

Web Accessibility and Automation

Combining accessibility and automated testing on iOS

Session 519

James Craig

iOS Accessibility Team

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

What Is Accessibility?

And why does it matter for my application?









50 million

Americans have a disability

U.S. Census Bureau

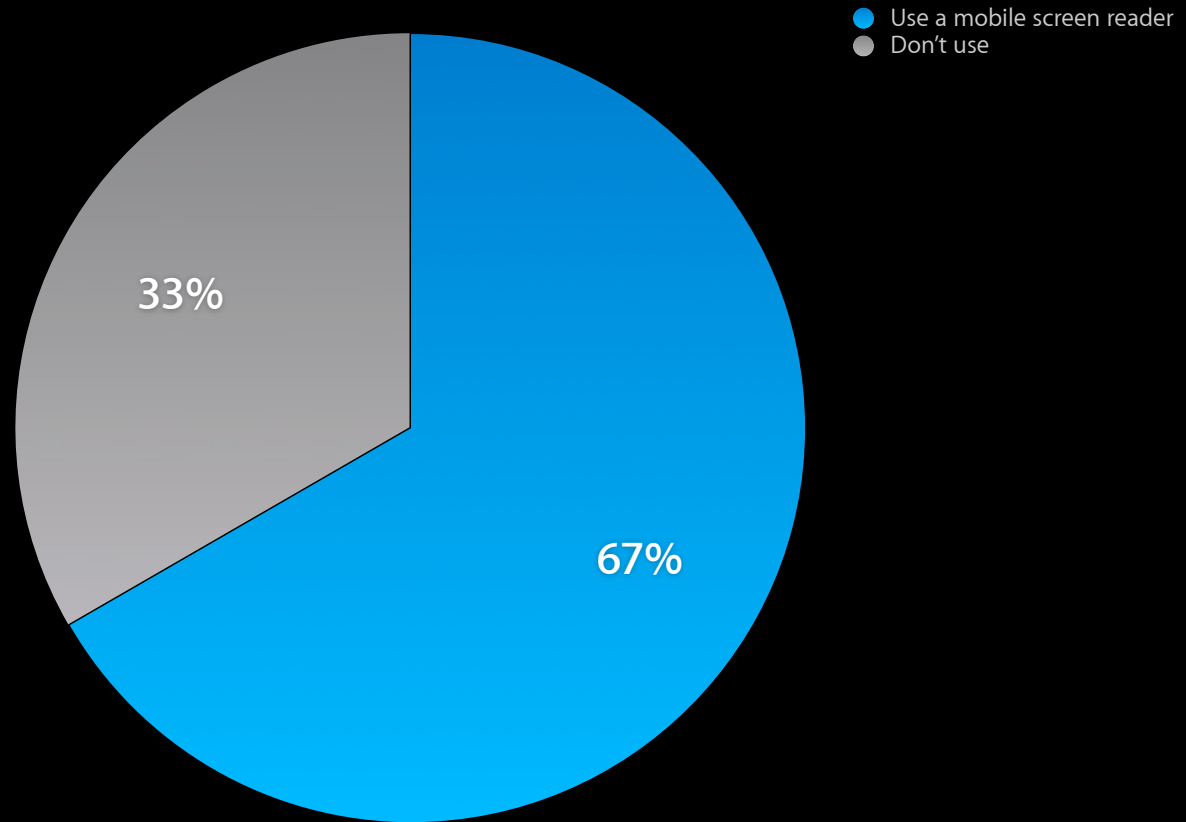
10 million

Americans are visually impaired

American Foundation for the Blind, Public Policy Center. Statistical Snapshots, March 2008

