

UIKit 无障碍函数索引

中国信息无障碍产品联盟&信息无障碍研究会 译制

20160909

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[无障碍函数索引](#)，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
UIKit 函数索引简介	1
1.无障碍.....	2
1.1 UIAccessibilityPostNotification	2
1.2 UIAccessibilityConvertFrameToScreenCoordinates	3
1.3 UIAccessibilityConvertPathToScreenCoordinates	4
1.4 UIAccessibilityRegisterGestureConflictWithZoom	5
1.5 UIAccessibilityRequestGuidedAccessSession	5
1.6 UIAccessibilityZoomFocusChanged	7
1.7 UIAccessibilityIsBoldTextEnabled.....	7
1.8 UIAccessibilityIsClosedCaptioningEnabled.....	8
1.9 UIAccessibilityDarkerSystemColorsEnabled.....	9
1.10 UIAccessibilityIsGrayscaleEnabled	9
1.11 UIAccessibilityIsGuidedAccessEnabled	10
1.12 UIAccessibilityIsInvertColorsEnabled	10
1.13 UIAccessibilityIsMonoAudioEnabled.....	11
1.14 UIAccessibilityIsReduceMotionEnabled.....	11
1.15 UIAccessibilityIsReduceTransparencyEnabled.....	12
1.16 UIAccessibilityIsSpeakScreenEnabled.....	13
1.17 UIAccessibilityIsSpeakSelectionEnabled.....	13
1.18 UIAccessibilityIsSwitchControlRunning	14
1.19 UIAccessibilityIsVoiceOverRunning.....	14

UIKit 函数索引简介

继承自: Not Application;

遵循: Not Application;

导入语句:

```
OBJECTIVE-C
```

```
@import UIKit;
```

可获得性: Not Application;

UIKit 框架定义了一批函数，其中很多函数被用于图形和绘图操作。

1.无障碍

1.1 UIAccessibilityPostNotification

向辅助应用发送一个通知。

声明

SWIFT

```
func UIAccessibilityPostNotification(_ notification: UIAccessibilityNotifications, _  
argument: AnyObject?)
```

OBJECTIVE-C

```
void UIAccessibilityPostNotification ( UIAccessibilityNotifications notification, id  
argument );
```

参数

notification : 要发送的通知（全部“通知”详见 [UIAccessibility Protocol Reference](#) 通知列表）。

argument : 被通知指定的参数。默认传递 `nil`，除非通知指定了另外的参数。

简介

如果应用具有频繁改变、出现或消失的用户界面元素，应用可能需要发送无障碍通知。

可获得性

在 iOS3.0 及更高的版本中可获得。

1.2

UIAccessibilityConvertFrameToScreenCoordinates

将指定矩形从视图坐标转换为屏幕坐标。

声明

SWIFT

```
func UIAccessibilityConvertFrameToScreenCoordinates(_ rect: CGRect, _ view: UIView) -> CGRect
```

OBJECTIVE-C

```
CGRect UIAccessibilityConvertFrameToScreenCoordinates ( CGRect rect, UIView *view );
```

参数

rect : 指定视图 (*view*) 坐标系统中的指定矩形。

view : 包含指定矩形的视图。该参数不能为 `nil`。

返回值

屏幕坐标中的矩形。

简介

使用该函数将无障碍框架矩形转换为屏幕坐标。

可获得性

在 iOS7.0 及更高的版本中可获得。

1.3 UIAccessibilityConvertPathToScreenCoordinates

将指定路径对象转换为屏幕坐标，并返回结果中的新路径对象。

声明

SWIFT

```
func UIAccessibilityConvertPathToScreenCoordinates(_ path: UIBezierPath, _  
view: UIView) -> UIBezierPath
```

OBJECTIVE-C

```
UIBezierPath * UIAccessibilityConvertPathToScreenCoordinates ( UIBezierPath  
*path, UIView *view );
```

参数

path : 想要转换的路径对象。被用来创建该路径对象的坐标值应该与指定视图 (*view*) 的坐标系相关联。该参数不能为 nil。

view : 该视图的坐标系被用来定义路径。该参数不能为 nil。

返回值

一个新路径对象，具有与 *path* 相同的形态，但 *path* 的点是 被指定在屏幕坐标中。

简介

该函数调整路径的点，开发者可将这些点提供给无障碍系统可使用的数值。在路径对象被发送到无障碍系统处理之前，开发者可以使用该函数转换应用用户界面使用的路径对象。

可获得性

在 iOS7.0 及更高的版本中可获得。

1.4 UIAccessibilityRegisterGestureConflictWithZoom

警告用户应用的特定手势与系统定义的无障碍缩放手势冲突。

声明

SWIFT

```
func UIAccessibilityRegisterGestureConflictWithZoom()
```

OBJECTIVE-C

```
void UIAccessibilityRegisterGestureConflictWithZoom ( void );
```

