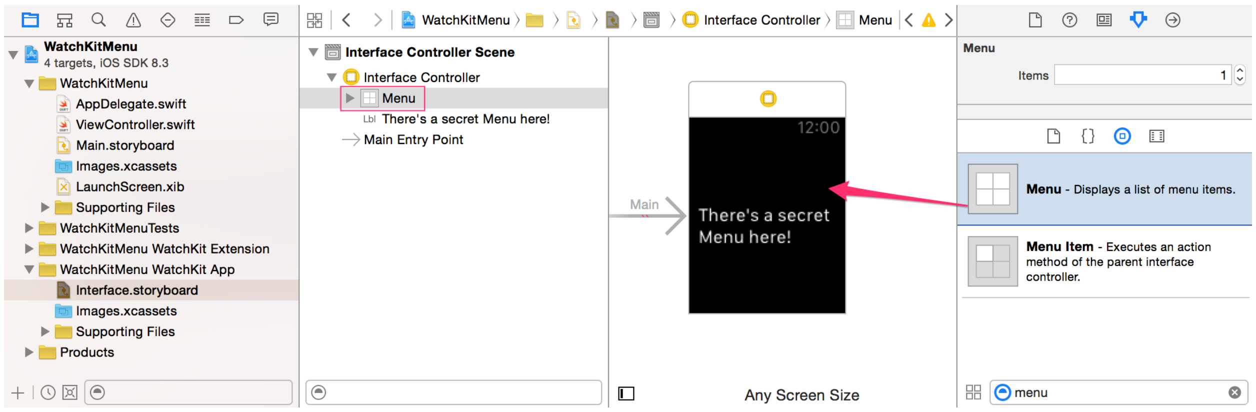
WatchKit Menu Tutorial

Start a New WatchKit Project

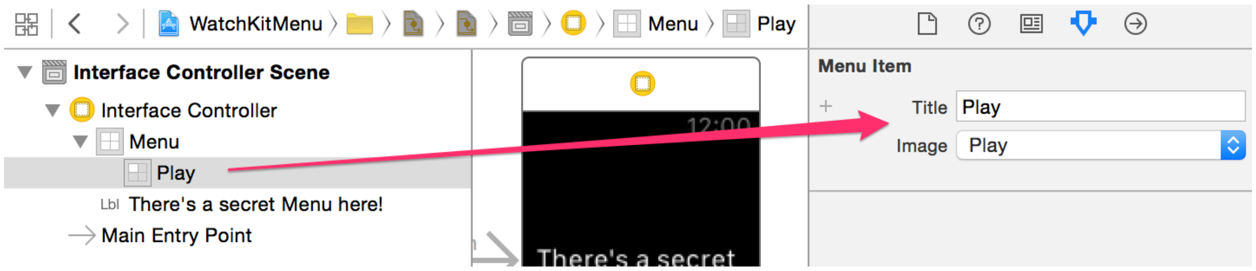
Follow the **Hello, WatchKit Tutorial** to set up a new WatchKit project. Make sure your Watch App runs properly.

Menu Setup

1. In **Interface.storyboard**, find the **Menu** in the Object Library and **drag it onto the Interface Controller**

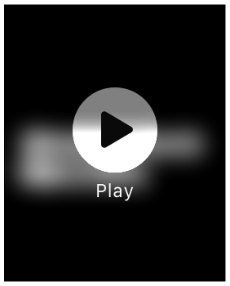


2. Note: There's no way to actually see the Menu in the Storyboard.
3. Configure the first **Menu Item** to be one of the default options



