

HarmonyOS

适老化适配指南

ArkUI 适老化适配指南
支持适老化 — 无障碍与适老化

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

更新时间：2026-03-09 10:50

概述

系统字体被放大后，应用应确保整体布局不出现错乱，组件不出现重叠。可以根据业务需要限制跟随的字体最大档位、改变布局来更好地适配更大字体等。本文旨在指导应用如何跟随系统字体大小和跟随到的最大倍数。

应用适配规则

- 在系统使用 1.75 倍及以上的大字体时，页面布局不得错乱，组件不得叠加，文字不得截断。
- 图标及图片不会随着字体的变大而变化。
- 页面中不重要的内容字体，可采用不跟随系统字体变化或限制字体最大尺寸的方法进行布局。
- 若应用跟随系统字体增大后导致页面内容位置挤压或截断等问题，可采取将X轴扩展至Y轴的措施，例如将左右布局调整为上下布局。
- 系统组件已针对适老化大字体进行了单独适配，尽可能在开发过程中使用系统组件。

应用适配适老化大字体

1. 开启路径

在「设置 → 辅助功能 → 关怀模式 → 放大显示」中开启。

各档位对应参数

档位	字体大小	字体粗细
----	------	------

档位	字体大小	字体粗细
标准	1倍	1倍
大1档	1.15倍	1倍
大2档	1.3倍	1.1倍
大3档	1.45倍	1.1倍
大4档	1.75倍	1.25倍
大5档	2.0倍	1.25倍
大6档	3.2倍	1.25倍

2. 适配方法

当前默认应用不跟随系统字体的变化。如需跟随系统字体变化，并设置最大跟随变化的倍数，请按以下步骤操作：

步骤一：配置 app.json5

通过配置 'configuration': "\$profile:configuration"，指向 base/profile/configuration.json 文件。

```
{
  "app": {
    "bundleName": "com.example.myapplication",
    "vendor": "example",
    "versionCode": 1000000,
    "versionName": "1.0.0",
    "icon": "$media:app_icon",
    "label": "$string:app_name",
    "configuration": "$profile:configuration"
  }
}
```

步骤二：新增 configuration.json

在 base 文件目录下新增 profile 文件夹，并在此目录下新增 configuration.json 文件。配置 'fontSizeScale': 'followSystem' 表示该应用的字体大小将根据系统设置进行缩放，'fontSizeMaxScale': '1.3' 表示应用字体大小随系统变化的最大缩放比例为 1.3 倍。

```
{
  "configuration": {
    "fontSizeScale": "followSystem",
    "fontSizeMaxScale": "1.3"
  }
}
```

步骤三：设置 Text 组件的 maxFontSize

若应用需适应系统字体大小的变化至 1.75 倍，但部分组件可调整至 2 倍。

首先配置 fontSizeMaxScale 为 1.75，然后为 Text 添加 maxFontSize 属性，传递参数为 2，表示该 Text 组件跟随系统字体大小变化的最大倍数为 2 倍。

```
{
  "configuration": {
    "fontSizeScale": "followSystem",
    "fontSizeMaxScale": "1.75"
  }
}
```

```
Text('hello world!')
  .fontSize($r('sys.float.Body_M'))
  .maxFontScale(2)
  .fontColor($r('sys.color.font_secondary'))
```

当 Text 组件配置了 maxFontScale 属性时，将采用组件设置的最大放大倍数，而非系统默认的最大放大倍数。若应用不需要跟随系统字体大小变化，无需配置。

3. 获取字体大小和粗细

生命周期回调方法 `onConfigurationUpdated` 的 `config` 参数可接收字体大小（`fontSizeScale`）与字体粗细（`fontWeightScale`）。注册系统环境变化的监听后，在系统环境变化时可触发回调。

应用冷启动查询系统字体大小档位：

```
let context = this.getUIContext().getHostContext() as common.UIAbilityContext;
let fontSizeScale: number = context.config?.fontSizeScale ?? 1;
```

适配适老化的系统组件及触发方式

适老化提供了一种通过鼠标或手指长按的方法来放大所选区域或组件。即如果系统字体大小大于 1 倍，当用户使用鼠标或手指长按装配了适老化方法的组件，需要从所选区域的组件中提取数据，并放入另一个弹窗组件中展示。

适老化规则

由于在系统字体大于 1 倍时，组件并没有默认放大，需要通过配置 `configuration` 标签，实现组件放大的适老化功能。

如何开启适老化

进入手机设置，点击辅助功能，开启关怀模式。

适老化操作

在已经支持适老化能力的组件上长按组件，能够触发弹窗，当用户释放时，适老化操作结束。当设置系统字体大于 1 倍时，组件自动放大；当系统字体恢复至 1 倍时组件恢复正常状态。

弹窗限制

当用户将系统字体设置为 2 倍以上时，弹窗内容包括 icon 和文字的放大倍数固定为 2 倍。

触发适老化的系统组件列表

长按系统组件触发

SideBarContainer、底部页签（tabBar）、Navigation、NavDestination、Tabs

设置系统字体默认放大

PickerDialog、Button、Menu、Stepper、BindSheet、TextInput、TextArea、Search、SelectionMenu、Chip、Dialog、Slider、Progress、Badge

SideBarContainer 示例

```
@Entry
@Component
struct SideBarContainerExample {
  @State currentFontSizeScale: number = 1
  normalIcon: Resource = $r('app.media.icon')
  selectedIcon: Resource = $r('app.media.icon')
  @State arr: number[] = [1, 2, 3]
  @State current: number = 1
  @State title: string = 'Index01';
  build() {
    SideBarContainer(SideBarContainerType.Embed) {
      Column() {
        ForEach(this.arr, (item: number) => {
          Column({ space: 5 }) {
            Image(this.current === item ? this.selectedIcon : this.normalIcon)
              .width(64).height(64)
            Text('0' + item).fontSize(25)
              .fontColor(this.current === item ? '#0A59F7' : '#999')
          }
          .onClick(() => {
            this.current = item;
            this.title = 'Index0' + item;
          })
        })
      }.width('100%')
      .justifyContent(FlexAlign.SpaceEvenly)
    }
    .controlButton({
      icons: {
        hidden: $r('sys.media.ohos_ic_public_drawer_open_filled'),
        shown: $r('sys.media.ohos_ic_public_drawer_close')
      }
    })
    .sideBarWidth(150)
    .minSideBarWidth(50).maxSideBarWidth(300)
  }
}
```

TextPickerDialog 示例

```
@Entry
@Component
struct TextPickerExample {
  private select: number | number[] = 0;
  private cascade: TextCascadePickerRangeContent[] = [
    { text: '辽宁省', children: [
      { text: '沈阳市', children: [{ text: '沈河区' }, { text: '和平区' }, { text: '浑南区' }] },
      { text: '大连市', children: [{ text: '中山区' }, { text: '金州区' }, { text: '长海县' }] }
    ]},
    { text: '吉林省', children: [
      { text: '长春市', children: [{ text: '南关区' }, { text: '宽城区' }, { text: '朝阳区' }] },
      { text: '四平市', children: [{ text: '铁西区' }, { text: '铁东区' }, { text: '梨树县' }] }
    ]}
  ]
}

@State v: string = '';
build() {
  Column() {
    Button('TextPickerDialog:' + this.v)
      .onClick(() => {
        TextPickerDialog.show({
          range: this.cascade,
          selected: this.select,
          onAccept: (value: TextPickerResult) => {
            this.select = value.index;
            this.v = value.value as string;
          },
          onCancel: () => {},
          onChange: (value: TextPickerResult) => {}
        })
      })
  }
}
```