



Accessibility on iOS

Make an app for everyone

Chris Fleizach
iOS Accessibility

About me

- B.S. Duke '02 
- Peace Corps (Tonga) 
- M.S. UCSD '06 
- Four years at Apple (including internship)
▪ VoiceOver (Mac OS X) 
▪ iOS Accessibility 

Accessibility

- Closed Captioning
- Zoom
- Screen readers
- *Many others...*



Settings > General > Accessibility



“I predict that the visually impaired community will agree that the iPhone was the single biggest game changing piece of technology for the assistive tech industry in modern times.”

Josh de Lioncourt, maccessibility.net

Demo

VoiceOver and Accessibility and Video

What You'll Learn

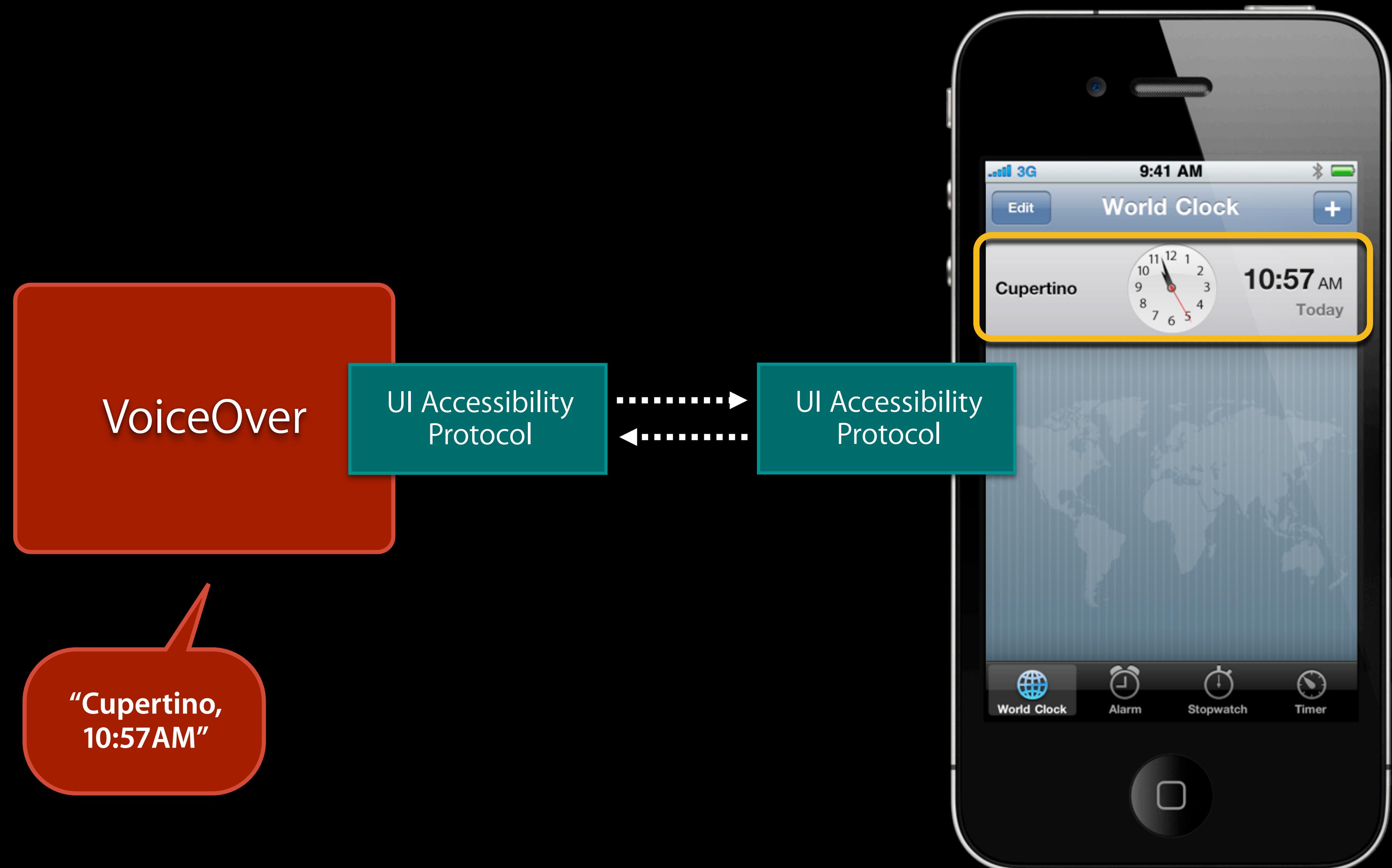


- iOS Accessibility Overview
- UIAccessibility Protocol
 - Accessibility Attributes
 - Accessibility Containers
 - Accessibility Actions
 - VoiceOver-specific API
- Best Practices
- Apps for users with disabilities

