

UI Automation JavaScript Reference

Contents

UI Automation JavaScript Reference 10

Accessing and Using User Interface Elements 10

Recording Results With the Log 11

Handling Alerts 11

Classes 13

UIActionSheet Class Reference 14

Overview 14

Tasks 17

Methods 17

UIActivityIndicator Class Reference 19

Overview 19

UIActivityIndicatorView Class Reference 23

Overview 23

Tasks 26

Methods 26

UIAlert Class Reference 27

Overview 27

Tasks 30

Methods 30

UIApplication Class Reference 32

Overview 32

Tasks 32

Methods 33

UIButton Class Reference 38

Overview 38

UICollectionView Class Reference 42

[Overview](#) 42

[Tasks](#) 43

[Methods](#) 43

[UIAEditingMenu Class Reference](#) 44

[Overview](#) 44

[UIAElement Class Reference](#) 48

[Overview](#) 48

[Tasks](#) 48

[Methods](#) 52

[UIAElementArray Class Reference](#) 70

[Overview](#) 70

[Tasks](#) 71

[Properties](#) 71

[Methods](#) 72

[Constants](#) 74

[UIAHost Class Reference](#) 75

[Overview](#) 75

[Tasks](#) 75

[Methods](#) 75

[UIAKey Class Reference](#) 77

[Overview](#) 77

[UIAKeyboard Class Reference](#) 81

[Overview](#) 81

[Tasks](#) 84

[Methods](#) 84

[UIALink Class Reference](#) 86

[Overview](#) 86

[Tasks](#) 89

[Methods](#) 89

[UIALogger Class Reference](#) 90

[Overview](#) 90

[Tasks](#) 90

[Methods](#) 91

[UINavigationController Class Reference](#) 94

[Overview](#) 94

[Tasks](#) 97

[Methods](#) 97

[UIPageIndicator Class Reference](#) 99

[Overview](#) 99

[Tasks](#) 102

[Methods](#) 103

[UIPicker Class Reference](#) 105

[Overview](#) 105

[Tasks](#) 108

[Methods](#) 108

[UIPickerWheel Class Reference](#) 109

[Overview](#) 109

[Tasks](#) 109

[Methods](#) 110

[UIPopover Class Reference](#) 111

[Overview](#) 111

[Tasks](#) 114

[Methods](#) 114

[UIProgressIndicator Class Reference](#) 116

[Overview](#) 116

[UIScrollView Class Reference](#) 120

[Overview](#) 120

[Tasks](#) 123

[Methods](#) 124

[UISearchBar Class Reference](#) 126

[Overview](#) 126

[UISecureTextField Class Reference](#) 127

[Overview](#) 127

[UISegmentedControl Class Reference](#) 128

[Overview](#) 128

[Tasks](#) 131

[Methods](#) 131

[UISlider Class Reference](#) 132

[Overview](#) 132

[Tasks](#) 135

[Methods](#) 135

[UIStaticText Class Reference](#) 136

[Overview](#) 136

[UIStatusBar Class Reference](#) 140

[Overview](#) 140

[UISwitch Class Reference](#) 144

[Overview](#) 144

[Tasks](#) 147

[Methods](#) 147

[UITabBar Class Reference](#) 148

[Overview](#) 148

[Tasks](#) 151

[Methods](#) 151

[UITableViewCell Class Reference](#) 152

[Overview](#) 152

[UITableViewGroup Class Reference](#) 156

[Overview](#) 156

[UITableView Class Reference](#) 160

[Overview](#) 160

[Tasks](#) 161

[Methods](#) 161

[UITarget Class Reference](#) 162

[Overview](#) 162

[Tasks](#) 162

[Methods](#) 166
[Event Handlers by Task](#) 182
[Event Handlers](#) 182
[Constants](#) 183

[UITextField Class Reference](#) 184

[Overview](#) 184
[Tasks](#) 187
[Methods](#) 187

[UITextView Class Reference](#) 188

[Overview](#) 188
[Tasks](#) 191
[Methods](#) 191

[UIToolbar Class Reference](#) 192

[Overview](#) 192

[UIWebView Class Reference](#) 196

[Overview](#) 196

[UIWindow Class Reference](#) 197

[Overview](#) 197
[Tasks](#) 200
[Methods](#) 200

[Document Revision History](#) 203

Tables

[UIActionSheet Class Reference](#) 14

[Table 1-1](#) [Methods inherited from UIAElement](#) 14

[UIActivityIndicator Class Reference](#) 19

[Table 2-1](#) [Methods inherited from UIAElement](#) 19

[UIActivityIndicatorView Class Reference](#) 23

[Table 3-1](#) [Methods inherited from UIAElement](#) 23

[UIAlert Class Reference](#) 27

[Table 4-1](#) [Methods inherited from UIAElement](#) 27

[UIButton Class Reference](#) 38

[Table 6-1](#) [Methods inherited from UIAElement](#) 38

[UICollectionViewController Class Reference](#) 42

[Table 7-1](#) [Methods inherited from UIScrollView](#) 42

[UIEditingMenu Class Reference](#) 44

[Table 8-1](#) [Methods inherited from UIAElement](#) 44

[UIKey Class Reference](#) 77

[Table 12-1](#) [Methods inherited from UIAElement](#) 77

[UIKeyboard Class Reference](#) 81

[Table 13-1](#) [Methods inherited from UIAElement](#) 81

[UILink Class Reference](#) 86

[Table 14-1](#) [Methods inherited from UIAElement](#) 86

[UINavigationController Class Reference](#) 94

[Table 16-1](#) [Methods inherited from UIAElement](#) 94

[UIPageIndicator Class Reference](#) 99

Table 17-1 [Methods inherited from UIAElement](#) 99

[UIAPicker Class Reference](#) 105

Table 18-1 [Methods inherited from UIAElement](#) 105

[UIAPickerView Class Reference](#) 109

Table 19-1 [Methods inherited from UIAPicker](#) 109

[UIAPopover Class Reference](#) 111

Table 20-1 [Methods inherited from UIAElement](#) 111

[UIAProgressIndicator Class Reference](#) 116

Table 21-1 [Methods inherited from UIAElement](#) 116

[UIAScrollView Class Reference](#) 120

Table 22-1 [Methods inherited from UIAElement](#) 120

[UISearchBar Class Reference](#) 126

Table 23-1 [Methods inherited from UIATextField](#) 126

[UISecureTextField Class Reference](#) 127

Table 24-1 [Methods inherited from UIATextField](#) 127

[UISegmentedControl Class Reference](#) 128

Table 25-1 [Methods inherited from UIAElement](#) 128

[UISlider Class Reference](#) 132

Table 26-1 [Methods inherited from UIAElement](#) 132

[UIAStaticText Class Reference](#) 136

Table 27-1 [Methods inherited from UIAElement](#) 136

[UIAStatusBar Class Reference](#) 140

Table 28-1 [Methods inherited from UIAElement](#) 140

[UISwitch Class Reference](#) 144

Table 29-1 [Methods inherited from UIAElement](#) 144

[UITabBar Class Reference](#) 148

Table 30-1 [Methods inherited from UIAElement](#) 148

UITableViewCell Class Reference 152

Table 31-1 Methods inherited from UIAElement 152

UITableViewGroup Class Reference 156

Table 32-1 Methods inherited from UIAElement 156

UITableView Class Reference 160

Table 33-1 Methods inherited from UIScrollView 160

UITextField Class Reference 184

Table 35-1 Methods inherited from UIAElement 184

TextView Class Reference 188

Table 36-1 Methods inherited from UIAElement 188

UIToolbar Class Reference 192

Table 37-1 Methods inherited from UIAElement 192

UIWebView Class Reference 196

Table 38-1 Methods inherited from UIScrollView 196

UIWindow Class Reference 197

Table 39-1 Methods inherited from UIAElement 197

UI Automation JavaScript Reference

Note: This document was previously titled UI Automation Reference Collection.

Use the UI Automation JavaScript library to write test scripts that exercise your app's user interface elements while the app runs on a connected device. You write the tests in JavaScript, calling the UI Automation API to simulate user interaction. The system returns log information to the host computer.

Note: UI Automation simulates all user interface actions initiated by the script. For the sake of brevity and clarity, this document describes those actions in terms of a user's perspective.

Accessing and Using User Interface Elements

In essence, your test script is an ordered set of commands, each of which accesses a user interface element in your app to perform a user action on it or to use the information associated within it. All the user interface elements in your app are represented to the script through an ordered hierarchy of objects defined by the `UIAElements` class and its subclasses. To reach a specified UI element, the script simply calls down the element hierarchy, starting with the top-level target object obtained by calling `UIATarget.localTarget()`. For example, the first button in the main window of your app might be referenced by index as follows:

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

If that first button is identified in your code as the Edit button, the following would also work:

```
UIATarget.localTarget().frontMostApp().mainWindow().buttons()["Edit"]
```

To tap that button, then, the script could use any of these three formats:

- `UIATarget.localTarget().frontMostApp().mainWindow().buttons()[0].tap();`
- `UIATarget.localTarget().frontMostApp().mainWindow().buttons()["Edit"].tap();`
- `var editButton=UIATarget.localTarget().frontMostApp().mainWindow().buttons()[0];
editButton.tap();`

The Automation instrument maintains a complete element hierarchy that represents your app's user interface. To view that hierarchy, use the `logElementTree` method to write an outline of it to the log:

```
UIATarget.localTarget().frontMostApp().logElementTree()
```

Recording Results With the Log

To record data during its tests, the script uses `UIALogger` class methods to send messages to the Automation instrument running on the host computer. Various methods are available to assist in organizing and analyzing the recorded data. For example:

- To indicate the initiation of a specified test, use the `logStart` method:
 - `UIALogger.logStart("Test1");`
- To end a test and mark it as failed, use the `logFail` method:
 - `UIALogger.logFail("Failed to foo.");`
- To send a general-purpose debug message, use the `logDebug` method:
 - `UIALogger.logDebug("Done with level 3.");`

You view the collected data in the Detail pane of the Automation instrument using Instruments.

Handling Alerts

When UI Automation encounters an alert during the execution of your script, it calls your alert handler, passing a reference to the `UIAAlert` object representing the alert. Your script should handle the alert appropriately and return a value of `true`, upon which normal script execution continues.

To ensure that alerts don't interfere with testing, the Automation instrument also implements a simple default alert handler. If your script's alert handler returns `false`, this default handler attempts to dismiss the alert by tapping the cancel button, if it exists; otherwise, it taps the default button.

The following code implements a simple alert handler that records a message to the log and returns `false`, thereby depending on the default handler to dismiss the alert:

```
UIATarget.onAlert = function onAlert(alert) {  
    var title = alert.name();  
  
    // add a warning to the log for each alert encountered
```

```
UIALogger.logWarning("Alert with title '" + title + "' encountered!");
UIATarget.localTarget().captureScreenWithName("alert_" + (new Date()).UTC());

// test if your script should handle the alert, and if so, return true

// otherwise, return false to use the default handler
return false;
}
```

Classes

UIActionSheet Class Reference

Inherits from	UIAElement
---------------	------------

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The `UIActionSheet` class allows access to, and control of, action sheets within your app.

For an explanation of how to use this and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 1-1 provides a list of methods inherited from `UIAElement`.

Table 1-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified action sheet.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified action sheet.
buttons (page 53)	Returns an array of buttons contained by the specified action sheet.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified action sheet.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified action sheet.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified action sheet.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified action sheet.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this action sheet.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified action sheet.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified action sheet.
popover (page 60)	Returns the popover object associated with the specified action sheet.

Method	Description
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified action sheet.
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified action sheet.
searchBars (page 62)	Returns an array of search bars contained by the specified action sheet.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified action sheet.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified action sheet.
sliders (page 62)	Returns an array of sliders contained by the specified action sheet.
staticText s (page 63)	Returns an array of static texts contained by the specified action sheet.
switches (page 63)	Returns an array of switches contained by the specified action sheet.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this action sheet.
tableView s (page 64)	Returns an array of table views contained by the specified action sheet.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified action sheet.
textViews (page 66)	Returns an array of text views contained by the specified action sheet.

Method	Description
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this action sheet.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified action sheet.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

Tasks

Working With Buttons ---

[cancelButton](#) (page 17)

Returns the Cancel button in the action sheet.

Methods

cancelButton ---

Returns the Cancel button in the action sheet.

(UIButton) cancelButton()

Availability

Available in iOS 4.0 and later.

UIAActivityIndicator Class Reference

Inherits from	UIAElement
Availability	Available in iOS 4.0 and later.

Overview

The `UIAActivityIndicator` class allows access to, and control of, activity indicator elements in your app. For an explanation of how to use this and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 2-1 provides a list of methods inherited from `UIAElement`.