Mobile Screen Reader Usage

Blind users of handheld devices



Source: WebAIM Screen Reader Survey 2011

Accessibility on iOS

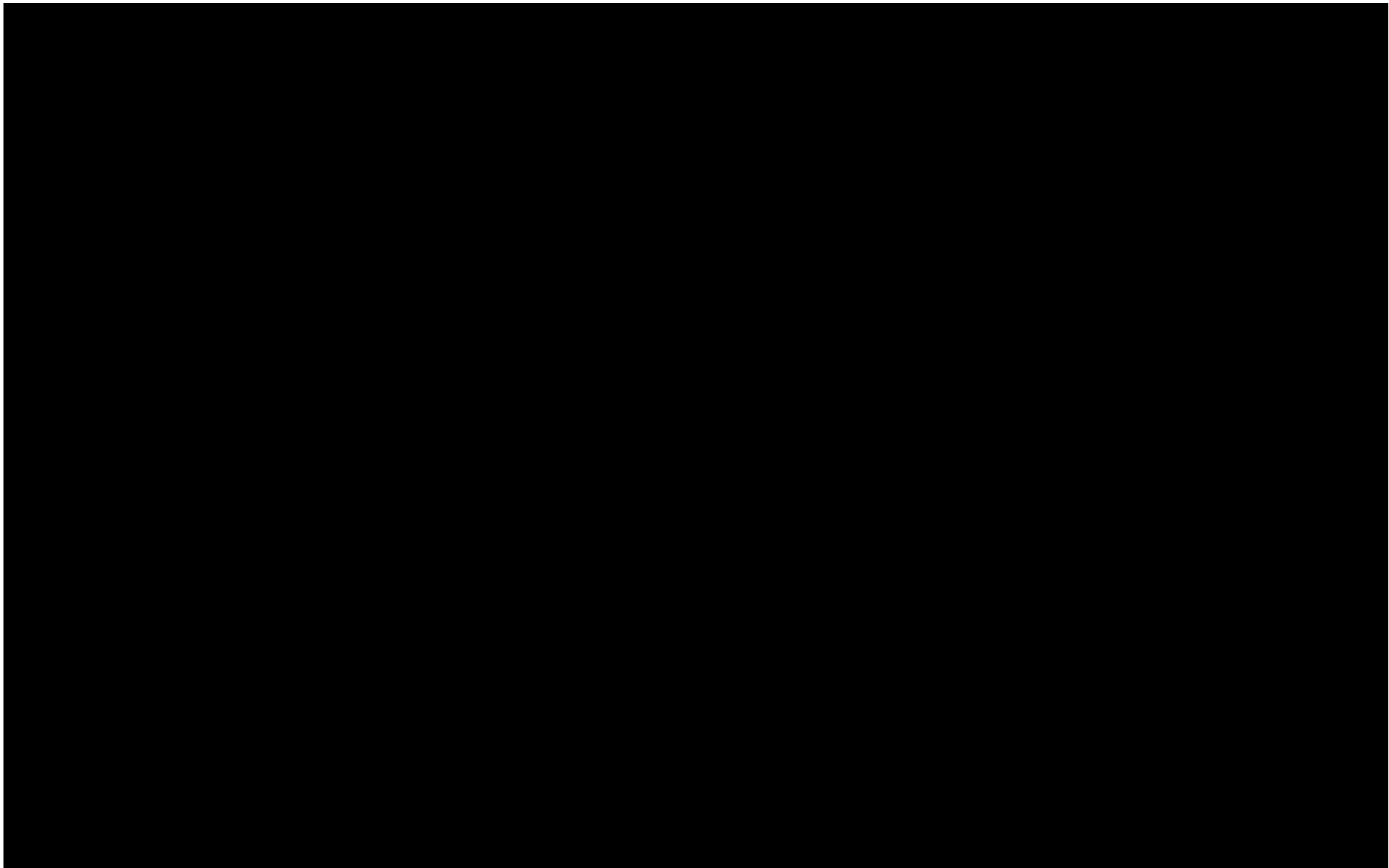
Assistive technology, built-in since iOS 3



“When it comes to embedding accessibility, Apple has set the standard in recent years...It is now up to other manufacturers to follow their lead.”

