Class

XCUIElement

A UI element in an application.

iOS | iPadOS | Mac Catalyst | macOS | tvOS | visionOS | watchOS | Xcode 16.3+

```
@MainActor
class XCUIElement
```

Overview

In macOS and iPadOS 15 and later, `XCUIElement` provides a way to test your app with keyboard and mouse interactions, such as typing, clicking, scrolling, and moving and pausing the pointer. In iOS, `XCUIElement` provides a way to test your app with gestures, such as tapping, swiping, pinching, and rotating.

> **Note**
>
> `XCUIElement` adopts the `XCUIElementAttributes` protocol, which provides additional properties for querying the current state of a UI element's attributes.

Topics

Querying element state

```
func waitForExistence(timeout: TimeInterval) -> Bool
```

Waits the specified amount of time for an element to exist.

```
func waitForNonExistence(timeout: TimeInterval) -> Bool
```

Waits the specified amount of time for an element to no longer exist.

```
func wait<V>(for: KeyPath<XCUIElement, V>, toEqual: V, timeout: TimeInterval) -> Bool
```

Waits a specified amount of time for a property value to equal a specified value.

```
var exists: Bool
```

Determines if the element exists.

```
var isHittable: Bool
```

Determines if the system can compute a hit point for the element.

```
var debugDescription: String
```

Provides debugging information about the element.

Querying descendant elements

```
func children(matching: XCUIElement.ElementType) -> XCUIElementQuery
```

Returns a query for all direct children of the element matching the type you specify.

```
func descendants(matching: XCUIElement.ElementType) -> XCUIElementQuery
```

Returns a query for all descendants of the element matching the type you specify.

Typing text

```
func typeText(String)
```

Types a string into the element.

Combining keystrokes

```
func typeKey(XCUIKeyboardKey, modifierFlags: XCUIElement.KeyModifierFlags)
```

Types a single key from the XCUIKeyboardKey enumeration with the specified modifier flags.

```
func typeKey(String, modifierFlags: XCUIElement.KeyModifierFlags)
```

Types a single key that a string represents with the flags you specify.

```
struct XCUIKeyboardKey
```

Constants to represent keys that have no typewritten equivalent.

```
class func perform(withKeyModifiers: XCUIElement.KeyModifierFlags, block: () -> Void)
```

Executes a block of code while holding a combination keystroke.

```
struct KeyModifierFlags
```

Flags for simulating combination keystrokes with keys, such as Control, Option, Shift, and Command.

Moving the pointer

```
func hover()
```

Moves the pointer over the element.

Clicking

```
func click()
```

Sends a click event to a hittable point computed for the element.

```
func click(forDuration: TimeInterval, thenDragTo: XCUIElement)
```

Clicks and holds an element for a duration you specify, and then drags it to another element.

```
func click(forDuration: TimeInterval, thenDragTo: XCUIElement, withVelocity: XCUIGestureVelocity, thenHoldForDuration: TimeInterval)
```

Clicks and holds an element for a duration, drags it at a velocity, and holds it over another element for a duration, all of which you specify.

```
func doubleClick()
```

Sends a double-click event to a hittable point the system computes for the element.

```
func rightClick()
```

Sends a Control-click event to a hittable point the system computes for the element.

Scrolling

```
func scroll(byDeltaX: CGFloat, deltaY: CGFloat)
```

Scrolls the view by the number of x and y pixels you specify.

Tapping and pressing

```
func tap()
```

Sends a tap event to a hittable point the system computes for the element.

`func doubleTap()`

Sends a double-tap event to a hittable point the system computes for the element.

`func press(forDuration: TimeInterval)`

Sends a press-and-hold gesture to a hittable point the system computes for the element, holding for the duration you specify.

`func press(forDuration: TimeInterval, thenDragTo: XCUIElement)`

Initiates a press-and-hold gesture, then drags to another element.

`func press(forDuration: TimeInterval, thenDragTo: XCUIElement, withVelocity: XCUIGestureVelocity, thenHoldForDuration: TimeInterval)`

Initiates a press-and-hold gesture, drags to another element at a velocity, and holds for a duration, all of which you specify.

Tapping multiple times

`func twoFingerTap()`

Sends a two-finger tap event to a hittable point the system computes for the element.

`func tap(withNumberOfTaps: Int, numberOfTouches: Int)`

Sends one or more taps with one or more touch points.

Performing gestures

`func swipeLeft()`

Sends a swipe-left gesture.

`func swipeLeft(velocity: XCUIGestureVelocity)`

Sends a swipe-left gesture with a velocity you specify.

`func swipeRight()`

Sends a swipe-right gesture.

`func swipeRight(velocity: XCUIGestureVelocity)`

Sends a swipe-right gesture with a velocity you specify.

`func swipeUp()`

Sends a swipe-up gesture.

`func swipeUp(velocity: XCUIGestureVelocity)`

Sends a swipe-up gesture with a velocity you specify.

`func swipeDown()`

Sends a swipe-down gesture.

`func swipeDown(velocity: XCUIGestureVelocity)`

Sends a swipe-down gesture with a velocity you specify.

`func pinch(withScale: CGFloat, velocity: CGFloat)`

Sends a pinching gesture with two touches.

`func rotate(CGFloat, withVelocity: CGFloat)`

Sends a rotation gesture with two touches.

`struct XCUIGestureVelocity`

A value that describes how fast a gesture moves across the screen, in pixels per second.

Interacting with sliders

`var normalizedSliderPosition: CGFloat`

Returns the position of the slider's indicator as a normalized value.

`func adjust(toNormalizedSliderPosition: CGFloat)`

Manipulates the UI to change the value the slider displays to a new value, based on a normalized position.

Interacting with pickers

`func adjust(toPickerWheelValue: String)`

Changes the value that the picker wheel displays.

Calculating coordinates

`func coordinate(withNormalizedOffset: CGVector) -> XCUICoordinate`

Creates and returns a new coordinate with a normalized offset.

Supporting types

`enum ElementType`

The types of UI elements that you find, inspect, and interact with in a UI test.

`enum SizeClass`

The user interface size classes you can inspect in a UI test.

`struct AttributeName`

A set of string constants that serve as keys for storing element attributes in a dictionary.

Deprecated methods

~~`func swipeDown(withVelocity: XCUIGestureVelocity)`~~

Sends a swipe-down gesture with a velocity you specify.

Deprecated

~~`func swipeUp(withVelocity: XCUIGestureVelocity)`~~

Sends a swipe-up gesture with a velocity you specify.

Deprecated

~~`func swipeLeft(withVelocity: XCUIGestureVelocity)`~~

Sends a swipe-left gesture with a velocity you specify.

Deprecated

~~`func swipeRight(withVelocity: XCUIGestureVelocity)`~~

Sends a swipe-right gesture with a velocity you specify.

Deprecated

Relationships

Inherits From

`NSObject`

Inherited By

`XCUIApplication`

Conforms To

CVarArg
Copyable
CustomDebugStringConvertible
CustomStringConvertible
Equatable
Hashable
NSObjectProtocol
Sendable
XCUIElementAttributes
XCUIElementSnapshotProviding
XCUIElementTypeQueryProvider
XCUIScreenshotProviding

See Also

UI elements

`protocol` XCUIElementAttributes

Attributes exposed by UI elements.

`protocol` XCUIElementSnapshot

A set of attributes to express a snapshot of an element's attributes and descendant user interface hierarchy.

`protocol` XCUIElementSnapshotProviding

A method to capture a snapshot of an element's attributes and descendant user interface hierarchy.

`class` XCUICoordinate

A location on screen relative to a UI element.