Table 2-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified activity indicator.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified activity indicator.
buttons (page 53)	Returns an array of buttons contained by the specified activity indicator.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified activity indicator.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified activity indicator.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified activity indicator.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified activity indicator.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this activity indicator.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified activity indicator.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified activity indicator.

Method	Description
popover (page 60)	Returns the popover object associated with the specified activity indicator.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified activity indicator.
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified activity indicator.
searchBars (page 62)	Returns an array of search bars contained by the specified activity indicator.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified activity indicator.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified activity indicator.
sliders (page 62)	Returns an array of sliders contained by the specified activity indicator.
staticText s (page 63)	Returns an array of static texts contained by the specified activity indicator.
switches (page 63)	Returns an array of switches contained by the specified activity indicator.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this activity indicator.
tableView s (page 64)	Returns an array of table views contained by the specified activity indicator.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified activity indicator.

Method	Description
textViews (page 66)	Returns an array of text views contained by the specified activity indicator.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this activity indicator.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified activity indicator.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

UIAActivityView Class Reference

Inherits from	UIAElement
---------------	------------

Availability	Available in iOS 6.0 and later.
--------------	---------------------------------

Overview

The `UIAActivityView` class allows access to, and control of, activity views within your app.

For an explanation of how to use this class and related classes, see the “Automating UI Testing” chapter in the *Instruments User Guide*.

Inherited Methods

Table 3-1 provides a list of methods inherited from `UIAElement`.

Table 3-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.

Method	Description
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticText s (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableView s (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textView s (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.

Method	Description
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

Tasks

Working With Buttons

[cancelButton](#) (page 26)

Returns the cancel button in the activity view.

Methods

cancelButton

Returns the cancel button in the activity view.

```
(UIButton) cancelButton()
```

Availability

Available in iOS 6.0 and later.

UIAlert Class Reference

Inherits from	UIAElement
Availability	Available in iOS 4.0 and later.

Overview

The `UIAlert` class allows access to, and control of, alerts within your app.

For an explanation of how to use this and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 4-1 provides a list of methods inherited from `UIAElement`.

Table 4-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified alert.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified alert.
buttons (page 53)	Returns an array of buttons contained by the specified alert.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified alert.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.
elements (page 55)	Returns an array of elements contained by the specified alert.

Method	Description
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified alert.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified alert.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by specified alert.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified alert.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified alert.
popover (page 60)	Returns the popover object associated with the specified alert.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified alert.
rect (page 60)	Returns the position of the object on the main screen.

Method	Description
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified alert.
searchBars (page 62)	Returns an array of search bars contained by the specified alert.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified alert.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified alert.
sliders (page 62)	Returns an array of sliders contained by the specified alert.
staticText s (page 63)	Returns an array of static texts contained by the specified alert.
switches (page 63)	Returns an array of switches contained by the specified alert.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this alert.
tableView s (page 64)	Returns an array of table views contained by the specified alert.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified alert.
textView s (page 66)	Returns an array of text views contained by the specified alert.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this alert.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.
value (page 67)	Returns a string containing a value attribute specific to the type of element.

Method	Description
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified alert.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

Tasks

Accessing Alert Buttons

[cancelButton](#) (page 30)

Returns the cancel button contained in the alert.

[defaultButton](#) (page 30)

Returns the default button contained in the alert.

Methods

cancelButton

Returns the cancel button contained in the alert.

```
(UIButton) cancelButton()
```

Availability

Available in iOS 4.0 and later.

defaultButton

Returns the default button contained in the alert.

```
(UIButton) defaultButton()
```

Availability

Available in iOS 4.0 and later.

UIApplication Class Reference

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The `UIApplication` class allows access to, and control of, app-level user interface elements.

For an explanation of how to use this and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Tasks

Working With App-level Elements

[actionSheet](#) (page 33)

Returns the action sheet.

[alert](#) (page 33)

Returns the alert.

[bundleID](#) (page 34)

Returns a string containing the app’s bundle ID in reverse-DNS format.

[editingMenu](#) (page 34)

Returns the app’s edit menu.

[interfaceOrientation](#) (page 34)

Returns the interface orientation.

[keyboard](#) (page 34)

Returns the app’s keyboard.

[mainWindow](#) (page 35)

Returns the object that represents the app’s main window.

[navigationBar](#) (page 35)

Returns the app’s navigation bar.

[preferencesValueForKey](#) (page 35)

Gets the value of a specified app user preference.

[setPreferencesValueForKey](#) (page 35)

Sets the value of a specified app user preference.

[statusBar](#) (page 36)

Returns the app's status bar.

[tabBar](#) (page 36)

Returns the app's tab bar.

[toolbar](#) (page 36)

Returns the app's toolbar.

[version](#) (page 36)

Returns a string representing the build version number of the app.

[windows](#) (page 37)

Returns an array of objects representing the app's windows.

Methods

actionSheet

Returns the action sheet.

```
(UIAActionSheet) actionSheet()
```

Availability

Available in iOS 4.0 and later.

alert

Returns the alert.

```
(UIAAlert) alert()
```

Availability

Available in iOS 4.0 and later.

bundleID

Returns a string containing the app's bundle ID in reverse-DNS format.

```
(String) bundleID()
```

Availability

Available in iOS 4.0 and later.

editingMenu

Returns the app's edit menu.

```
(UIAEditingMenu) editingMenu()
```

Availability

Available in iOS 4.0 and later.

interfaceOrientation

Returns the interface orientation.

```
(Number) interfaceOrientation()
```

Discussion

Interface orientation represents the rotation required to keep the interface right-side up upon device rotation. Note that in landscape mode, device orientation and interface orientation are opposite, because rotating the device requires rotating the content in the opposite direction.

Availability

Available in iOS 4.0 and later.

keyboard

Returns the app's keyboard.

```
(UIAKeyboard) keyboard()
```

Availability

Available in iOS 4.0 and later.

mainWindow

Returns the object that represents the app's main window.

(UIWindow) mainWindow()

Availability

Available in iOS 4.0 and later.

navigationBar

Returns the app's navigation bar.

(UINavigationController) navigationBar()

Availability

Available in iOS 4.0 and later.

preferencesValueForKey

Gets the value of a specified app user preference.

(NotTyped) preferencesValueForKey(key)

Parameters

key

The key representing the preference to be set.

Availability

Available in iOS 4.0 and later.

setPreferencesValueForKey

Sets the value of a specified app user preference.

(undefined) setPreferencesValueForKey(NotTyped value, String key)

Parameters

value

The new value to set for the specified preference.

key

The key representing the preference to be set.

Availability

Available in iOS 4.0 and later.

statusBar

Returns the app's status bar.

```
(UIStatusBar) statusBar()
```

Availability

Available in iOS 4.0 and later.

tabBar

Returns the app's tab bar.

```
(UITabBar) tabBar()
```

Availability

Available in iOS 4.0 and later.

toolbar

Returns the app's toolbar.

```
(UIToolbar) toolbar()
```

Availability

Available in iOS 4.0 and later.

version

Returns a string representing the build version number of the app.

```
(String) version()
```

Discussion

The build version is represented by a string comprising three period-separated integers.

Availability

Available in iOS 4.0 and later.

windows

Returns an array of objects representing the app's windows.

(UIAElementArray) windows()

Availability

Available in iOS 4.0 and later.

UIButton Class Reference

Inherits from	UIAElement
Availability	Available in iOS 4.0 and later.

Overview

The `UIButton` class allows access to, and control of, button elements in your app.

For an explanation of how to use this and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 6-1 provides a list of methods inherited from `UIAElement`.

Table 6-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified button.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified button.
buttons (page 53)	Returns an array of buttons contained by the specified button.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified button.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.
elements (page 55)	Returns an array of elements contained by the specified button.

Method	Description
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified button.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified button.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this button.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified button.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified button.
popover (page 60)	Returns the popover object associated with the specified button.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified button.
rect (page 60)	Returns the position of the object on the main screen.

Method	Description
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified button.
searchBar s (page 62)	Returns an array of search bars contained by the specified button.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified button.
segmentedControl s (page 62)	Returns an array of segmented controls contained by the specified button.
slider s (page 62)	Returns an array of sliders contained by the specified button.
staticText s (page 63)	Returns an array of static texts contained by the specified button.
switch s (page 63)	Returns an array of switches contained by the specified button.
tabBar (page 63)	Returns the specified tab bar.
tabBar s (page 63)	Returns an array of tab bars contained by this button.
tableView s (page 64)	Returns an array of table views contained by the specified button.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textField s (page 65)	Returns an array of text fields contained by the specified button.
textView s (page 66)	Returns an array of text views contained by the specified button.
toolbar (page 66)	Returns the specified toolbar.
toolbar s (page 66)	Returns an array of toolbars contained by this button.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.
value (page 67)	Returns a string containing a value attribute specific to the type of element.

Method	Description
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified button.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

UICollectionView Class Reference

Inherits from	UIScrollView
Availability	Available in iOS 6.0 and later

Overview

The `UICollectionView` class allows access to, and control of, elements within a collection view in your app.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 7-1 provides a list of methods inherited from `UIScrollView`.

Table 7-1 Methods inherited from `UIScrollView`

Method	Description
<code>scrollDown</code> (page 124)	Scrolls down within the specified collection view.
<code>scrollLeft</code> (page 124)	Scrolls left within the specified collection view.
<code>scrollRight</code> (page 124)	Scrolls right within the specified collection view.
<code>scrollToElementWithName</code> (page 124)	Scrolls within the collection view until the named element is displayed on the screen.
<code>scrollToElementWithPredicate</code> (page 125)	Scrolls within the collection view until the matching element is displayed on the screen.
<code>scrollToElementWithValueForKey</code> (page 125)	Scrolls within the collection view until the element with the specified value for the specified key is displayed on the screen.
<code>scrollUp</code> (page 125)	Scrolls up within the specified collection view.

Tasks

Working With Cells

`cells` (page 43)

Returns an array of elements within the collection view.

`visibleCells` (page 43)

Returns an array visible elements within the collection view.

Methods

`cells`

Returns an array of elements within the collection view.

(UIAElementArray) `cells()`

Availability

Available in iOS 6.0 and later.

`visibleCells`

Returns an array visible elements within the collection view.

(UIAElementArray) `visibleCells()`

Availability

Available in iOS 6.0 and later.

UIAEditingMenu Class Reference

Inherits from	UIAElement
Availability	Available in iOS 4.0 and later.

Overview

The `UIAEditingMenu` class allows access to, and control of, your app's edit menu.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 8-1 provides a list of methods inherited from `UIAElement`.

Table 8-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified menu.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified menu.
buttons (page 53)	Returns an array of buttons contained by the specified menu.
checkIsValid (page 53)	Returns the specified element's current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified menu.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified menu.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified menu.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified menu.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this menu.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified menu.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified menu.
popover (page 60)	Returns the popover object associated with the specified menu.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified menu.

Method	Description
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollViewViews (page 62)	Returns an array of scroll views contained by the specified menu.
searchBars (page 62)	Returns an array of search bars contained by the specified menu.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified menu.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified menu.
sliders (page 62)	Returns an array of sliders contained by the specified menu.
staticTexts (page 63)	Returns an array of static texts contained by the specified menu.
switches (page 63)	Returns an array of switches contained by the specified menu.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this menu.
tableViewes (page 64)	Returns an array of table views contained by the specified menu.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified menu.
textViewes (page 66)	Returns an array of text views contained by the specified menu.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this menu.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.

Method	Description
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified menu.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

UIAElement Class Reference

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The `UIAElement` class is the superclass for all user interface elements in the context of the Automation instrument for automating user interface testing of iOS apps.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Tasks

Determining Element Positioning

[hitpoint](#) (page 56)

Returns the screen position to tap for the specified element.

[rect](#) (page 60)

Returns the position of the object on the main screen.

Determining and Manipulating Element Hierarchy

[activityIndicators](#) (page 52)

Returns an array of the activity indicators contained by the specified object.

[activityView](#) (page 52)

Returns an object representing an activity view.

[ancestry](#) (page 53)

Returns an array containing the parents of the specified object.

[buttons](#) (page 53)

Returns an array of buttons contained by the specified object.

[collectionViews](#) (page 53)

Returns an array of collection views contained by the specified object.

[elements](#) (page 55)

Returns an array of elements contained by the specified object.

[images](#) (page 56)

Returns an array of images contained by the specified object.

[links](#) (page 58)

Returns an array of links contained by the specified object.

[navigationBar](#) (page 59)

Returns the app's navigation bar.

[navigationBars](#) (page 59)

Returns an array of navigation bar objects contained by this object.

[pageIndicators](#) (page 59)

Returns an array of page indicators contained by the specified object.