Robin Spinks, Royal National Institute of Blind People (RNIB), United Kingdom

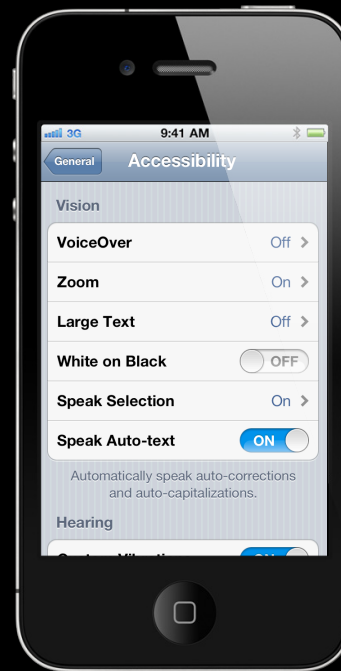




Your
App

Accessibility Settings

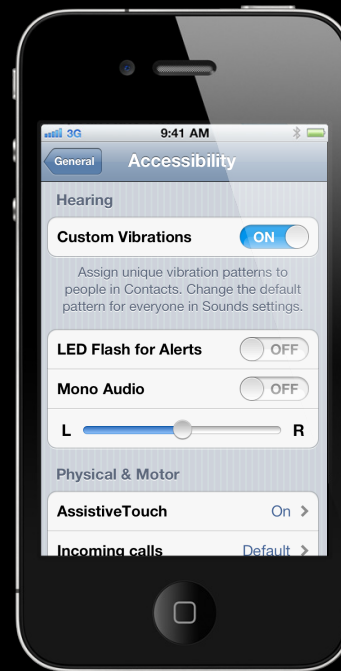
Many new features in iOS 5



- ← VoiceOver
 - ← Zoom
 - ← Large Text
 - ← White on Black
 - ← Speak Selection
 - ← Speak Auto-text
- 5eText magnification
Zoom reader for the visually impaired
Text magnification
font size in Mail, Notes, Messages, Calendar, and other apps
Speak Selection
Speak Auto-text
with certain visual impairments, without including some forms of color blindness
Hear auto-corrections and capitalizations

Accessibility Settings

Many new features in iOS 5



5

← Custom Vibrations

5

← Assign unique vibration patterns for incoming calls, tap to record your own patterns

← LED Flash for Alerts

← Mono Audio

← Turn both camera LED on calls or alerts when the screen is locked
← Turn both stereo channels into each earphone, or set a custom balance

Accessibility Settings

Many new features in iOS 5



- ← AssistiveTouch
- ← Route Incoming Calls
- ← Control the entire iOS interface to a Bluetooth headset or the phone speaker

