Building Watch Apps

Session 108

Neil Desai watchOS Engineer
Agenda
Agenda

Architecture
Agenda

Architecture

Interface elements
Agenda

Architecture
Interface elements
Debugging
Agenda

Architecture
Interface elements
Debugging
watchOS 2 Features
Architecture
Architecture

iPhone

Apple Watch
Architecture

iPhone

Apple Watch

Watch App
Architectural Overview

- **iPhone**
- **Apple Watch**
  - **Watch App**
  - **Watch Extension**
Architecture

iPhone

iOS App

Apple Watch

Watch App

Watch Extension

WatchConnectivity
Architecture

Interface

UI is stored in Watch app
Architecture

Interface

UI is stored in Watch app

Plan your UI in advance
Architecture

Interface

UI is stored in Watch app
Plan your UI in advance
Code updates UI from the extension
Architecture

WatchKit Framework

One controller per screen of content
Architecture

WatchKit Framework

One controller per screen of content
Manages UI elements through outlets
Architecture

WatchKit Framework

One controller per screen of content
Manages UI elements through outlets
Uses target-action design pattern
Mom
Phone Calls: 10

Dad
Phone Calls: 4

Lisa
Phone Calls: 5

Ethan
Phone Calls: 7

Evan
Phone Calls: 3
Adding a Watch App
Adding a Watch App

Add Watch App target
Adding a Watch App

Add Watch App target
Configure interface elements
Adding a Watch App

Add Watch App target
Configure interface elements
Build and run
Simulator
Simulator
Overview
Simulator
Overview

Full simulator
Simulator

Overview

Full simulator

38mm and 42mm
Simulator

Overview

Full simulator

38mm and 42mm

Simulate Touch Pressure
Simulator
Overview

Full simulator
38mm and 42mm
Simulate Touch Pressure
Clock faces
Simulator

Overview

Full simulator
38mm and 42mm
Simulate Touch Pressure
Clock faces
Multiple apps
Simulator

Touch Pressure
Simulator

Touch Pressure
Demo
App Lifecycle
Watch Extension

WKExtensionDelegate

DidFinishLaunching()

Watch App

Initialize app
Watch Extension

WKExtensionDelegate

DidFinishLaunching()

DidBecomeActive()

Watch App

Initialize app

Update model
Watch Extension

WKExtensionDelegate

DidFinishLaunching()

DidChangeActive()

WKInterfaceController

Watch App

Initialize app

Update model
Watch Extension

WKExtensionDelegate
- didFinishLaunching()
- DidBecomeActive()

WKInterfaceController
- awakeWithContext()

Watch App
- Initialize app
- Update model
- Initialize UI
Watch Extension

WKExtensionDelegate
- DidFinishLaunching()
- DidBecomeActive()

WKInterfaceController
- awakeWithContext()
- willActivate()

Watch App
- Initialize app
- Update model
- Initialize UI
- Display UI

Initialize app
Update model
Initialize UI
Display UI
Watch Extension

WKExtensionDelegate
DidFinishLaunching()
DidBecomeActive()

WKInterfaceController
awakeWithContext()
willActivate()
(various action methods)

Watch App

Initialize app
Update model
Initialize UI
Display UI
Wearer interacts with UI
Watch Extension

WKExtensionDelegate
DidFinishLaunching()
DidBecomeActive()

WKInterfaceController
awakeWithContext()
willActivate()
(various action methods)
didDeactivate()

Watch App

Initialize app
Update model

Initialize UI
Display UI
Wearer interacts with UI
Wearer stops interacting

Wearer interacts with UI
Wearer stops interacting
Watch Extension

WKExtensionDelegate
DidFinishLaunching()

WKInterfaceController
awakeWithContext()
willActivate()
(various action methods)
didDeactivate()

WKExtensionDelegate

Watch App

Initialize app
Update model

Initialize UI
Display UI
Wearer interacts with UI
Wearer stops interacting
Watch Extension

WKExtensionDelegate
- didFinishLaunching()
- DidBecomeActive()

