

HarmonyOS 无障碍开发指南

ArkUI 无障碍属性开发指南
HarmonyOS API Version 10+

华为 HarmonyOS 开发者文档
来源：developer.huawei.com
生成日期：2026-07-26

概述

组件可以设置相应的无障碍属性和事件，来更好地使用无障碍能力。通过合理设置这些属性，屏幕朗读（Screen Reader）等辅助功能服务能够准确地向用户描述界面内容，提升应用的无障碍体验。

【说明】
从 API Version 10 开始支持。后续版本如有新增内容，则采用上角标单独标记该内容的起始版本。

【说明】
卡片能力：从 API version 12 开始，该接口支持在 ArkTS 卡片中使用。
原子化服务 API：从 API version 11 开始，该接口支持在原子化服务中使用。
系统能力：SystemCapability.ArkUI.ArkUI.Full

无障碍属性总览

属性名	作用	关键说明
accessibilityGroup	无障碍组	将多个子组件合并为一个焦点单元
accessibilityText	无障碍文本	读屏器朗读的核心文本
accessibilityDescription	无障碍说明	告知用户如何操作或后果
accessibilityLevel	无障碍重要性	控制节点是否参与无障碍遍历
accessibilityVirtualNode	虚拟子节点	为自绘制组件提供无障碍节点
accessibilityChecked	多选选中状态	用于多选场景的选中状态维护

属性名	作用	关键说明
accessibilitySelected	单选选中状态	用于单选场景的选中状态维护
accessibilityTextHint	文本提示	供无障碍辅助应用查询的提示

accessibilityGroup — 无障碍组

accessibilityGroup(value: boolean)

设置是否启用无障碍分组。启用后该组件及其所有子组件将作为一个可以选中的组件，无障碍服务将不再关注其子组件内容。

若组件启用无障碍分组，当组件不包含通用文本属性，同时未设置无障碍文本时，将默认拼接其子组件的通用文本属性作为组件的合并文本（深度优先），此时合并文本不使用子组件的无障碍文本。

参数名	类型	必填	说明
value	boolean	是	无障碍分组，true=启用；默认值：false

accessibilityGroup14+(isGroup, accessibilityOptions)

设置是否启用无障碍分组，支持优先拼接无障碍文本进行朗读。通过 `accessibilityPreferred` 启用优先拼接后，将优先拼接子组件的无障碍文本。

参数名	类型	必填	说明
isGroup	boolean	是	无障碍分组，true=启用；默认值：false
accessibilityOptions	AccessibilityOptions	是	包含 <code>accessibilityPreferred</code> 字段

AccessibilityOptions14+ 对象说明

`accessibilityPreferred: boolean` — 若设置为 `true`，则深度遍历每个子节点时优先选择子节点的无障碍文本 `accessibilityText`。若无障碍文本为空则选择本身 `Text` 文本。默认值：`false`。

```
// 示例：将一组文本标记为一个无障碍组
Column() {
  Text("文本1").fontSize(50).fontWeight(FontWeight.Bold)
  Text("文本2").fontSize(50).fontWeight(FontWeight.Bold)
}
.width('100%')
.accessibilityGroup(true)
.accessibilityLevel("yes")
.accessibilityText("分组")
```

```
// 示例：优先使用子组件的无障碍文本进行朗读
Column({ space: 10 }) {
  Text('123456')
    .accessibilityText("有accessibility有text优先读accessibility")
    .accessibilityLevel("yes")
  Button().accessibilityLevel("yes")
    .accessibilityText("accessibility无text 读accessibility")
  Button("无accessibility有text 读text").accessibilityLevel("yes")
}
.accessibilityGroup(true, { accessibilityPreferred: true })
```

accessibilityText — 无障碍文本

accessibilityText(value: string)

设置无障碍文本。当组件不包含文本属性时，屏幕朗读选中此组件时不播报，使用者无法清楚地知道当前选中了什么组件。为了解决此场景，开发人员可为不包含文字信息的组件设置无障碍文本。

