Creating Complications with ClockKit

Session 209

Eliza Block watchOS Engineer
Paul Salzman watchOS Engineer
WED 10
11:05
12:30–2:00PM
Lunch with Paul
Zero Zero
8:27
68°
11:00 AM
Haight – Twin Peaks
Group D
11:00AM
Haight – Twin Peaks
Group D

1:30PM
Bernal – Presidio
Group C

4:00PM
SOMA – Castro
Group A

6:00PM
Sunset – Marina
Group B
11:00 AM
Haight – Twin Peaks
Group D

1:30 PM
Bernal – Presidio
Group C

4:00 PM
SOMA – Castro
Group A

6:00 PM
Sunset – Marina
Group B
11:00AM
Group D
Haight – Twin Peaks

1:20
3:45
6:15

6:00PM
Group B
Sunset – Marina
Bernal – Presidio

4:00PM
SOMA – Castro

Group A

Sunset – Marina
Group B
Time Travel
Time Travel
Agenda
Agenda

Getting Started

Complication Layout
Agenda

Getting Started

Complication Layout

Providing Images and Text
Agenda

Getting Started

Complication Layout

Providing Images and Text

Timeline
Agenda

- Getting Started
- Complication Layout
- Providing Images and Text
- Timeline
- Complication Data Source
Agenda

Getting Started

Complication Layout

Providing Images and Text

Timeline

Complication Data Source

Keeping Your Data Fresh
Getting Started
Complication Families
Complication Families
Modular
Complication Families

Modular

Modular small
Complication Families

Modular

Modular small

Modular large
Complication Families

Utilitarian

Modular small

Modular large
Complication Families

Utilitarian

Modular small

Modular large
Complication Families

Utilitarian

Modular small
Modular large
Utilitarian small
Complication Families

Utilitarian

Modular small
Modular large
Utilitarian small
Utilitarian large
Complication Families

Circular

Modular small
Modular large
Utilitarian small
Utilitarian large
Complication Families

Circular

- Modular small
- Modular large
- Utilitarian small
- Utilitarian large
Complication Families

Circular

Modular small
Modular large
Utilitarian small
Utilitarian large
Circular small
enum CLKComplicationFamily : Int {
    case ModularSmall
    case ModularLarge
    case UtilitarianSmall
    case UtilitarianLarge
    case CircularSmall
}

Complication Families

Modular small
Modular large
Utilitarian small
Utilitarian large
Circular small
Complication Layout
CLKComplicationTemplate

Modular Small

NYC
5:54

8:21

75°

85

NYC
2
SF 0
Utilitarian Small

NYC
5:54
8:21
75°
85
2
SF 0

Modular Small

NYC
5:54
8:21
75°
85
2
SF 0

Modular Large

12:00–2:00PM
Lunch with Lucas Saru Sushi

8:26PM

J inbound 1 min
L outbound 4 min
K inbound 11 min

Distant weekdays
Dec 29, 0001 Fri
Dec 31, 4000 Sun

Kitchen Timer
5:17

Haight – Twin Peaks
Group D

85
<table>
<thead>
<tr>
<th>Time</th>
<th>Location</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>12:00–2:00PM</td>
<td>Modular Large</td>
<td>Lunch with Lucas Saru Sushi</td>
</tr>
<tr>
<td>11:00AM</td>
<td>Modular Large</td>
<td>Haight – Twin Peaks Group D</td>
</tr>
<tr>
<td></td>
<td></td>
<td>J inbound: 1 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>L outbound: 4 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K inbound: 11 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Distant weekdays</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dec 29, 0001 Fri</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dec 31, 4000 Sun</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Kitchen Timer: 5:17</td>
</tr>
<tr>
<td>8:26PM</td>
<td>Modular Small</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Utilitarian Small</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>8:26PM</td>
</tr>
<tr>
<td></td>
<td>Utilitarian Large</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>8:26PM 10hrs 8mins</td>
</tr>
</tbody>
</table>
CLKComplicationTemplate

