

UIAccessibility 类

中国信息无障碍产品联盟&信息无障碍研究会 译制

20160902

翻译声明

翻译机构：信息无障碍研究会 (ARA) 中国信息无障碍产品联盟 (CAPA)

译者：刘辉

审阅：刘彪、沈广荣

本文档翻译自苹果 iOS Developer Library 的官方文档[无障碍类](#)，如您对翻译文档内容有异议，请将原文文档作为主要参考，原文版权由 Apple Inc 持有并保留。

本翻译文档使用请参见 [CC BY-NC-SA 3.0](#)。文档可以免费使用、分享，但请保留本链接，如您有任何内容上的修改，请发送邮件至 liuhui@siaa.org.cn，我们只是希望文档内容能够统一完整，真正帮助开发者完善产品的信息无障碍。

目录

翻译声明.....	0
1.UIAccessibilityCustomAction	1
1.1 初始化自定义操作.....	2
1.1.1 - initWithName:target:selector:.....	2
1.2 访问操作参数.....	3
1.2.1 name 属性.....	3
1.2.2 target 属性	3
1.2.3 selector 属性	4
2.UIAccessibilityElement	5
2.1 创建一个无障碍元素.....	6
2.1.1 - initWithAccessibilityContainer:	6
2.2 访问包含视图.....	7
2.2.1 accessibilityContainer 属性	7
2.3 判定无障碍.....	7
2.3.1 isAccessibilityElement 属性.....	7
2.4 访问无障碍元素的属性.....	8
2.4.1 accessibilityLabel 属性.....	8
2.4.2 accessibilityHint 属性.....	8
2.4.3 accessibilityValue 属性.....	9
2.4.4 accessibilityFrame 属性.....	9
2.4.5 accessibilityTraits 属性.....	10

1.UIAccessibilityCustomAction

继承自: `NSObject` `UIAccessibilityCustomAction`;

遵循: `NSObject`;

导入语句:

```
OBJECTIVE-C
```

```
@import UIKit;
```

可获得性: 在 iOS8.0 及更高版本中可获得;

一个 `UIAccessibilityCustomAction` 对象代表一个在无障碍对象上执行的自定义操作。支持自定义操作的应用可以创建该类的实例, 当执行操作的时候, 指定操作的用户可读名称、对象和使用的选择器。为了响应特定用户提示, 辅助技术了呈现自定义操作。例如, `VoiceOver` 让用户使用转子快速访问操作。

创建该类的实例后, 将其添加到适当无障碍对象的 `accessibilityCustomActions` 属性中。

1.1 初始化自定义操作

1.1.1 - initWithName:target:selector:

返回指定名称的初始化自定义操作对象。

声明

SWIFT

```
init(name name: String,  
      target target: AnyObject?,  
      selector selector: Selector)
```

OBJECTIVE-C

```
-(instancetype)initWithName:(NSString *)name  
                        target:(id)target  
                        selector:(SEL)selector
```

参数

name : 操作的本地化名称。为操作提供一个简短的描述性名称。

target : 执行操作的对象。

selector : 当想要执行操作时, *target* 调用的选择器。方法签名必须实现以下形式之一:

- | | |
|---|---|
| 1 | - (BOOL)myPerformActionMethod |
| 2 | - (BOOL)myPerformActionMethod:(UIAccessibilityCustomAction *)action |

返回值

一个初始化自定义操作对象。

可获得性

在 iOS8.0 及更高的版本中可获得。

1.2 访问操作参数

1.2.1 name 属性

操作的本地化名称。

声明

SWIFT

```
var name: String
```

OBJECTIVE-C

```
@property(nonatomic, copy) NSString *name
```

可获得性

在 iOS8.0 及更高的版本中可获得。

1.2.2 target 属性

执行操作的对象。

声明

SWIFT

```
weak var target: AnyObject?
```

OBJECTIVE-C

@property(nonatomic, weak) id target

可获得性

在 iOS8.0 及更高的版本中可获得。

1.2.3 selector 属性

执行操作的方法。

声明

SWIFT

```
var selector: Selector
```

OBJECTIVE-C

```
@property(nonatomic, assign) SEL selector
```

简介

选择器的签名必须采用以下方式之一：

- | | |
|---|---|
| 1 | - (BOOL)myPerformActionMethod |
| 2 | - (BOOL)myPerformActionMethod:(UIAccessibilityCustomAction *)action |

当用户选择一个自定义操作，辅助技术调用 `target` 属性对象中的指定方法。使用该方法执行已标识的操作。

可获得性

在 iOS8.0 及更高的版本中可获得。

2. UIAccessibilityElement

继承自: Not Application;

遵循: Not Application;

导入语句: Not Application;

可获得性: Not Application;

UIAccessibilityElement 类封装了一个项目的相关信息，该项目应该对残障用户无障碍，但却不是默认可访问。例如，一个图标或文本图像不能自动无障碍，因为其不是继承自 UIView（或 UIControl）。一个包含非视图项目的视图可以创建一个 UIAccessibilityElement 实例，来呈现每个需要可访问的项目。