简介

如果应用使用的多指手势与系统缩放手势（也就是，三指手势）冲突，使用该函数。在该情景下，会呈现给用户关闭缩放或继续的选择，

可获得性

在 iOS5.0 及更高的版本中可获得。

1.5 UIAccessibilityRequestGuidedAccessSession

将应用异步转换为单 App 模式或退出单 App 模式。

声明

SWIFT

```
func  UIAccessibilityRequestGuidedAccessSession(_ enable: Bool, _  
completionHandler: (Bool) -> Void)
```

OBJECTIVE-C

```
void  UIAccessibilityRequestGuidedAccessSession ( BOOL enable, void  
(^completionHandler)(BOOL didSucceed) );
```

参数

enable : 值为 YES 时，为该应用将设备置为单 App 模式；值为 NO 时，退出单 App 模式。

completionHandler : 告知应用操作成功或失败的块。该块具有以下参数：

didSucceed : 如果值为 YES，则应用成功转换为或退出单 App 模式。如果值为 NO，应用或设备没有资格获得单 App 模式，或存在一些其他错误。

简介

开发者可以使用该方法将应用锁定为单 App 模式，并稍后将其从该模式释放。例如，一个测试应用可能在测试开始时进入该模式，当用户完成测试时退出该模式。进入单 App 模式仅支持使用移动设备管理（MDM）监管的设备，且应用自己必须可以由 MDM 激活该模式。开发者必须平衡进入单 App 模式和退出该模式的调用。

因为进入或退出单 App 模式可能会需要一些时间，该方法异步执行并使用 *completionHandler* 块告知执行结果。

可获得性

在 iOS7.0 及更高的版本中可获得。

参见

[UIAccessibilityIsGuidedAccessEnabled](#)

1.6 UIAccessibilityZoomFocusChanged

告知系统应用的焦点已经变换到一个新位置。

声明

SWIFT

```
func UIAccessibilityZoomFocusChanged(_ type: UIAccessibilityZoomType, _  
frame: CGRect, _ view: UIView)
```

OBJECTIVE-C

```
void UIAccessibilityZoomFocusChanged ( UIAccessibilityZoomType type, CGRect  
frame, UIView *view );
```

