



Automating User Interface Testing with Instruments

Or “How to find bugs while you sleep”

Michael Creasy

QA Manager, UI Automation & iPhone SDK

Introduction

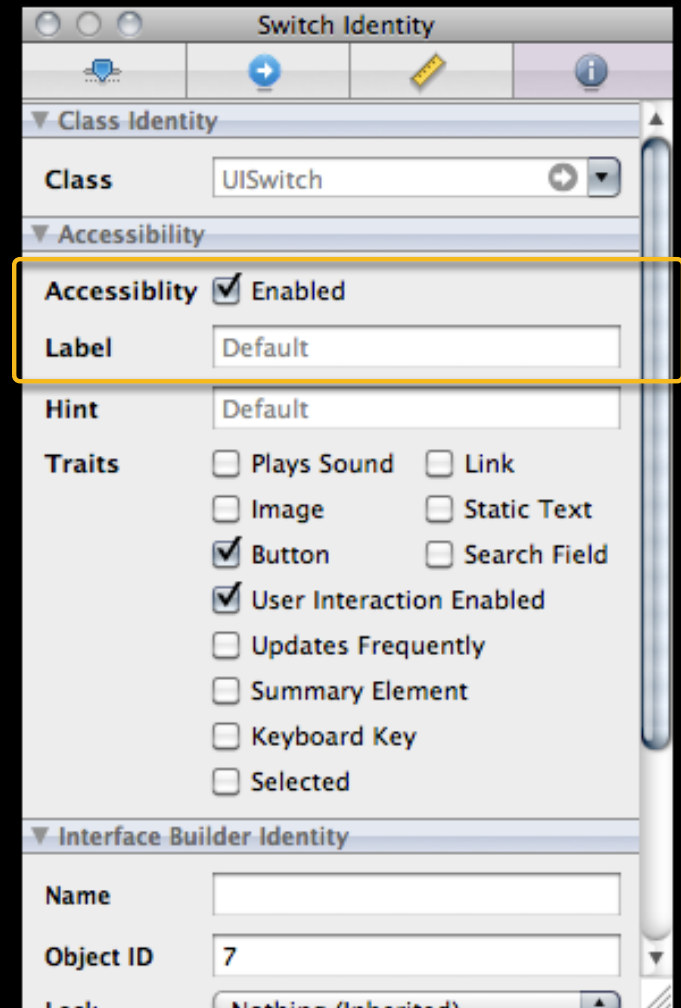
- Why create automated tests?
- What is UI Automation?
- How to automate an application
- Advanced Automation

Why Create Automated Tests?

- Find bugs while you sleep
- Repeatable regressions tests
- Quickly turn around updates to your application

What Is UI Automation?

- Automates UIKit based applications
- Touch based
- iPhone, iPod touch and iPhone Simulator
- Integrated in Instruments
- Accessibility based
- JavaScript automation scripts



