

HUAWEI TECHNOLOGIES CO.,LTD.

# 华为 MATEX 显示适配 指导

---



## 目录

1	背景和范围.....	3
2	适用范围.....	3
3	规格描述.....	3
3.1	MATEX物理形态概述.....	3
3.2	静态适配需求规格描述.....	4
3.2.1	大屏静态显示规格.....	4
3.2.2	主屏静态显示规格.....	4
3.2.3	副屏静态显示规格.....	5
3.3	动态适配需求规格描述.....	5
4	设计和开发指导.....	6
4.1	大屏布局UI设计指导.....	6
4.1.1	适配建议一：X轴方向自适应.....	6
4.1.2	适配建议二：布局内容扩展.....	7
4.1.3	适配建议三：分栏布局.....	8
4.1.4	适配建议四：横竖屏布局一致.....	9
4.2	适配开发指导.....	10
4.2.1	开发约束.....	10
4.2.2	适配规格：支持热切换能力.....	10
4.3	调试.....	13
4.3.1	模拟调试.....	13

## 1 背景和范围

此文档针对华为MATEX 以及后续可能的折叠屏产品，提供应用适配的规格要求描述，以及适配指导。以确保应用在折叠屏显示各种形态下，皆可以布局正确，显示美观，并且屏幕物理形态切换时，应用尽可能做到不重启，不中断当前用户业务。