WKInterfaceController
- awakeWithContext()
- willActivate()
- (various action methods)
- didDeactivate()

WKExtensionDelegate
- WillResignActive()

Watch App

- Initialize app
- Update model
- Initialize UI
- Display UI
- Wearer interacts with UI
- Wearer stops interacting
- Active app to inactive
WatchKit Layout
WatchKit Layout

Model
WatchKit Layout Model

Different from UIKit and AppKit
WatchKit Layout

Model

Different from UIKit and AppKit

Flow-based layout
WatchKit Layout

Model

Different from UIKit and AppKit
Flow-based layout
UI created in Interface Builder
WatchKit Layout Model

Flow-based layout

Interface Controller

- **Group** - A container that manages the layout of other items.
- **Table** - Displays one or more rows of data.
- **Image** - Displays a static or animated image.
- **Separator** - A line for separating content in your interface.
- **Button** - A tappable area with a title and/or image.
- **Switch** - A control for indicating a binary value.
- **Slider** - A control for selecting a floating-point value from a range of continuous or discrete values.
- **Picker** - A control for selecting an item from a list.
- **Label** - Displays a static text string.
WatchKit Layout Model

Flow-based layout

Interface Controller

- **Group**: A container that manages the layout of other items.
- **Table**: Displays one or more rows of data.
- **Image**: Displays a static or animated image.
- **Separator**: A line for separating content in your interface.
- **Button**: A tappable area with a title and/or image.
- **Switch**: A control for indicating a binary value.
- **Slider**: A control for selecting a floating-point value from a range of continuous or discrete values.
- **Picker**: A control for selecting an item from a list.
- **Label**: Displays a static text string.
WatchKit Layout Model

Flow-based layout

Interface Controller

Image

- Group - A container that manages the layout of other items.
- Table - Displays one or more rows of data.
- Image - Displays a static or animated image.
- Separator - A line for separating content in your interface.
- Button - A tappable area with a title and/or image.
- Switch - A control for indicating a binary value.
- Slider - A control for selecting a floating-point value from a range of continuous or discrete values.
- Picker - A control for selecting an item from a list.
- Label - Displays a static text string.
WatchKit Layout Model
Flow-based layout

- **Interface Controller**

- **Image**

- **Group** - A container that manages the layout of other items.
- **Table** - Displays one or more rows of data.
- **Image** - Displays a static or animated image.
- **Separator** - A line for separating content in your interface.
- **Button** - A tappable area with a title and/or image.
- **Switch** - A control for indicating a binary value.
- **Slider** - A control for selecting a floating-point value from a range of continuous or discrete values.
- **Picker** - A control for selecting an item from a list.
- **Label** - Displays a static text string.
WatchKit Layout Model

Flow-based layout

Interface Controller

Image

Button
WatchKit Layout Model

Flow-based layout

Interface Controller

Image

Button

Group - A container that manages the layout of other items.
Table - Displays one or more rows of data.
Image - Displays a static or animated image.
Separator - A line for separating content in your interface.
Button - A tappable area with a title and/or image.
Switch - A control for indicating a binary value.
Slider - A control for selecting a floating-point value from a range of continuous or discrete values.
Picker - A control for selecting an item from a list.
Label - Displays a static text string.
WatchKit Layout Model

Groups are containers of elements

- Group - A container that manages the layout of other items.
- Table - Displays one or more rows of data.
- Image - Displays a static or animated image.
- Separator - A line for separating content in your interface.
- Button - A tappable area with a title and/or image.
- Switch - A control for indicating a binary value.
- Slider - A control for selecting a floating-point value from a range of continuous or discrete values.
- Picker - A control for selecting an item from a list.
- Label - Displays a static text string.
WatchKit Layout Model