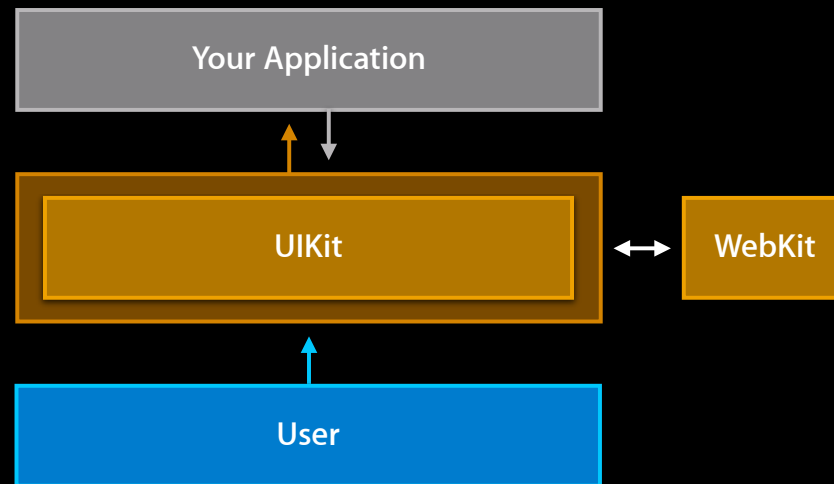
Demo

Using VoiceOver on iOS

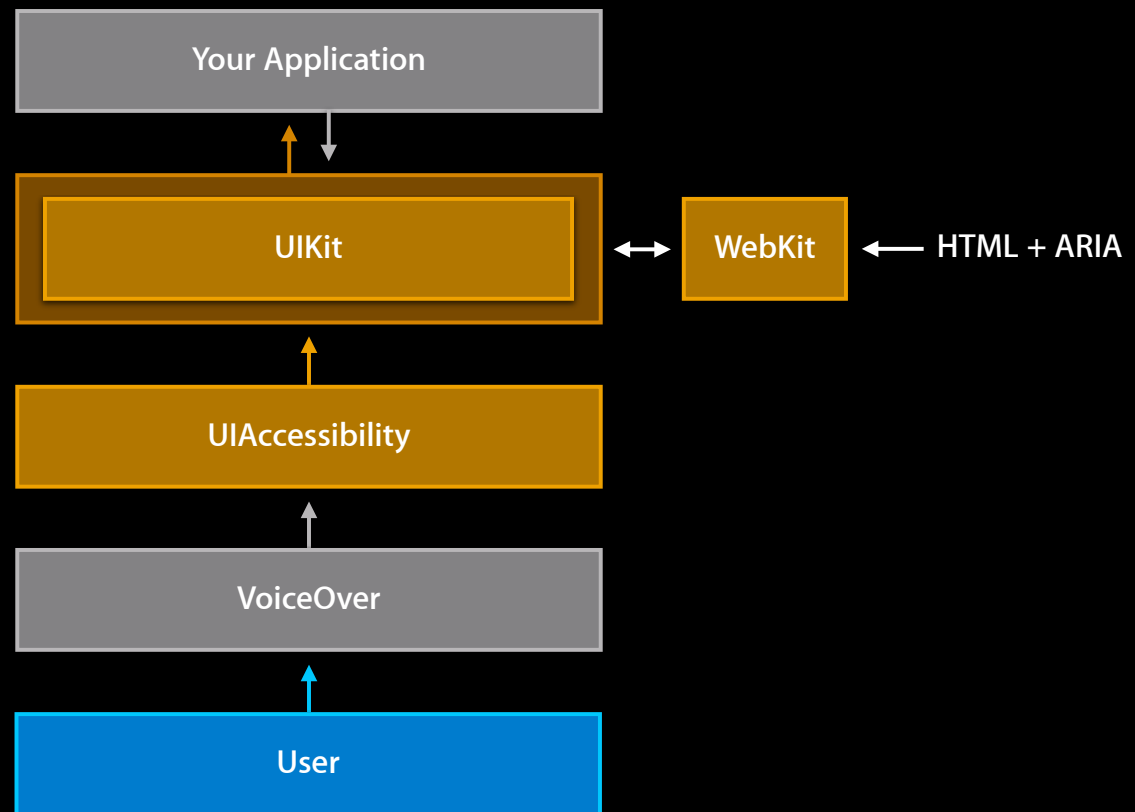
Accessibility in Web Views

Leveraging WebKit and UIAccessibility

Accessibility Architecture on iOS



Accessibility Architecture on iOS



Web Accessibility

Techniques to apply across the board

- Sites displayed in Safari and other accessible browsers
- Sites displayed in native applications like feed readers
- Web views displaying local content using web technologies




HTML Accessibility

Label all images, form elements, etc.



- Images

 ``

 ``

- Form elements

 `User: <input type="text">`

 `<label for="user">User:</label> <input id="user" type="text">`

HTML Accessibility

Use native HTML5 form controls



- Date, datetime, time, month, week, year, etc.



```
<input type="date">
```

- Range (HTML 5 "slider" controls)

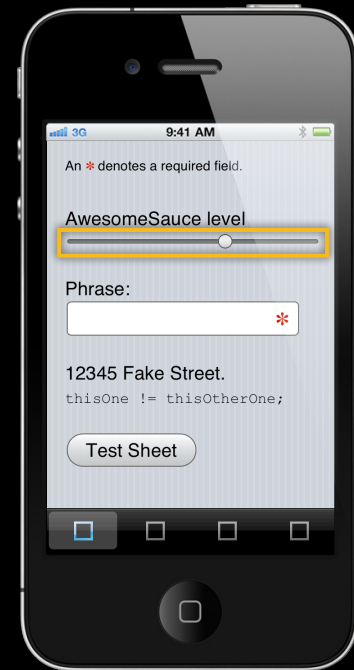


```
<input type="range">
```

- Required attribute



```
<input type="text" required>
```



HTML Accessibility

Avoid custom controls and gesture overrides

- New! Single-finger scrolling with `overflow:auto;`
- New! Fixed positioning (really) with `position:fixed;`

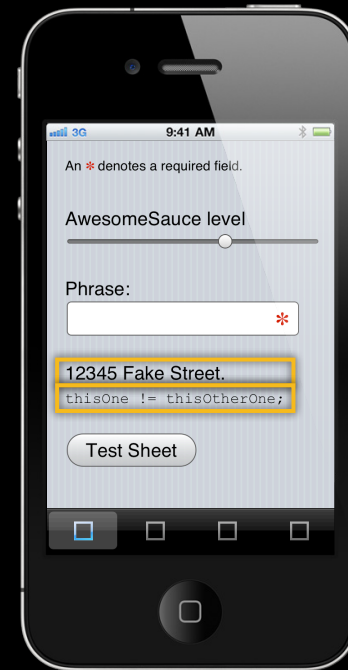


CSS 3 Speech

Control screen reader verbosity

```
.address, .phone, .zip {  
    speak: digits;  
}  
code {  
    speak: literal-punctuation;  
}
```