Demo

Matt Dreisbach

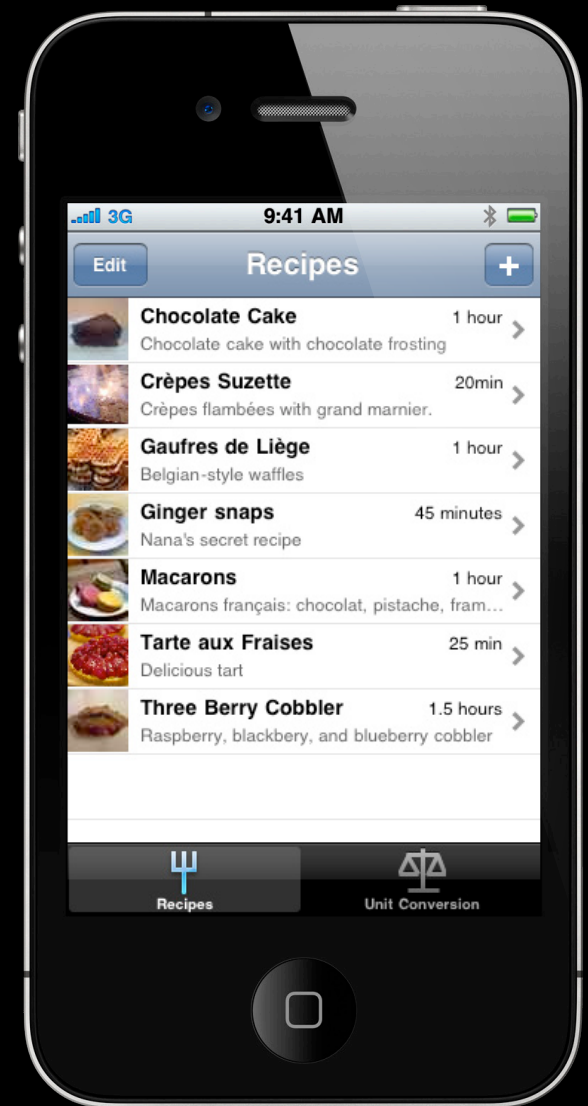
Lead Engineer, UI Automation

How to Automate an Application

Introduction to Elements

- UIAElement—the base element
 - Name
 - Value
 - Elements
 - Parent

Control Hierarchy



Control Hierarchy

- Target application

`UIATarget.localTarget().frontMostApp()`



Control Hierarchy

- Target application
 - Main window

```
UIATarget.localTarget().frontMostApp().mainWindow  
( )
```



Control Hierarchy

- Target application
 - Main window
 - View

```
UIATarget.localTarget().frontMostApp().mainWindow  
().tableViews()[0]
```



Control Hierarchy

- Target application
 - Main window
 - View
 - Element

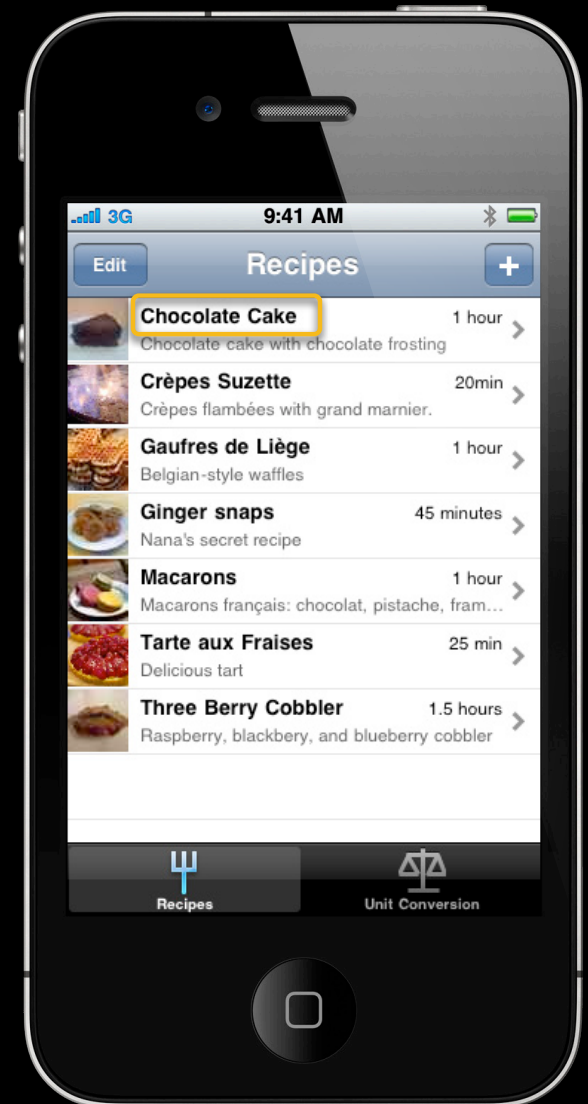
```
UIATarget.localTarget().frontMostApp().mainWindow  
().tableViews()[0].cells()[0]
```