11:00 AM
Haight – Twin Peaks
Group D
11:00 AM
Haight – Twin Peaks
Group D
CLKComplicationTemplate

Header image

Football at 11:00 AM
Haight – Twin Peaks
Group D
CLKComplicationTemplate

11:00 AM
Haight – Twin Peaks
Group D
CLKComplicationTemplate

Header image -> Header text

11:00 AM

Body 1 text -> Haight – Twin Peaks
Group D
CLKComplicationTemplate

11:00 AM

Haight – Twin Peaks

Group D
CLKComplicationTemplate

Header image  ↓  Header text

11:00 AM

Body 1 text  →  Haight – Twin Peaks
Body 2 text  →  Group D

class CLKComplicationTemplateModularLargeStandardBody {
    var headerImageProvider: CLKImageProvider?
    var headerTextProvider:  CLKTextProvider
    var body1TextProvider:   CLKTextProvider
    var body2TextProvider:   CLKTextProvider?
}

class CLKComplicationTemplateModularLargeStandardBody {
    var headerImageProvider: CLKImageProvider?
    var headerTextProvider:  CLKTextProvider
    var body1TextProvider:   CLKTextProvider
    var body2TextProvider:   CLKTextProvider?
}

CLKComplicationTemplate

11:00 AM

Group D
Haight – Twin Peaks
Body 1 text
Body 2 text

class CLKComplicationTemplateModularLargeStandardBody {
    var headerImageProvider: CLKImageProvider?
    var headerTextProvider:  CLKTextProvider
    var body1TextProvider:   CLKTextProvider
    var body2TextProvider:   CLKTextProvider?
}
Providing Images and Text
Image Providers

...not UIImages
Image Providers
...not UIImages
Image Providers

…not UIImages
Image Providers
...not UIImages
Image Providers

...not UnitImages
Image Providers

…not UllImages
Image Providers
...not UlImages
Image Providers

...not UIImages

```swift
let bgImage = UIImage(named: "Background")
let fgImage = UIImage(named: "Foreground")

let imageProvider = CLKImageProvider(backgroundImage: bgImage,
  backgroundColor: aColor,
  foregroundImage: fgImage,
  foregroundColor: CLKImageProviderForegroundColor.White)
```
Text Providers

...not NSStrings
Text Providers

...not NSStrings

Space is very constrained
Text Providers

...not NSStrings

Space is very constrained
Declare your intentions
Text Providers

...not NSStrings

Space is very constrained
Declare your intentions
Let us handle formatting and fitting
Example: Formatting Dates
CLKDateTextProvider
Example: Formatting Dates

CLKDateTextProvider

Wednesday, September 23
Example: Formatting Dates

CLKDateTextProvider

Wednesday, Se…
Example: Formatting Dates
CLKDateTextProvider

Wednesday, September 23
Example: Formatting Dates
CLKDateTextProvider

Wednesday, September 23
Wednesday, Sep 23
Example: Formatting Dates
CLKDateTextProvider

Wednesday, September 23
Wednesday, Sep 23
Wed, Sep 23
Example: Formatting Dates
CLKDateTextProvider

Wednesday, September 23
Wednesday, Sep 23
Wed, Sep 23
Sep 23
Example: Formatting Dates
CLKDateTextProvider

Wednesday, September 23
Wednesday, Sep 23
Wed, Sep 23
Sep 23
23
Example: Formatting Dates
CLKDateTextProvider

Wednesday, September 23
Wednesday, Sep 23
Wed, Sep 23
Sep 23
23

let sep23: NSDate = ...
let units = [NSCalendarUnit.Weekday,
            NSCalendarUnit.Month,
            NSCalendarUnit.Day]

textProvider = CLKDateTextProvider(sep23, units)
Example: Formatting Dates

CLKDateTextProvider

Wednesday, September 23
Wednesday, Sep 23
Wed, Sep 23
Sep 23
23

let sep23: NSDate = ...
let units = [NSCalendarUnit.Weekday,
NSCalendarUnit.Month,
NSCalendarUnit.Day]
textProvider = CLKDateTextProvider(sep23, units)
Example: Formatting Dates
CLKDateTextProvider