## 2 适用范围

文档适用于Android平台APP相关的设计，开发，测试人员等。

## 3 规格描述

### 3.1 MATEX 物理形态概述

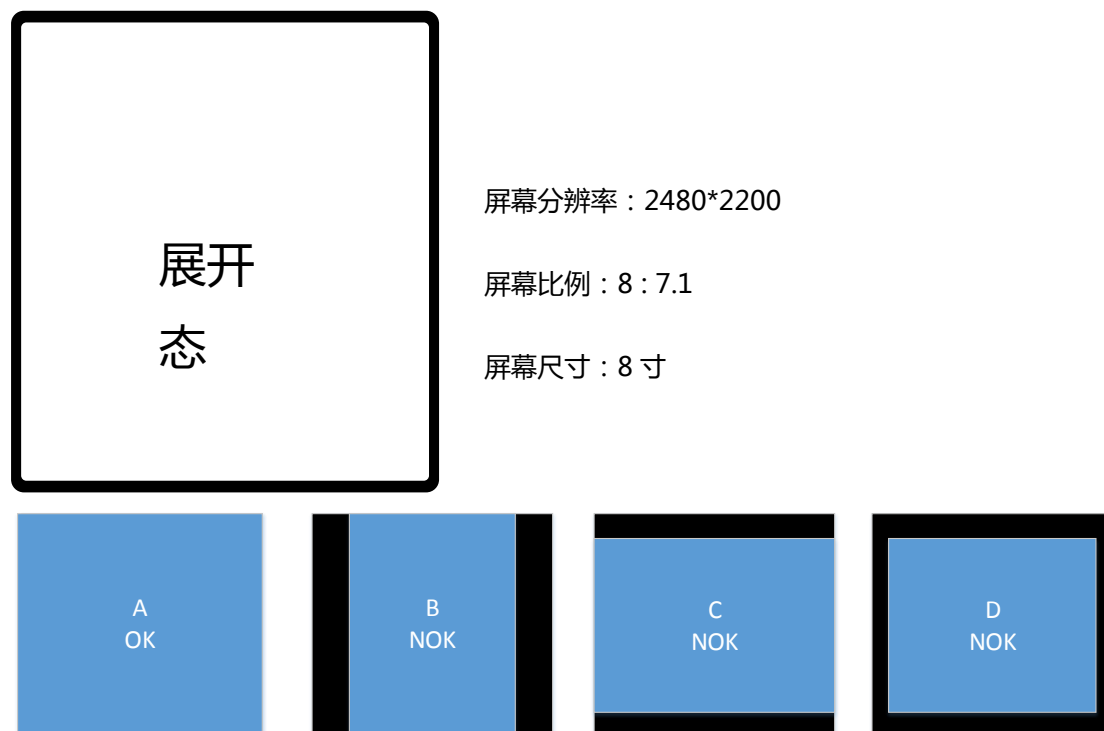


对于通用软件，可能存在如下形态，即：

- ✓ 展开态，全屏点亮工作
- ✓ 折叠主屏态：折叠态，主屏工作
- ✓ 折叠副屏态：折叠态：副屏工作

## 3.2 静态适配需求规格描述

### 3.2.1 大屏静态显示规格

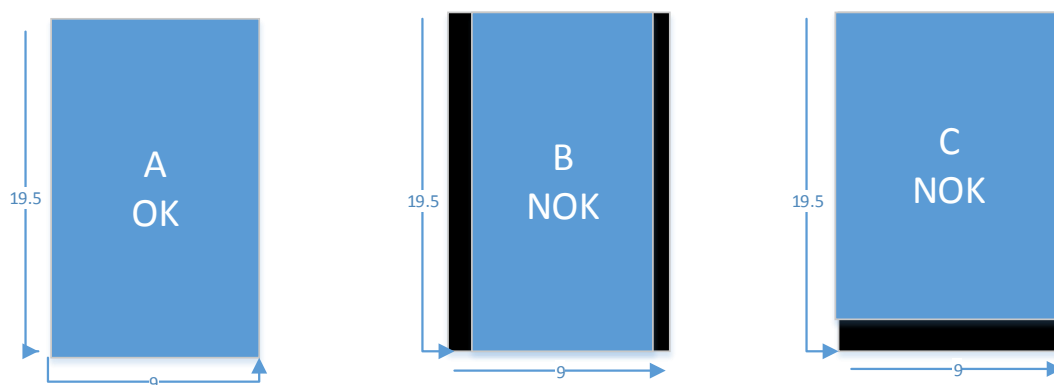


- ✓ 大屏规格定义：应用在大屏展开下，可以撑满全屏显示，且在横竖屏切换形态下，布局和操作按键都正常，不出现任意方向的黑边。
- ✓ 如上图，A图为撑满全屏，OK，BCD都是错误的适配规格。

### 3.2.2 主屏静态显示规格

屏幕比例：19.5 : 9

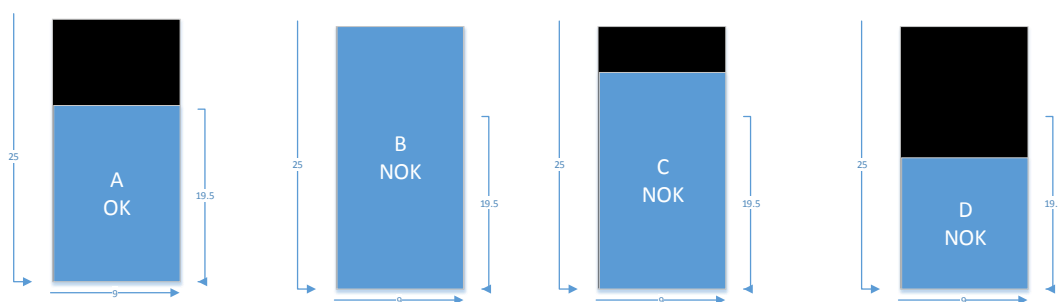
参照下图：



### 3.2.3 副屏静态显示规格

副屏规格：25 : 9

副屏下，默认不以全屏显示，以主屏比例显示，即在副屏下也显示19.5:9居下显示。应用只需要做好19.5:9的适配即可，副屏以19.5:9 在屏幕下方显示由系统统一控制。

