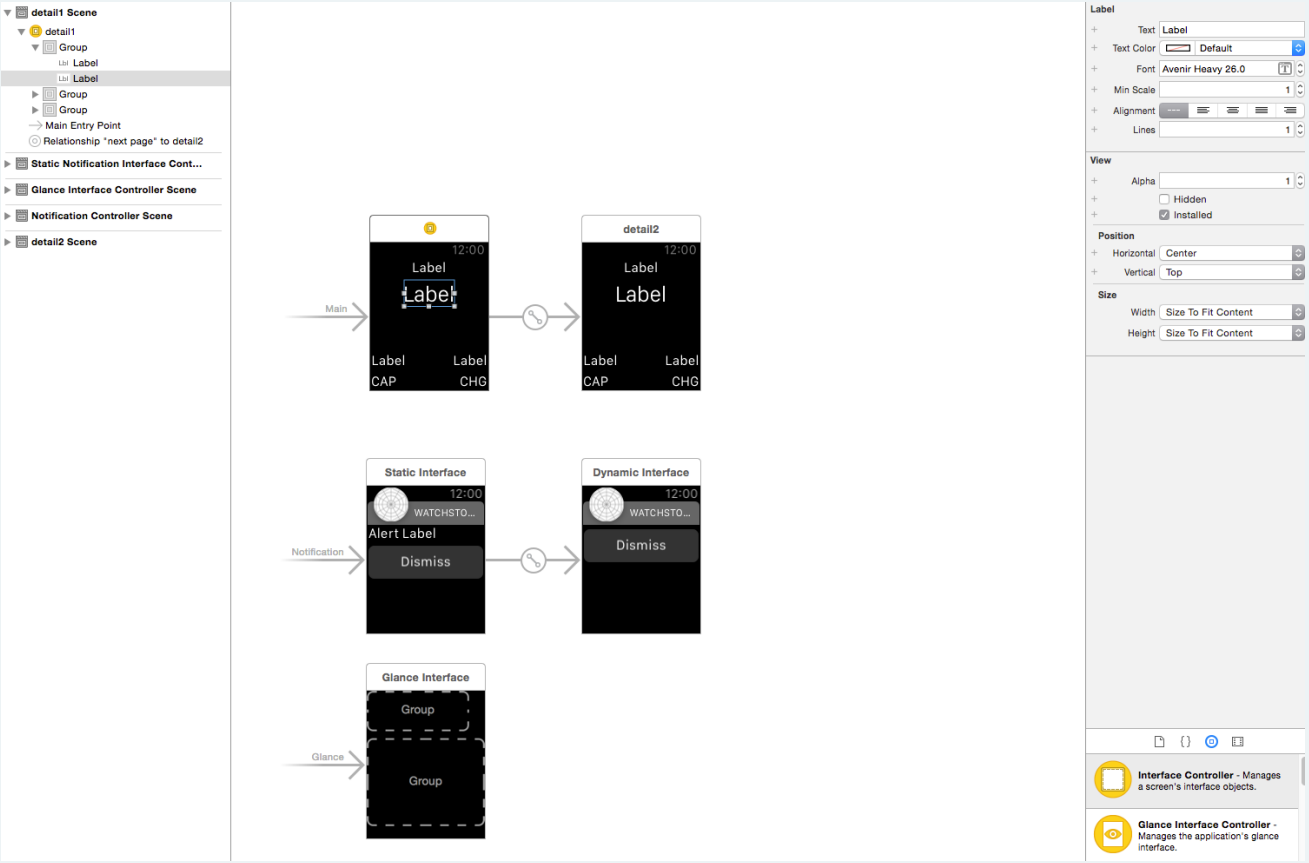
Leveraging iOS Technologies

Xamarin

- Use of some iOS technologies may be restricted
- Some general guidelines:
  - Avoid using technologies that request user permission (e.g. Core Location)
  - Do not use background execution modes for a technology
  - Avoid performing long-running tasks with a technology
    - Perform any long-running tasks on your iOS app