Control Hierarchy

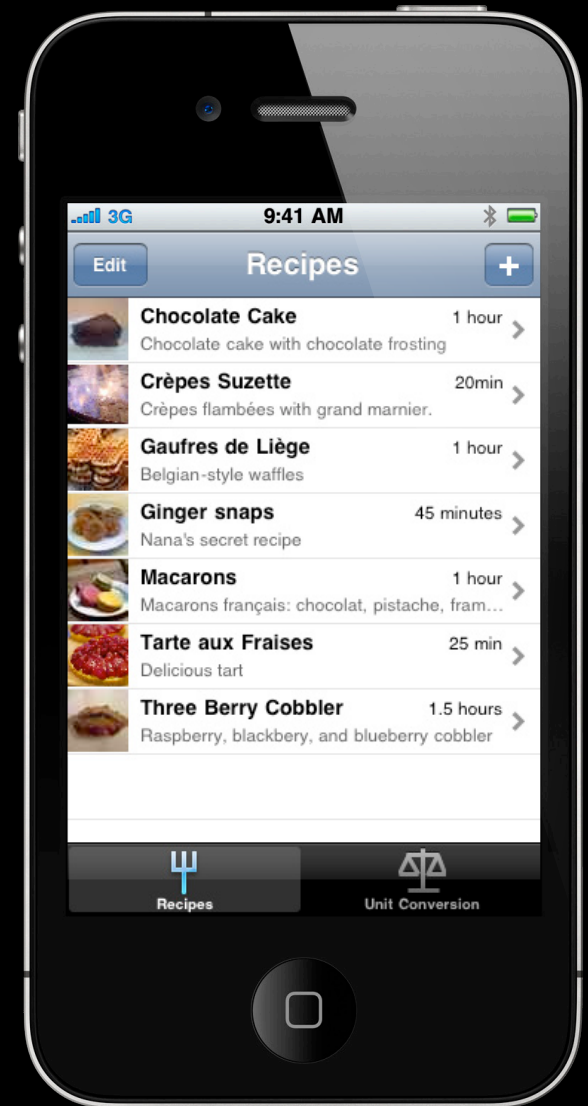
- Target application
 - Main window
 - View
 - Element
 - Child element

```
UITarget.localTarget().frontMostApp().mainWindow  
().tableViews()[0].cells()[0].elements()["Chocolate  
Cake"]
```



Tapping Buttons

```
UITarget.localTarget().frontMostApp  
().navigationBar().buttons()["Add"].tap();
```



Tapping Buttons

```
UITarget.localTarget().frontMostApp()  
  .navigationBar().buttons()["Add"].tap();
```



Text Input

```
var name = "Turtle Pie";  
UITarget.localTarget().frontMostApp()  
.mainWindow().textFields()[0].setValue(name);
```



Text Input

```
var name = "Turtle Pie";  
UITarget.localTarget().frontMostApp()  
.mainWindow().textFields()[0].setValue(name);
```



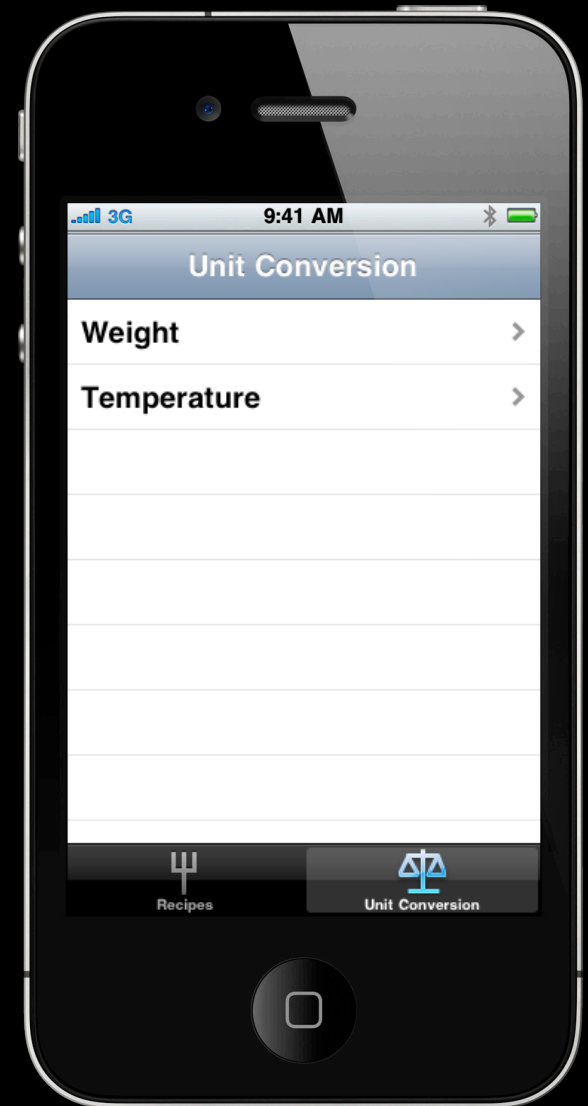
Switching Tabs

```
var tabBar = UIATarget.localTarget().frontMostApp()
    .tabBar();
var selectedTabName = tabBar.selectedButton().name();
if (selectedTabName != "Unit Conversion") {
    tabBar.buttons()["Unit Conversion"].tap();
}
```