Wednesday, September 23
Wednesday, Sep 23
Wed, Sep 23
Sep 23
23

let sep23: NSDate = ...
let units = [NSCalendarUnit.Weekday,
            NSCalendarUnit.Month,
            NSCalendarUnit.Day]

textProvider = CLKDateTextProvider(sep23, units)
Example: Formatting Dates

CLKDateTextProvider

Wednesday, September 23
Wednesday, Sep 23
Wed, Sep 23
Sep 23
23

let sep23: NSDate = ...
let units = [NSCalendarUnit.Weekday,
            NSCalendarUnit.Month,
            NSCalendarUnit.Day]

textProvider = CLKDateTextProvider(sep23, units)
More Text Providers
More Text Providers

Simple text provider
More Text Providers

Simple text provider
Time text provider
More Text Providers

Simple text provider
Time text provider
More Text Providers

Simple text provider
Time text provider
Time interval text provider
More Text Providers

Simple text provider
Time text provider
Time interval text provider
Relative Dates
Relative Dates
Relative Dates
Relative Dates
Relative Dates

Waning crescent
Moonset 2:19PM
2HRS 57MINS

8:27 75°
Relative Dates
Relative Dates
Relative Date Text Provider

Waning crescent
Moonset 2:19PM
2HRS 56MINS
let moonset : NSDate = ... // 2:19pm
Relative Date Text Provider

Waning crescent
Moonset 2:19PM
2HRS 56MINS

let moonset : NSDate = ... // 2:19pm
let units : NSCalendarUnit = [.Hour, .Minute]
Relative Date Text Provider

Waning crescent
Moonset 2:19PM
2hrs 56mins

let moonset : NSDate = ... // 2:19pm
let units : NSCalendarUnit = [.Hour, .Minute]
let style : CLKRelativeDateStringStyle = .Natural
let moonset : NSDate = ... // 2:19pm
let units : NSCalendarUnit = [.Hour, .Minute]
let style : CLKRelativeDateStyle = .Natural
let textProvider = CLKRelativeDateTextProvider(date: moonset,
                                           style: style,
                                           units: units)
Relative Date Text Provider

let moonset : NSDate = ... // 2:19pm
let units : NSCalendarUnit = [.Hour, .Minute]
let style : CLKRelativeDateStyle = .Natural
let textProvider = CLKRelativeDateTextProvider(date: moonset,
    style: style,
    units: units)

template.body2TextProvider = textProvider
## Relative Date Text Provider

### Styles

<table>
<thead>
<tr>
<th>Natural</th>
<th>Offset</th>
<th>Timer</th>
</tr>
</thead>
<tbody>
<tr>
<td>2HRS 7MINS</td>
<td>+14 SECONDS</td>
<td>02:32</td>
</tr>
<tr>
<td>13MINS 5SECS</td>
<td>-59 MINUTES</td>
<td>21:15:59</td>
</tr>
<tr>
<td>3WKS 4DAYS</td>
<td>+2 HOURS</td>
<td></td>
</tr>
<tr>
<td>2YRS 3MOS</td>
<td>-6 DAYS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+12 WEEKS</td>
<td></td>
</tr>
</tbody>
</table>
NY: 2
SF: 0

11:00 AM
Haight – Twin Peaks
Group D
11:00AM
Haight – Twin Peaks
Group D
11:00AM
Haight – Twin Peaks
Group D
11:00 AM
Haight – Twin Peaks
Group D
Timeline