Creating the User Interface

Xcode-Beta Interface Builder



Xamarin.iOS Designer

Xamarin

Xamarin Studio File Edit View Search Project Build Run Version Control Tools Window Help

WatchApp - Interface.storyboard* - Xamarin Studio

Debug | iPhone 5s iOS 8.2 | Project saved. | Press '⌘.' to search

Unit Tests

Solution

- WatchBlog
  - WatchApp
    - References
    - Components
    - Resources
    - Entitlements.plist
    - Info.plist
    - Interface.storyboard
  - WatchBlog
  - **WatchExtension**
    - References
    - Components
    - Resources
    - Entitlements.plist
    - GlanceController.cs
    - Info.plist
    - InterfaceController.cs
    - ModalIC.cs
    - MyModalInterfaceController.cs
    - NotificationController.cs
    - PushedIC.cs
    - PushNotificationPayload.json

Interface.storyboard

Any Apple Watch | ZOOM

12:00
My First Watch
Push
Interface Contr...

12:00
Show Modal
Interface Contr...

12:00
Hello Watch!

Toolbox

Controls

- Button
- Date
- Group

Properties | Document Outline

Widget | Layout | Events

Identity

Class: MyModalInterfaceController
Localization ID: 60

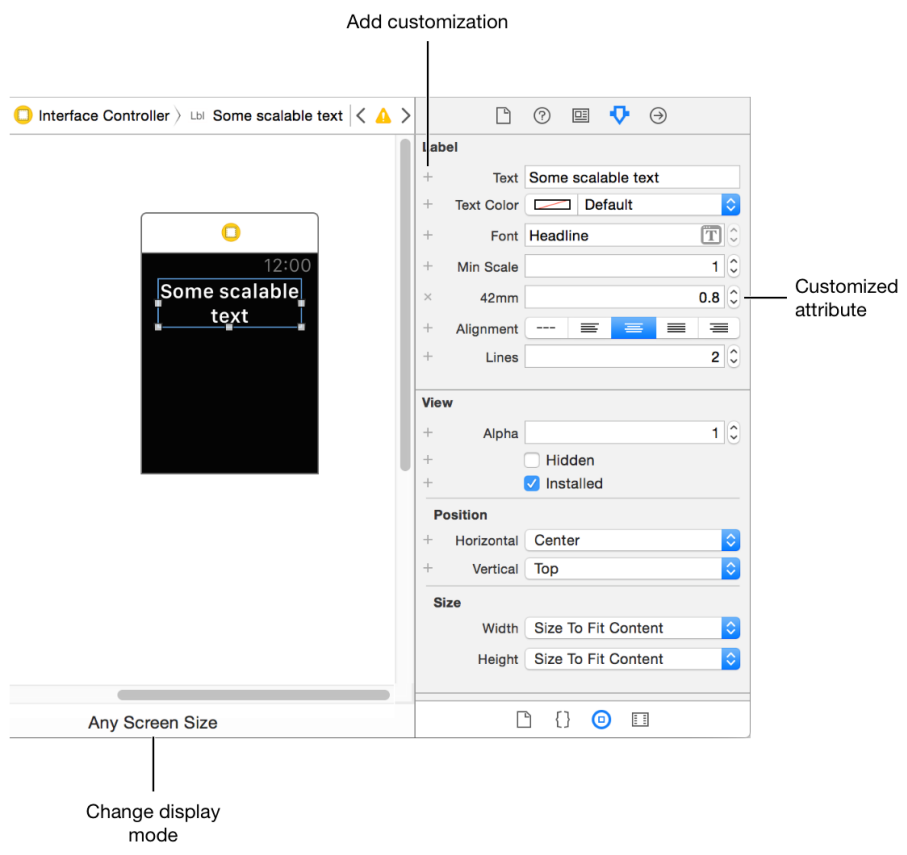
Interface Controller

Identifier:
Title:

Background: No Image
Mode: Scale To Fill
Animate: No
Color: Default

Errors | Tasks

Differences



Assembling Your Storyboard Scenes

Xamarin

- WatchKit does not use the same layout model as iOS
- Xcode positions the items for you
  - Stacking UI elements vertically on different lines
- At runtime, WatchKit takes UI elements and lays them out for you based on the available space
- Size and position of most items can be configured using the Attributes inspector
- Groups are the primary tool for positioning items (horizontal and vertical positioning)

Assembling Your Storyboard Scenes

Xamarin

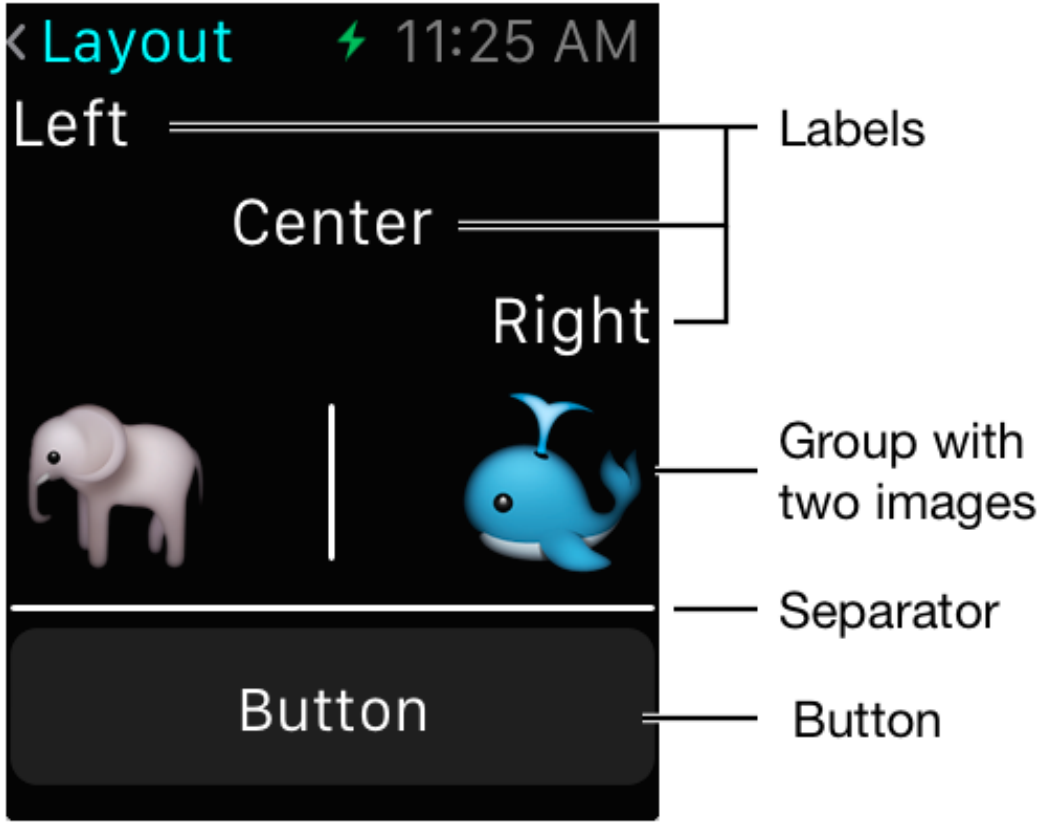
- *** IMPORTANT ***
- You cannot dynamically add objects at runtime
- You can hide objects
- All objects need to be created through the Interface Builder / Xamarin.iOS Designer
- An app can only have one Glance interface controller