Note: CSS 3 Speech Module is a draft and is subject to change



Demo

Resolving accessibility errors using HTML5 and CSS3



Using WAI-ARIA

Accessible Rich Internet Applications

- Retrofit old content without gutting it

```
<a href="#" role="button"> Save </a>
```

- Redefine roles where needed

```
<div role="dialog"> ... </div>  
el.setAttribute('role', 'dialog');
```

- Override or assign states and properties

```
<div role="checkbox" aria-checked="mixed"> ... </div>  
el.setAttribute('aria-checked', 'mixed');
```

- Update your application state when you change element style

```
el.className = 'selected';  
el.setAttribute('aria-selected', 'true');
```



Focus Management

Setting VoiceOver cursor focus on iOS

- VoiceOver cursor focus is not the same as keyboard focus!
- Set focus in a web view

```
var button = document.getElementsByTagName('button')[0];  
button.focus();
```

- Using tabIndex if needed

```
<div id="cb" role="checkbox"> Spam </div>  
var checkbox = document.getElementById('cb');  
checkbox.tabIndex = 0; // make the div focusable first  
checkbox.focus(); // or this would cause an error in desktop browsers
```

Demo

Resolving accessibility errors using ARIA
and managed focus

UI Automation in Instruments

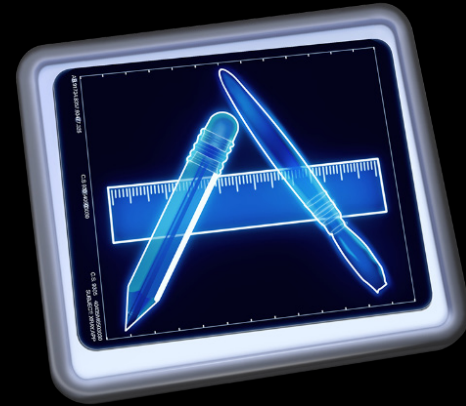
Creating automated tests for your app

Why Create Automated Tests?

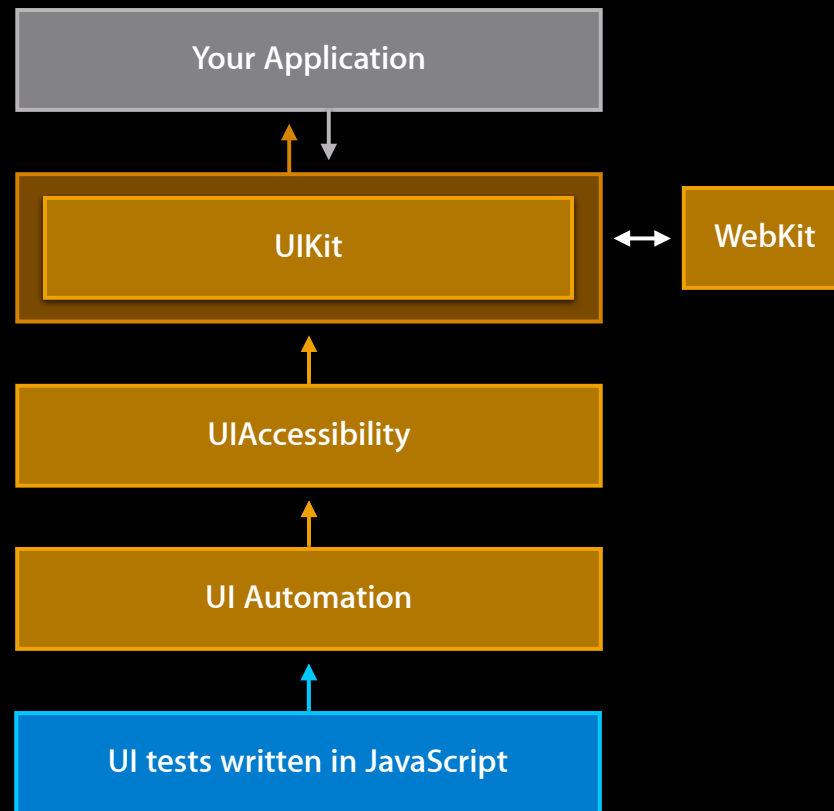
- Easily repeatable regression tests
- Stress your app with longevity tests
- Faster testing for iOS application updates
- Confidence in product quality
- Bonus: Catch some accessibility errors

What Is UI Automation on iOS?