Accessibility API

- An accessibility interface allows another process to
 - Access individual items in the user interface
 - Query for status, state and description
 - Perform actions and events
- The remote process is an Assistive Technology (i.e. VoiceOver)

Accessibility Architecture



Making Accessible Apps

- Accessibility API introduced in iOS 3.0
 - Lives in UIKit in UIAccessibility.h
- Allows an app to provide UI information to VoiceOver
- Standard controls in UIKit already accessible

UIAccessibility API: Attributes

Accessibility attributes return information about the element

- (BOOL)isAccessibilityElement
- (NSString *)accessibilityLabel
- (UIAccessibilityTraits)accessibilityTraits
- (CGRect)accessibilityFrame
- (NSString *)accessibilityHint
- (NSString *)accessibilityValue

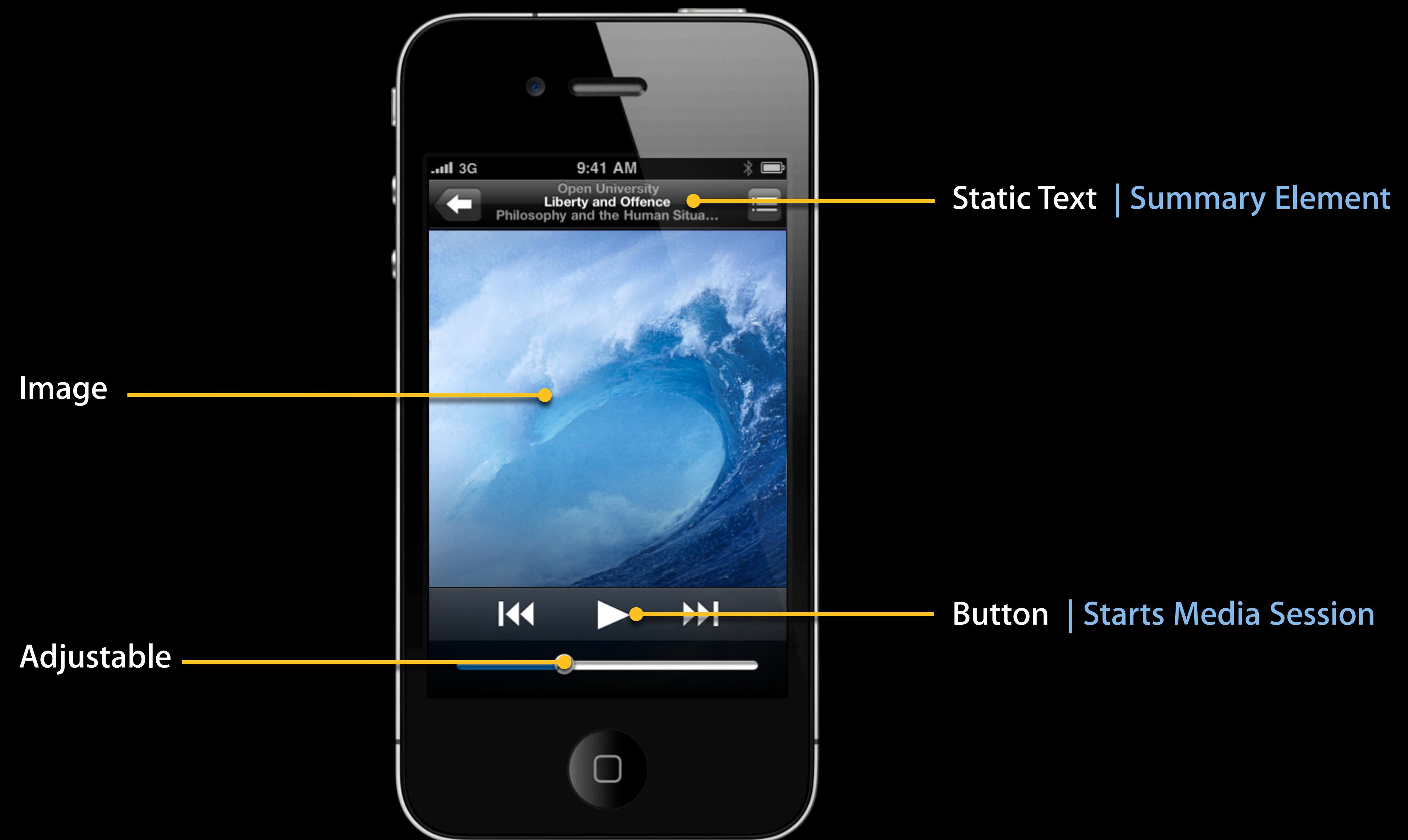
Common Accessibility Attributes

- `(BOOL)isAccessibilityElement`
 - Return YES to make VoiceOver see this element
- `(NSString *)accessibilityLabel`
 - A textual representation of the element