说明：若组件既拥有文本属性，又拥有无障碍文本属性，则仅播报无障碍文本内容。

参数名	类型	必填	说明
value	string	是	无障碍文本内容；默认值：""

accessibilityText12+(text: Resource)

设置无障碍文本，支持通过 Resource 引用资源文件。

```
// 示例：为图标按钮设置无障碍文本
Image($r('app.media.search_icon'))
  .width(40).height(40)
  .accessibilityText('搜索按钮')
  .onClick(() => { /* 搜索逻辑 */ })
```

```
// 示例：为图片设置无障碍文本
Image($r('app.media.avatar'))
  .accessibilityText('用户头像')
```

accessibilityDescription — 无障碍说明

accessibilityDescription(value: string)

用于为用户进一步说明当前组件，帮助用户理解将要执行的操作及其后果。若组件既拥有文本属性又拥有无障碍说明属性，则先播报文本属性，再播报无障碍说明。

```
// 示例：为滑块设置无障碍说明
Slider({ value: 50, min: 0, max: 100 })
  .accessibilityText('音量滑块')
  .accessibilityDescription('上下滑动调节音量，当前值为 50%')
  .onChange((value: number) => { /* 音量调节 */ })
```

accessibilityLevel — 无障碍重要性

控制某个组件是否可被无障碍辅助服务所识别。

值	说明
"auto"	由无障碍分组服务和 ArkUI 综合判断（默认值）
"yes"	当前组件可被无障碍辅助服务识别
"no"	当前组件不可被无障碍辅助服务识别
"no-hide-descendants"	当前组件及其所有子组件均不可被识别

```
// 示例：隐藏装饰性分隔线
Divider().accessibilityLevel('no')

// 示例：强制文本可聚焦
Text('系统提示：操作成功').accessibilityLevel('yes')
```

accessibilityVirtualNode11+ — 虚拟子节点

对自绘制组件传入一个自定义的 `CustomBuilder`，该 `Builder` 中的组件在后端仅做布局不做显示，辅助应用获取无障碍节点信息时会返回 `CustomBuilder` 中的节点信息。

accessibilityChecked13+ / accessibilitySelected13+ — 选中状态

`accessibilityChecked` 用于多选场景。设置 `true` 或 `false` 后会默认修改组件的 `checkable` 属性为 `true`。

`accessibilitySelected` 用于单选场景。

说明：如果同时使用两个接口，需要将不需要的接口参数设为 `undefined`。

accessibilityTextHint12+ — 文本提示

设置组件的文本提示信息，供无障碍辅助应用查询。

最佳实践

- 优先使用标准组件：Button、TextInput、Checkbox 等标准组件已自动设置角色和状态。
- 关键场景必设：无文本图标按钮、自定义控件必须设置 `accessibilityText`。
- 合理分组：使用 `accessibilityGroup(true)` 将逻辑相关的元素组合，提升导航效率。
- 避免信息冗余：父组件已设组时，子组件应使用 `accessibilityLevel('no')` 隐藏。
- 状态同步：`accessibilityState` 必须与组件的实际交互状态保持实时同步。

完整示例

```
@Entry
@Component
struct Index {
  @Builder customAccessibilityNode() {
    Column() {
      Text('virtual node')
    }.width(10).height(10)
  }

  build() {
    Row() {
      Column() {
        Text('文本1').fontSize(50).fontWeight(FontWeight.Bold)
        Text('文本2').fontSize(50).fontWeight(FontWeight.Bold)
      }
      .width('100%')
      .accessibilityGroup(true)
      .accessibilityLevel('yes')
      .accessibilityText('分组')
      .accessibilityDescription('Column可以被选中')
      .accessibilityVirtualNode(this.customAccessibilityNode)
      .accessibilityChecked(true)
    }.height('100%')
  }
}
```