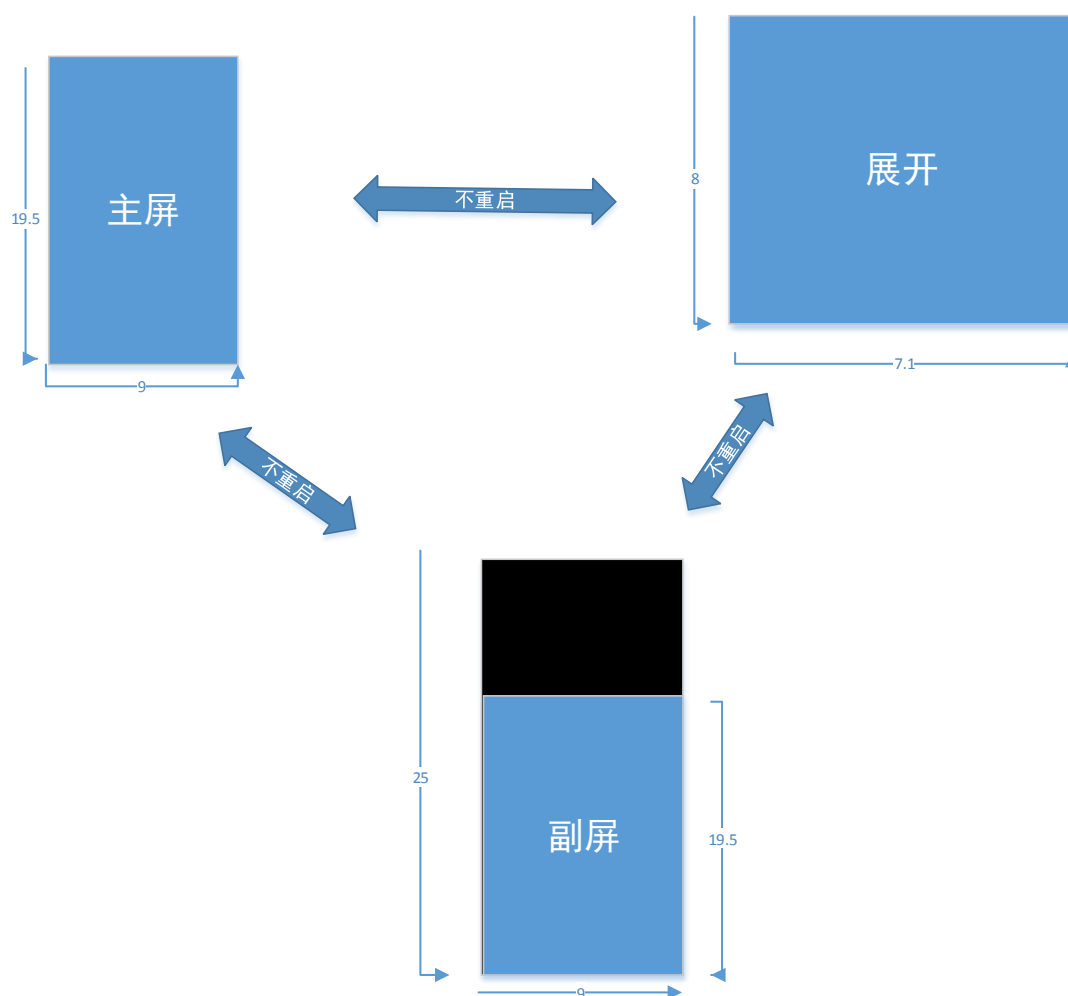


### 3.3 动态适配需求规格描述

说明：

- 1) 任意形态下打开一个应用，满足以上静态显示规格
- 2) 当用户物理形态切换时，应用需要做到不重启应用，并自动在新的设备形态下按照静态规格布局。