Accessibility Traits

- `(UIAccessibilityTraits)accessibilityTraits`
 - Defines behavior
 - Bitmask of integers

Accessibility Traits

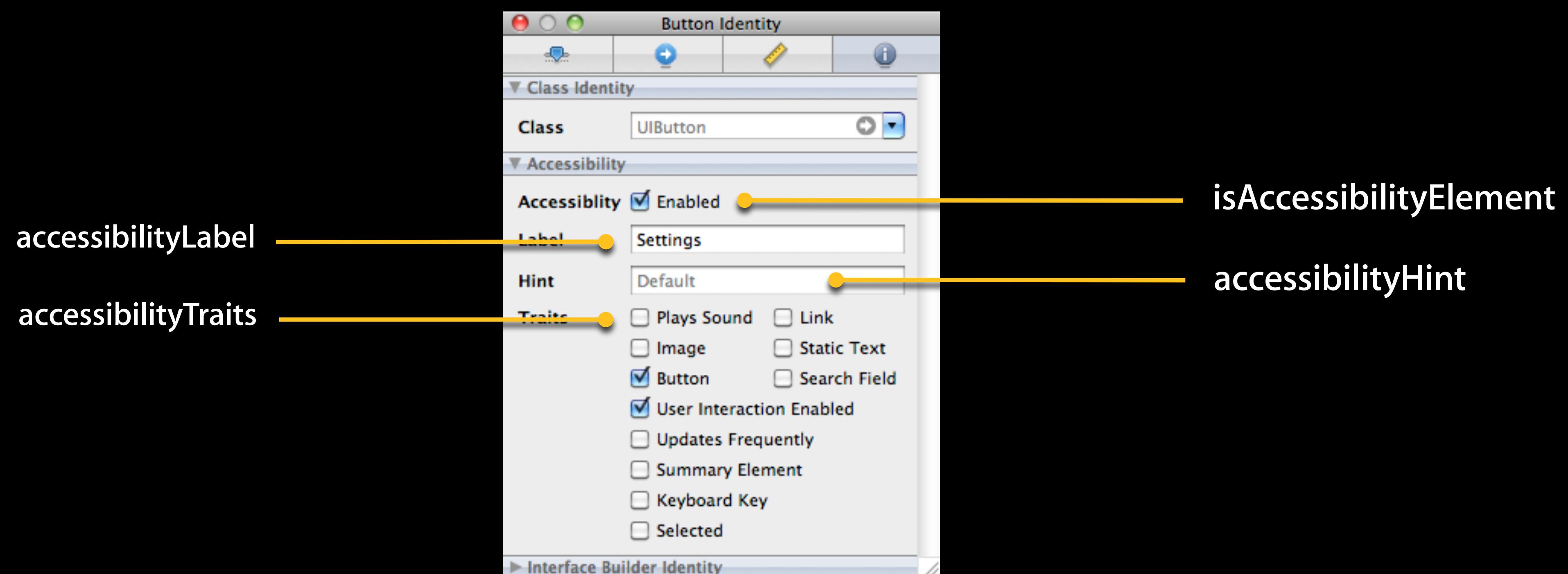


Other Accessibility Attributes

- `(CGRect)accessibilityFrame`
 - The onscreen rectangle for the UI element
- `(NSString *)accessibilityValue`
 - A dynamically changing value that describes the UI element
- `(NSString *)accessibilityHint`
 - A string that provides additional help

Adding Accessibility with IB

Change simple accessibility values



Demo

Introduction to iPhone Accessibility

What You'll Learn



- UIAccessibility Protocol
 - Accessibility Attributes
 - Adding Accessibility in Code
 - Accessibility Containers
 - Accessibility Actions
 - VoiceOver-specific API
- Best Practices
- Apps for users with disabilities

Adding Accessibility in Code

If accessibility values don't change

```
- (void)awakeFromNib {
    ...
    UIControl *control = [[UIControl alloc] initWithFrame:frame];
    control.isAccessibilityElement = YES;
    control.accessibilityLabel = @"Play";
    [window addSubview:control];
    ...
}
```