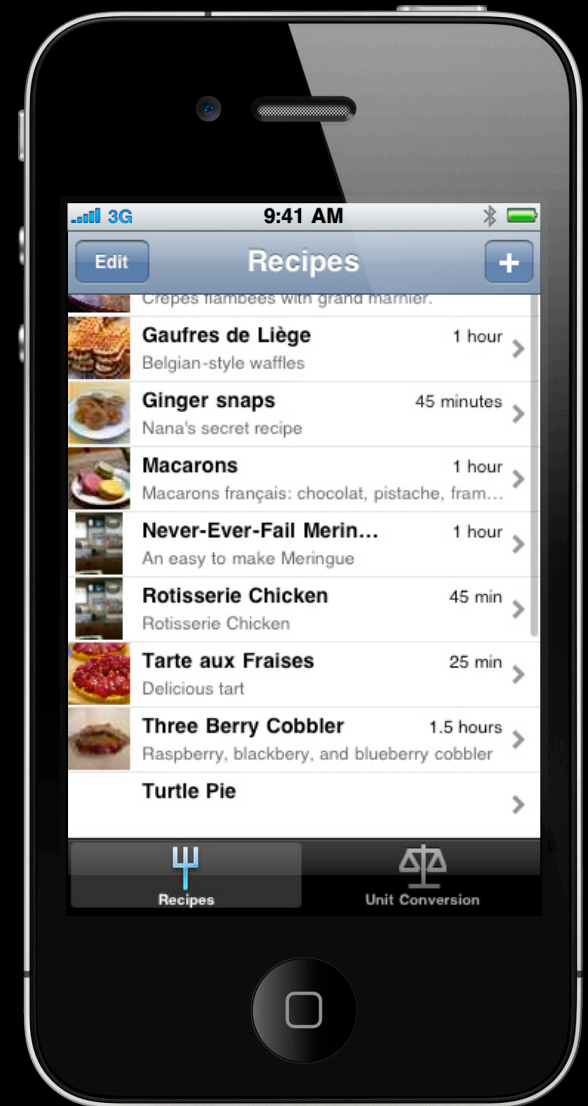
Switching Tabs

```
var tabBar = UIATarget.localTarget().frontMostApp()
    .tabBar();
var selectedTabName = tabBar.selectedButton().name();
    if (selectedTabName != "Unit Conversion") {
        tabBar.buttons()["Unit Conversion"].tap();
    }
}
```



Scrolling

```
UIATarget.localTarget().frontMostApp()  
.mainWindow().tableViews()[0]  
.scrollToElementWithPredicate("name  
beginswith 'Turtle Pie'");
```



Verification



Verification

```
var cell = UIATarget.localTarget().frontMostApp()
    .mainWindow().tableViews()[0].cells()
    .firstWithPredicate("name beginswith 'Turtle Pie'")
if (cell.isValid())
{
    UIALogger.logPass(testName);
}
else
{
    UIALogger.logFail(testName);
}
```



Logging

- Start and end tests

```
var testName = "My first test";
UIALogger.logStart(testName);
...
// test code

...
UIALogger.logPass(testName);
```

Logging

- Logging during a test

```
var testName = "My first test";
UIALogger.logStart(testName);
...
UIALogger.logMessage("Logging about my test");

...
UIALogger.logPass(testName);
```


Logging

- Capture a screenshot

```
var testName = "My first test";
UIALogger.logStart(testName);
...
UIALogger.logMessage("Logging about my test");
UIATarget.localTarget().captureScreenWithName("screenshot1"); // specify a name
for the screenshot
...
UIALogger.logPass(testName);
```