即：在用户进行展开，折叠等操作，应用任务不中断，自动适应各种屏幕下的静态布局规格。



## 4 设计和开发指导

### 4.1 大屏布局 UI 设计指导

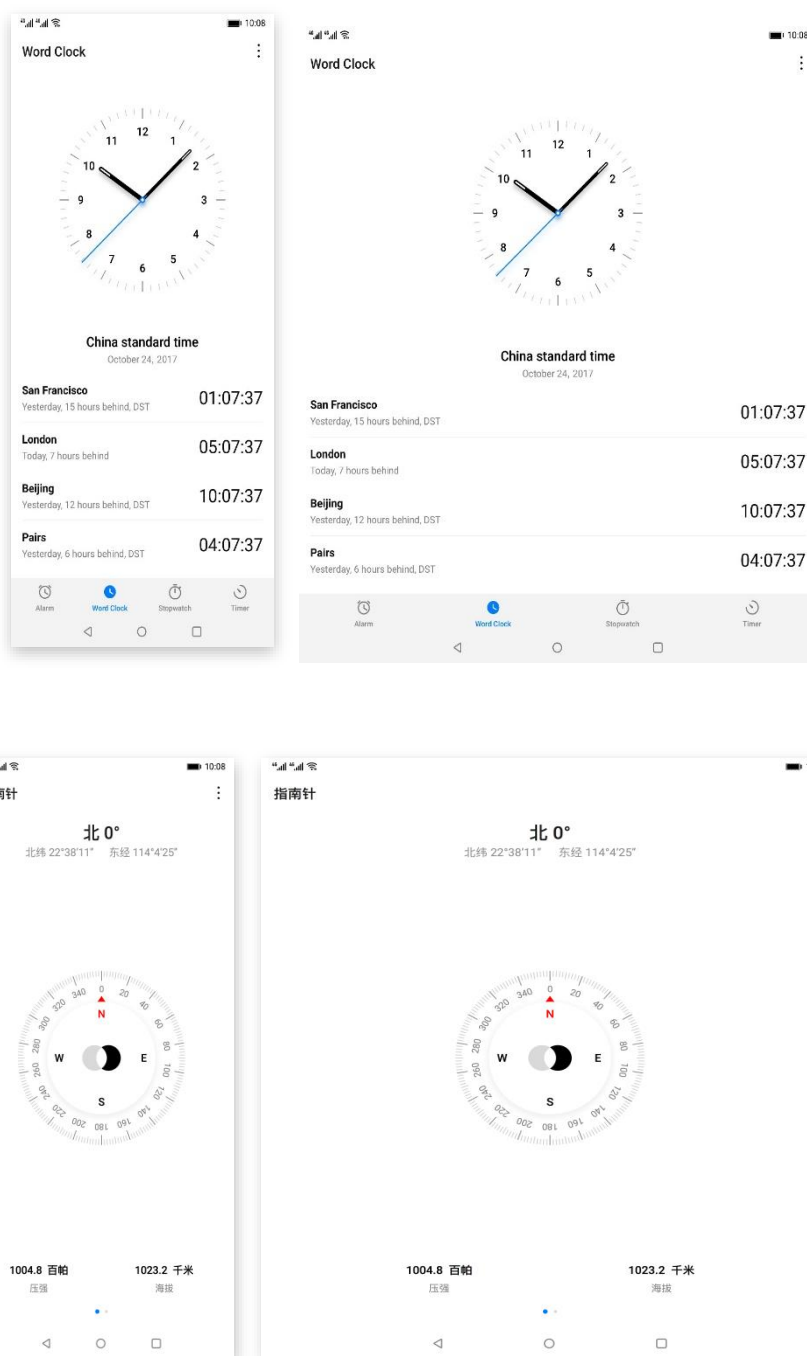
如下适配建议以及各应用适合适配类型，仅作为推荐性建议，具体由应用根据各自业务特性进行针对性设计。