Paul Salzman watchOS Engineer
Timeline

57°
Timeline

57°
Timeline

57°
Timeline

Brunch with K.K.
Joanie’s Café
11:00 AM – 1:00 PM
Timeline

11:00 AM–1:00 PM
Brunch with K.K.
Joanie’s Café

4:00–5:00 PM
Haircut
Nancy Bee

8:00–10:30 PM
Movie Night w/
David & Scott
Timeline

11:00AM–1:00PM
Brunch with K.K.
Joanie’s Café

4:00–5:00PM
Haircut
Nancy Bee

8:00–10:30PM
Movie Night w/
David & Scott
Timeline

11:00 AM - 1:00 PM
Brunch with K.K.
Joanie’s Café

4:00 - 5:00 PM
Haircut
Nancy Bee

8:00 - 10:30 PM
Movie Night w/
David & Scott
Timeline

11:00 AM–1:00 PM
Brunch with K.K.
Joanie’s Café

4:00–5:00 PM
Haircut
Nancy Bee

8:00–10:30 PM
Movie Night w/
David & Scott

Current Time
Timeline

11:00 AM–1:00 PM
Brunch with K.K.
Joanie’s Café

4:00–5:00 PM
Haircut
Nancy Bee

8:00–10:30 PM
Movie Night w/
David & Scott
Timeline

4:00–5:00 PM
Haircut
Nancy Bee

11:00 AM–1:00 PM
Brunch with K.K.
Joanie’s Café

4:00–5:00 PM
Haircut
Nancy Bee

8:00–10:30 PM
Movie Night w/
David & Scott

Current Time
Timeline

11:00 AM–1:00 PM
Brunch with K.K.
Joanie’s Café

4:00–5:00 PM
Haircut
Nancy Bee

8:00–10:30 PM
Movie Night w/
David & Scott
Timeline

11:00 AM – 1:00 PM
Brunch with K.K.
Joanie’s Café

4:00 – 5:00 PM
Haircut
Nancy Bee

8:00 – 10:30 PM
Movie Night w/ David & Scott
Timeline

11:00 AM–1:00 PM
Brunch with K.K.
Joanie’s Café

4:00–5:00 PM
Haircut
Nancy Bee

8:00–10:30 PM
Movie Night w/
David & Scott
Timeline

11:00 AM–1:00 PM
Brunch with K.K.
Joanie’s Café

4:00–5:00 PM
Haircut
Nancy Bee

8:00–10:30 PM
Movie Night w/
David & Scott

10:30 PM–1 PM
Brunch with K.K.
Joanie’s Café
Timeline

11:00 AM–1:00 PM
Brunch with K.K.
Joanie’s Café

4:00–5:00 PM
Haircut
Nancy Bee

8:00–10:30 PM
Movie Night w/
David & Scott

Calendar
No more events today
Timeline

11:00 AM–1:00 PM
Brunch with K.K.
Joanie’s Café

4:00–5:00 PM
Haircut
Nancy Bee

8:00–10:30 PM
Movie Night w/
David & Scott

11:00 AM–1:00 PM
Brunch with K.K.
Joanie’s Café

Current Time

Calendar
No more events today
4:00–5:00PM
Haircut
Nancy Bee

11:00AM–1:00PM
Brunch with K.K.
Joanie’s Café

4:00–5:00PM
Haircut
Nancy Bee

8:00–10:30PM
Movie Night w/
David & Scott

Calendar
No more events
today

Current Time

MIDNIGHT

1PM

5PM

10:30PM
Timeline

8:00–10:30PM
Movie Night w/ David & Scott

11:00AM–1:00PM
Brunch with K.K.
Joanie’s Café

4:00–5:00PM
Haircut
Nancy Bee

8:00–10:30PM
Movie Night w/ David & Scott

Calendar
No more events today

Current Time
Timeline

Calendar
No more events today

11:00 AM–1:00 PM
Brunch with K.K.
Joanie’s Café

4:00–5:00 PM
Haircut
Nancy Bee

8:00–10:30 PM
Movie Night w/
David & Scott

Calendar
No more events today

Current Time
Timeline
CLKComplicationTimelineEntry
Timeline
CLKComplicationTimelineEntry
CLKComplicationTemplate
CLKImageProvider
CLKTextProvider
CLKTextProvider
Timeline
CLKComplicationTimelineEntry