Demo

Matt Dreisbach

Lead Engineer, UI Automation

Advanced Automation

Handling Unexpected Alerts



Handling Unexpected Alerts

- Automatically dismiss alerts



Handling Unexpected Alerts

```
UIATarget.onAlert = function onAlert(alert) {  
    var title = alert.name();  
  
    UIALogger.logWarning("Alert with title '" +  
title + "' encountered!");  
  
    return false; // use default handler  
}
```



Handling Expected Alerts



Handling Expected Alerts



Handling Expected Alerts

```
UIATarget.onAlert = function onAlert(alert) {  
    var title = alert.name();  
  
    UIALogger.logWarning("Alert with title '" +  
title + "' encountered!");  
  
    return false; // use default handler  
}
```



Handling Expected Alerts

```
UITarget.onAlert = function onAlert(alert) {  
    var title = alert.name();  
  
    UIALogger.logWarning("Alert with title '" +  
title + "' encountered!");  
  
    if (title == "Duplicate Recipe Name") {  
        alert.buttons()["Continue"].tap();  
  
        return true; // bypass default handler  
    }  
  
    return false; // use default handler  
}
```



Multitasking

- Deactivate application for a given time
- Reactivating the application is handled automatically

```
UIALogger.logMessage("Deactivating app");  
UIATarget.localTarget().deactivateAppForDuration(10);  
UIALogger.logMessage("Resuming test after deactivation");
```

Orientation



Orientation

```
var target = UIATarget.localTarget();  
var app = target.frontMostApp();  
  
// set landscape left  
target.setDeviceOrientation(UIA_DEVICE_ORIENTATION_LANDSCAPELEFT);  
UIALogger.logMessage("Current orientation is " + app.interfaceOrientation());
```

Orientation

```
var target = UIATarget.localTarget();
var app = target.frontMostApp();

// set landscape left
target.setDeviceOrientation(UIA_DEVICE_ORIENTATION_LANDSCAPELEFT);
UIALogger.logMessage("Current orientation is " + app.interfaceOrientation());

// portrait
target.setDeviceOrientation(UIA_DEVICE_ORIENTATION_PORTRAIT);
UIALogger.logMessage("Current orientation is " + app.interfaceOrientation());
```

Advanced Touch Interaction

- Taps
- Pinches
- Drags and Flicks

Advanced Touch Interaction

- Taps

```
UIATarget.localTarget().tap({x:100, y:200});
```

```
UIATarget.localTarget().doubleTap({x:100, y:200});
```

```
UIATarget.localTarget().twoFingerTap({x:100, y:200});
```


Advanced Touch Interaction

- Pinches

```
UIATarget.localTarget().pinchOpenFromToForDuration({x:20, y:200}, {x:300, y:200},  
2);
```

```
UIATarget.localTarget().pinchCloseFromToForDuration({x:20, y:200}, {x:300, y:  
200}, 2);
```

Advanced Touch Interaction

- Drag and Flick

```
UIATarget.localTarget().dragFromToForDuration({x:160, y:200}, {x:160, y:400}, 1);
```

```
UIATarget.localTarget().flickFromTo({x:160, y:200}, {x:160, y:400});
```

Timeout

- Wait for elements to appear

```
UIATarget.localTarget().frontMostApp().navigationBar().buttons()["Add"];
```

- Five second default timeout

- Custom timeouts

```
UIATarget.localTarget().pushTimeout(2);
```

```
...
```

```
UIATarget.localTarget().popTimeout();
```

More Information

Mike Jurewitz

Developer Tools and Performance Evangelist

jurewitz@apple.com

Documentation

UI Automation Reference Collection

<https://developer.apple.com/wwdc/iphone/library/documentation/DeveloperTools/Reference/UIAutomationRef/index.html>

Instruments User Guide

<https://developer.apple.com/wwdc/iphone/library/documentation/DeveloperTools/Conceptual/InstrumentsUserGuide/index.html>

Apple Developer Forums

<http://devforums.apple.com>

Related Sessions

Accessibility on iPhone OS

Nob Hill
Wednesday 4:30PM

Labs

User Interface Automation Testing Lab

Developer Tools Lab A
Wednesday 4:30PM

User Interface Automation Testing Lab

Developer Tools Lab A
Thursday 9:00AM

Q&A