#### 4.1.1 适配建议一：X 轴方向自适应

##### ✓ 说明

即，尽量保持Y轴方向上元素不变，X轴方向上自适应。

##### ✓ 示例：



✓ 范围（推荐性）：

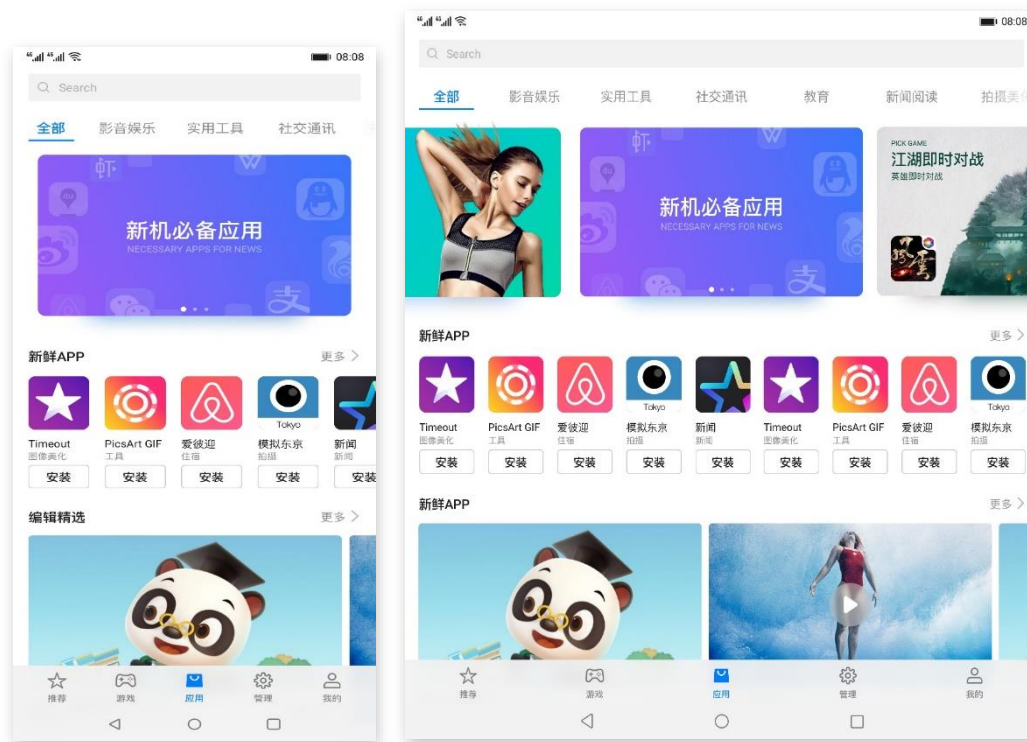
适用于界面元素相对单一，没有大量列表类、或较多显示元素的页面

#### 4.1.2 适配建议二：布局内容扩展

- ✓ 说明：参考pad布局显示更多内容，对于区分了手机和pad布局的应用，在展开态优先考虑参考pad的大屏布局适配展开态界面，显示更多内容；尽量保证Y轴方向元素的不