Adding Accessibility in Code

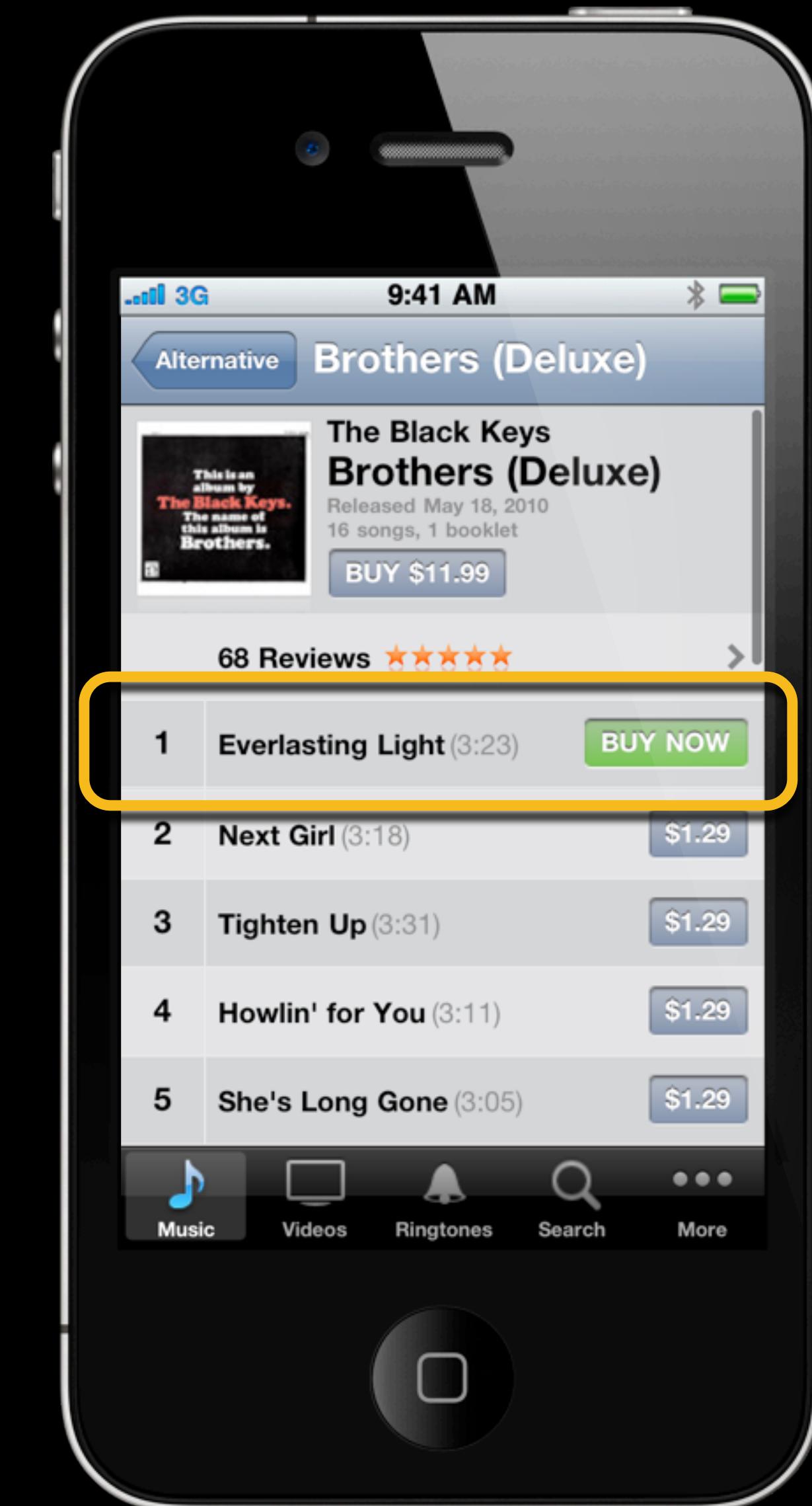
If accessibility values change

```
@implementation MyTemperatureView  
  
- (BOOL)isAccessibilityElement {  
    return YES;  
}  
  
- (NSString *)accessibilityLabel {  
    return @"Current Temperature";  
}  
  
- (NSString *)accessibilityValue {  
    return [cityWeather currentTemperatureAsString];  
}  
  
@end
```

Notifications

Tell VoiceOver something happened

```
UIAccessibilityPostNotification(  
    UIAccessibilityScreenChangedNotification,  
    nil);  
  
UIAccessibilityPostNotification(  
    UIAccessibilityLayoutChangedNotification,  
    nil);
```



Notifications

Tell VoiceOver something happened

```
UIAccessibilityPostNotification(  
    UIAccessibilityAnnouncementNotification,  
    @"89 degrees east");
```



Demo

Setting Accessibility Attributes

What You'll Learn



- UIAccessibility Protocol
 - Accessibility Attributes
 - Accessibility Containers
 - Accessibility Actions
 - VoiceOver-specific API
- Best Practices
- Apps for users with disabilities

Accessibility Containers

- For custom views with multiple parts
- Returns accessibility “sub-elements”

MonthView : UIView



Accessibility Containers

- For custom views with multiple parts
- Returns accessibility “sub-elements”

“October 14th”