[parent](#) (page 59)

Returns the parent of the specified element.

[pickers](#) (page 60)

Returns an array of picker objects contained by the specified object.

[popover](#) (page 60)

Returns the popover object associated with the specified object, if one exists.

[progressIndicators](#) (page 60)

Returns an array of progress indicators contained by the specified object.

[scrollViews](#) (page 62)

Returns an array of scroll views contained by the specified object.

[searchBars](#) (page 62)

Returns an array of search bars contained by the specified object.

[secureTextFields](#) (page 62)

Returns an array of secure text fields contained by the specified object.

[segmentedControls](#) (page 62)

Returns an array of segmented controls contained by the specified object.

[sliders](#) (page 62)

Returns an array of sliders contained by the specified object.

[staticTexts](#) (page 63)

Returns an array of static texts contained by the specified object.

[switches](#) (page 63)

Returns an array of switches contained by the specified object.

[tabBar](#) (page 63)

Returns the specified tab bar.

[tabBars](#) (page 63)

Returns an array of tab bars contained by this object.

[tableViews](#) (page 64)

Returns an array of table views contained by the specified object.

[textFields](#) (page 65)

Returns an array of text fields contained by the specified object.

[textViews](#) (page 66)

Returns an array of text views contained by the specified object.

[toolbar](#) (page 66)

Returns the specified toolbar.

[toolbars](#) (page 66)

Returns an array of toolbars contained by this object.

[webViews](#) (page 67)

Returns an array of web views contained by the specified object.

Gestures and Actions

These methods allow you to effect the common gestures and actions a user can perform through the user interface. Options are available for use with some of these methods to give you flexibility in defining and varying the attributes of the gesture or action to be performed.

[doubleTap](#) (page 54)

Double-taps the specified element.

[dragInsideWithOptions](#) (page 54)

Drags within the bounds of an element.

[flickInsideWithOptions](#) (page 55)

Flicks within the bounds of an element.

[rotateWithOptions](#) (page 61)

Perform a rotation gesture centered on the specified element.

[scrollToVisible](#) (page 61)

Scrolls until the specified element is visible in a container view.

[tap](#) (page 64)

Taps the specified element.

[tapWithOptions](#) (page 64)

Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.

[touchAndHold](#) (page 66)

Touches the specified element and holds for the specified duration.

[twoFingerTap](#) (page 67)

Performs a two-finger (two-touch) tap on this element.

Determining Element State

Use these methods to determine whether an element is still valid.

[checkIsValid](#) (page 53)

Returns the specified element's current validity status.

[hasKeyboardFocus](#) (page 56)

Determines whether the specified element receives keyboard input.

[isEnabled](#) (page 56)

Determines whether the specified element is enabled.

[isValid](#) (page 57)

Returns the specified element's validity status as of the most recent access.

[isVisible](#) (page 57)

Determines whether the specified element is visible on the screen.

[waitForInvalid](#) (page 67)

Waits for the specified element to become invalid.

Identifying Elements

[label](#) (page 57)

Returns a string containing the label attribute of the element.

[name](#) (page 58)

Returns a string containing the name attribute of the element.

[value](#) (page 67)

Returns a string containing a value attribute specific to the type of element.

[withName](#) (page 68)

Returns an element whose name attribute matches a specified string.

[withPredicate](#) (page 68)

Returns the element matching the specified criteria.

[withValueForKey](#) (page 68)

Returns the element containing the specified property with the specified value.

Logging Element Information

[logElement](#) (page 58)

Logs information about the specified element.

[logElementTree](#) (page 58)

Logs information about the specified element and all of its subelements.

Methods

activityIndicators

Returns an array of the activity indicators contained by the specified object.

```
(UIAElementArray) activityIndicators()
```

Availability

Available in iOS 4.0 and later.

activityView

Returns an object representing an activity view.

```
(UIAActivityView) activityView()
```

Availability

Available in iOS 6.0 and later.

ancestry

Returns an array containing the parents of the specified object.

(UIAElementArray) ancestry()

Availability

Available in iOS 4.0 and later.

buttons

Returns an array of buttons contained by the specified object.

(UIAElementArray) buttons()

Availability

Available in iOS 4.0 and later.

checkIsValid

Returns the specified element's current validity status.

(Boolean) checkIsValid()

Discussion

Use this method to determine whether the user interface element represented by the specified UIAElement currently exists. You should use `checkIsValid`, for example, if you're referencing an element after having performed some action that may have changed the UI state of that element in some way. This requires a call to the underlying Accessibility framework to ensure the validity of the result.

See Also

`isValid`

Availability

Available in iOS 4.0 and later.

collectionViews

Returns an array of collection views contained by the specified object.

(UIAElementArray) collectionViews()

Availability

Available in iOS 6.0 and later.

doubleTap

Double-taps the specified element.

```
(undefined) doubleTap()
```

Availability

Available in iOS 4.0 and later.

dragInsideWithOptions

Drags within the bounds of an element.

```
(undefined) dragInsideWithOptions(Object options)
```

Parameters

`options`

A dictionary that specifies characteristics of the gesture. Valid keys are as follows:

<code>touchCount</code>	The number of touches to use in the specified gesture. (Effectively, the number of fingers a user would use to make the specified gesture.) The default touch count value is 1.
<code>duration</code>	The length of hold time for the specified gesture. The default duration value for a tap is 0. The default value for touch-and-hold gestures (such as drag, pinch open, and pinch close) is 1.
<code>startOffset</code>	The first offset to use for a multiple-point gesture. The default value is {x:0.0, y:0.0}. See the discussion for details.
<code>endOffset</code>	The last offset to use for a multiple-point gesture. The default value is {x:0.0, y:0.0}. See the discussion for details.

Discussion

You can use offsets to achieve finer precision in specifying the hitpoint within the `rect` for the specified element. The offset comprises a pair of x and y values, each ranging from 0.0 to 1.0. These values represent, respectively, relative horizontal and vertical positions within the `rect`, with {x:0.0, y:0.0} as the top left and {x:1.0, y:1.0} as the bottom right. Thus, {x:0.3, y:0.6} specifies a position just below and to the left of center, and {x:1.0, y:0.5} specifies a position centered vertically at the far right.

This example performs a slow drag within the target element from left edge to right edge, just below the top:

```
target.dragInsideWithOptions({startOffset:{x:0.0, y:0.1}, endOffset:{x:1.0, y:0.1},  
duration:1.5});
```

Availability

Available in iOS 4.0 and later.

elements

Returns an array of elements contained by the specified object.

```
(UIAElementArray) elements()
```

Availability

Available in iOS 4.0 and later.

flickInsideWithOptions

Flicks within the bounds of an element.

```
(undefined) flickInsideWithOptions(Object options)
```

Parameters

`options`

A dictionary that specifies characteristics of the gesture. Valid keys are as follows:

<code>touchCount</code>	The number of touches to use in the specified gesture. (Effectively, the number of fingers a user would use to make the specified gesture.) The default touch count value is 1.
<code>startOffset</code>	The first offset to use for a multiple-point gesture. The default value is {x:0.0, y:0.0}. See the discussion for details.
<code>endOffset</code>	The last offset to use for a multiple-point gesture. The default value is {x:0.0, y:0.0}. See the discussion for details.

Discussion

You can use offsets to achieve finer precision in specifying the hitpoint within the `rect` for the specified element. The offset comprises a pair of x and y values, each ranging from 0.0 to 1.0. These values represent, respectively, relative horizontal and vertical positions within the `rect`, with {x:0.0, y:0.0} as the top left and {x:1.0, y:1.0} as the bottom right. Thus, {x:0.3, y:0.6} specifies a position just below and to the left of center, and {x:1.0, y:0.5} specifies a position centered vertically at the far right.

This example performs a flick just above the bottom edge of the target element, from center to right edge:

```
target.flickInsideWithOptions({startOffset:{x:0.5, y:0.9}, endOffset:{x:1.0, y:0.9}});
```

Availability

Available in iOS 4.0 and later.

hasKeyboardFocus

Determines whether the specified element receives keyboard input.

(Number) hasKeyboardFocus()

Return Value

Returns 1 if the specified element is the receiver of keyboard input, 0 if not. If the status is not available, it returns null.

Availability

Available in iOS 4.0 and later.

hitpoint

Returns the screen position to tap for the specified element.

(Point) hitpoint()

Availability

Available in iOS 4.0 and later.

images

Returns an array of images contained by the specified object.

(UIAElementArray) images()

Availability

Available in iOS 4.0 and later.

isEnabled

Determines whether the specified element is enabled.

(Number) isEnabled()

Return Value

Returns 1 if the specified element is enabled, 0 if not. If the status is not available, it returns null.

Availability

Available in iOS 4.0 and later.

isValid

Returns the specified element's validity status as of the most recent access.

(Boolean) isValid()

Discussion

Use this method to determine whether the user interface element represented by the specified UIAElement existed as of the last attempt to access it. To be certain that the element exists, use `checkIsValid` instead.

See Also

`checkIsValid`

Availability

Available in iOS 4.0 and later.

isVisible

Determines whether the specified element is visible on the screen.

(Number) isVisible()

Return Value

Returns 1 if the user interface element represented by the specified element is visible on screen, 0 if not. If the status is not available, it returns null.

Availability

Available in iOS 4.0 and later.

label

Returns a string containing the label attribute of the element.

(String) label()

Discussion

This method always returns the label attribute string. (Contrast with the [name](#) (page 58) method.)

Availability

Available in iOS 5.0 and later.

links

Returns an array of links contained by the specified object.

(UIAElementArray) links()

Availability

Available in iOS 4.0 and later.

logElement

Logs information about the specified element.

(undefined) logElement()

Discussion

This method can be used with any element.

Availability

Available in iOS 4.0 and later.

logElementTree

Logs information about the specified element and all of its subelements.

(undefined) logElementTree()

Discussion

This method can be used with any element.

Availability

Available in iOS 4.0 and later.

name

Returns a string containing the name attribute of the element.

(String) name()

Discussion

The element name is derived from the accessibility attribute of the underlying view. If an identifier attribute string is specified, that string is used as the name; otherwise, the label attribute string is used as the name. Contrast with the [label](#) (page 57) method.

For more information, see *UIAccessibilityIdentification Protocol Reference*.

Availability

Available in iOS 4.0 and later.

navigationBar

Returns the app's navigation bar.

```
(UIAElement) navigationBar()
```

Discussion

This method has been moved up to this class from the UIAWindow Class.

Availability

Available in iOS 4.0 and later.

navigationBars

Returns an array of navigation bar objects contained by this object.

```
(UIAElementArray) navigationBars()
```

Discussion

This method has been moved up to this class from the UIAWindow Class.

Availability

Available in iOS 4.0 and later.

pageIndicators

Returns an array of page indicators contained by the specified object.

```
(UIAElementArray) pageIndicators()
```

Availability

Available in iOS 4.0 and later.

parent

Returns the parent of the specified element.

```
(UIAElement) parent()
```

Availability

Available in iOS 4.2 and later.

pickers

Returns an array of picker objects contained by the specified object.

```
(UIAElementArray) pickers()
```

Availability

Available in iOS 4.0 and later.

popover

Returns the popover object associated with the specified object, if one exists.

```
(UIAPopover) popover()
```

Availability

Available in iOS 4.2 and later.

progressIndicators

Returns an array of progress indicators contained by the specified object.

```
(UIAElementArray) progressIndicators()
```

Availability

Available in iOS 4.0 and later.

rect

Returns the position of the object on the main screen.

```
(Rect) rect()
```

Discussion

Your script should treat the `rect` object as a generic JavaScript object whose properties for `origin`, `x`, `y`, `size`, `width`, and `height` correspond to those of the analogous `CGRect` Cocoa structure. The `rect` object has the form `{origin:{x:xposition,y:yposition}, size:{width:widthvalue,height:heightvalue}}`. The relevant coordinates are screen-relative and are adjusted to account for device orientation.

Availability

Available in iOS 4.0 and later.

rotateWithOptions

Perform a rotation gesture centered on the specified element.

(undefined) rotateWithOptions(Object options)

Parameters

options

A dictionary that specifies characteristics of the rotation gesture. Valid keys are as follows:

centerOffset	The offset to use for the center of the rotate gesture. The default offset value is {x:0.0, y:0.0}.
duration	The length of hold time for the specified gesture, in seconds. The default duration value is 1.
radius	The distance in points from the center to the edge of the circular path.
rotation	The length of rotation in radians. The default is pi (π).
touchCount	The number of touches to use in the specified gesture. (Effectively, the number of fingers a user would use to make the specified gesture.) Valid values are 1 to 5. The default is 2.

Discussion

This gesture is generated such that each touch is equidistant from the others.

Availability

Available in iOS 4.0 and later.

scrollToVisible

Scrolls until the specified element is visible in a container view.