Groups are containers of elements

- **Group**: A container that manages the layout of other items.
- **Table**: Displays one or more rows of data.
- **Image**: Displays a static or animated image.
- **Separator**: A line for separating content in your interface.
- **Button**: A tappable area with a title and/or image.
- **Switch**: A control for indicating a binary value.
- **Slider**: A control for selecting a floating-point value from a range of continuous or discrete values.
- **Picker**: A control for selecting an item from a list.
- **Label**: Displays a static text string.
WatchKit Layout Model
Groups are containers of elements

- **Group**: A container that manages the layout of other items.
- **Table**: Displays one or more rows of data.
- **Image**: Displays a static or animated image.
- **Separator**: A line for separating content in your interface.
- **Button**: A tappable area with a title and/or image.
- **Switch**: A control for indicating a binary value.
- **Slider**: A control for selecting a floating-point value from a range of continuous or discrete values.
- **Picker**: A control for selecting an item from a list.
- **Label**: Displays a static text string.
WatchKit Layout
Programming Model

You don’t write object creation code
WatchKit Layout
Programming Model

You don't write object creation code
Fine tuned control of
WatchKit Layout
Programming Model

You don’t write object creation code
Fine tuned control of
• Positioning and Sizing
WatchKit Layout
Programming Model

You don’t write object creation code
Fine tuned control of
• Positioning and Sizing
• Layout hierarchy
WatchKit Layout
Programming Model

You don’t write object creation code
Fine tuned control of
• Positioning and Sizing
• Layout hierarchy
• Animation
WatchKit Layout
Programming Model

You don’t write object creation code
Fine tuned control of
• Positioning and Sizing
• Layout hierarchy
• Animation
<table>
<thead>
<tr>
<th>Name</th>
<th>Phone Calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mom</td>
<td>10</td>
</tr>
<tr>
<td>Dad</td>
<td>4</td>
</tr>
<tr>
<td>Lisa</td>
<td>5</td>
</tr>
<tr>
<td>Ethan</td>
<td>7</td>
</tr>
<tr>
<td>Eyan</td>
<td>3</td>
</tr>
</tbody>
</table>
Creating My UI
Creating My UI

Main App UI
Glance
Notification
Demo
Mom
- Phone Calls: 10

Dad
- Phone Calls: 4

Lisa
- Phone Calls: 5

Ethan
- Phone Calls: 7

Evan
- Phone Calls: 3
<table>
<thead>
<tr>
<th>Name</th>
<th>Phone Calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mom</td>
<td>10</td>
</tr>
<tr>
<td>Dad</td>
<td>4</td>
</tr>
<tr>
<td>Lisa</td>
<td>5</td>
</tr>
<tr>
<td>Ethan</td>
<td>7</td>
</tr>
<tr>
<td>Eya</td>
<td>3</td>
</tr>
</tbody>
</table>
WatchConnectivity
WatchConnectivity Framework
WatchConnectivity Framework
WatchConnectivity Framework

Application Context
WatchConnectivity Framework

Application Context
Send Message
WatchConnectivity Framework

- Application Context
- Send Message
- File Transfer
WatchConnectivity Framework

Application Context
Send Message
File Transfer
Transfer User info
WatchConnectivity Framework

Application Context
Send Message
File Transfer
Transfer User info

Introducing Watch Connectivity
WatchConnectivity Framework
WatchConnectivity Framework

Add WatchConnectivity
WatchConnectivity Framework

Add WatchConnectivity
Debug my app
WatchConnectivity Framework

Add WatchConnectivity
Debug my app
Profile with Instruments
Instruments
Choose a profiling template for: Apple Watch - 38mm (2.0 Simulator) > CallMeLately Extension.appex

Standard  Custom  Recent

Blank  Activity Monitor  Allocations  Automation  Cocoa Layout  Core Animation

Core Data  Counters  Energy Diagnostics  File Activity  GPU Driver Leaks

Time Profiler
Performs low-overhead time-based sampling of processes running on the system’s CPUs.