Accessibility Containers

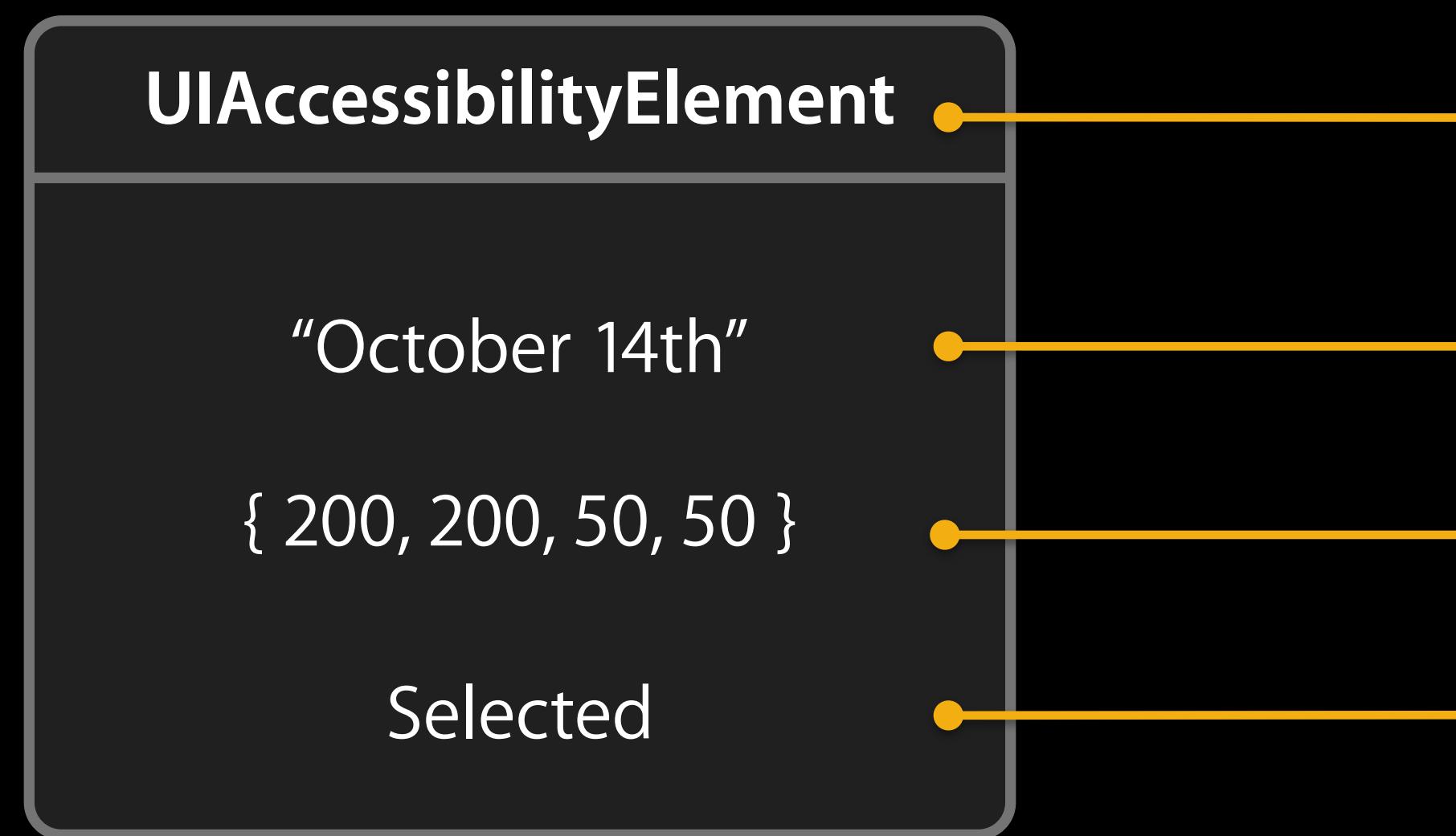
- For custom views with multiple parts
- Returns accessibility “sub-elements”

“October 11th”



Creating a Sub Element

Make a UIAccessibilityElement



```
UIAccessibilityElement *dayPart = [[UIAccessibilityElement alloc]
    initWithAccessibilityContainer:monthView];
dayPart.accessibilityLabel = @"October 14th";
dayPart.accessibilityFrame = CGRectMake(200, 200, 50, 50);
dayPart.accessibilityTraits = UIAccessibilityTraitSelected;
[accessibilityElements addObject:dayPart];
```

Accessibility Container

Mirrors NSArray

```
@implementation MonthView
- (NSInteger)accessibilityElementCount {
    return [accessibilityElements count];
}
- (id)accessibilityElementAtIndex:(NSInteger)index {
    return [accessibilityElements objectAtIndex:index];
}
- (NSInteger)indexOfAccessibilityElement:(id)element {
    return [accessibilityElements indexOfObject:element];
}
@end
```