(undefined) scrollToVisible()

Discussion

Use this method with tables and web views.

Availability

Available in iOS 4.0 and later.

scrollViewViews

Returns an array of scroll views contained by the specified object.

(UIAElementArray) scrollViewViews()

Availability

Available in iOS 4.0 and later.

searchBars

Returns an array of search bars contained by the specified object.

(UIAElementArray) searchBars()

Availability

Available in iOS 4.0 and later.

secureTextFields

Returns an array of secure text fields contained by the specified object.

(UIAElementArray) secureTextFields()

Availability

Available in iOS 4.0 and later.

segmentedControls

Returns an array of segmented controls contained by the specified object.

(UIAElementArray) segmentedControls()

Availability

Available in iOS 4.0 and later.

sliders

Returns an array of sliders contained by the specified object.

(UIAElementArray) sliders()

Availability

Available in iOS 4.0 and later.

staticTexts

Returns an array of static texts contained by the specified object.

```
(UIAElementArray) staticTexts()
```

Availability

Available in iOS 4.0 and later.

switches

Returns an array of switches contained by the specified object.

```
(UIAElementArray) switches()
```

Availability

Available in iOS 4.0 and later.

tabBar

Returns the specified tab bar.

```
(UIAElement) tabBar()
```

Discussion

This method has been moved up to this class from the UIWindow Class.

Availability

Available in iOS 4.0 and later.

tabBars

Returns an array of tab bars contained by this object.

```
(UIAElementArray) tabBars()
```

Discussion

This method has been moved up to this class from the UIWindow Class.

Availability

Available in iOS 4.0 and later.

tableViews

Returns an array of table views contained by the specified object.

```
(UIAElementArray) tableViews()
```

Availability

Available in iOS 4.0 and later.

tap

Taps the specified element.

```
(undefined) tap()
```

Availability

Available in iOS 4.0 and later.

tapWithOptions

Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.

```
(undefined) tapWithOptions(Object options)
```


Parameters

options

A dictionary that specifies characteristics of the gesture. Valid keys are as follows:

tapCount	The number of taps that compose the specified gesture. The default value is 1 (single tap).
touchCount	The number of touches to use in the specified gesture. (Effectively, the number of fingers a user would use to make the specified gesture.) The default touch count value is 1.
duration	The length of hold time for the specified gesture. The default duration value for a tap is 0. The default value for touch-and-hold gestures (such as drag, pinch open, and pinch close) is 1.
tapOffset	The offset to use for the specified tap gesture. The default offset value is {x:0.0, y:0.0}. See the discussion for details.

Discussion

For example, you could specify a triple tap with two fingers at the center of the screen (on an iPhone in portrait orientation), as follows:

```
element.tapWithOptions({touchCount:2, tapCount:3});
```

```
element.tapWithOptions({touchCount:2, tapCount:3, tapOffset:{x:0.75, y:0.25}});
```

You can use offsets to achieve finer precision in specifying the hitpoint within the `rect` for the specified element. The offset comprises a pair of x and y values, each ranging from 0.0 to 1.0. These values represent, respectively, relative horizontal and vertical positions within the `rect`, with {x:0.0, y:0.0} as the top left and {x:1.0, y:1.0} as the bottom right. Thus, {x:0.3, y:0.6} specifies a position just below and to the left of center, and {x:1.0, y:0.5} specifies a position centered vertically at the far right.

Availability

Available in iOS 4.0 and later.

textFields

Returns an array of text fields contained by the specified object.

```
(UIAElementArray) textFields()
```

Availability

Available in iOS 4.0 and later.

textViews

Returns an array of text views contained by the specified object.

```
(UIAElementArray) textViews()
```

Availability

Available in iOS 4.0 and later.

toolbar

Returns the specified toolbar.

```
(UIAElement) toolbar()
```

Discussion

This method has been moved up to this class from the UIWindow Class.

Availability

Available in iOS 4.0 and later.

toolbars

Returns an array of toolbars contained by this object.

```
(UIAElementArray) toolbars()
```

Discussion

This method has been moved up to this class from the UIWindow Class.

Availability

Available in iOS 4.0 and later.

touchAndHold

Touches the specified element and holds for the specified duration.

```
(undefined) touchAndHold(Number duration)
```

Parameters

duration

The length of time to hold the touch on the element, in seconds. The default duration value for a tap is 0. The default value for touch-and-hold gestures (such as drag, pinch open, and pinch close) is 1.

Availability

Available in iOS 4.0 and later.

twoFingerTap

Performs a two-finger (two-touch) tap on this element.

(undefined) twoFingerTap()

Availability

Available in iOS 4.0 and later.

value

Returns a string containing a value attribute specific to the type of element.

(String) value()

Discussion

For example, a switch has a value of 1 for ON and 0 for OFF.

Availability

Available in iOS 4.0 and later.

waitForInvalid

Waits for the specified element to become invalid.

(Boolean) waitForInvalid()

Discussion

Waits for the user interface element represented by the specified UIAElement to become invalid. Uses the current `timeout` value for the wait time interval.

Availability

Available in iOS 4.0 and later.

webViews

Returns an array of web views contained by the specified object.

(UIAElementArray) webViews()

Availability

Available in iOS 4.0 and later.

withName

Returns an element whose name attribute matches a specified string.

```
(UIAElement) withName(String name)
```

Parameters

name

A string containing the name to test for.

Discussion

Tests if the name attribute of the element has the given string value. If the match fails, the test is retried until the current `timeout` expires.

Availability

Available in iOS 4.0 and later.

withPredicate

Returns the element matching the specified criteria.

```
(UIAElement) withPredicate(PredicateString predicateString)
```

Parameters

predicateString

A string specifying the match criteria.

Discussion

Uses the specified predicate string to test for a match. If the match fails, the test is retried until the current `timeout` expires. See *Predicate Programming Guide* for information about using predicates.

Availability

Available in iOS 4.0 and later.

withValueForKey

Returns the element containing the specified property with the specified value.

```
(UIAElement) withValueForKey(NotTyped value, String key)
```

Parameters

`value`

A string specifying the value that the specified property, if it exists, should match.

`key`

A string specifying the property to test for.

Discussion

Tests if the element has a specified property with the specified value. If the match fails, the test is retried until the current `timeout` expires.

Availability

Available in iOS 4.0 and later.

UIAElementArray Class Reference

Availability

Available in iOS 4.0 and later.

Overview

The `UIAElementArray` class supports operations with arrays containing multiple `UIAElement` objects. You can search the array by name or key/value pairs, or by custom criteria that you specify using a predicate. For detailed information about using predicates, see *Predicates Programming Guide*.

`UIAElementArray` supports the traditional shorthand syntax for accessing items in native JavaScript objects:

- dot reference by element name, for example:

```
var okButton = buttons.OK;
```

- bracket reference by index, for example:

```
var firstElement = elements[0];
```

- bracket reference by element name, for example:

```
var helloWorldText = staticTexts["Hello World"];
```

It is important to note one potentially confusing limitation in using bracket references by element name. If the name of an element happened to be a number, JavaScript would interpret that name as an index, likely yielding incorrect results. For example, consider this array:

```
var elements = ["DoIt", "Cancel", "1"];
```

If you attempt to access the third element by name with the bracket reference syntax:

```
elements()["1"]
```

it is actually treated as a reference to the first element, yielding the first element (named “DoIt”) instead. In cases where this problem might occur, you should use the dot reference syntax instead:

```
elements().firstWithName("1")
```

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Tasks

Working With Arrays

`length` (page 71)

Returns the number of user interface elements in the array.

`firstWithName` (page 72)

Returns the first element in the array with this name.

`firstWithPredicate` (page 72)

Returns the first element in the array matching the given criteria.

`firstWithValueForKey` (page 72)

Returns the first element in the array with a value that matches the property key.

`toArray` (page 73)

Converts the array into a standard JavaScript array.

`withName` (page 73)

Returns all elements in the array with this name.

`withPredicate` (page 73)

Returns all elements in the array matching the given.

`withValueForKey` (page 73)

Returns all elements in the array with a value that matches the property key.

Properties

`length`

Returns the number of user interface elements in the array.

(Number) `length`

Methods

firstWithName

Returns the first element in the array with this name.

```
(UIAElement) firstWithName(String name)
```

Parameters

name

A string whose value is the name of the element match on.

Availability

Available in iOS 4.0 and later.

firstWithPredicate

Returns the first element in the array matching the given criteria.

```
(UIAElement) firstWithPredicate(PredicateString predicateString)
```

Parameters

predicateString

A predicate specifying the criteria to match.

Discussion

For detailed information about predicate matching, see *Predicate Programming Guide*.

Availability

Available in iOS 4.0 and later.

firstWithValueForKey

Returns the first element in the array with a value that matches the property key.

```
(UIAElement) firstWithValueForKey(NotTyped value, String key)
```

Availability

Available in iOS 4.0 and later.

toArray

Converts the array into a standard JavaScript array.

(Array) toArray()

Availability

Available in iOS 4.0 and later.

withName

Returns all elements in the array with this name.

(UIAElementArray) withName(String name)

Availability

Available in iOS 4.0 and later.

withPredicate

Returns all elements in the array matching the given.

(UIAElementArray) withPredicate(PredicateString predicateString)

Discussion

Predicate matching follows the same rules as NSPredicate.

Availability

Available in iOS 4.0 and later.

withValueForKey

Returns all elements in the array with a value that matches the property key.

(UIAElementArray) withValueForKey(NotTyped value, String key)

Parameters

value

A string specifying the value that the specified property, if it exists, should match.

key

A string specifying the property to test for.

Availability

Available in iOS 4.0 and later.

Constants

UIAElementNil

Constants

UIAElementNil

Returned by a function with return type `UIAElement` or `UIAElementArray` if the requested element is not available after the `timeout` grace period you specify. This mechanism allows your scripting expressions to complete even if an intermediate function in the expression fails temporarily during the grace period. For example, the following code does not raise an exception if the navigation bar does not exist; instead it returns `UIAElementNil`.

```
var backButton = UIATarget.localTarget().frontMostApp().navigationBar().buttons()["Back"];

if (backButton.isValid()) {
    backButton.tap();
} else {
    UIALogger.logError("Could not find 'Back' button!");
}
```

Available in iOS 4.0 and later.

UIAHost Class Reference

Availability	Available in iOS 5.0 and later.
---------------------	---------------------------------

Overview

The `UIAHost` class allows your script to exercise limited control over the Automation instrument process running on the host computer.

For an explanation of how to use this and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Tasks

Performing a Task on the Host Computer

`performTaskWithPathArgumentsTimeout` (page 75)

Executes a task from the Automation instrument process running on the host.

Methods

`performTaskWithPathArgumentsTimeout`

Executes a task from the Automation instrument process running on the host.

(object) `performTaskWithPathArgumentsTimeout(path, args, timeout)`

Parameters

`path`

The pathname of the code to run, relative to the root level of the host’s boot drive.

`args`

An array that specifies the arguments for the code to be run.

`timeout`

The length, in seconds, of the grace period in which the task is expected to execute, before script execution resumes.

Discussion

The process executes from within the context of the Instruments application parent process. The code below runs the `echo` command to display “Hello World” with a grace period of 5 seconds.

The returned object contains the properties `exitCode`, `stdout`, and `stderr`. The example uses these properties to capture the exit code, standard output stream, and standard error stream, displaying each in a log message with a debug severity level.

```
var target = UIATarget.localTarget();
var host = target.host();

var result = host.performTaskWithPathArgumentsTimeout("/usr/bin/echo", ["Hello
World"], 5);

UIALogger.logDebug("exitCode: " + result.exitCode);
UIALogger.logDebug("stdout: " + result.stdout);
UIALogger.logDebug("stderr: " + result.stderr);
```

Availability

Available in iOS 5.0 and later.

UIAKey Class Reference

Inherits from	UIAElement
Availability	Available in iOS 4.0 and later.

Overview

The UIAKey class allows access to, and control, of key elements within your app’s keyboard.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 12-1 provides a list of methods inherited from UIAElement.

Table 12-1 Methods inherited from UIAElement

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.

Method	Description
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticTexts (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableView s (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textView s (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.

Method	Description
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

UIAKeyboard Class Reference

Inherits from	UIAElement
---------------	------------

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The UIAKeyboard class allows access to, and control of, elements within your app’s keyboard.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 13-1 provides a list of methods inherited from UIAElement.

Table 13-1 Methods inherited from UIAElement

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.

Method	Description
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticTexts (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableView s (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textView s (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.

Method	Description
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

Tasks

Retrieving Keyboard Information

[keys](#) (page 84)

Returns an array representing the keys of the specified keyboard.

Exercising the Keyboard

[typeString](#) (page 85)

Taps the keys of the specified keyboard as required to generate the specified string.