参数

type : 一个介绍性常量，标识缩放类型。

frame : 在屏幕坐标中，当前缩放的框架。

view : 包含缩放框架的视图。

可获得性

在 iOS5.0 及更高的版本中可获得。

1.7 UIAccessibilityIsBoldTextEnabled

返回一个布尔值，标识是否启用了粗体文本。

声明

SWIFT

```
func UIAccessibilityIsBoldTextEnabled() -> Bool
```

OBJECTIVE-C

```
BOOL UIAccessibilityIsBoldTextEnabled ( void );
```

返回值

如果用户在设置中启用了粗体文本，返回 YES；否则，返回 NO。

可获得性

在 iOS8.0 及更高的版本中可获得。

1.8

UIAccessibilityIsClosedCaptioningEnabled

返回一个布尔值，标识是否启用了隐藏式字幕。

声明

SWIFT

```
func UIAccessibilityIsClosedCaptioningEnabled() -> Bool
```

OBJECTIVE-C

```
BOOL UIAccessibilityIsClosedCaptioningEnabled ( void );
```

返回值

如果用户在设置中启用了隐藏式字幕，返回 YES；否则，返回 NO。

可获得性

在 iOS5.0 及更高的版本中可获得。

1.9

UIAccessibilityDarkerSystemColorsEnabled

返回一个布尔值，标识是否启用了加深颜色。

声明

SWIFT

```
func UIAccessibilityDarkerSystemColorsEnabled() -> Bool
```

OBJECTIVE-C

```
BOOL UIAccessibilityDarkerSystemColorsEnabled ( void );
```

返回值

如果用户在设置中启用了加深颜色，返回 YES；否则，返回 NO。

可获得性

在 iOS8.0 及更高的版本中可获得。

1.10 UIAccessibilityIsGrayscaleEnabled

返回一个布尔值，标识是否启用了灰度。

声明

SWIFT

```
func UIAccessibilityIsGrayscaleEnabled() -> Bool
```

OBJECTIVE-C

```
BOOL UIAccessibilityIsGrayscaleEnabled ( void );
```

返回值

如果用户在设置中启用了灰度，返回 YES；否则，返回 NO。

可获得性

在 iOS8.0 及更高的版本中可获得。

1.11 UIAccessibilityIsGuidedAccessEnabled

返回一个布尔值，标识是否启用了引导式访问。

声明

SWIFT

```
func UIAccessibilityIsGuidedAccessEnabled() -> Bool
```

OBJECTIVE-C

```
BOOL UIAccessibilityIsGuidedAccessEnabled ( void );
```

返回值

如果用户在设置中启用了引导式访问，返回 YES；否则，返回 NO。

可获得性

在 iOS6.0 及更高的版本中可获得。

1.12 UIAccessibilityIsInvertColorsEnabled

返回一个布尔值，标识是否启用了反转颜色。

声明

SWIFT

```
func UIAccessibilityIsInvertColorsEnabled() -> Bool
```

OBJECTIVE-C

```
BOOL UIAccessibilityIsInvertColorsEnabled ( void );
```

返回值

如果用户在设置中启用了反转颜色，返回 YES；否则，返回 NO。

可获得性

在 iOS6.0 及更高的版本中可获得。

1.13 UIAccessibilityIsMonoAudioEnabled

返回一个布尔值，标识系统音频是否设置为单声道音频。

声明

SWIFT

```
func UIAccessibilityIsMonoAudioEnabled() -> Bool
```

OBJECTIVE-C

```
BOOL UIAccessibilityIsMonoAudioEnabled ( void );
```

返回值

如果当前已启用了单声道音频，返回 YES；否则，返回 NO。

可获得性

在 iOS5.0 及更高的版本中可获得。

1.14 UIAccessibilityIsReduceMotionEnabled

返回一个布尔值，标识是否启用了减弱动态效果。

声明

SWIFT

```
func UIAccessibilityIsReduceMotionEnabled() -> Bool
```

OBJECTIVE-C

```
BOOL UIAccessibilityIsReduceMotionEnabled( void );
```

返回值

如果用户在设置中启用了减弱动态效果，返回 YES；否则，返回 NO。

可获得性

在 iOS8.0 及更高的版本中可获得。

1.15

UIAccessibilityIsReduceTransparencyEnabled

返回一个布尔值，标识是否启用了降低透明度。

声明

SWIFT

```
func UIAccessibilityIsReduceTransparencyEnabled() -> Bool
```

OBJECTIVE-C

```
BOOL UIAccessibilityIsReduceTransparencyEnabled ( void );
```

返回值

如果用户在设置中启用了降低透明度，返回 YES；否则，返回 NO。

可获得性

在 iOS8.0 及更高的版本中可获得。

1.16 UIAccessibilityIsSpeakScreenEnabled

返回一个布尔值，标识是否启用了朗读屏幕。

声明

SWIFT

```
func UIAccessibilityIsSpeakScreenEnabled() -> Bool
```

OBJECTIVE-C

```
BOOL UIAccessibilityIsSpeakScreenEnabled ( void );
```

返回值

如果用户在设置中启用了朗读屏幕，返回 YES；否则，返回 NO。

可获得性

在 iOS8.0 及更高的版本中可获得。

1.17 UIAccessibilityIsSpeakSelectionEnabled

返回一个布尔值，标识是否启用了朗读所选项。

声明

SWIFT

```
func UIAccessibilityIsSpeakSelectionEnabled() -> Bool
```


OBJECTIVE-C

```
BOOL UIAccessibilityIsSpeakSelectionEnabled ( void );
```

返回值

如果用户在设置中启用了朗读所选项，返回 YES；否则，返回 NO；

可获得性

在 iOS8.0 及更高的版本中可获得。

1.18 UIAccessibilityIsSwitchControlRunning

返回一个布尔值，标识是否启用了切换控制。

声明

SWIFT

```
func UIAccessibilityIsSwitchControlRunning() -> Bool
```

OBJECTIVE-C

```
BOOL UIAccessibilityIsSwitchControlRunning ( void );
```

返回值

如果用户在设置中启用了切换控制，返回 YES；否则，返回 NO。

可获得性

在 iOS8.0 及更高的版本中可获得。

1.19 UIAccessibilityIsVoiceOverRunning

返回一个布尔值，标识 VoiceOver 是否正在运行。

声明

SWIFT

```
func UIAccessibilityIsVoiceOverRunning() -> Bool
```

OBJECTIVE-C

```
BOOL UIAccessibilityIsVoiceOverRunning ( void );
```

返回值

如果 VoiceOver 正在运行，返回 YES；否则，返回 NO。

简介

可以使用该函数为 VoiceOver 自定义应用用户界面。例如，开发者可能想要为 VoiceOver 用户将快速消失的 UI 元素一直保持在屏幕上。注意，开发者也可以监听 UIAccessibilityVoiceOverStatusChanged 通知，来获得 VoiceOver 启动和停止的时间。

可获得性

在 iOS4.0 及更高的版本中可获得。