Demo

Accessibility Container Protocol

What You'll Learn



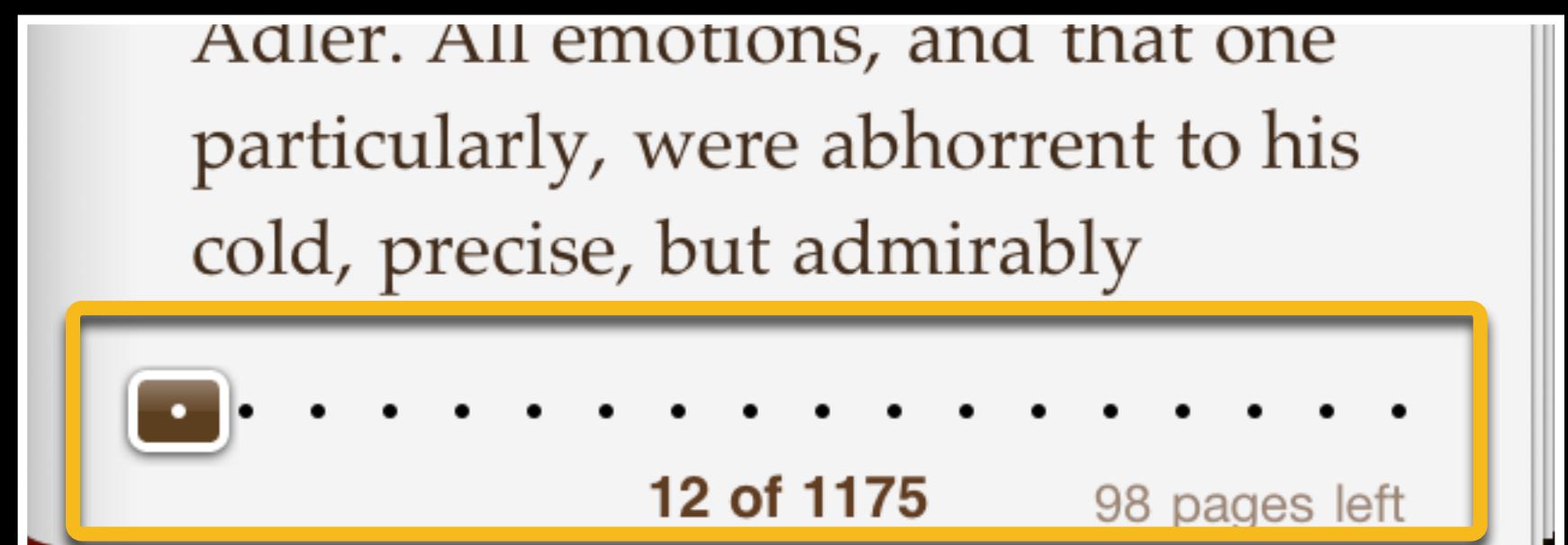
- UIAccessibility Protocol
 - Accessibility Attributes
 - Accessibility Containers
 - Accessibility Actions
 - VoiceOver-specific API
- Best Practices
- Apps for users with disabilities

Accessibility Actions

Adjustable elements



```
@implementation PageControl
- (UIAccessibilityTraits)accessibilityTraits {
    return [super accessibilityTraits] |
        UIAccessibilityTraitAdjustable;
}
- (void)accessibilityIncrement {
    [self moveToNextPage];
}
- (void)accessibilityDecrement {
    [self moveToPreviousPage];
}
@end
```



Accessibility Actions

Scrollable elements



```
@implementation CityView  
  
-(BOOL)accessibilityScroll:(UIAccessibilityScrollDirection)direction {  
  
if (direction == UIAccessibilityScrollDirectionLeft) {  
    [self scrollLeft];  
    UIAccessibilityPostNotification (UIAccessibilityPageScrolled,  
        @"Showing Cupertino");  
    return YES;  
}  
else if (direction == UIAccessibilityScrollDirectionRight) {  
    [self scrollRight];  
    UIAccessibilityPostNotification (UIAccessibilityPageScrolled,  
        @"Showing San Diego");  
    return YES;  
}  
return NO;  
}
```



Demo

Accessibility Actions

What You'll Learn



- UIAccessibility Protocol
 - Accessibility Attributes
 - Accessibility Containers
 - Accessibility Actions
 - VoiceOver-specific API
- Best Practices
- Apps for users with disabilities

VoiceOver-specific API



`BOOL UIAccessibilityIsVoiceOverRunning()`

- Is VoiceOver on?