Title: Play
Image: Play

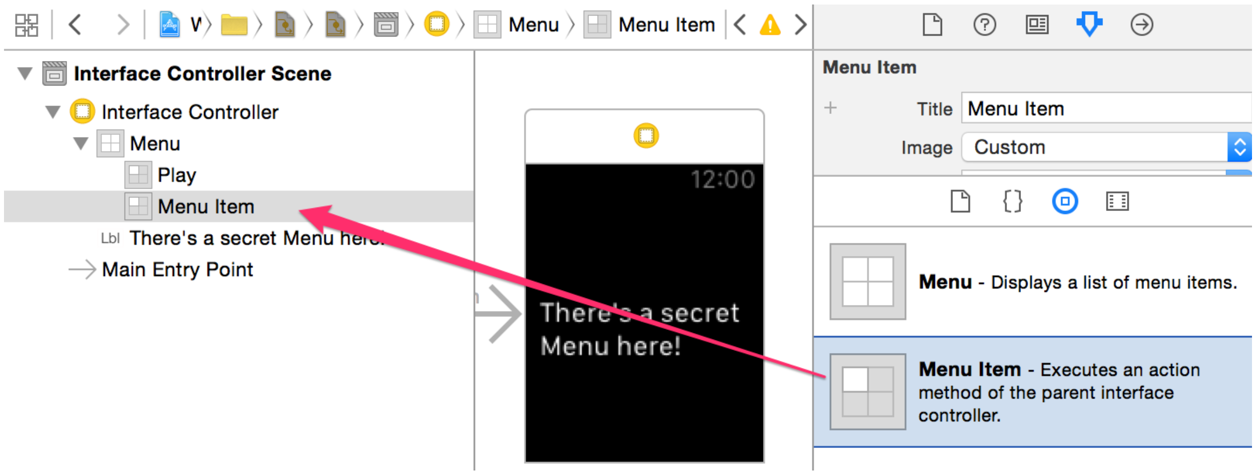
4. Run the Watch App. **Press and hold** with your mouse until the Menu option comes up (remember, this is activated via **Force Touch** on the Apple Watch)



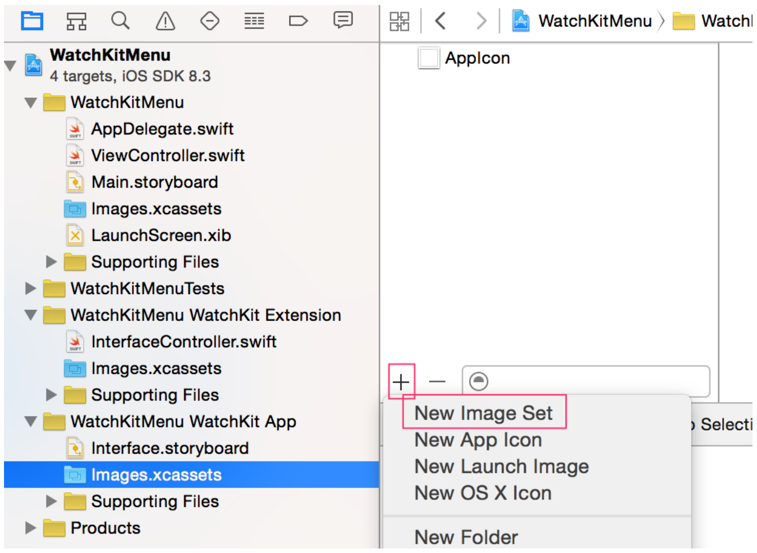
Adding Additional Menu Items

In the Storyboard

1. To add additional Menu Items, drag another **Menu Item** from the Object Library into the Menu in the Interface Controller


Menu
Menu Item
Interface Controller Scene
Interface Controller
Menu
Play
Menu Item
There's a secret Menu here!
Main Entry Point
12:00
There's a secret Menu here!
Menu Item
Title Menu Item
Image Custom
Menu - Displays a list of menu items.
Menu Item - Executes an action method of the parent interface controller.

2. This time, we're going to add a **Custom Image** as our Menu Item
   - Open the **Images.xcassets** folder in your **WatchKit App** (NOT the WatchKit Extension! This will make sure that the asset is readily available on the Watch instead of having to get the image from the iPhone via Bluetooth.)
   - Add a **New Image Set**



- Select the **Apple Watch** in the **Device Specific** settings for your new Image