 CLKComplicationTimelineEntry
   ↓
   NSDate

 CLKComplicationTimelineEntry
   ↓
   CLKComplicationTemplate

   ↓
   CLKImageProvider

   ↓
   CLKTextProvider

   ↓
   CLKTextProvider
class CLKComplicationTimelineEntry : NSObject {
    var date: NSDate
    var complicationTemplate: CLKComplicationTemplate
}
Complication Data Source
Complication Data Source

CLKComplicationDataSource protocol
Complication Data Source
CLKComplicationDataSource protocol

Object annotated in extension target settings
Complication Data Source
CLKComplicationDataSource protocol

Object annotated in extension target settings
Callbacks per complication
Complication Data Source

CLKComplicationDataSource protocol

Object annotated in extension target settings

Callbacks per complication

• Check CLKComplication’s family property
Complication Data Source
CLKComplicationDataSource protocol

Object annotated in extension target settings

Callbacks per complication
  • Check CLKComplication’s family property

Handler parameter for completion and passing data
Complication Data Source
Current timeline entry

Clock Face

Extension

Now
Complication Data Source

Current timeline entry

What is your current timeline entry?

Clock Face

Extension

Now
Complication Data Source

Current timeline entry

What is your current timeline entry?

Clock Face → Extension
Complication Data Source

Current timeline entry

Clock Face → What is your current timeline entry? → Extension

Now
func getCurrentTimelineEntryForComplication(
    complication: CLKComplication,
    withHandler handler: (CLKComplicationTimelineEntry?) -> Void)
func getCurrentTimelineEntryForComplication(
    complication: CLKComplication,
    withHandler handler: (CLKComplicationTimelineEntry?) -> Void)
func getCurrentTimelineEntryForComplication(
    complication: CLKComplication,
    withHandler handler: (CLKComplicationTimelineEntry?) -> Void)
Complication Data Source

Extend timeline

Clock Face

Extension

Now
Complication Data Source

Extend timeline

Clock Face

What entries do you have after a specific point in time?

Extension

After Date
Complication Data Source

Extend timeline

Clock Face

What entries do you have after a specific point in time?

Extension

Array

CLKComplicationTimelineEntry

CLKComplicationTimelineEntry

...
Complication Data Source

Extend timeline

Clock Face → What entries do you have after a specific point in time? → Extension

After Date
Complication Data Source

Extend timeline

Clock Face → What entries do you have after a specific point in time? → Extension

After Date
Complication Data Source

Extend timeline

Clock Face

What entries do you have after a specific point in time?

Extension
func getTimelineEntriesForComplication(
    complication: CLKComplication,
    afterDate date: NSDate,
    limit: Int,
    withHandler handler: ([CLKComplicationTimelineEntry]?) -> Void)
func getTimelineEntriesForComplication(
    complication: CLKComplication,
    afterDate date: NSDate,
    limit: Int,
    withHandler handler: ([CLKComplicationTimelineEntry]?) -> Void)

Extend timeline
func getTimelineEntriesForComplication(
    complication: CLKComplication,
    afterDate date: NSDate,
    limit: Int,
    withHandler handler: ([CLKComplicationTimelineEntry]?) -> Void)
func getTimelineEntriesForComplication(
    complication: CLKComplication,
    afterDate date: NSDate,
    limit: Int,
    withHandler handler: ([CLKComplicationTimelineEntry]?) -> Void)
func getTimelineEntriesForComplication(
    complication: CLKComplication,
    afterDate date: NSDate,
    limit: Int,
    withHandler handler: ([CLKComplicationTimelineEntry]?) -> Void)
func getTimelineEntriesForComplication(
    complication: CLKComplication,
    beforeDate date: NSDate,
    limit: Int,
    withHandler handler: ([CLKComplicationTimelineEntry]?) -> Void)
Complication Data Source

Time Travel directions

Clock Face

Extension

Now
Complication Data Source

Time Travel directions

Which directions of Time Travel do you support?
Complication Data Source

Time Travel directions

Clock Face → Which directions of Time Travel do you support? → Extension

Now
Which directions of Time Travel do you support?