变

✓ 示例：



✓ 范围（推荐性）：

一般适用于WEB类应用，页面特征一般为元素多，适配原则以尽量显示较多元素优先。

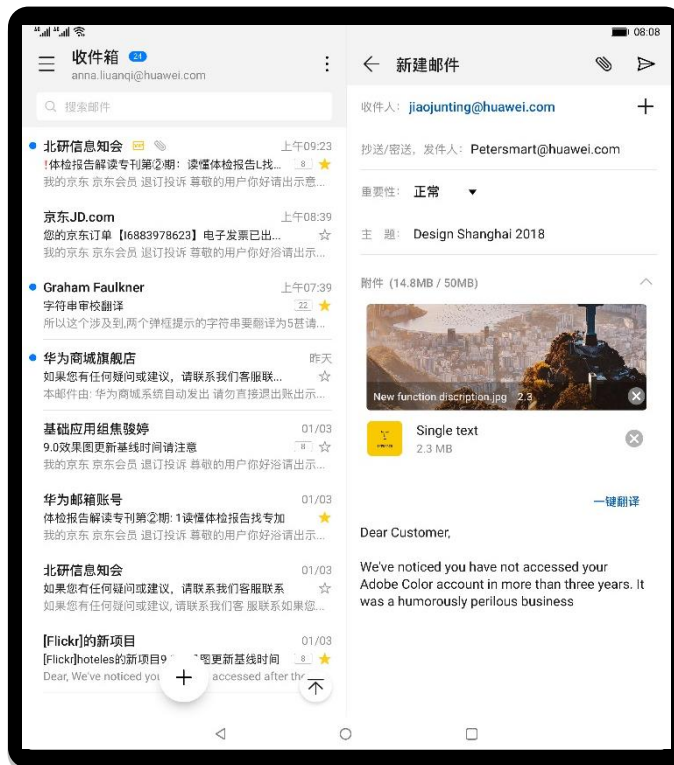
#### 4.1.3 适配建议三：分栏布局

✓ 说明：

对于设计过分栏能力的模块，需要在展开态体现分栏布局。

✓ 示例：





✓ 范围：

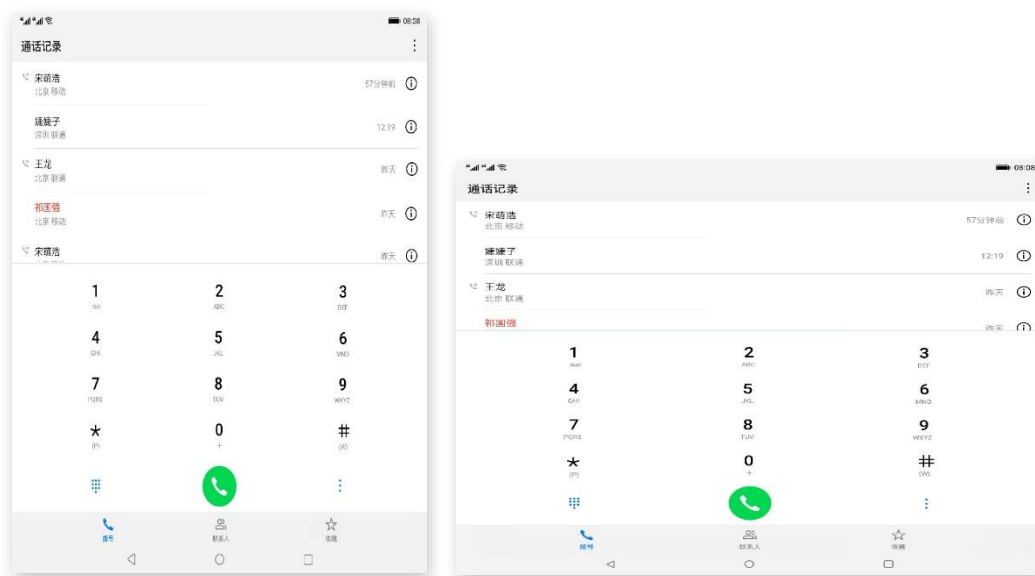
一般有明显list二级菜单的元素结构比较适合。

#### 4.1.4 适配建议四：横竖屏布局一致

✓ 说明：

考虑到展开态8:7.1的比例，展开态的横屏和竖屏建议一套布局。横竖一致；不对展开态的横屏特殊处理，挪移布局不用体现

✓ 示例：



## 4.2 适配开发指导

### 4.2.1 开发约束

#### ✓ 说明：

具备8:7.1， 19.5:9， 25:9， 三种可能的显示形态。

强烈建议应用支持可变比例显示（resizeable），在可预见的屏幕比例范围内，都可以做到良好适配。且在比例切换时，应用能够自己处理 screenSize 变化，不需要重走生命周期。总结来说，即要求应用无论静态还是场景切换下都可以完全支持自适应布局。