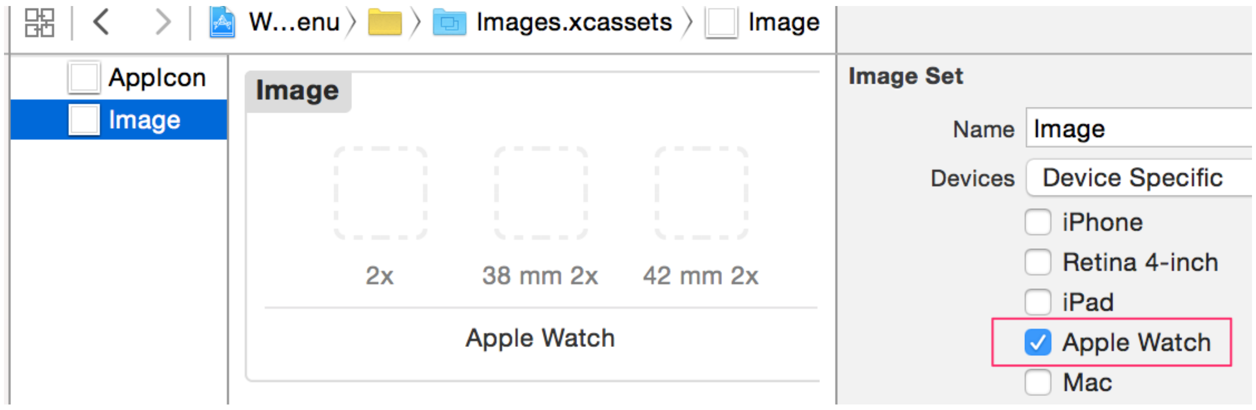
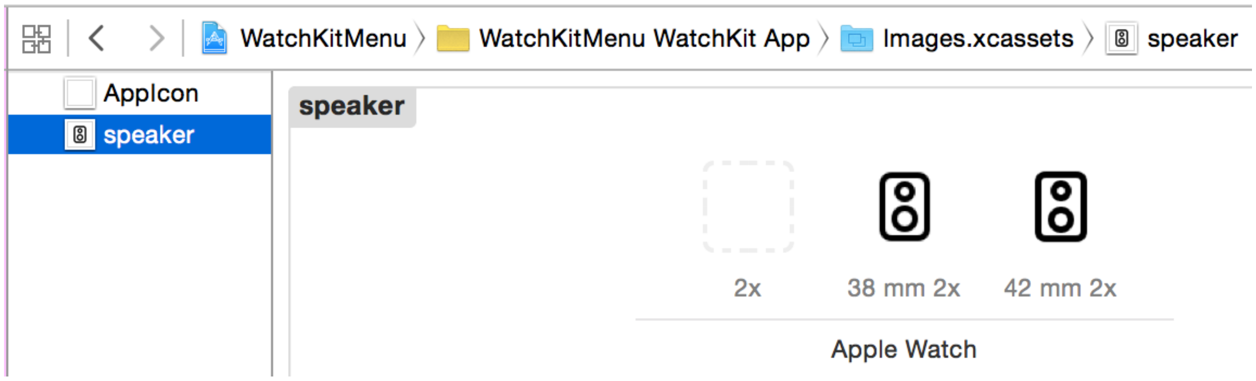
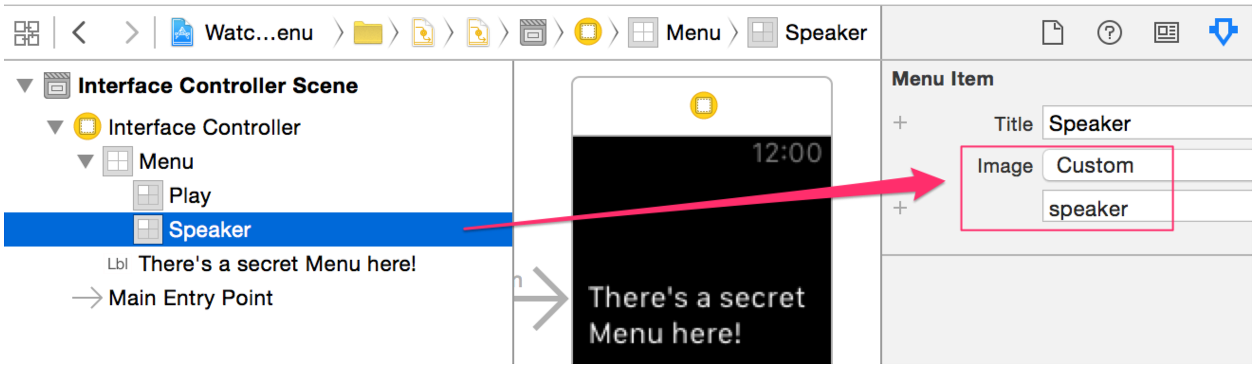