Clock Face → Extension

Now
Complication Data Source

Time Travel directions

Which directions of Time Travel do you support?

Clock Face → Extension

Now
Complication Data Source

Time Travel directions

Clock Face → Which directions of Time Travel do you support? → Extension

Now
func getSupportedTimeTravelDirectionsForComplication(
    complication: CLKComplication,
    withHandler handler: (CLKComplicationTimeTravelDirections) -> Void)
func getSupportedTimeTravelDirectionsForComplication(
    complication: CLKComplication,
    withHandler handler: (CLKComplicationTimeTravelDirections) -> Void)
func getSupportedTimeTravelDirectionsForComplication(
    complication: CLKComplication,
    withHandler handler: (CLKComplicationTimeTravelDirections) -> Void)
Brunch with K.K.
Joanie’s Café
11:00 AM – 1:00 PM
Movie Night w/ David & Scott
8:00–10:30 PM
Complication Data Source

Timeline boundaries

Clock Face

Extension

Now
Complication Data Source

Timeline boundaries

Clock Face

When is the end of your timeline?

Extension

Now
Complication Data Source

Timeline boundaries

Clock Face

When is the end of your timeline?

Extension

Now
Complication Data Source

Timeline boundaries

Clock Face → When is the beginning of your timeline? → Extension

Now
Complication Data Source
Timeline boundaries

Clock Face

When is the beginning of your timeline?

Extension

Now
func getTimelineEndDateForComplication(
    complication: CLKComplication,
    withHandler handler: (NSDate?) -> Void)
func getTimelineEndDateForComplication(
    complication: CLKComplication,
    withHandler handler: (NSDate?) -> Void)
Complication Data Source

Timeline boundaries

```swift
func getTimelineEndDateForComplication(
    complication: CLKComplication,
    withHandler handler: (NSDate?) -> Void)
```
func getTimelineStartDateForComplication(
    complication: CLKComplication,
    withHandler handler: (NSDate?) -> Void)
Complication Data Source

Placeholder templates

Clock Face

Extension
Complication Data Source

Placeholder templates

Clock Face

What is your placeholder template?

Extension
Complication Data Source
Placeholder templates

Clock
Face

What is your placeholder template?

Extension
Complication Data Source
Placeholder templates
func getPlaceholderTemplateForComplication(
    complication: CLKComplication,
    withHandler handler: (CLKComplicationTemplate?) -> Void)
func getPlaceholderTemplateForComplication(
    complication: CLKComplication,
    withHandler handler: (CLKComplicationTemplate?) -> Void)
func getPlaceholderTemplateForComplication(
    complication: CLKComplication,
    withHandler handler: (CLKComplicationTemplate?) -> Void)
Demo

Building a complication
Keeping Your Data Fresh
Data Lifecycle

WatchConnectivity
Data Lifecycle
Reload Timeline

Clock Face

Extension

Now
Reload Timeline

Clock Face

Extension

Now

Clock Face

Extension

Now
Reload Timeline

Clock Face

Extension
CLKComplicationServer

Now
Reload Timeline

Clock Face

Please reload my timeline

Extension
CLKComplicationServer

Now
Reload Timeline

Clock Face

Please reload my timeline

Extension
CLKComplicationServer

Now
Reload Timeline

Clock Face

What is your current timeline entry?