Choose  Cancel
Instruments
Instruments

Engagement times are short
Instruments

Engagement times are short
Performance is critical
Instruments

Engagement times are short
Performance is critical
Device
Instruments

Engagement times are short
Performance is critical
Device
Simulator
WWDC CallMeLately
WWDC CallMeLately

Added a Watch app
WWDC CallMeLately

Added a Watch app

Created an interface for our app and glance
WWDC CallMeLately

Added a Watch app
Created an interface for our app and glance
WatchConnectivity
WWDC CallMeLately

Added a Watch app
Created an interface for our app and glance
WatchConnectivity
Debugged our iOS and watchOS apps
Added a Watch app
Created an interface for our app and glance
WatchConnectivity
Debugged our iOS and watchOS apps
Profiled in Instruments
watchOS 2 Features
Complications
Complications
Digital Crown
Digital Crown
Animations
Heart Rate Sensors
Haptics
Media Playback
Media Playback
Media Playback
Audio Recording
Audio Recording
Recap

Architecture
Recap

Architecture
Adding a Watch app
Recap

Architecture
Adding a Watch app
App Lifecycle
Recap

Architecture
Adding a Watch app
App Lifecycle
Use interface elements
Recap

Architecture
Adding a Watch app
App Lifecycle
Use interface elements
Build and run
Recap

Architecture
Adding a Watch app
App Lifecycle
Use interface elements
Build and run
Debugging
Recap

Architecture
Adding a Watch app
App Lifecycle
Use interface elements
Build and run
Debugging
Use watchOS capabilities
More Information

Documentation
watchOS 2 Transition Guide
WatchKit Programming Guide

Sample Code
Lister
WatchKit Catalog
http://developer.apple.com/watchOS

Technical Support
Apple Developer Forums
Developer Technical Support

General Inquiries
Jake Behrens, watchOS Frameworks Evangelist
behrens@apple.com
## Related Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Date &amp; Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple Watch Accessibility</td>
<td>Pacific Heights</td>
<td>Tuesday 1:30PM</td>
</tr>
<tr>
<td>WatchKit In-Depth, Part 1</td>
<td>Pacific Heights</td>
<td>Wednesday 9:00AM</td>
</tr>
<tr>
<td>WatchKit In-Depth, Part 2</td>
<td>Pacific Heights</td>
<td>Wednesday 10:00AM</td>
</tr>
<tr>
<td>Creating Complications with ClockKit</td>
<td>Pacific Heights</td>
<td>Wednesday 11:00AM</td>
</tr>
<tr>
<td>Layout and Animation Techniques for WatchKit</td>
<td>Pacific Heights</td>
<td>Thursday 10:00AM</td>
</tr>
<tr>
<td>WatchKit Tips and Tricks</td>
<td>Presidio</td>
<td>Friday 10:00AM</td>
</tr>
<tr>
<td>Introducing Watch Connectivity</td>
<td>Pacific Heights</td>
<td>Thursday 11:00AM</td>
</tr>
<tr>
<td>Designing for Apple Watch</td>
<td>Presidio</td>
<td>Wednesday 4:30PM</td>
</tr>
<tr>
<td>Apple Watch Design Tips and Tricks</td>
<td>Presidio</td>
<td>Friday 3:30PM</td>
</tr>
<tr>
<td>Labs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ClockKit Complications Lab</strong></td>
<td>Frameworks Lab D</td>
<td>Wednesday 1:30PM</td>
</tr>
<tr>
<td><strong>WatchKit Lab</strong></td>
<td>Frameworks Lab D</td>
<td>Wednesday 3:30PM</td>
</tr>
<tr>
<td><strong>Watch Connectivity Lab</strong></td>
<td>Frameworks Lab B</td>
<td>Thursday 1:30PM</td>
</tr>
<tr>
<td><strong>WatchKit Layout and Animation Lab</strong></td>
<td>Frameworks Lab A</td>
<td>Thursday 3:30PM</td>
</tr>
<tr>
<td><strong>WatchKit and ClockKit Complications Lab</strong></td>
<td>Frameworks Lab A</td>
<td>Friday 1:30PM</td>
</tr>
</tbody>
</table>