Image Set
Name: Image
Devices: Device Specific
- [ ] iPhone
- [ ] Retina 4-inch
- [ ] iPad
- [x] Apple Watch
- [ ] Mac

- Notice that there are custom Chronicons in the **Chronicons folder** in the same folder as this tutorial
- Add the Chronicon of your choice into the correct size slot and rename the Image to the appropriate name



- In the Storyboard, set the correct custom image name

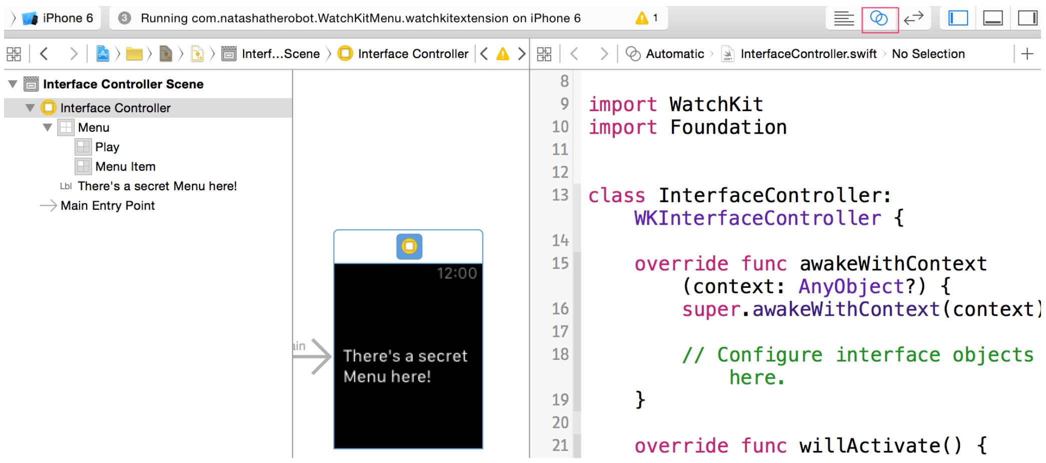

Watc...enu
Menu
Speaker
Interface Controller Scene
Interface Controller
Menu
Play
Speaker
Lbl There's a secret Menu here!
Main Entry Point
12:00
There's a secret Menu here!

Menu Item
Title: Speaker
Image: Custom
speaker

3. Run the app to see your new custom menu item in action!
4. Note: You can only have **up to 4 Menu Items per Interface Controller**