Extension
CLKComplicationServer

Now
Extend Timeline

Clock Face

Extension
CLKComplicationServer

Now
Extend Timeline

Clock Face

Please extend my timeline

Extension
CLKComplicationServer

Now
What entries do you have after a specific point in time?
// get the shared instance
let server = CLKComplicationServer.sharedInstance()
// get the shared instance
let server = CLKComplicationServer.sharedInstance()

// extend the timeline for all complications
for complication in server.activeComplications {
}

// get the shared instance
let server = CLKComplicationServer.sharedInstance()

// extend the timeline for all complications
for complication in server.activeComplications {
    server.extendTimelineForComplication(complication)
}
// get the shared instance
let server = CLKComplicationServer.sharedInstance()

// extend the timeline for all complications
for complication in server.activeComplications {
    server.extendTimelineForComplication(complication)
}

// reload the timeline for all complications
for complication in server.activeComplications {
    server.reloadTimelineForComplication(complication)
}
When Can You Update?
Whenever extension runs
Whenever extension runs

• When Watch app runs
Whenever extension runs

- When Watch app runs
- Locally scheduled wake
Whenever extension runs

- When Watch app runs
- Locally scheduled wake
- iPhone initiated wake
Whenever extension runs

- When Watch app runs
- Locally scheduled wake  
  Budgeted
- iPhone initiated wake  
  Budgeted
When Can You Update?

Whenever extension runs

• When Watch app runs

• Locally scheduled wake  Budgeted

• iPhone initiated wake  Budgeted
CLKComplicationDataSource

One more call

```swift
func getNextRequestedUpdateDateWithHandler(
    handler: (NSDate?) -> Void)
```
CLKComplicationDataSource

One more call

```swift
func getNextRequestedUpdateDateStringWithHandler(
    handler: (NSDate?) -> Void)
```

One date returned across all active complications
CLKComplicationDataSource

One more call

```swift
func getNextRequestedUpdateDateWithHandler(
    handler: (NSDate?) -> Void)
```

One date returned across all active complications

Hint for budgeted background launch of extension
CLKComplicationDataSource

One more call

```swift
func getNextRequestedUpdateDateWithHandler(
    handler: (NSDate?) -> Void)
```

One date returned across all active complications

Hint for budgeted background launch of extension

Allows data delegate callbacks
Summary
Summary

Build timeline of templates and providers
Summary

Build timeline of templates and providers
Take advantage of formatting and fitting
Summary

Build timeline of templates and providers
Take advantage of formatting and fitting
Refresh data using delegate events
Summary

Build timeline of templates and providers
Take advantage of formatting and fitting
Refresh data using delegate events
Do less work to run more
More Information

Documentation
- watchOS 2 Transition Guide
- WatchKit Programming Guide

Sample Code
- Lister
- WatchKit Catalog

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</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introducing Watch Connectivity</td>
<td>Pacific Heights</td>
<td>Thursday 11:00AM</td>
</tr>
<tr>
<td>WatchKit Tips and Tricks</td>
<td>Presidio</td>
<td>Friday 10:00AM</td>
</tr>
<tr>
<td>Introducing WatchKit for watchOS 2</td>
<td>Presidio</td>
<td>Tuesday 10:00AM</td>
</tr>
<tr>
<td>Building Watch Apps</td>
<td>Pacific Heights</td>
<td>Tuesday 4:30PM</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>
</tbody>
</table>
## Related Labs

<table>
<thead>
<tr>
<th>Lab</th>
<th>Frameworks Lab</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClockKit Complications Lab</td>
<td>D</td>
<td>Wednesday 1:30PM</td>
</tr>
<tr>
<td>Watch Connectivity Lab</td>
<td>B</td>
<td>Thursday 1:30PM</td>
</tr>
<tr>
<td>WatchKit and ClockKit Complications Lab</td>
<td>A</td>
<td>Friday 1:30PM</td>
</tr>
</tbody>
</table>
Apple WWDC 2015