HarmonyOS

屏幕朗读应用适配指导

Accessibility Kit（无障碍服务）

屏幕朗读应用适配指导

华为 HarmonyOS 开发者文档

来源：developer.huawei.com

生成日期：2026-07-26

概述

屏幕朗读软件（Screen Reader）主要帮助视障人士使用移动智能设备，通过语音输出，获取屏幕或界面中的信息。

开发原则

- 确保视障用户可以通过手势快速、符合使用逻辑顺序地导航至页面内所有有效 UI 对象。
- 确保用户在当前获焦的 UI 对象下接收到适当的语音朗读反馈，朗读内容应简洁清晰地告知用户当前所在 UI 对象内容、功能以及可执行的操作。

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。

1. 标注屏幕朗读内容的场景

显示文本（text）：用户界面上呈现的信息。
无障碍文本（accessibilityText）：无障碍专有的朗读信息，不在界面上显示。
优先级：accessibilityText > text。当无障碍文本不为空时，朗读无障碍文本，否则朗读显示文本。

建议

- 文本类控件尽量使用显示文本，使视障和健全用户获取相同信息。
- 如果显示文本外还有颜色等额外信息，可使用 accessibilityText 为视障用户提供更多朗读信息。
- 非文本类控件，可使用 accessibilityText 提供朗读信息。

```
@Component
export struct Rule_2_1_1 {
  build() {
    NavDestination() {
      Column() {
        Button('Button')
          .accessibilityText('Accessibility text')
          .fontSize(20)
      }.width('100%').height('100%')
    }.title('Rule 2.1.1')
  }
}
```

朗读效果：聚焦 button 时，朗读为「按钮，Accessibility text」。

2. 禁用屏幕朗读焦点的场景

适用对象：装饰性控件（分隔符、占位符、美化图标等），不传达有效信息或提供交互功能。

方法

- accessibilityGroup(true)：用于多个组件的组合，组合内的组件默认没有焦点。
- accessibilityLevel('no')：用于组件设置不可聚焦，不被无障碍感知。
- accessibilityLevel('no-hide-descendants')：用于容器，容器及其所有子组件均不可聚焦。

```
@Component
export struct Rule_2_1_3 {
  @State message: string = 'Broadcast';
  @State message1: string = 'No broadcast';
  build() {
    NavDestination() {
      Column() {
        Row() {
          Text(this.message).fontSize(40)
            .fontWeight(FontWeight.Bold).fontColor(Color.Blue)
        }.width('100%').height('50%')
        Row() {
          Text(this.message1).fontSize(40)
            .fontWeight(FontWeight.Bold).fontColor(Color.Grey)
            .accessibilityLevel('no') // 此组件不可聚焦
        }.width('100%').height('50%')
      }.height('100%')
    }.title('Rule 2.1.3')
  }
}
```

说明：当 ScreenReader 打开时，message 可被聚焦，message1 不被聚焦。

3. 多维嵌套场景

问题：在嵌套组中，应避免两个可获焦对象的功能或朗读内容重复。
解决方案：将完整的信息描述统一标注在父控件上。

```
@Component
export struct Rule_2_1_4 {
  build() {
    NavDestination() {
      Column() {
        Text('Incorrect behavior:')
        // 播报 'Time Group 12:05 Beijing' + '12:05' + 'Beijing', 重复
        Row() {
          Text('12:05').fontSize(32).fontColor(Color.Red)
          Text('Beijing').fontSize(20).fontColor(Color.Green)
        }.accessibilityText('Time Group')

        Text('Correct behavior:')
        // 只朗读 '07:05 Moscow', 不重复
        Row() {
          Text('07:05').fontSize(32).fontColor(Color.Red)
          Text('Moscow').fontSize(20).fontColor(Color.Green)
        }.accessibilityGroup(true) // 父控件组合, 子控件不可聚焦
      }.alignItems(HorizontalAlign.Start).padding(10)
    }.title('Rule 2.1.4')
  }
}
```

4. 组合场景

原则：对于表示同一个对象信息的多个组件，需要进行组合标注，对外只暴露一个无障碍焦点。
方法：将多个控件设置为一个组，通过对组设置朗读标签，达到整组播报的效果，组内的子控件设置不可获取焦点。

```
@Component
export struct Rule_2_1_5 {
  build() {
    NavDestination() {
      Column() {
        // 错误：每个子组件独立获取焦点
        Row() {
          Text('23 Dec 2023').fontSize(32)
          Column().backgroundColor(Color.Blue).width(50).height(50)
            .accessibilityText('Snow')
          Text('-1').fontSize(20)
        }.height(50)

        // 正确：父组件使用 accessibilityGroup(true)
        Row() {
          Text('24 Dec 2023').fontSize(32)
          Column().backgroundColor(Color.Yellow).width(50).height(50)
            .accessibilityText('Sunny')
          Text('-7').fontSize(20)
        }.height(50).accessibilityGroup(true)
      }.alignItems(HorizontalAlign.Start).padding(10)
    }.title('Rule 2.1.5')
  }
}
```