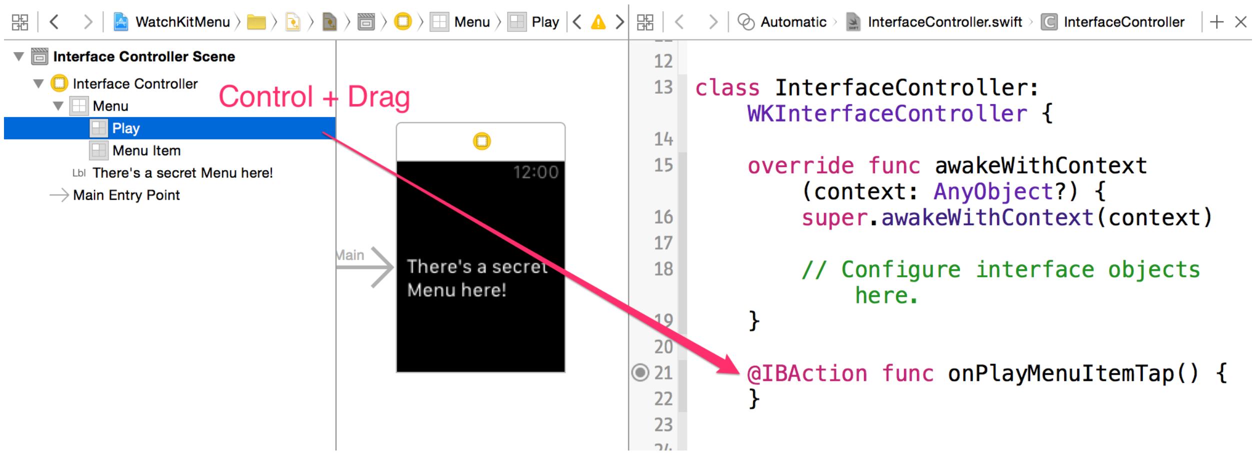
Create an @IBAction

1. **Select the Interface Controller** in the Storyboard, and tap the **Assistant Editor** button. It should automatically take you to the **InterfaceController.swift** file, located in your WatchKit Extension Target


iPhone 6
Running com.natashatherobot.WatchKitMenu.watchkitextension on iPhone 6
Interf...Scene
Interface Controller
Automatic
InterfaceController.swift
No Selection
Interface Controller Scene
Interface Controller
Menu
Play
Menu Item
There's a secret Menu here!
Main Entry Point
12:00
There's a secret Menu here!

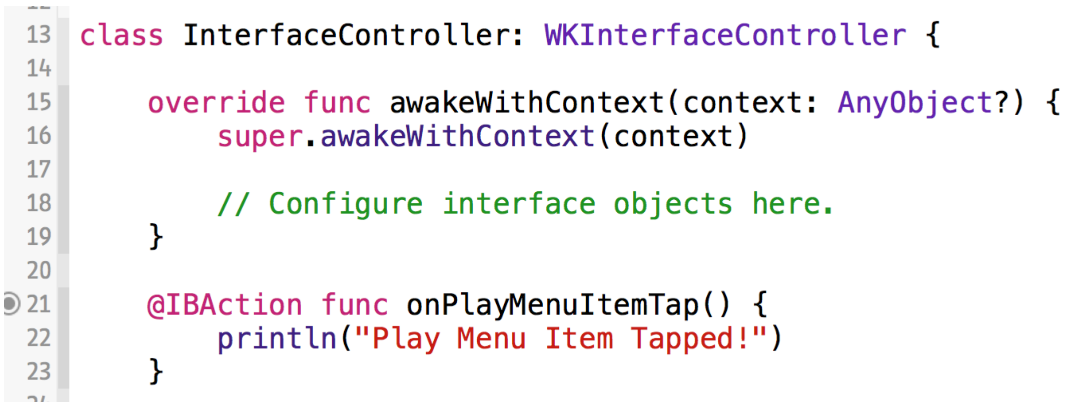
```
import WatchKit
import Foundation

class InterfaceController:
    WKInterfaceController {

    override func awakeWithContext
        (context: AnyObject?) {
        super.awakeWithContext(context)

        // Configure interface objects
            here.
    }

    override func willActivate() {
```

2. **Control + Drag** from the Menu Item (Play in my case) into the InterfaceController.swift file to create an **@IBAction**.



```
class InterfaceController:
    WKInterfaceController {

    override func awakeWithContext
        (context: AnyObject?) {
        super.awakeWithContext(context)

        // Configure interface objects
            here.
    }

    @IBAction func onPlayMenuItemTap() {
    }
```

