NSAccessibility

A legacy, informal protocol that Apple doesn't recommend for active use.

Overview

The `NSAccessibility` informal protocol defines an old, key-based API. For the most part, Apple doesn't recommend using this API. Use the method-based API in NSAccessibilityProtocol instead. However, there are a few methods and properties that are still relevant. You can combine the accessibilityHitTest(_:) method, and the accessibilityFocusedUIElement and accessibilityNotifiesWhenDestroyed properties with the new NSAccessibility Protocol protocol.

Topics

Available Methods and Properties

`var accessibilityFocusedUIElement: Any? { get }`

`func accessibilityHitTest(_ point: NSPoint) -> Any?`

`var accessibilityNotifiesWhenDestroyed: Bool { get }`

A Boolean value that indicates whether a custom accessibility object sends a notification when its corresponding UI element is destroyed.

Constants

☰ Standard Attributes

Standard attributes that can be adopted by any accessibility object.

:≡ Text-Specific Attributes

Attributes that are specific to text.

:≡ Text-Specific Parameterized Attributes

Parameterized attributes specific to text.

:≡ Text Attributed-String Attributes and Constants

Attributes and key constants used with attributed strings.

:≡ Window-Specific Attributes

Attributes specific to windows.

:≡ App-Specific Attributes

Attributes that are specific to the app object.

:≡ Grid View Attributes

Attributes that are used with grid views, such as thumbnails and media browsers that present a grid of items. The children of a grid are ordered.

:≡ Table View and Outline View Attributes

Attributes that are specific to tables and outlines.

:≡ Outline View Attributes

Attributes that are used in outline views.

:≡ Cell-Based Table Attributes

Attributes that are specific to cell-based tables.

:≡ Cell-Based Table Parameterized Attributes

Parameterized attributes specific to cell-based tables.

:≡ Cell Attributes

Attributes that are specific to individual table cells.

:≡ Layout Area Attributes

Attributes that are specific to layout areas.

:≡ Layout Area Parameterized Attributes

Parameterized attributes that are specific to layout areas.

:≡ Layout Item Attributes

Attributes that are specific to the items in a layout area.

≡ Slider Attributes

Attributes that are specific to sliders.

≡ Screen Matte Attributes

Attributes that are specific to screen mattes.

≡ Ruler View Attributes

Attributes that are specific to ruler views.

≡ Linkage Elements

Constants that specify links between accessibility elements.

≡ Miscellaneous Attributes

Miscellaneous attributes that can apply to various types of elements.

≡ Column Sort Direction

Values that indicate the sort direction of a column.

≡ Measurement Unit Attributes

Values that indicate the unit values of a ruler or layout area.

≡ Orientations

Values that indicate the orientation of elements, such as scroll bars and split views.

≡ Ruler Marker Type Values

Values that indicate the marker type of an element.

≡ Actions

Standard actions that accessibility objects can perform.

Deprecated

func accessibilityActionDescription(_ action: NSAccessibility.Action) -> String?

Returns a localized description of the specified action.

`Deprecated`

func accessibilityActionNames() -> [NSAccessibility.Action]

Returns an array of action names supported by the accessibility element.

`Deprecated`

```
func accessibilityArrayAttributeCount(_ attribute: NSAccessibility.
Attribute) -> Int
```

Returns the count of the specified accessibility array attribute.

```
func accessibilityArrayAttributeValues(_ attribute: NSAccessibility.
Attribute, index: Int, maxCount: Int) -> [Any]
```

Returns a subarray of values of an accessibility array attribute.

```
func accessibilityAttributeNames() -> [NSAccessibility.Attribute]
```

Returns an array of attribute names supported by the receiver.

Deprecated

```
func accessibilityAttributeValue(_ attribute: NSAccessibility.Attribute
) -> Any?
```

Returns the value of the specified attribute in the receiver.

Deprecated

```
func accessibilityAttributeValue(_ attribute: NSAccessibility.
ParameterizedAttribute, forParameter parameter: Any?) -> Any?
```

Returns the value of the receiver's parameterized attribute corresponding to the specified
attribute name and parameter.

Deprecated

```
func accessibilityIndex(ofChild child: Any) -> Int
```

Returns the index of the specified accessibility child in the parent.

```
func accessibilityIsAttributeSettable(_ attribute: NSAccessibility.
Attribute) -> Bool
```

Returns a Boolean value that indicates whether the value for the specified attribute in the
receiver can be set.

Deprecated

```
func accessibilityIsIgnored() -> Bool
```

Returns a Boolean value indicating whether the receiver should be ignored in the parent-child
accessibility hierarchy.

Deprecated

```
func accessibilityParameterizedAttributeNames() -> [NSAccessibility.
ParameterizedAttribute]
```

Returns a list of parameterized attribute names supported by the receiver.

Deprecated

~~func **accessibilityPerformAction**(_ action: NSAccessibility.Action)~~

Performs the action associated with the specified action.

Deprecated

~~func **accessibilitySetOverrideValue**(_ value: Any?, **forAttribute** attribute: NSAccessibility.Attribute) -> Bool~~

Overrides the specified attribute in the receiver or adds it if it does not exist, and sets its value to the specified value.

Deprecated

~~func **accessibilitySetValue**(_ value: Any?, **forAttribute** attribute: NSAccessibility.Attribute)~~

Sets the value of the specified attribute in the receiver to the specified value.

Deprecated

See Also

Protocols

protocol `NSEditorRegistration`

A set of methods that controllers can implement to enable an editor view to inform the controller when it has uncommitted changes.

protocol `NSInputServiceProvider`

protocol `NSInputServerMouseTracker`

~~protocol NSDrawerDelegate~~

A set of methods that drawer delegates implement to open, close, and resize the drawer.

Deprecated