`UIAccessibilityVoiceOverStatusChanged`

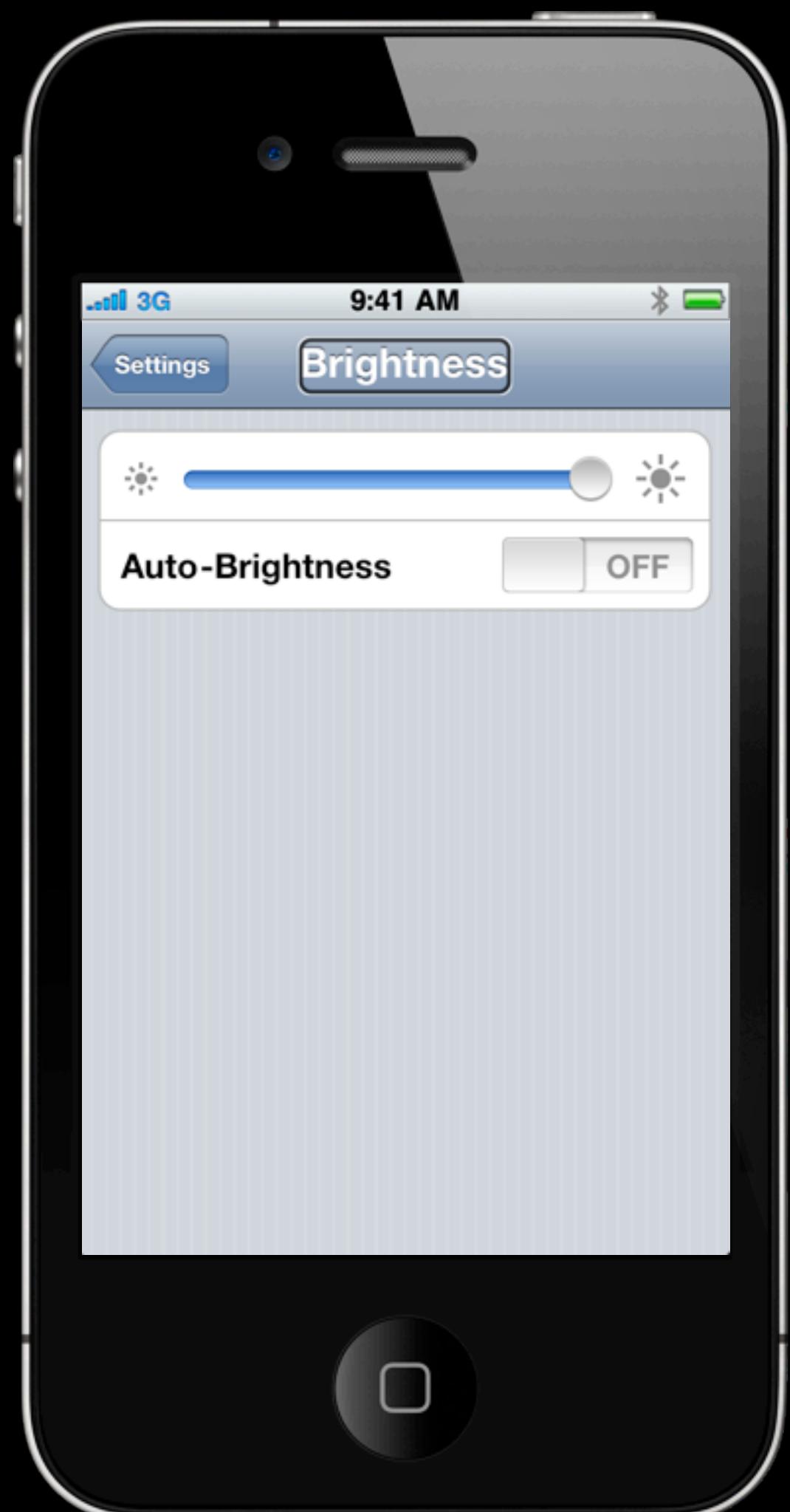
- A notification to determine when VoiceOver is enabled.



VoiceOver-specific API



- `(BOOL)accessibilityElementIsFocused`
 - Is VoiceOver focused on this element?
- `(void)accessibilityElementDidBecomeFocused`
 - Did VoiceOver focus on this element?
- `(void)accessibilityElementDidLoseFocus`
 - Did VoiceOver focus leave this element?



Best Practices

- Use short, concise labels
 - ✓ Good: “Add city”
 - ✗ Bad: “Adds a city to the list of cities”

Best Practices

- Don't include the type information in the label
 - ✓ Good: "Remove city"
 - ✗ Bad: "Remove city button"