5. 按钮标注场景

要求：对于非文本类按钮（如图标按钮），必须提供标注信息。标注文本不要包含控件类型（如「按钮」）或操作指引（如「单指双击即可打开」），这些由系统自动追加。

```
@Component
export struct Rule_2_1_6 {
  @State isPlaying: boolean = false
  build() {
    NavDestination() {
      Column() {
        Row() {
          Image(this.isPlaying ? $r('app.media.pause') : $r('app.media.play'))
            .width(50).height(50)
            .accessibilityText(this.isPlaying ? 'Pause' : 'Play')
            .onClick(() => { this.isPlaying = !this.isPlaying })
          Text('Good_morning.mp3').margin({ left: 10 })
        }
      }.width('100%').height('100%').backgroundColor(Color.White)
    }.title('Rule 2.1.6')
  }
}
```

6. 插画/视频/动画的播报场景

原则：插画和对应的功能介绍应该组合在一起，当焦点落到插画或者包含插画的复合控件时，需要朗读出对应的功能描述。借助 accessibilityGroup(true) 属性实现。

```
@Component
export struct Rule_2_1_7 {
  private description: string = 'gesture swipe left then up'
  build() {
    NavDestination() {
      Column() {
        Column() {
          Image($r('app.media.gesture_swipe_left_then_up')).width(220).height(220)
          Text(this.description).fontSize(22).fontColor(Color.Red)
        }.accessibilityGroup(true) // 将图像和文本合并为一个无障碍对象
      }.width('100%').height('100%').backgroundColor(Color.White)
    }.title('Rule 2.1.7')
  }
}
```

7. 内容动态变化场景

场景：界面上重要内容动态变化后，需要实时发送变化后的朗读内容。
方法：调用无障碍提供的主动朗读接口 accessibility.sendAccessibilityEvent。

```
import accessibility from '@ohos.accessibility';

let eventInfo: accessibility.EventInfo = ({
  type: 'announceForAccessibility',
  bundleName: 'com.example.pagesrouter',
  triggerAction: 'common',
  textAnnouncedForAccessibility: 'test123 text'
});

accessibility.sendAccessibilityEvent(eventInfo).then(() => {
  console.info('Succeed in send event');
});
```

EventInfo 说明

属性	类型	说明	示例
type	EventType	主动播报事件类型	announceForAccessibility
bundleName	string	目标应用名	当前应用包名
triggerAction	Action	触发事件的Action	click
textAnnouncedForAccessibility	string	主动播报的内容	test123 text

8. 控件状态变化场景

要求：在控件状态（如播放/暂停按钮）切换时，需要实时更新对应的标注信息。

9. 操作错误场景

要求：对于错误或警告信息（如网络连接错误），不能仅靠颜色区分，需要实时告知用户错误提示和改进方法。

```
@Component
export struct Rule_2_1_14 {
  build() {
    NavDestination() {
      Column() {
        Text('Connection state').fontSize(30)
        Row() {
          Radio({ value: 'Radio1', group: 'radioGroup' }).checked(true)
          Text('Connection interrupted').fontColor(Color.Red)
        }.width('80%').accessibilityGroup(true)
      }.width('100%').height('100%').backgroundColor(Color.White)
    }.title('Rule 2.1.14')
  }
}
```

10. 多语种场景

要求：对标注字符串进行多语种翻译，支持的语种与应用界面一致。若拼接字符串，需考虑语种特性（如阿拉伯语的从右到左）。

11. 控件位置调整场景（拖拽）

要求：移动过程中需要实时播报即将移动到的位置，放置后播报已放置的位置信息。应用可调用主动播报接口来实现。

12. 重新设置新焦点位置的场景

场景：当前焦点所在的控件消失或隐藏后，需要重新设置新的焦点位置。
说明：新焦点应在原控件位置的下一个控件上，不应跳变到前面的控件。
方法：使用主动聚焦接口 accessibility.sendAccessibilityEvent，设置 type 为 requestFocusForAccessibility。

```
// 组件定义
build() {
  Column() {
    Button('待聚焦组件').id('abc345')
  }
}

// 发送聚焦事件
import accessibility from '@ohos.accessibility';
let eventInfo: accessibility.EventInfo = ({
  type: 'requestFocusForAccessibility',
  bundleName: 'com.example.pagesrouter',
  triggerAction: 'common',
  customId: 'abc345'
});
accessibility.sendAccessibilityEvent(eventInfo);
```

EventInfo 说明

属性	类型	说明	示例
type	EventType	主动聚焦事件类型	requestFocusForAccessibility
bundleName	string	目标应用名	当前应用包名
triggerAction	Action	触发事件的Action	click
customId	string	组件id	abc345

开发要点总结

- 文本类控件尽量使用显示文本，使视障和健全用户获取相同信息。
- 装饰性控件使用 accessibilityLevel('no') 禁用焦点。
- 多维嵌套场景下，将完整信息标注在父控件上，子控件使用 accessibilityGroup(true) 隐藏。
- 同一对象信息的多个组件需进行组合标注，对外只暴露一个无障碍焦点。
- 非文本类按钮必须设置 accessibilityText，内容不要包含控件类型和操作指引。
- 插画和功能描述需组合在一起朗读。
- 内容动态变化后需调用 sendAccessibilityEvent 主动播报。
- 控件状态切换时需实时更新标注信息。
- 错误/警告信息不能仅靠颜色区分，需有文本提示。
- 标注字符串需进行多语种翻译。

- 拖拽操作需实时播报位置变化。
- 控件消失后，新焦点应在原位置的下一个控件上。