

WatchKit

Hands-On Challenges

WatchKit Hands-On Challenges

Copyright © 2014 Razeware LLC.

All rights reserved. No part of this book or corresponding materials (such as text, images, or source code) may be reproduced or distributed by any means without prior written permission of the copyright owner.

This book and all corresponding materials (such as source code) are provided on an "as is" basis, without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

All trademarks and registered trademarks appearing in this book are the property of their respective owners.



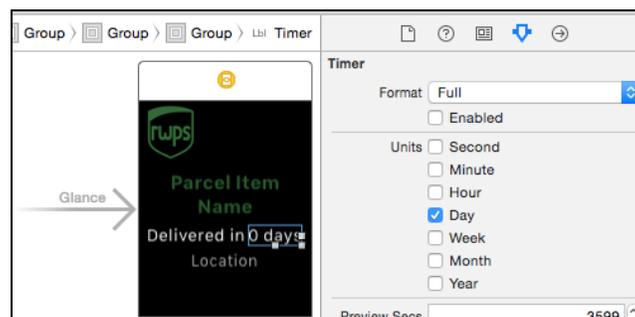
Challenge I: Countdown

On the Glance you've set up a label to display the number of days remaining before the current parcel is delivered, and you generate its content using `daysRemainingBeforeDelivery()` and string interpolation. But is there a better way?



There certainly is, and it's called **WKInterfaceTimer**. This is a special kind of label that displays a countdown to a provided date, using a set of units you specify in Interface Builder, such as minute, hour, day, week, and so-on.

Your challenge is simple: update the Glance so it uses **WKInterfaceTimer** to display the number of days remaining before the parcel is delivered.



But there is a caveat; when using just the Day unit, **WKInterfaceTimer** only takes into account the remaining full days. You'll need to add a new method to the Parcel class to calculate an adjusted delivery date specifically for use with **WKInterfaceTimer**.

Note: you can use `NSCalendar`, date components, and `dateByAddingTimeInterval(_:)` to help you achieve this.

Before you turn the page for our solution, be sure to give it a try for yourself first!



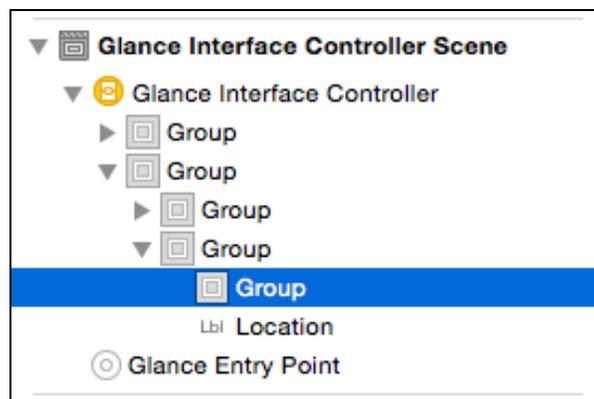
Solution

Open **Interface.storyboard** from the **RWPS Watch App\Storyboards** group, and delete the “Delivered in 99 days” label from the Glance Interface Controller Scene.

Once you’ve done that, open **ParcelGlanceController.swift** and delete the following line:

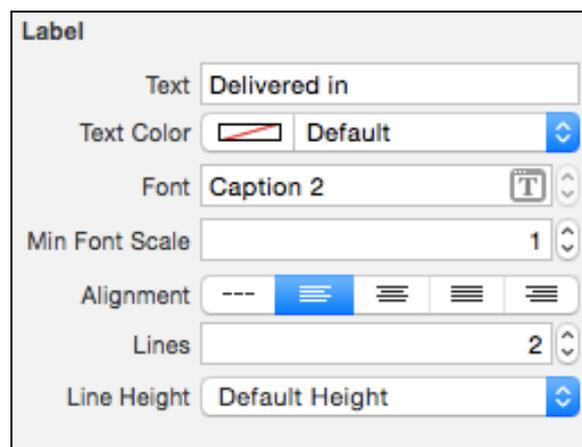
```
@IBOutlet weak var deliveredInLabel: WKInterfaceLabel!
```

Re-open the storyboard, and drag a new Group from the Object Library into the Document Outline, dropping it just above the “Location” label.



In the Attributes inspector, set its **Horizontal** position to **Center**.

Next, drag a Label from the Object Library into the new group. In the Attributes inspector, set its Text to “Delivered in”, its Font to the text style Caption 2, its Alignment to left-justified, and the number of Lines to 2.



Then drag a Timer from the Object Library into the new group, just below the label.





In the Attributes inspector, set its Font to the text style Caption 2 and its Alignment to left-justified.

Open **Parcel.swift** and add the following method:

```
func deliveryDateForUseWithTimer() -> NSDate? {
    let calendar = NSCalendar.currentCalendar()
    let components = calendar.components(.CalendarUnitHour |
        .CalendarUnitMinute | .CalendarUnitSecond, fromDate: NSDate())
    let adjustedDeliveryDate =
        calendar.dateByAddingComponents(components, toDate:
            deliveryDate, options: nil)?.dateByAddingTimeInterval(60)
    return adjustedDeliveryDate
}
```

Here you get the hour, minute, and second components from the current date, and add them to the delivery date, as the delivery date currently has no time attached. You then use `dateByAddingTimeInterval(_:)` to add a further 60 seconds. This simply manipulates the delivery date into being a whole number of days from the current time, so the timer will display it correctly.

Open **ParcelGlanceContainer.swift** and add the following just above the other outlets:

```
@IBOutlet weak var deliveryTimer: WKInterfaceTimer!
```

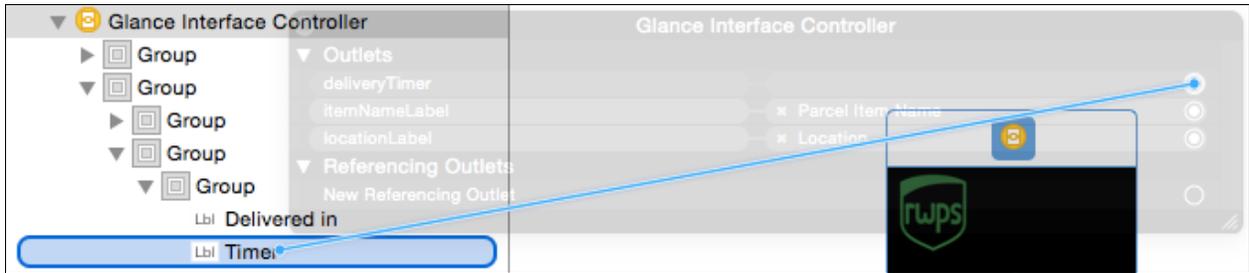
Then replace the existing `willActivate()` with the following:

```
override func willActivate() {
    super.willActivate()
    if let parcel = CoreDataStack.nextParcel() {
        if let deliveryDate = parcel.deliveryDateForUseWithTimer() {
            deliveryTimer.setDate(deliveryDate)
            deliveryTimer.start()
        }
        itemNameLabel.setText(parcel.itemName)
        locationLabel.setText("Currently at \(parcel.location)")
    }
}
```

This simply retrieves a delivery date suitable for use with the `WKInterfaceTimer`, and then sets it on the timer. The rest of the code remains unchanged.



Finally, open **Interface.Storyboard** and right-click on the Glance Interface Controller. Use the popup menu to connect the **deliveryTimer** outlet to the **Timer** object in the group.



Build and run. The Glance will now be displaying the correct number of days until the parcel is delivered, but this time using `WKInterfaceTimer`.