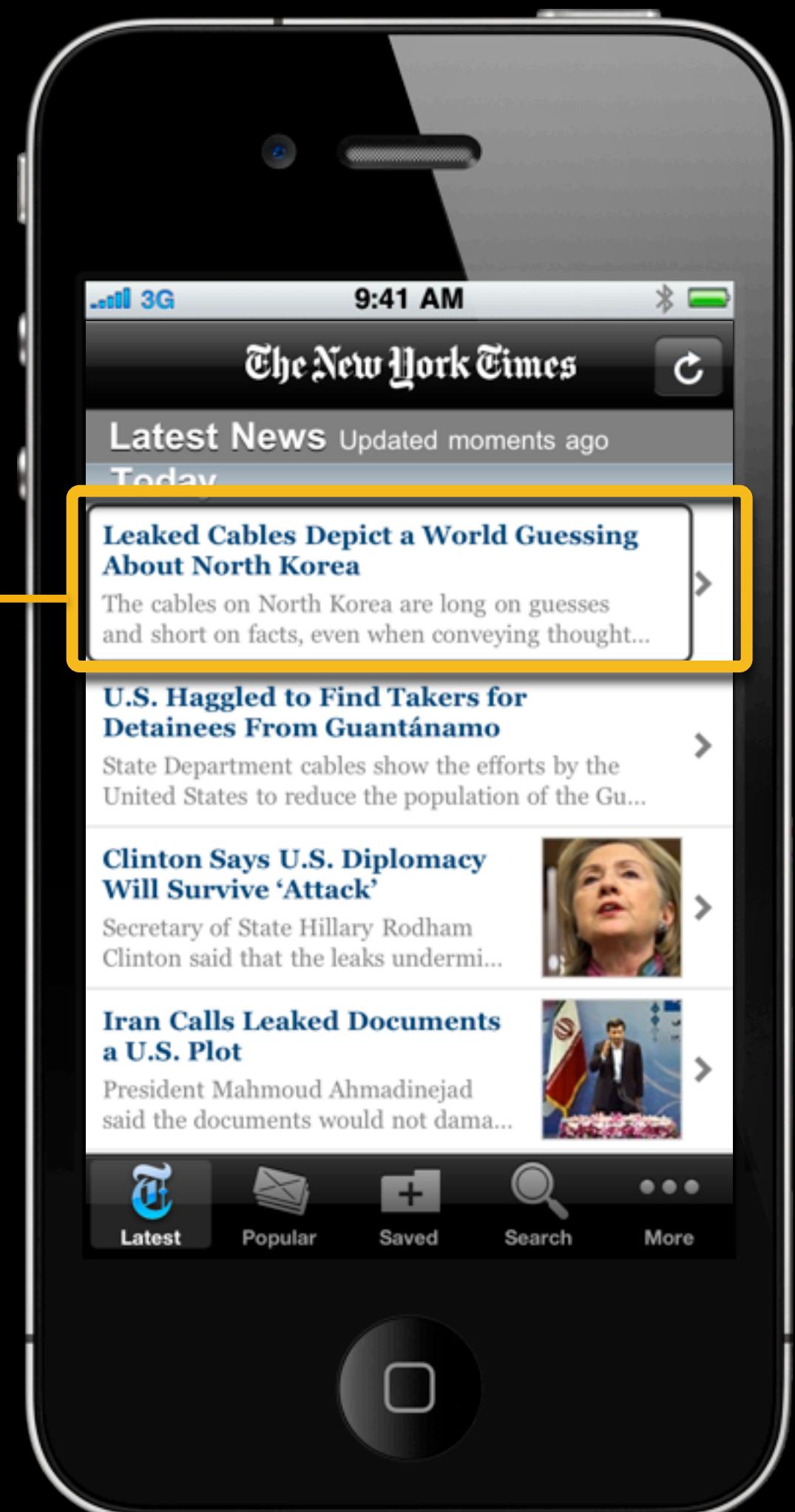
Best Practices

- Use localized strings for labels and hints
-  VoiceOver works in over 30 languages

Best Practices

- With custom table view cells
 - ✓ Only need to add accessibilityLabel
 - ✓ If there are multiple strings, concatenate with commas

```
@implementation NewsTableViewCell
- (NSString *)accessibilityLabel
{
    return [NSString stringWithFormat:@"%@, %@",  
        [self header], [self summary]];
}
@end
```



Demo

Accessible Bowling

What You'll Learn



- UIAccessibility Protocol
 - Accessibility Attributes
 - Accessibility Containers
 - Accessibility Actions
 - VoiceOver-specific API
- Best Practices
- Apps for users with disabilities

Apps for users with disabilities

- Accessibility features on iOS has created a large user base
- Opportunity to develop apps addressing specific needs



Digit-Eyes



SayText



oMoby



iCanSee



iPrompts



Proloquo2Go

Summary

- Add accessibility
 - Increases user base
 - Heaps of praise
 - Karmic improvement
- Apps for users with disabilities
 - Incredible innovation awaits



More Information

Documentation

Accessibility Programming Guideline for iOS

Search on <http://developer.apple.com/> for Accessibility

UIAccessibility Protocol Reference

Search on <http://developer.apple.com/> for UIAccessibility

VoiceOver User Manual

<http://support.apple.com/manuals/iphone>