Available Interface Elements

Xamarin

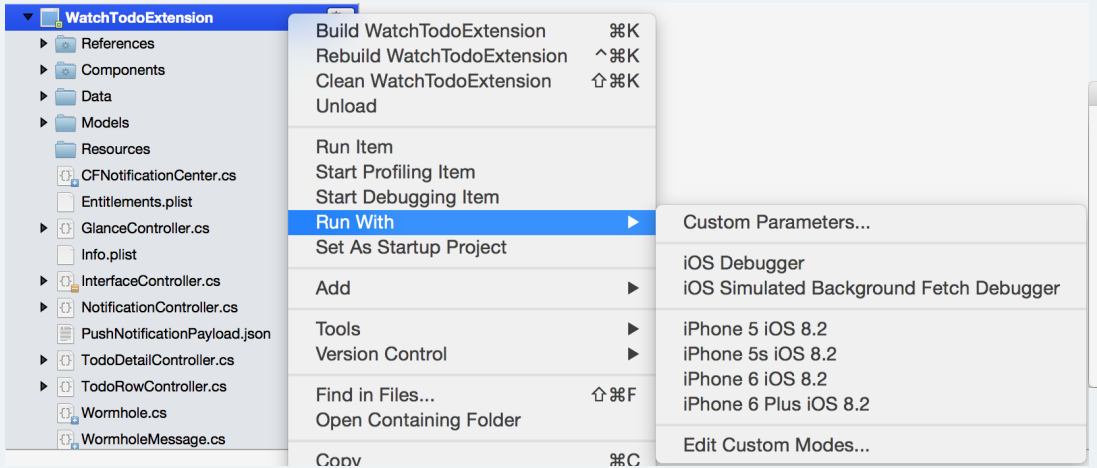
- You cannot subclass or customize interface elements
- These are the controls that are currently available to developers:
  - WKInterfaceLabel, WKInterfaceButton, WKInterfaceImage
  - WKInterfaceGroup, WKInterfaceSeparator, WKInterfaceTable
  - WKInterfaceSwitch, WKInterfaceMap, WKInterfaceSlider
  - WKInterfaceTimer

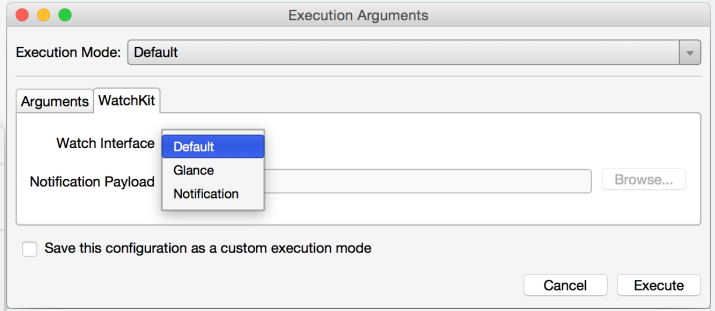
Assembling Your Storyboard Scenes



Launching in other modes

- Launching in Glance and Notification Modes
  - Run With -> Custom Parameters…





Xamarin

DEMO

Get to the dang code already!

Useful Links

Xamarin

- Apple's WatchKit Site - *THE* source (Programming Guide)
  - https://developer.apple.com/watchkit/
- Apple Watch Programming Guide
  - https://developer.apple.com/library/prerelease/ios/documentation/General/Conceptual/WatchKitProgrammingGuide/
- Craig Dunn's wealth of WatchKit knowledge
  - http://conceptdev.blogspot.com/2015/02/apple-watch-kit-round-up.html
- Xamarin's Developer Documentation
  - http://developer.xamarin.com/guides/ios/watch/intro-to-watchkit/
- Xamarin's Blog Post
  - http://blog.xamarin.com/xamarin-releases-watch-kit-support-like-clockwork/

Xamarin

Try Xamarin Today

Learn more

Xamarin

Xamarin University

Unrivaled Mobile Development Training

Live unlimited mobile development training from mobile experts, in your time-zone, on your schedule and as often as you'd like

Xamarin

Thank You

Mike Berlin
@mikeberlin
http://mike-berlin.com