Methods

keys

Returns an array representing the keys of the specified keyboard.

(UIAElementArray) keys()

Availability

Available in iOS 4.0 and later.

typeString

Taps the keys of the specified keyboard as required to generate the specified string.

(undefined) typeString(String string)

Parameters

string

The string to be typed on the keyboard.

Availability

Available in iOS 5.0 and later.

UIALink Class Reference

Inherits from	UIAElement
Availability	Available in iOS 4.0 and later.

Overview

The `UIALink` class allows access to, and control of, link elements.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 14-1 provides a list of methods inherited from `UIAElement`.

Table 14-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.

Method	Description
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollViewViews (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticTexts (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableViewes (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textViewes (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.

Method	Description
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

Tasks

Retrieving Link Information

[url](#) (page 89)

Returns a string containing a URL.

Methods

url

Returns a string containing a URL.

```
(String) url()
```

Availability

Available in iOS 4.0 and later.

UIALogger Class Reference

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The `UIALogger` class provides test and error information on retrieval functionality.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Tasks

Logging With Test Status

`logFail` (page 91)

Logs a message and indicates a test has completed unsuccessfully.

`logIssue` (page 92)

Logs a message and indicates a test has terminated abnormally.

`logPass` (page 92)

Logs a message and indicates a test has completed successfully.

`logStart` (page 93)

Logs a message and indicates a test has started.

Logging With Severity Levels

These methods log a message and set a severity level to support filtering in the detail pane.

`logDebug` (page 91)

Logs the specified message and sets the severity level to debug.

[logError](#) (page 91)

Logs the specified message and sets the severity level to error.

[logMessage](#) (page 92)

Logs the specified message and sets the severity level to message.

[logWarning](#) (page 93)

Logs the specified message and sets the severity level to warning.

Methods

logDebug

Logs the specified message and sets the severity level to debug.

(undefined) logDebug(String message)

Parameters

message

A string containing the message to log.

Availability

Available in iOS 4.0 and later.

logError

Logs the specified message and sets the severity level to error.

(undefined) logError(String message)

Parameters

message

A string containing the message to log.

Availability

Available in iOS 4.0 and later.

logFail

Logs a message and indicates a test has completed unsuccessfully.

(undefined) logFail(String message)

Availability

Available in iOS 4.0 and later.

logIssue

Logs a message and indicates a test has terminated abnormally.

```
(undefined) logIssue(String message)
```

Parameters

message

A string containing the message to log.

Availability

Available in iOS 4.0 and later.

logMessage

Logs the specified message and sets the severity level to message.

```
(undefined) logMessage(String message)
```

Parameters

message

A string containing the message to log.

Availability

Available in iOS 4.0 and later.

logPass

Logs a message and indicates a test has completed successfully.

```
(undefined) logPass(String message)
```

Parameters

message

A string containing the message to log.

Availability

Available in iOS 4.0 and later.

logStart

Logs a message and indicates a test has started.

(undefined) logStart(String message)

Parameters

message

A string containing the message to log.

Availability

Available in iOS 4.0 and later.

logWarning

Logs the specified message and sets the severity level to warning.

(undefined) logWarning(String message)

Parameters

message

A string containing the message to log.

Availability

Available in iOS 4.0 and later.

UINavigationController Class Reference

Inherits from	UIAElement
---------------	------------

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The `UINavigationController` class allows access to, and control of, buttons in your app's navigation bar.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 16-1 provides a list of methods inherited from `UIAElement`.

Table 16-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element's current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.

Method	Description
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticTexts (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableViews (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textViews (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.

Method	Description
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

Tasks

Accessing Buttons

[leftButton](#) (page 97)

Returns the left button in the navigation bar.

[rightButton](#) (page 98)

Returns the right button in the navigation bar.

Methods

leftButton

Returns the left button in the navigation bar.

```
(UIButton) leftButton()
```

Availability

Available in iOS 4.0 and later.

rightButton

Returns the right button in the navigation bar.

(UIButton) rightButton()

Availability

Available in iOS 4.0 and later.

UIPageIndicator Class Reference

Inherits from	UIAElement
---------------	------------

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The `UIPageIndicator` class allows access to, and control of, page indicator elements in your app.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 17-1 provides a list of methods inherited from `UIAElement`.

Table 17-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.

Method	Description
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticTexts (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableView s (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textView s (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.

Method	Description
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

Tasks

Handling Page Navigation

[goToNextPage](#) (page 103)

Goes to the next (logically, to the right) open view .

[goToPreviousPage](#) (page 103)

Goes to the previous (logically, to the left) open view .

[pageCount](#) (page 103)

Returns the number of open views.

[pageIndex](#) (page 103)

Returns the index of the currently open view.

[selectPage](#) (page 103)

Goes to the page specified by the index value.

Methods

goToNextPage

Goes to the next (logically, to the right) open view .

(undefined) goToNextPage()

Availability

Available in iOS 4.0 and later.

goToPreviousPage

Goes to the previous (logically, to the left) open view .

(undefined) goToPreviousPage()

Availability

Available in iOS 4.0 and later.

pageCount

Returns the number of open views.

(Number) pageCount()

Availability

Available in iOS 4.0 and later.

pageIndex

Returns the index of the currently open view.

(Number) pageIndex()

Availability

Available in iOS 4.0 and later.

selectPage

Goes to the page specified by the index value.

(undefined) selectPage(Number index)

Parameters

index

The value identifying the selected page.

Availability

Available in iOS 4.0 and later.

UIAPicker Class Reference

Inherits from	UIAElement
---------------	------------

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The `UIAPicker` class allows access to, and control of, wheel elements within a picker.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 18-1 provides a list of methods inherited from `UIAElement`.

Table 18-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.

Method	Description
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticTexts (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableView s (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textView s (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.

Method	Description
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

Tasks

Manipulating Pickers

[wheels](#) (page 108)

Returns an array representing the wheels of the specified picker.

Methods

wheels

Returns an array representing the wheels of the specified picker.

```
(UIAElementArray) wheels()
```

Availability

Available in iOS 4.0 and later.

UIPickerWheel Class Reference

Inherits from	UIPicker
Availability	Available in iOS 4.0 and later.

Overview

The `UIPickerWheel` class allows access to, and control of, wheel elements within a picker.

For an explanation of how to use this class and related classes, see the UI Automation section of “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 19-1 provides a list of methods inherited from `UIPicker`.

Table 19-1 Methods inherited from `UIPicker`

Method	Description
wheels (page 108)	Returns an array representing the wheels of the specified picker.

Tasks

Manipulating Wheels

[selectValue](#) (page 110)

Drags the wheel to the first row with the specified value.

[values](#) (page 110)

Returns an array representing the possible item values to select for the wheel.

Methods

selectValue

Drags the wheel to the first row with the specified value.

(undefined) selectValue()

Special Considerations

This method is unsupported for UIPickerView objects backed by a UIDatePicker view in iOS 5 and earlier. Since picker wheels representing years and eras have hundreds of thousands of values, you can only set values that are between the min and max supported values.

Availability

Available in iOS 4.0 and later.

values

Returns an array representing the possible item values to select for the wheel.

(Array) values()

Special Considerations

This method is unsupported for UIPickerView objects backed by a UIDatePicker view in iOS 5 and earlier; in such cases, it returns nil. Since picker wheels representing years and eras have hundreds of thousands of values, only the min and max supported values are returned in these cases.

Availability

Available in iOS 4.0 and later.

UIAPopover Class Reference

Inherits from	UIAElement
Availability	Available in iOS 4.0 and later.

Overview

The `UIAPopover` class provides methods for accessing and manipulating popovers and the elements they contain.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 20-1 provides a list of methods inherited from `UIAElement`.

Table 20-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.
rect (page 60)	Returns the position of the object on the main screen.

Method	Description
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticTexts (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableView s (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textView s (page 66)	Returns an array of text views contained by the specified object.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.

Method	Description
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

Tasks

Retrieving Popover Contents

[actionSheet](#) (page 114)

Returns the action sheet contained by the popover.

[navigationBar](#) (page 115)

Returns the navigation bar contained by the popover.

[tabBar](#) (page 115)

Returns the tab bar contained by the popover.

[toolbar](#) (page 115)

Returns the toolbar contained by the popover.

Dismissing the Popover

[dismiss](#) (page 115)

Dismisses a popover by tapping outside the popover and within the region defined for dismissal.

Methods

[actionSheet](#)

Returns the action sheet contained by the popover.

(UIAActionSheet) `actionSheet()`

Availability

Available in iOS 4.0 and later.

dismiss

Dismisses a popover by tapping outside the popover and within the region defined for dismissal.

```
(void) dismiss()
```

Availability

Available in iOS 4.2 and later.

navigationBar

Returns the navigation bar contained by the popover.

```
(UINavigationController) navigationBar()
```

Availability

Available in iOS 4.0 and later.

tabBar

Returns the tab bar contained by the popover.

```
(UITabBar) tabBar()
```

Availability

Available in iOS 4.0 and later.

toolbar

Returns the toolbar contained by the popover.

```
(UIToolbar) toolbar()
```

Availability

Available in iOS 4.0 and later.

UIAProgressIndicator Class Reference

Inherits from	UIAElement
---------------	------------

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The `UIAProgressIndicator` class allows access to, and control of, progress indicator elements in your app.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 21-1 provides a list of methods inherited from `UIAElement`.

Table 21-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.

Method	Description
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticTexts (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableView s (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textView s (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.

Method	Description
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

UIScrollView Class Reference

Inherits from	UIAElement
Availability	Available in iOS 4.0 and later.

Overview

The `UIScrollView` class allows access to, and control of, the elements of a scroll view.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 22-1 provides a list of methods inherited from `UIAElement`.

Table 22-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.

Method	Description
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticTexts (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableViews (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textViews (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.

Method	Description
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

Tasks

Scrolling ---

[scrollUp](#) (page 125)

Scrolls up within the specified scroll view.

[scrollDown](#) (page 124)

Scrolls down within the specified scroll view.

[scrollLeft](#) (page 124)

Scrolls left within the specified scroll view.

[scrollRight](#) (page 124)

Scrolls right within the specified scroll view.

[scrollToElementWithName](#) (page 124)

Scrolls within the specified scroll view until the named element is displayed on the screen.

[scrollToElementWithPredicate](#) (page 125)

Scrolls within the specified scroll view until the matching element is displayed on the screen.

[scrollToElementWithValueForKey](#) (page 125)

Scrolls within the specified scroll view until the element with the specified value for the specified key is displayed on the screen.

Methods

scrollDown

Scrolls down within the specified scroll view.

(undefined) scrollDown()

Availability

Available in iOS 4.0 and later.

scrollLeft

Scrolls left within the specified scroll view.

(undefined) scrollLeft()

Availability

Available in iOS 4.0 and later.

scrollRight

Scrolls right within the specified scroll view.

(undefined) scrollRight()

Availability

Available in iOS 4.0 and later.

scrollToElementWithName

Scrolls within the specified scroll view until the named element is displayed on the screen.

(UIAElement) scrollToElementWithName(String name)

Parameters

name

The name of the element to scroll to.

Availability

Available in iOS 4.0 and later.

scrollToElementWithPredicate

Scrolls within the specified scroll view until the matching element is displayed on the screen.

```
(UIAElement) scrollToElementWithPredicate(PredicateString predicateString)
```

Parameters

predicateString

The predicate to define the match criteria.

Availability

Available in iOS 4.0 and later.

scrollToElementWithValueForKey

Scrolls within the specified scroll view until the element with the specified value for the specified key is displayed on the screen.

```
(UIAElement) scrollToElementWithValueForKey(NotTyped value, String key)
```

Parameters

value

The value for the specified key.

key

The key for the specified value.

Availability

Available in iOS 4.0 and later.

scrollUp

Scrolls up within the specified scroll view.

```
(undefined) scrollUp()
```

Availability

Available in iOS 4.0 and later.

UISearchBar Class Reference

Inherits from	UITextField
Availability	Available in iOS 4.0 and later.

Overview

The `UISearchBar` class allows access to, and control of, search bar elements in your app.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 23-1 provides a list of methods inherited from `UITextField`.

Table 23-1 Methods inherited from `UITextField`

Method	Description
setValue (page 187)	Sets the specified text field to the specified value.

UIASecureTextField Class Reference

Inherits from	UITextField
Availability	Available in iOS 4.0 and later.

Overview

The `UIASecureTextField` class allows access to, and control of, secure text field elements in your app.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 24-1 provides a list of methods inherited from `UITextField`.

Table 24-1 Methods inherited from `UITextField`

Method	Description
setValue (page 187)	Sets the specified text field to the specified value.

UISegmentedControl Class Reference

Inherits from	UIAElement
---------------	------------

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The `UISegmentedControl` class allows access to, and control of, elements within segmented controls in your app.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 25-1 provides a list of methods inherited from `UIAElement`.

