

# Android. Accessibility. Interface

中国信息无障碍产品联盟&信息无障碍研究会 译制

20161019

## 目录

翻译声明.....	1
1.AccessibilityEventSource .....	2
1.1 开发指南.....	2
1.2 概要.....	2
1.3 公有方法.....	3
1.3.1 sendAccessibilityEvent .....	3
1.3.2 sendAccessibilityEventUnchecked .....	3
2. AccessibilityManager.AccessibilityStateChangeListener .....	5
2.1 概要.....	5
2.2 公有方法.....	5
2.2.1 onAccessibilityStateChanged .....	5
3. AccessibilityManager.TouchExplorationStateChangeListener .....	7
3.1 概要.....	7
3.2 公有方法.....	7
3.2.1 onTouchExplorationStateChanged .....	7

# 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：**李鸿利、刘彪、沈广荣

本文档翻译自谷歌官方文档《[Android Accessibility Interface](#)》。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 Google 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，如您对内容上有任何的意见或疑问，请发送邮件至 [liuhui@siaa.org.cn](mailto: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。

# 1. AccessibilityEventSource

添加于 [API 级别 4](#)。

public interface AccessibilityEventSource

android.view.accessibility.AccessibilityEventSource

已知间接子类

[AbsListView](#), [AbsSeekBar](#), [AbsSpinner](#), [AbsoluteLayout](#), [ActionMenuView](#), [AdapterView<T extends Adapter>](#), [AdapterViewAnimator](#), [AdapterViewFlipper](#), 和 117 个其他子类。

[AccessibilityEvents](#) 的类源实现该接口。

## 1.1 开发指南

更多关于让应用无障碍的信息，详见 [Android 无障碍开发指南](#)。

## 1.2 概要

公有方法	
abstract void	<a href="#">sendAccessibilityEvent(int eventType)</a> 处理发送给定事件类型 <a href="#">AccessibilityEvent</a> 的请求。
abstract void	<a href="#">sendAccessibilityEventUnchecked(AccessibilityEvent event)</a> 处理发送 <a href="#">AccessibilityEvent</a> 的请求。

## 1.3 公有方法

### 1.3.1 sendAccessibilityEvent

添加于 [API 级别 4](#)。

void sendAccessibilityEvent (int eventType)

处理发送给定事件类型 [AccessibilityEvent](#) 的请求。该方法必须首先调用 [AccessibilityManager.isEnabled\(\)](#)，检测是否开启了无障碍，通过调用 [AccessibilityEvent.obtain\(int\)](#) 从事件池中获取一个 [AccessibilityEvent](#)，然后填充事件并通过调用 [AccessibilityManager.sendAccessibilityEvent\(AccessibilityEvent\)](#) 将其发送出去。

参数：

eventType	int: 事件类型。
-----------	------------

参见：

[AccessibilityEvent](#)

[AccessibilityManager](#)

### 1.3.2 sendAccessibilityEventUnchecked

添加于 [API 级别 4](#)。

void sendAccessibilityEventUnchecked (AccessibilityEvent event)

处理发送 [AccessibilityEvent](#) 的请求。该方法在发送事件前，不保证已检查无障碍是否开启。通过调用 [AccessibilityManager.isEnabled\(\)](#) 检测是否开启了无障碍是调用者的责任。

参数：

Event	AccessibilityEvent: 事件。
-------	-------------------------

参见：

AccessibilityEvent

AccessibilityManager

## 2.

# AccessibilityManager.AccessibilityStateChangeListener

添加于 [API 级别 4](#)。

```
public static interface AccessibilityManager.AccessibilityStateChangeListener
```

```
android.view.accessibility.AccessibilityManager.AccessibilityStateChangeListener
```

系统无障碍状态的监听器。为了监听设备上无障碍状态的改变，调用 [addAccessibilityStateChangeListener\(AccessibilityManager.AccessibilityStateChangeListener\)](#) 实现该接口并在系统中将其注册。

## 2.1 概要

公有方法	
abstract void	<a href="#">onAccessibilityStateChanged</a> (boolean enabled) 当无障碍开启状态改变时调用。

## 2.2 公有方法

### 2.2.1 onAccessibilityStateChanged

添加于 [API 级别 4](#)。

```
void onAccessibilityStateChanged (boolean enabled)
```

当无障碍开启状态改变时调用。

**参数：**

enabled	boolean: 是否已开启无障碍。
---------	--------------------



## 3. AccessibilityManager.TouchExplorationStateChangeListener

添加于 [API 级别 19](#)。

```
public static interface AccessibilityManager.TouchExplorationStateChangeListener
android.view.accessibility.AccessibilityManager.TouchExplorationStateChangeListener
```

系统触摸浏览状态的监听器。为了监听设备上触摸浏览状态的改变，调用 [addTouchExplorationStateChangeListener\(AccessibilityManager.TouchExplorationStateChangeListener\)](#) 实现该接口并在系统中将其注册。

### 3.1 概要

公有方法	
abstract void	<a href="#">onTouchExplorationStateChanged(boolean enabled)</a> 当触摸浏览开启状态改变时调用。

### 3.2 公有方法

#### 3.2.1 onTouchExplorationStateChanged

添加于 [API 级别 19](#)。

```
void onTouchExplorationStateChanged (boolean enabled)
```

当触摸浏览开启状态改变时调用。

参数:

enabled	Boolean: 是否开启了触摸浏览。
---------	---------------------