2016

UIAccessibilityAction

非正式协议

中国信息无障碍产品联盟&信息无障碍研究会　译制

刘辉 20160815

原文地址

目录

UIAccessibilityAction 简介

继承自：Not Applicable;

遵循：Not Applicable;

导入语句：

```
OBJECTIVE-C

@import UIKit;
```

可获得性：在 iOS4.0 及更高版本中可获得；

UIAccessibilityAction 非正式协议为无障碍元素提供一种方式，以支持特定操作，例如在范围内选择值，或在屏幕上滚动信息。例如，为了响应滚动手势，开发者需要实现 accessibilityScroll: 方法并发送具有新页面状态的 UIAccessibilityPageScrolledNotification（例如“第三页共九页（Page 3 of 9）”）。或者，为了实现一个元素可访问，例如滑动条或选择器视图，首先需要包含 UIAccessibilityTraitAdjustable 特性来描述该元素的特征。然后，开发者必须实现 accessibilityIncrement 和 accessibilityDecrement 方法。当做完这些，辅助技术用户可以使用辅助技术特殊手势调整该元素。

1. 执行操作

1.1 – accessibilityActivate

告知元素激活自己，并报告操作成功或失败。

声明

SWIFT func accessibilityActivate() -> Bool
OBJECTIVE-C - (BOOL)accessibilityActivate

返回值

如果元素被激活，返回 YES，如果未被激活，返回 NO。

简介

开发者可以使用该方法让用户更容易的访问复杂控件。当 VoiceOver 用户双击选定元素时，无障碍系统调用该方法。实现该方法应该可以激活元素并执行其认为合理的其他任务。例如，开发者可能使用该方法激活一个控件，该控件激活需要一个复杂手势，且 VoiceOver 用户很难完成，可能是因为当 VoiceOver 运行时，该手势有不一样的含义。

执行任务之后，返回一个适当的布尔值标识操作成功或失败。

可获得性

在 iOS7.0 及更高的版本中可获得。

1.2 – accessibilityIncrement

告知无障碍元素增加其内容的值。

声明

SWIFT
func accessibilityIncrement()
OBJECTIVE-C
- (void)accessibilityIncrement

简介

如果元素具有 UIAccessibilityTraitAdjustable 特性，开发者必须实现该方法。使用该方法增加元素的值。例如，一个 UISlider 对象使用该方法以合适的数量增加该元素的值。

可获得性

在 iOS4.0 及更高的版本中可获得。

1.3 – accessibilityDecrement

告知无障碍元素减少其内容的值。

声明

SWIFT
func accessibilityDecrement()
OBJECTIVE-C
- (void)accessibilityDecrement

简介

如果元素具有 UIAccessibilityTraitAdjustable 特性，开发者必须实现该方法。使用该方法减少元素的值。例如，一个 UISlider 对象使用该方法以合适的数量减少该元素的值。

可获得性

在 iOS4.0 及更高的版本中可获得。

1.4 – accessibilityScroll:

以一种应用指定的方式滚动屏幕内容，并返回操作成功或失败。

声明

SWIFT
func accessibilityScroll(_ *direction*: UIAccessibilityScrollDirection) -> Bool
OBJECTIVE-C
- (BOOL)accessibilityScroll:(UIAccessibilityScrollDirection)*direction*

参数

direction ：指定滚定操作方向的常量。有效常量的描述，详见 UIAccessibilityScrollDirection。

返回值

如果滚动操作成功，返回 YES；否则，返回 NO。该方法默认返回 NO。

简介

如果一个在视图层次中的视图支持页面滚动操作，需要实现该方法。

- 如果特定方向的滚动操作成功，返回 YES 并发送 UIAccessibilityPageScrolledNotification 通知。

- 如果滚动操作失败，在视图层次中由父视图调用 accessibilityScroll: 。

可获得性

在 iOS4.2 及更高的版本中可获得。

1.5 – accessibilityPerformEscape

关闭一个模态视图，并返回该操作的成功或失败。

声明

SWIFT func accessibilityPerformEscape() -> Bool
OBJECTIVE-C - (BOOL)accessibilityPerformEscape

返回值

如果模态视图被成功关闭，返回 YES；否则，返回 NO。该方法默认返回 NO。

简介

在一个元素上、或在可以模态出现或视图层次中的包含视图上，实现该方法。当 VoiceOver 用户执行关闭操作，该方法关闭视图。例如，为了给用户一个从容的关闭操作来执行关闭浮窗，开发者可能需要为该浮窗实现该方法。

可获得性

在 iOS5.0 及更高的版本中可获得。

1.6 – accessibilityPerformMagicTap

执行一个突出的操作。

声明

SWIFT

```
func accessibilityPerformMagicTap() -> Bool
```

OBJECTIVE-C

```
- (BOOL)accessibilityPerformMagicTap
```

返回值

如果魔法轻拍操作成功，返回 YES；否则，返回 NO。该方法默认返回 NO。

简介

该方法执行的精确操作来自应用，一般会弹出应用最重要的状态。例如，在电话应用中，魔法轻拍执行应答和结束通话；在音乐应用中，魔法轻拍执行播放和暂停音乐；在计时应用中，魔法轻拍执行启动和停止计时器；在相机应用中，执行拍照操作。

可获得性

在 iOS6.0 及更高的版本中可获得。

2. 访问自定义操作

2.1 accessibilityCustomActions 属性

与内置操作一起呈现的自定义操作数组。

声明

SWIFT

```
var accessibilityCustomActions: [UIAccessibilityCustomAction]?
```

OBJECTIVE-C

@property(nonatomic, strong) NSArray <UIAccessibilityCustomAction *> *accessibilityCustomActions

简介

该数组包含一个或多个定义已支持操作的 UIAccessibilityCustomAction 对象。辅助技术，例如 VoiceOver，会在合适的时间呈现自定义操作。

可获得性

在 iOS8.0 及更高的版本中可获得。

3. 数据类型

3.1 UIAccessibilityScrollDirection

一个滚动操作的方向。

声明

SWIFT

```
enum UIAccessibilityScrollDirection : Int {
    case Right
    case Left
    case Up
    case Down
    case Next
    case Previous
```

```
}
```

OBJECTIVE-C

```
typedef enum {
    UIAccessibilityScrollDirectionRight = 1,
    UIAccessibilityScrollDirectionLeft,
    UIAccessibilityScrollDirectionUp,
    UIAccessibilityScrollDirectionDown,
    UIAccessibilityScrollDirectionNext,
    UIAccessibilityScrollDirectionPrevious
} UIAccessibilityScrollDirection;
```

常量

- UIAccessibilityScrollDirectionRight

  用户执行向右的滚动操作。

  在 iOS4.2 及更高的版本中可获得。

- UIAccessibilityScrollDirectionLeft

  用户执行向左的滚动操作。

  在 iOS4.2 及更高的版本中可获得。

- UIAccessibilityScrollDirectionUp

  用户执行向上的滚动操作。

  在 iOS4.2 及更高的版本中可获得。

- UIAccessibilityScrollDirectionDown

  用户执行向下的滚动操作。

  在 iOS4.2 及更高的版本中可获得。

- UIAccessibilityScrollDirectionNext

  用户以有序视图集的顺序执行滚动到下一个视图的操作。

  在 iOS5.0 及更高的版本中可获得。

- UIAccessibilityScrollDirectionPrevious

  用户以有序视图集的顺序执行滚动到上一个视图的操作。

  在 iOS5.0 及更高的版本中可获得。

引入语句

OBJECTIVE-C @import UIKit;
SWIFT import UIKit

可获得性

在 iOS4.2 及更高的版本中可获得。