Table 25-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.

Method	Description
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticTexts (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableView s (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textView s (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.

Method	Description
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

Tasks

Accessing Buttons

[selectedButton](#) (page 131)

Returns the currently selected button within the segmented control.

Methods

[selectedButton](#)

Returns the currently selected button within the segmented control.

```
(UIAElement) selectedButton()
```

Availability

Available in iOS 4.0 and later.

UISlider Class Reference

Inherits from	UIAElement
---------------	------------

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The `UISlider` class allows access to, and control of, slider elements in your app.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 26-1 provides a list of methods inherited from `UIAElement`.

Table 26-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.

Method	Description
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticTexts (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableView s (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textView s (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.

Method	Description
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

Tasks

Setting Slider Values ---

[dragToValue](#) (page 135)

Drags the slider to the specified value.

Methods

[dragToValue](#) ---

Drags the slider to the specified value.

(undefined) `dragToValue(Number value)`

Parameters

`value`

The desired decimal value from 0 to 1, inclusive. A 0 value represents far left and a value of 1 represents far right.

Availability

Available in iOS 4.0 and later.

UIAStaticText Class Reference

Inherits from	UIAElement
---------------	------------

Availability	Available in iOS 5.0 and later.
--------------	---------------------------------

Overview

The `UIAStaticText` class allows access to, and control of, static text views.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 27-1 provides a list of methods inherited from `UIAElement`.

Table 27-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.

Method	Description
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticText s (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableView s (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textView s (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.

Method	Description
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

UIAStatusBar Class Reference

Inherits from	UIAElement
---------------	------------

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The `UIAStatusBar` class allows access to, and control of, your app's status bar.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 28-1 provides a list of methods inherited from `UIAElement`.

Table 28-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element's current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.

Method	Description
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticText s (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableView s (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textView s (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.

Method	Description
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

UISwitch Class Reference

Inherits from	UIAElement
---------------	------------

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The `UISwitch` class allows access to, and control of, switch elements in your app.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 29-1 provides a list of methods inherited from `UIAElement`.

Table 29-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.

Method	Description
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollViewViews (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticTexts (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableViewes (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textViewes (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.

Method	Description
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

Tasks

Setting Switch Values

[setValue](#) (page 147)

Sets the specified switch to the specified value.

Methods

setValue

Sets the specified switch to the specified value.

(undefined) setValue(Boolean value)

Parameters

value

A boolean value to represent the desired state, true for On, false for Off.

Availability

Available in iOS 4.0 and later.

UITabBar Class Reference

Inherits from	UIAElement
---------------	------------

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The `UIAElement` class allows access to, and control of, elements within your app's tab bar.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 30-1 provides a list of methods inherited from `UIAElement`.

Table 30-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element's current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.

Method	Description
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticTexts (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableView s (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textView s (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.

Method	Description
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

Tasks

Selecting a Button ---

[selectedButton](#) (page 151)

Returns the button currently selected in the tab bar.

Methods

[selectedButton](#) ---

Returns the button currently selected in the tab bar.

```
(UIButton) selectedButton()
```

Availability

Available in iOS 4.0 and later.

UITableViewCell Class Reference

Inherits from	UIAElement
---------------	------------

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The `UITableViewCell` class allows access to, and control of, table cell elements within a table view.

Important: In certain test scenarios, particularly with new off-screen table cell elements, UI Automation is unable to immediately derive a label for an element, rendering that element inaccessible to your script. To avoid resultant problems with automated tests, your application should always set the `accessibilityLabel` property for new table cell elements. See *UIAccessibilityElement Class Reference* for more information.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 31-1 provides a list of methods inherited from `UIAElement`.

Table 31-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element’s current validity status.

Method	Description
collectionView (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.

Method	Description
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticText s (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableView s (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textView s (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.

Method	Description
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

UITableGroup Class Reference

Inherits from	UIAElement
---------------	------------

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The `UITableGroup` class allows access to, and control of, group elements within a table view.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 32-1 provides a list of methods inherited from `UIAElement`.

Table 32-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.

Method	Description
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticTexts (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableView s (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textView s (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.

Method	Description
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

UITableView Class Reference

Inherits from	UIScrollView
Availability	Available in iOS 4.0 and later.

Overview

The `UITableView` class allows access to, and control of, elements within a table view in your app.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 33-1 provides a list of methods inherited from `UIScrollView`.

Table 33-1 Methods inherited from `UIScrollView`

Method	Description
scrollDown (page 124)	Scrolls down within the specified table view.
scrollLeft (page 124)	Scrolls left within the specified table view.
scrollRight (page 124)	Scrolls right within the specified table view.
scrollToElementWithName (page 124)	Scrolls within the table view until the named element is displayed on the screen.
scrollToElementWithPredicate (page 125)	Scrolls within the table view until the matching element is displayed on the screen.
scrollToElementWithValueForKey (page 125)	Scrolls within the table view until the element with the specified value for the specified key is displayed on the screen.
scrollUp (page 125)	Scrolls up within the specified table view.

Tasks

Retrieving Information

[cells](#) (page 161)

Returns an array of the cells within the table view.

[groups](#) (page 161)

Returns an array of the groups within the table view.

[visibleCells](#) (page 161)

Returns an array of the cells that are visible within the table view.

Methods

[cells](#)

Returns an array of the cells within the table view.

(UIAElementArray) `cells()`

Availability

Available in iOS 4.0 and later.

[groups](#)

Returns an array of the groups within the table view.

(UIAElementArray) `groups()`

Availability

Available in iOS 4.0 and later.

[visibleCells](#)

Returns an array of the cells that are visible within the table view.

(UIAElementArray) `visibleCells()`

Availability

Available in iOS 4.0 and later.

UITarget Class Reference

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The `UITarget` class represents high-level user interface elements of the system under test (SUT)—that is, your app, the iOS, and the connected device on which they’re running. Your test scripts, written in JavaScript and running in conjunction with the UI Automation instrument, use this class and related UI Automation classes to exercise the SUT and log results.

For the sake of simplicity and consistency with other Apple documentation, this document describes device operations and user interface actions as though they were performed by a user. In practice, the Automation instrument simulates these operations and actions.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Tasks

Getting the Base Target and Host Objects

`host` (page 171)

Returns an object representing the machine that is host to the current target.

`localTarget` (page 171)

Returns an object representing the system under test (SUT).

Managing Your App

`deactivateAppForDuration` (page 167)

Renders your app inactive for the specified duration.

[frontMostApp](#) (page 170)

Returns an object representing your app.

Obtaining Device Property Information

Use these methods to obtain information specific to the device, such as assigned name, device model, and operating-system name and version.

[model](#) (page 172)

Returns the device model.

[name](#) (page 172)

Returns the device name.

[rect](#) (page 175)

Returns the rectangle surrounding the device's main screen.

[systemName](#) (page 179)

Returns the name of the operating system running on the device.

[systemVersion](#) (page 179)

Returns the current version of the operating system running on the device.

Determining and Changing Device Orientation

[deviceOrientation](#) (page 168)

Returns the current orientation of the device.

[setDeviceOrientation](#) (page 176)

Changes the device orientation to the specified new `deviceOrientation` value.

Changing the Device Location

[setLocation](#) (page 177)

Specifies a change in device's latitude and longitude.

[setLocationWithOptions](#) (page 177)

Specifies a change in the device's latitude, longitude, and other characteristics.

Device Controls and Actions

[clickVolumeDown](#) (page 167)

Presses the volume down hardware button.

[clickVolumeUp](#) (page 167)

Presses the volume up hardware button.

[holdVolumeDown](#) (page 170)

Holds down the volume down hardware button for the specified duration.

[holdVolumeUp](#) (page 171)

Presses and holds the volume up hardware button for the specified duration.

[lockForDuration](#) (page 172)

Locks the device, using a lock event, for the specified duration.

[shake](#) (page 178)

Performs a shake action on the device.

[lock](#) (page 171) **Deprecated in iOS 5.0**

Locks the device, using a lock event.

[unlock](#) (page 181) **Deprecated in iOS 5.0**

Unlocks the device using an unlock event followed by a drag of the slider.

Interacting with the Screen

The `rect` and `point` objects used with these screen interaction methods have properties for `origin`, `size`, `x`, `y`, `height`, and `width` corresponding to the analogous `CGRect`, `CGPoint`, and `CGSize` Cocoa structures. Your script should treat methods with `rect`, `point`, or `size` arguments or return types as JavaScript objects with those properties defined. The relevant coordinates are screen-relative and are adjusted to account for device orientation.

[dragFromToForDuration](#) (page 169)

Drags from a specified starting screen location to a specified ending screen location, for a specified length of time.

[doubleTap](#) (page 169)

Double-taps the specified element or at the specified screen location.

[flickFromTo](#) (page 170)

Flicks from the specified starting screen location to the specified ending screen location.

[pinchCloseFromToForDuration](#) (page 173)

Pinches (performs a pinch-close gesture) from a specified starting screen location to a specified ending screen location, for a specified length of time.

[pinchOpenFromToForDuration](#) (page 173)

Stretches (performs a pinch-open gesture) from a specified starting screen location to a specified ending screen location, for a specified length of time.

[rotateWithOptions](#) (page 175)

Performs a rotation gesture at the specified location.

[tap](#) (page 179)

Taps the specified element or the specified screen location.

[tapWithOptions](#) (page 180)

Taps the specified element with the specified options.

[touchAndHold](#) (page 181)

Touches the specified element, or the specified screen location, and holds for the specified duration.

Capturing Screen Images

These methods allow you to record the appearance of the screen (or some portion of it). Such images can be helpful in tracking progress in a test and in diagnosing problems.

[captureRectWithName](#) (page 166)

Takes a screen shot of the specified rectangular portion of the device screen.

[captureScreenWithName](#) (page 166)

Takes a screen shot of the entire device screen.

Manipulating Timeouts

[popTimeout](#) (page 174)

Retrieves the previous timeout value from a stack, restores it as the current timeout value, and returns it.

[pushTimeout](#) (page 174)

Stores the current timeout value on a stack and sets a new timeout value.

[setTimeout](#) (page 178)

Sets a new timeout value.

[timeout](#) (page 180)

Returns the current timeout value.

Miscellaneous

[delay](#) (page 168)

Delays script execution for the specified time.

Methods

captureRectWithName

Takes a screen shot of the specified rectangular portion of the device screen.

```
(undefined) captureRectWithName(Rect rect, String imageName)
```

Parameters

rect

The `rect` that defines the area of the screen to capture.

imageName

A string to use as the name for the resultant image file.

Discussion

Your script should treat the `rect` object as a generic JavaScript object whose properties for origin, x, y, size, width, and height correspond to those of the analogous `CGRect` Cocoa structure. The `rect` object has the form `{origin:{x:xposition, y:yposition}, size:{width:widthvalue, height:heightvalue}}`. The relevant coordinates are screen-relative and are adjusted to account for device orientation.

The image is saved as a file in .PNG graphic format, with the specified name, in the log.

Availability

Available in iOS 4.0 and later.

captureScreenWithName

Takes a screen shot of the entire device screen.

```
(undefined) captureScreenWithName(String imageName)
```

Parameters

`imageName`

A string to use as the name for the resultant image file.

Discussion

The image is saved as a file in .PNG graphic format, with the specified name, in the log.

Availability

Available in iOS 4.0 and later.

clickVolumeDown

Presses the volume down hardware button.

(undefined) `clickVolumeDown()`

Availability

Available in iOS 4.0 and later.

clickVolumeUp

Presses the volume up hardware button.

(undefined) `clickVolumeUp()`

Availability

Available in iOS 4.0 and later.

deactivateAppForDuration

Renders your app inactive for the specified duration.

(Boolean) `deactivateApp(Number duration)`

Parameters

`duration`

The time, in seconds, for the app to remain inactive.

Discussion

Use this method to test shifting your app to and from the background execution context. Note that apps built using iOS SDK 4.0 or later and running in iOS 4.0 and later aren't necessarily terminated when the user presses the Home button. See *iOS App Programming Guide* for details of multitasking and background execution context.

Availability

Available in iOS 4.0 and later.

delay

Delays script execution for the specified time.

(Boolean) delay(Number timeInterval)

Parameters

timeInterval

The time to delay, in seconds.

Discussion

You can use this method to provide enough time for lengthy operations to complete.

Availability

Available in iOS 4.0 and later.

deviceOrientation

Returns the current orientation of the device.