3. Add some printing code in your IBAction.

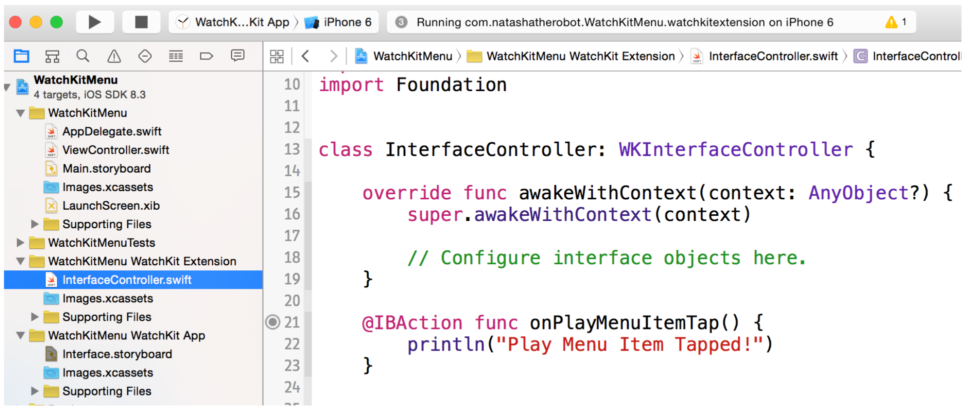


```
class InterfaceController: WKInterfaceController {

    override func awakeWithContext(context: AnyObject?) {
        super.awakeWithContext(context)

        // Configure interface objects here.
    }

    @IBAction func onPlayMenuItemTap() {
        println("Play Menu Item Tapped!")
    }
```

4. **Run** your Watch App. **Tap on the Menu Item** with the IBAction to see it printing your message!

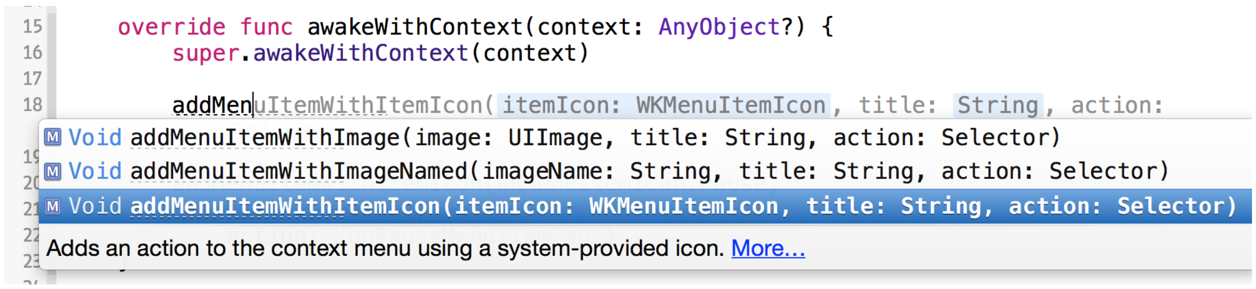
In Code

1. Go to the **IntefaceController.swift file** in your **WatchKit Extension**



```
import Foundation

class InterfaceController: WKInterfaceController {

    override func awakeWithContext(context: AnyObject?) {
        super.awakeWithContext(context)

        // Configure interface objects here.
    }

    @IBAction func onPlayMenuItemTap() {
        println("Play Menu Item Tapped!")
    }
```

2. Start typing **addMenu**…. Notice the options that come up to add Menu Items in code. Select **addMenuItemWithItemIcon:title:action:**



```
    override func awakeWithContext(context: AnyObject?) {
        super.awakeWithContext(context)

        addMenuItemWithItemIcon(itemIcon: WKMenuItemIcon, title: String, action:
```

```
M Void addMenuItemWithImage(image: UIImage, title: String, action: Selector)
M Void addMenuItemWithImageNamed(imageName: String, title: String, action: Selector)
M Void addMenuItemWithItemIcon(itemIcon: WKMenuItemIcon, title: String, action: Selector)
```

Adds an action to the context menu using a system-provided icon. More...

3. Configure this method with the default **Pause** icon

```
override func awakeWithContext(context: AnyObject?) {
    super.awakeWithContext(context)

    addMenuItemWithItemIcon(.Pause,
        title: "Pause",
        action: "onPauseMenuItemTap")
}
```

4. Add the **onPauseMenuItemTap** function, which will get called when the Pause Menu Item is tapped

```
class InterfaceController: WKInterfaceController {

    override func awakeWithContext(context: AnyObject?) {
        super.awakeWithContext(context)

        addMenuItemWithItemIcon(.Pause,
            title: "Pause",
            action: "onPauseMenuItemTap")
    }

    func onPauseMenuItemTap() {
        println("Pause Menu Item Tapped!")
    }
```

5. Run the app to make sure the Pause Menu Item works as expected!
6. **Challenge**: Create a Menu Item with a Custom Image in Code