- Integrated into Instruments app
- Allows automation of iOS apps
- Leverages UIAccessibility
- Tests written in JavaScript



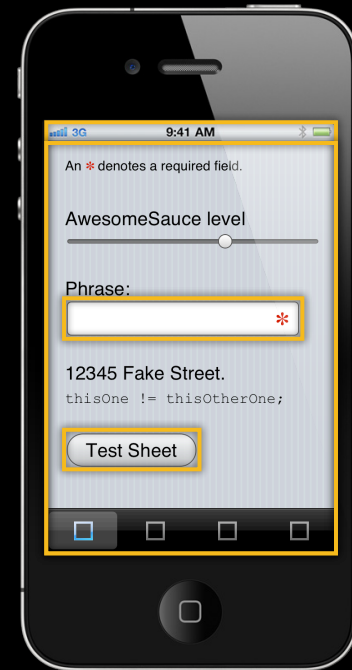
UI Automation and UIAccessibility



UI Automation

Assign references to views and elements

```
var target    = UIATarget.localTarget();
var app      = target.frontMostApp();
var win      = app.mainWindow();
var web      = win.scrollViews()[0].webViews()[0];
var textField = web.textFields()[0];
var myButton = web.buttons()['Test Sheet'];
```



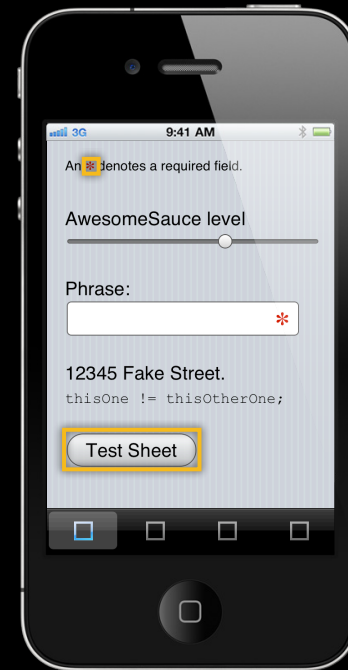
UI Automation

Inspect the interface

```
// output accessible name of the first image
web.images()[0].label();
> "asterisk"

// make sure it's not disabled
myButton.isEnabled();
> true

// is the Cancel sheet button in the accessible tree?
web.buttons()['Cancel'].checkIsValid();
> false
```



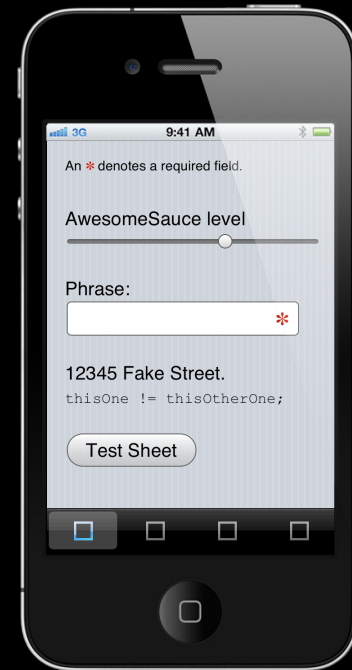
UI Automation

Trigger user events

```
// tap a button  
myButton.tap();
```

```
// touch and hold, then release  
web.textFields()[0].touchAndHold(2);
```

```
// simulate a shake event  
UIATarget.localTarget().shake();
```



UI Automation

Logging with UIALogger and target methods

```
var testName = 'Add User Scenario';
UIALogger.logStart(testName);
...
UIALogger.logMessage('Logging info: ' + withSomeVariable);
...
UIATarget.localTarget().logElementTree();
UIATarget.localTarget().captureScreenWithName('Edit Dialog');
...
if (someFailureCase) UIALogger.logFail(testName); // explicit failure
UIALogger.logPass(testName);
```

Demo

Using UI Automation with Instruments

More Information

Vicki Murley

Internet & Web Technologies Evangelist
vicki@apple.com

Michael Jurewitz

Developer Tools and Frameworks Evangelist
jurewitz@apple.com

Apple Accessibility

<http://www.apple.com/accessibility>

UI Automation on iOS

<http://developer.apple.com/>

Apple Developer Forums

<http://devforums.apple.com>

Takeaways

- Accessibility bugs are usually easy to fix
- You should be testing already; accessibility makes it easier
- Accessibility is **not** about code or compliance; it's about **people**

Q&A