(Number deviceOrientation) deviceOrientation()

Discussion

The returned value is a constant that represents the physical orientation of the device and may be different from the current orientation of your app's user interface. The possible values are as follows:

```
UIA_DEVICE_ORIENTATION_UNKNOWN  
UIA_DEVICE_ORIENTATION_PORTRAIT  
UIA_DEVICE_ORIENTATION_PORTRAIT_UPSIDEDOWN  
UIA_DEVICE_ORIENTATION_LANDSCAPELEFT  
UIA_DEVICE_ORIENTATION_LANDSCAPERIGHT  
UIA_DEVICE_ORIENTATION_FACEUP  
UIA_DEVICE_ORIENTATION_FACEDOWN
```

See the Constants section for descriptions of these values.

Availability

Available in iOS 4.0 and later.

doubleTap

Double-taps the specified element or at the specified screen location.

(undefined) doubleTap(Object tapPointObject)

Parameters

tapPointObject

A rect, point, or UIAElement.

Discussion

The rect and point objects have properties for origin, size, x, y, height, and width corresponding to the analogous CGRect, CGPoint, and CGSize Cocoa structures. Your script should treat methods with rect, point, or size arguments or return types as JavaScript objects with those properties defined. The relevant coordinates are screen-relative and are adjusted to account for device orientation.

Availability

Available in iOS 4.0 and later.

dragFromToForDuration

Drags from a specified starting screen location to a specified ending screen location, for a specified length of time.

(undefined) dragFromToForDuration(fromPointObject, toPointObject, Number duration)

Parameters

fromPointObject

The rect or point from which the drag action is to begin.

toPointObject

The rect or point at which the drag action is to end.

duration

The length of time, in seconds, between starting and stopping the gesture.

Discussion

The rect and point objects have properties for origin, size, x, y, height, and width corresponding to the analogous CGRect, CGPoint, and CGSize Cocoa structures. Your script should treat methods with rect, point, or size arguments or return types as JavaScript objects with those properties defined. The relevant coordinates are screen-relative and are adjusted to account for device orientation.

Availability

Available in iOS 4.0 and later.

flickFromTo

Flicks from the specified starting screen location to the specified ending screen location.

```
(undefined) flickFromTo(fromPointObject, toPointObject)
```

Parameters

`fromPointObject`

The rect or point from which the flick action is to begin.

`toPointObject`

The rect or point at which the flick action is to end.

Discussion

The rect and point objects have properties for `origin`, `size`, `x`, `y`, `height`, and `width` corresponding to the analogous `CGRect`, `CGPoint`, and `CGSize` Cocoa structures. Your script should treat methods with `rect`, `point`, or `size` arguments or return types as JavaScript objects with those properties defined. The relevant coordinates are screen-relative and are adjusted to account for device orientation.

Availability

Available in iOS 4.0 and later.

frontMostApp

Returns an object representing your app.

```
(UIApplication) frontMostApp()
```

Discussion

This `UIApplication` object is the centralized point of control and coordination for your app.

Availability

Available in iOS 4.0 and later.

holdVolumeDown

Holds down the volume down hardware button for the specified duration.

```
(undefined) holdVolumeDown(Number duration)
```

Availability

Available in iOS 4.0 and later.

holdVolumeUp

Presses and holds the volume up hardware button for the specified duration.

(undefined) holdVolumeUp(Number duration)

Availability

Available in iOS 4.0 and later.

host

Returns an object representing the machine that is host to the current target.

(UIAHost) host()

Availability

Available in iOS 5.0 and later.

localTarget

Returns an object representing the system under test (SUT).

(UITarget) localTarget()

Availability

Available in iOS 4.0 and later.

lock

Locks the device, using a lock event. (Deprecated in iOS 5.0.)

(undefined) lock()

Special Considerations

This method, and its counterpart, `unlock`, are deprecated. Use `lockForDuration` instead.

Availability

Available in iOS 4.0 and later.

Deprecated in iOS 5.0.

lockForDuration

Locks the device, using a lock event, for the specified duration.

(undefined) lockForDuration(Number duration)

Parameters

Duration

The length of time, in seconds, for the lock to persist.

Discussion

This method replaces the deprecated `lock` and `unlock` methods.

Availability

Available in iOS 5.0 and later.

model

Returns the device model.

(String) model()

Discussion

Examples of model strings are `iPhone` and `iPod touch`.

Availability

Available in iOS 4.0 and later.

name

Returns the device name.

(String) name()

Discussion

The device name is an arbitrary string specified for the device by the user. On an iPhone, for example, you can see the name on the device in the General > About settings or in iTunes on the Summary > iPhone tab.

Availability

Available in iOS 4.0 and later.

pinchCloseFromToForDuration

Pinches (performs a pinch-close gesture) from a specified starting screen location to a specified ending screen location, for a specified length of time.

```
(undefined) pinchCloseFromToForDuration(fromPointObject, toPointObject, Number duration)
```

Parameters

`fromPointObject`

The rect or point from which the pinch-close action is to begin.

`toPointObject`

The rect or point at which the pinch-close action is to end.

`duration`

The length of time, in seconds, between starting and stopping the gesture.

Discussion

The rect and point objects have properties for `origin`, `size`, `x`, `y`, `height`, and `width` corresponding to the analogous `CGRect`, `CGPoint`, and `CGSize` Cocoa structures. Your script should treat methods with rect, point, or size arguments or return types as JavaScript objects with those properties defined. The relevant coordinates are screen-relative and are adjusted to account for device orientation.

Availability

Available in iOS 4.0 and later.

pinchOpenFromToForDuration

Stretches (performs a pinch-open gesture) from a specified starting screen location to a specified ending screen location, for a specified length of time.

```
(undefined) pinchOpenFromToForDuration(fromPointObject, toPointObject, Number duration)
```

Parameters

`fromPointObject`

The rect or point from which the pinch-open action is to begin.

`toPointObject`

The rect or point at which the pinch-open action is to end.

`duration`

The length of time, in seconds, between starting and stopping the gesture.

Discussion

The `rect` and `point` objects have properties for `origin`, `size`, `x`, `y`, `height`, and `width` corresponding to the analogous `CGRect`, `CGPoint`, and `CGSize` Cocoa structures. Your script should treat methods with `rect`, `point`, or `size` arguments or return types as JavaScript objects with those properties defined. The relevant coordinates are screen-relative and are adjusted to account for device orientation.

Availability

Available in iOS 4.0 and later.

popTimeout

Retrieves the previous timeout value from a stack, restores it as the current timeout value, and returns it.

(Number) `popTimeout()`

Return Value

The timeout value last stored on the stack with `pushTimeout`.

Discussion

Use this method to revert to the previous grace period duration.

If an object representing a UI element becomes available within the grace period, an attempt is made to instantiate that object from information retained by the instrument.

Availability

Available in iOS 4.0 and later.

pushTimeout

Stores the current timeout value on a stack and sets a new timeout value.

(undefined) `pushTimeout(timeoutValue)`

Parameters

`timeout`

The length of the grace period, in seconds.

Discussion

This method, in conjunction with `popTimeout`, allows you to temporarily change the duration of the grace period for object resolution. This code changes the timeout period to 2 seconds before attempting to access an element, then restores the previous timeout period.

```
target = UIATarget.localTarget();

target.pushTimeout(2);
    // attempt element access
target.popTimeout();
```

If an object representing a UI element becomes available within the grace period, an attempt is made to instantiate that object from information retained by the instrument.

Availability

Available in iOS 4.0 and later.

rect

Returns the rectangle surrounding the device's main screen.

```
(Rect) rect()
```

Availability

Available in iOS 4.0 and later.

rotateWithOptions

Performs a rotation gesture at the specified location.

```
(undefined) rotateWithOptions(Object location, Object options)
```

Parameters

`location`

The `point` object at center of the rotation gesture, with properties for `x` and `y`, corresponding to the analogous `CGPoint` Cocoa structure. The relevant coordinates are screen-relative and are adjusted to account for device orientation.

options

A dictionary that specifies characteristics of the rotation gesture. Valid keys are as follows:

duration	The length of hold time, in seconds, for the specified gesture. The default duration value is 1.
radius	The distance in points from the center to the edge of the circular path.
rotation	The length of rotation in radians. The default is <code>pi</code> (ffl).
touchCount	The number of touches to use in the specified gesture (effectively, the number of fingers a user would use to make the specified gesture.) Valid values are 1 to 5. The default is 2.

Discussion

This gesture is generated such that each touch is equidistant from the others.

Availability

Available in iOS 5.0 and later.

setDeviceOrientation

Changes the device orientation to the specified new `deviceOrientation` value.

(undefined) `setDeviceOrientation(Number deviceOrientation)`

Discussion

The specified `deviceOrientation` value must be one of the following constants:

```
UIA_DEVICE_ORIENTATION_UNKNOWN  
UIA_DEVICE_ORIENTATION_PORTRAIT  
UIA_DEVICE_ORIENTATION_PORTRAIT_UPSIDEDOWN  
UIA_DEVICE_ORIENTATION_LANDSCAPELEFT  
UIA_DEVICE_ORIENTATION_LANDSCAPERIGHT  
UIA_DEVICE_ORIENTATION_FACEUP  
UIA_DEVICE_ORIENTATION_FACEDOWN
```

See the “Constants” section for descriptions of these values.

Availability

Available in iOS 4.0 and later.

setLocation

Specifies a change in device's latitude and longitude.

(boolean) setLocation(coordinates)

Parameters

coordinates

A dictionary that specifies the new location. Valid keys are as follows:

latitude	The latitude in degrees. Positive values indicate latitudes north of the equator. Negative values indicate latitudes south of the equator.
longitude	The longitude in degrees. Measurements are relative to the zero meridian, with positive values extending east of the meridian and negative values extending west of the meridian.

Availability

Available in iOS 5.0 and later.

setLocationWithOptions

Specifies a change in the device's latitude, longitude, and other characteristics.

(boolean) setLocationWithOptions(coordinates, options)

Parameters

coordinates

A dictionary that specifies the new location. Valid keys are as follows:

latitude	The latitude in degrees. Positive values indicate latitudes north of the equator. Negative values indicate latitudes south of the equator.
longitude	The longitude in degrees. Measurements are relative to the zero meridian, with positive values extending east of the meridian and negative values extending west of the meridian.

options

A dictionary that specifies additional characteristics of the location change. Valid keys are as follows:

altitude	The height, in meters, relative to sea level. Positive values indicate altitudes above sea level. Negative values indicate altitudes below sea level.
horizontalAccuracy	The radius, in meters, of the horizontal circle of uncertainty centered at the specified location. Negative values are invalid.
verticalAccuracy	The radius, in meters, of the horizontal circle of uncertainty centered at the specified location. Negative values are invalid.
course	The direction in which the device is moving, regardless of the device orientation.
speed	The speed, in meters per second, at which the device is moving.

Availability

Available in iOS 5.0 and later.

setTimeout

Sets a new timeout value.

(undefined) setTimeout(Number timeout)

Parameters

timeout

A number representing the length,in seconds, of the grace period.

Discussion

The timeout value establishes a grace period for object resolution. If an object representing a UI element becomes available within the grace period, an attempt is made to instantiate that object from information retained by the instrument.

Availability

Available in iOS 4.0 and later.

shake

Performs a shake action on the device.

(undefined) shake()

Availability

Available in iOS 4.0 and later.

systemName

Returns the name of the operating system running on the device.

(String) systemName()

Availability

Available in iOS 4.0 and later.

systemVersion

Returns the current version of the operating system running on the device.

(String) systemVersion()

Discussion

An example of a system version string is 1.2.

Availability

Available in iOS 4.0 and later.

tap

Taps the specified element or the specified screen location.

(undefined) tap(Object tapPointObject)

Parameters

tapPointObject

A rect, point, or UIAElement.

Discussion

The rect and point objects have properties for origin, size, x, y, height, and width corresponding to the analogous CGRect, CGPoint, and CGSize Cocoa structures. Your script should treat methods with rect, point, or size arguments or return types as JavaScript objects with those properties defined. The relevant coordinates are screen-relative and are adjusted to account for device orientation.

Availability

Available in iOS 4.0 and later.

tapWithOptions

Taps the specified element with the specified options.

(undefined) tapWithOptions(Object tapPointObject, Object options)

Parameters

tapPointObject

A rect, point, or UIAElement.

options

A dictionary that specifies characteristics of the gesture. Valid keys are as follows:

tapCount	The number of taps that compose the specified gesture. The default value is 1 (single tap).
touchCount	The number of touches to use in the specified gesture. (Effectively, the number of fingers a user would use to make the specified gesture.) The default touch count value is 1.
duration	The length of hold time for the specified gesture. The default duration value for a tap is 0.

Discussion

The rect and point objects have properties for origin, size, x, y, height, and width corresponding to the analogous CGRect, CGPoint, and CGSize Cocoa structures. Your script should treat methods with rect, point, or size arguments or return types as JavaScript objects with those properties defined. The relevant coordinates are screen-relative and are adjusted to account for device orientation.