无障碍元素的属性向辅助应用提供该元素的信息，例如位置和当前值。尽管可能不需要去为该元素创建一个 UIAccessibilityElement 实例，但开发者可能需要设置元素的属性。例如，如果应用包含一个自定义保存图标的按钮，该按钮已被作为一个无障碍元素呈现，因为它是 UIButton 的子类。但是，开发者需要为 label 和 hint 属性提供信息，因为该信息专属于该按钮。开发者可以在 Interface Builder 中，或在 UIAccessibility 非正式协议中，设置该属性。

2.1 创建一个无障碍元素

2.1.1 - `initWithAccessibilityContainer:`

创建和初始化一个无障碍元素，在指定容器中呈现一个项目。

声明

```
- (id)initWithAccessibilityContainer:(id)container
```

参数

container : 包含项目的视图。

返回值

在容器中呈现一个非视图项目的无障碍元素。

简介

一般来说，开发者不需要在应用中为项目创建无障碍元素，因为标准 `UIKit` 控件和视图默认可访问。但是，如果存在一个视图包含非视图项目，例如图标和文本图像，且需要对残障用户可访问，开发者需要创建无障碍元素去呈现它们。在该情景下，包含视图应该实现 `UIAccessibilityContainer` 非正式协议，并使用该方法创建一个无障碍元素，来呈现每个应该暴露给辅助应用的项目。

2.2 访问包含视图

2.2.1 accessibilityContainer 属性

包含无障碍元素的视图。

声明

```
@property(nonatomic, assign) id accessibilityContainer
```

2.3 判定无障碍

2.3.1 isAccessibilityElement 属性

一个布尔值，标识项目是否是个辅助应用可访问的无障碍元素。

声明

```
@property(nonatomic, assign) BOOL isAccessibilityElement
```

简介

该属性的默认值为 NO。如果接收者是个 UIKit 控件，默认值为 YES。

2.4 访问无障碍元素的属性

2.4.1 accessibilityLabel 属性

一个简洁标识无障碍元素的字符串。

声明

```
@property(n nonatomic, retain) NSString *accessibilityLabel
```

简介

Label 是个简短的、本地化的字符串，用来标识无障碍元素，但应不包含控件或视图的类型。例如，保存按钮的 label 应该为“保存”，不是“保存按钮”。

标准 UIKit 控件和视图的 label 默认是从其标题属性中获取的。如果提供了一个自定义控件或视图，开发者需要适当地设置该属性，这样辅助应用可以为残障用户提供准确信息。

2.4.2 accessibilityHint 属性

一个字符串，简洁地描述无障碍元素执行操作的结果。

声明

```
@property(n nonatomic, retain) NSString *accessibilityHint
```

简介

Hint 是个简洁的、本地化描述文本，描述元素上执行操作的结果，而不是标识元素或操作。例如，一个包含邮箱信息的表行的 hint 可能为“选择该信息”，但不能是“点击该行选择信息”。

标准 UIKit 控件和视图默认由系统提供 hints。如果提供一个自定义控件或视图，开发者需要适当地设置该属性，这样辅助技术可以为残障用户提供准确信息。

2.4.3 accessibilityValue 属性

一个字符串，呈现无障碍元素的当前值。

声明

```
@property(n nonatomic, retain) NSString *accessibilityValue
```

简介

Value 是个包含元素当前值的本地化字符串。例如，一个滑块的值可能是 9.5 或 35%，文本域的值是其包含的文本。

当一个无障碍元素具有一个不能使用 label 呈现的值的时候，使用该属性。例如，一个声音滑块的 label 可能是“音量”，但是其 value 为当前音量级。在该情景下，对用户来说仅仅知道滑块的标识是不够的，因为用户也需要知道其当前值。保存按钮的 label，换句话说，告知用户需要知道的该按钮的所有信息，将“保存”作为 value 是不必要且令人困扰的。

2.4.4 accessibilityFrame 属性

屏幕坐标中，无障碍元素的框架。

声明

```
@property(n nonatomic, assign) CGRect accessibilityFrame
```

简介

当在应用中创建一个无障碍元素来呈现元素，开发者必须在 CGRect 结构中设置该属性，CGRect 结构指定对象的屏幕位置和大小。（继承自 UIView 的对象默认包含该信息。）

参见

[@property\(n nonatomic, assign\) CGRect accessibilityFrame](#)

2.4.5 accessibilityTraits 属性

Traits 的组合，能够最好地描述无障碍元素的特征。

声明

```
@property(nonatomic, assign) UIAccessibilityTraits accessibilityTraits
```

简介

一个 trait 描述元素行为、状态或用法的一个方面。在该属性中，几个 traits 被组合起来（使用一个 OR 运算）为辅助应用提供元素的完整画像。完整 traits 列表，参见 [UIAccessibility Protocol Reference](#) 中的“无障碍特性(Accessibility Traits)”。

UIKit 为所有标准控件和视图提供适当的 traits 组合。当为一个自定义无障碍元素组合 traits 时，保证：

- 注意常识信息。请勿将相互排斥的 trait 组合在一起，例如组合按钮和搜索框 trait。
- 组合已选择 trait 和父类 trait。特别是，在用来设置自定义元素 trait 的方法中，总是组合自定义 trait 和[super accessibilityTraits]。