Availability

Available in iOS 4.0 and later.

timeout

Returns the current timeout value.

(Number) timeout()

Discussion

The timeout value establishes a grace period for object resolution. If an object representing a UI element becomes available within the grace period, an attempt is made to instantiate that object from information retained by the instrument.

Availability

Available in iOS 4.0 and later.

touchAndHold

Touches the specified element, or the specified screen location, and holds for the specified duration.

(undefined) touchAndHold(Object tapPointObject, Number duration)

Parameters

tapPointObject

A rect, point, or UIAElement.

duration

The length of time, in seconds, to hold the touch.

Discussion

The rect and point objects have properties for origin, size, x, y, height, and width corresponding to the analogous CGRect, CGPoint, and CGSize Cocoa structures. Your script should treat methods with rect, point, or size arguments or return types as JavaScript objects with those properties defined. The relevant coordinates are screen-relative and are adjusted to account for device orientation.

Availability

Available in iOS 4.0 and later.

unlock

Unlocks the device using an unlock event followed by a drag of the slider. (Deprecated in iOS 5.0.)

(undefined) unlock()

Discussion

Simulating passcode entry is currently unsupported. Set the Settings > General > Passcode Lock feature to Off prior to running your tests.

Special Considerations

This method, and its counterpart, lock, are deprecated. Use lockForDuration instead.

Availability

Available in iOS 4.0 and later.

Deprecated in iOS 5.0.

Event Handlers by Task

Handling Alerts

[onAlert](#) (page 182)

Called by UI Automation to allow your script to respond to alerts.

Event Handlers

onAlert

Called by UI Automation to allow your script to respond to alerts.

(Boolean) `onAlert(UIAAlert alert)`

Parameters

`alert`

An object representing the alert encountered.

Return Value

Returns `true` if successful. Returns `false` to cause the default alert handler to run.

Discussion

Your `onAlert` handler is called if an alert is encountered at any time during the execution of the script. If you do not have a declared `onAlert` handler, the UI Automation default alert handler runs instead.

This default handler attempts to dismiss the alert by first tapping the cancel button, if the button exists, then tapping the default button, if one is identifiable. If the alert is still not dismissed, an exception is thrown.

Returning `false` from your own handler also causes the default handler to run. For cursory tests, the script handler might only log an alert message and return `false` to let the default handler dismiss the alert.

Constants

Constants

UIA_DEVICE_ORIENTATION_UNKNOWN

The orientation of the device cannot be determined.

UIA_DEVICE_ORIENTATION_PORTRAIT

The device is in portrait mode, with the device upright and the home button at the bottom.

UIA_DEVICE_ORIENTATION_PORTRAIT_UPSIDEDOWN

The device is in portrait mode but upside down, with the device upright and the home button at the top.

UIA_DEVICE_ORIENTATION_LANDSCAPELEFT

The device is in landscape mode, with the device upright and the home button on the right side.

UIA_DEVICE_ORIENTATION_LANDSCAPERIGHT

The device is in landscape mode, with the device upright and the home button on the left side.

UIA_DEVICE_ORIENTATION_FACEUP

The device is parallel to the ground with the screen facing upward.

UIA_DEVICE_ORIENTATION_FACEDOWN

The device is parallel to the ground with the screen facing downward.

UITextField Class Reference

Inherits from	UIAElement
---------------	------------

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The `UITextField` class allows access to, and control of, text field elements in your app.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 35-1 provides a list of methods inherited from `UIAElement`.

Table 35-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.

Method	Description
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticTexts (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableView s (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textView s (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.

Method	Description
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

Tasks

Setting Text Field Values

[setValue](#) (page 187)

Sets the specified text field to the specified value.

Methods

setValue

Sets the specified text field to the specified value.

(undefined) setValue(String value)

Parameters

value

A string containing the text to populate the text field.

Availability

Available in iOS 4.0 and later.

UITextView Class Reference

Inherits from	UIAElement
Availability	Available in iOS 4.0 and later.

Overview

The UITextView class allows access to, and control of, text view elements in your app.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 36-1 provides a list of methods inherited from UIAElement.

Table 36-1 Methods inherited from UIAElement

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element’s current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.

Method	Description
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticTexts (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableView s (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textView s (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.

Method	Description
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

Tasks

Setting Text View Values

[setValue](#) (page 191)

Sets the specified text view to the specified value.

Methods

setValue

Sets the specified text view to the specified value.

(undefined) setValue(String value)

Parameters

value

A string containing the text to populate the text view.

Availability

Available in iOS 4.0 and later.

UIToolbar Class Reference

Inherits from	UIAElement
---------------	------------

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The `UIToolbar` class allows access to, and control of, your app's toolbar.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 37-1 provides a list of methods inherited from `UIAElement`.

Table 37-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element's current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
navigationBar (page 59)	Returns the app's navigation bar.
navigationBars (page 59)	Returns an array of navigation bar objects contained by this object.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.

Method	Description
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified object.
searchBars (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControls (page 62)	Returns an array of segmented controls contained by the specified object.
sliders (page 62)	Returns an array of sliders contained by the specified object.
staticText s (page 63)	Returns an array of static texts contained by the specified object.
switches (page 63)	Returns an array of switches contained by the specified object.
tabBar (page 63)	Returns the specified tab bar.
tabBars (page 63)	Returns an array of tab bars contained by this object.
tableView s (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textFields (page 65)	Returns an array of text fields contained by the specified object.
textView s (page 66)	Returns an array of text views contained by the specified object.
toolbar (page 66)	Returns the specified toolbar.
toolbars (page 66)	Returns an array of toolbars contained by this object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.

Method	Description
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webViews (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

UIAWebView Class Reference

Inherits from	UIScrollView
Availability	Available in iOS 4.0 and later.

Overview

The `UIAWebView` class allows access to, and control of, web views.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 38-1 provides a list of methods inherited from `UIScrollView`.

Table 38-1 Methods inherited from `UIScrollView`

Method	Description
scrollDown (page 124)	Scrolls down within the specified collection view.
scrollLeft (page 124)	Scrolls left within the specified collection view.
scrollRight (page 124)	Scrolls right within the specified collection view.
scrollToElementWithName (page 124)	Scrolls within the collection view until the named element is displayed on the screen.
scrollToElementWithPredicate (page 125)	Scrolls within the collection view until the matching element is displayed on the screen.
scrollToElementWithValueForKey (page 125)	Scrolls within the collection view until the element with the specified value for the specified key is displayed on the screen.
scrollUp (page 125)	Scrolls up within the specified collection view.

UIAWindow Class Reference

Inherits from	UIAElement
---------------	------------

Availability	Available in iOS 4.0 and later.
--------------	---------------------------------

Overview

The `UIAWindow` class allows access to, and control of, your app's window elements.

For an explanation of how to use this class and related classes, see “Automating UI Testing” in *Instruments User Guide*.

Inherited Methods

Table 39-1 provides a list of methods inherited from `UIAElement`.

Table 39-1 Methods inherited from `UIAElement`

Method	Description
activityIndicators (page 52)	Returns an array of the activity indicators contained by the specified object.
activityView (page 52)	Returns an object representing an activity view.
ancestry (page 53)	Returns an array containing the parents of the specified object.
buttons (page 53)	Returns an array of buttons contained by the specified object.
checkIsValid (page 53)	Returns the specified element's current validity status.
collectionViews (page 53)	Returns an array of collection views contained by the specified object.
doubleTap (page 54)	Double-taps the specified element.
dragInsideWithOptions (page 54)	Drags within the bounds of an element.

Method	Description
elements (page 55)	Returns an array of elements contained by the specified object.
flickInsideWithOptions (page 55)	Flicks within the bounds of an element.
hasKeyboardFocus (page 56)	Determines whether specified element receives keyboard input.
hitpoint (page 56)	Returns the screen position to tap for the specified element.
images (page 56)	Returns an array of images contained by the specified object.
isEnabled (page 56)	Determines whether the specified element is enabled.
isValid (page 57)	Returns the specified element's validity status as of the most recent access.
isVisible (page 57)	Determines whether the specified element is visible on the screen.
label (page 57)	Returns a string containing the label attribute of the element.
links (page 58)	Returns an array of links contained by the specified object.
logElement (page 58)	Logs information about the specified element.
logElementTree (page 58)	Logs information about the specified element and all of its subelements.
name (page 58)	Returns a string containing the name attribute of the element.
pageIndicators (page 59)	Returns an array of page indicators contained by the specified object.
parent (page 59)	Returns the parent of the specified element.
pickers (page 60)	Returns an array of picker objects contained by the specified object.
popover (page 60)	Returns the popover object associated with the specified object.
progressIndicators (page 60)	Returns an array of progress indicators contained by the specified object.
rect (page 60)	Returns the position of the object on the main screen.
rotateWithOptions (page 61)	Perform a rotation gesture centered on the specified element.

Method	Description
scrollToVisible (page 61)	Scrolls until the specified element is visible in a container view.
scrollView s (page 62)	Returns an array of scroll views contained by the specified object.
searchBar s (page 62)	Returns an array of search bars contained by the specified object.
secureTextFields (page 62)	Returns an array of secure text fields contained by the specified object.
segmentedControl s (page 62)	Returns an array of segmented controls contained by the specified object.
slider s (page 62)	Returns an array of sliders contained by the specified object.
staticText s (page 63)	Returns an array of static texts contained by the specified object.
switch s (page 63)	Returns an array of switches contained by the specified object.
tableView s (page 64)	Returns an array of table views contained by the specified object.
tap (page 64)	Taps the specified element.
tapWithOptions (page 64)	Performs the specified gesture on the specified element using a dictionary to specify gesture attributes.
textField s (page 65)	Returns an array of text fields contained by the specified object.
textView s (page 66)	Returns an array of text views contained by the specified object.
touchAndHold (page 66)	Touches the specified element and holds for the specified duration.
twoFingerTap (page 67)	Performs a two-finger (two-touch) tap on this element.
value (page 67)	Returns a string containing a value attribute specific to the type of element.
waitForInvalid (page 67)	Waits for the specified element to become invalid.
webView s (page 67)	Returns an array of web views contained by the specified object.
withName (page 68)	Returns an element whose name attribute matches a specified string.
withPredicate (page 68)	Returns the element matching the specified criteria.

Method	Description
withValueForKey (page 68)	Returns the element containing the specified property with the specified value.

Tasks

Working with Window-level Elements

[contentArea](#) (page 200)

Returns the content area of the window (below the navigation bar and above the tab bar or toolbar).

[navigationBar](#) (page 201)

Returns the the app's navigation bar.

[navigationBars](#) (page 201)

Returns an array of the navigation bars contained by this object.

[tabBar](#) (page 201)

Returns the app's tab bar.

[tabBars](#) (page 201)

Returns an array of tab bars contained by this object.

[toolbar](#) (page 201)

Returns the the app's toolbar.

[toolbars](#) (page 202)

Returns an array of toolbars contained by this object.

Methods

contentArea

Returns the content area of the window (below the navigation bar and above the tab bar or toolbar).

(Rect) `contentArea()`

Availability

Available in iOS 4.0 and later.

navigationBar

Returns the app's navigation bar.

(UINavigationController) `navigationBar()`

Availability

Available in iOS 4.0 and later.

navigationBars

Returns an array of the navigation bars contained by this object.

(UIAElementArray) `navigationBars()`

Availability

Available in iOS 4.2 and later.

tabBar

Returns the app's tab bar.

(UITabBar) `tabBar()`

Availability

Available in iOS 4.0 and later.

tabBars

Returns an array of tab bars contained by this object.

(UIAElementArray) `tabBars()`

Availability

Available in iOS 4.2 and later.

toolbar

Returns the app's toolbar.

(UIToolbar) `toolbar()`

Availability

Available in iOS 4.0 and later.

toolbars

Returns an array of toolbars contained by this object.

(UIAElementArray) toolbars()

Availability

Available in iOS 4.2 and later.

Document Revision History

This table describes the changes to *UI Automation JavaScript Reference*.

Date	Notes
2012-09-19	Added links to two new classes. Formerly titled UI Automation Reference Collection.
2012-02-16	Fixed a broken link.
2011-10-12	Added new UIAHost class, described new features.
2010-11-15	Updated to include new UIAPopover Class Reference.
2010-09-01	Updated to include minor corrections.
2010-05-27	New document that describes the JavaScript classes used to support automated iPhone application user interface testing.



Apple Inc.

© 2012 Apple Inc.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Apple Inc.

1 Infinite Loop

Cupertino, CA 95014

408-996-1010

Apple, the Apple logo, Cocoa, Instruments, iPhone, iPod, iPod touch, and iTunes are trademarks of Apple Inc., registered in the U.S. and other countries.

Java is a registered trademark of Oracle and/or its affiliates.

iOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.