### 4.2.2 适配规格

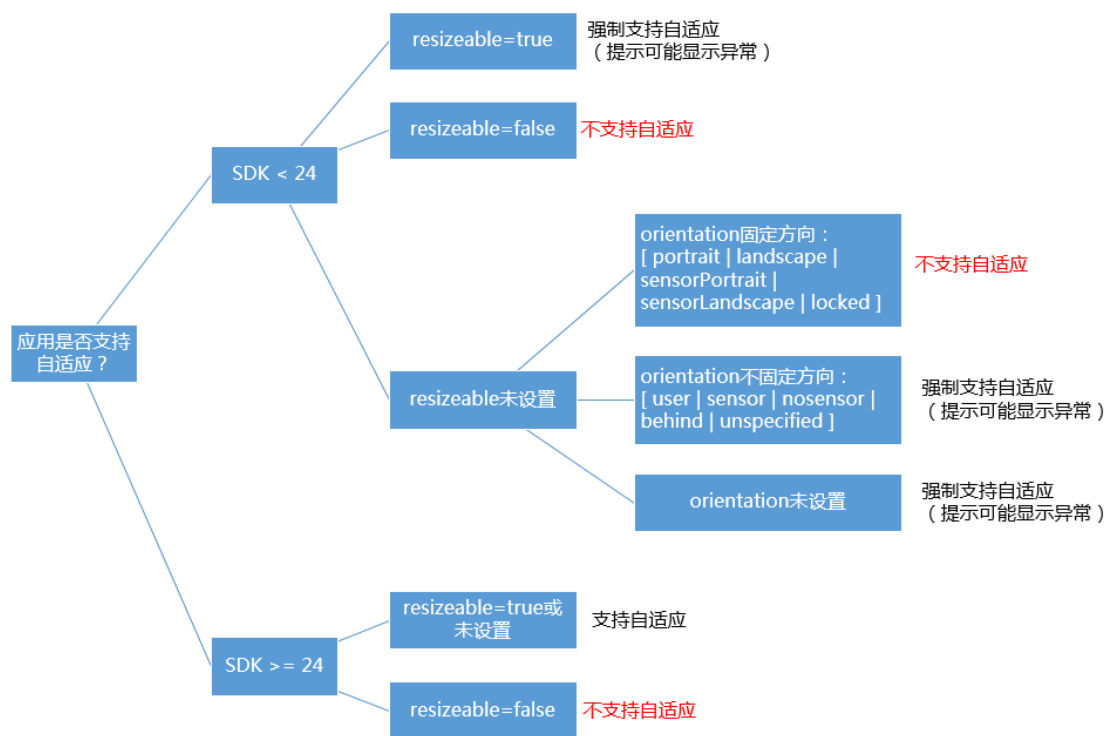
应用适配折叠屏需要做“屏幕比例适配”和“切换显示比例应用不重启适配”。

#### 1. 屏幕比例适配

##### ■ 适配方式1：应用支持自适应能力（推荐）

应用只要能做到自适应，在任意比例的屏幕都能做到全屏显示。

#### ✓ 应用如何申明自适应



✓ 设置方法:

在 manifest 文件的 `<activity>` 或 `<application>` 节点中设置 `android:resizeableActivity` 的值为 `true`, 可声明应用支持自适应显示, Activity 将能以分屏和 `freeform` 模式启动。

```
android:resizeableActivity=["true" | "false"]
```

■ 适配方式2: 设置应用支持的最大比例和最小比例适配

为保证应用在主副屏或者大屏态能保证满屏显示, 请尽快确保应用可以支撑较高范围的宽高比, 避免后续反复适配, 华为建议最高宽高比支持到2.4或者更大(可以确保在主屏、辅屏下满屏显示, 如果设置的比例小于2.4可能因为最终屏幕比例与当前规格存在差异引起应用无法全屏显示), 最小宽高比为1:1(可以确保在展开态大屏下满屏显示)

✓ 最大宽高比申明: `maxaspectratio: 2.4` (2.4表明在主副屏下满屏显示)

Android 7.1 及以下版本, 在 manifest 文件的 `<application>` 节点中增加 `<meta-data>` 数据, 设置最大支持比例:

```
<meta-data android:name="android.max_aspect" android:value="2.4" />
```

Android 8.0 及以上版本，在 manifest 文件的 `<activity>` 节点中增加 `android:MaxAspectRatio` 属性，声明最大支持比例：

```
<activity android:maxAspectRatio="2.4">
...
</activity>
```

✓ 最小宽高比申明： `minAspectRatio: 1.0` （App | Activity Level ）

Q之前版本：在 manifest 文件的 `<application>` 节点中增加 `<meta-data>` 数据，设置最小支持比例（声明为1.0即表示在展开态大屏下全屏显示）：

```
■ <meta-data android:name="android.min_aspect" android:value="1.0" />
```

Q版本之后支持：在 manifest 文件的 `<activity>` 节点中增加 `android:MinAspectRatio` 属性，声明最小支持比例：

```
<activity android:minAspectRatio="1.0">
...
</activity>
```

## 2. 切换显示比例应用不重启适配

■ 应用如何自处理 `screenSize` 事件，确保热切换时不重走生命周期

在 manifest 文件的 `<activity>` 节点中的 `android:configChanges` 属性增加 `screenSize|smallestScreenSize|screenLayout` 字符串，当屏幕比例变化时，系统会回调 Activity 的 `onConfigurationChanged()` 方法，而避免 Activity 重新启动。

```
■ android:configChanges="screenSize|smallestScreenSize|screenLayout"
```

应用复写 `onConfigurationChanged()` 方法，通过该方法的 `Configuration` 参数获得屏幕的分辨率等信息，就可以针对不同比例屏幕下的应用界面布局做相应调整，如切换布局、调整控件位置和间距等。

```
■ @Override
public void onConfigurationChanged(Configuration newConfig) {
```

```
super.onConfigurationChanged(newConfig);  
Log.i("test", "newConfig.screenHeightDp:" + newConfig.screenHeightDp  
    + ", newConfig.screenWidthDp" + newConfig.screenWidthDp);  
//根据屏幕分辨率信息调整应用的布局  
}
```

## 4.3 调试

### 4.3.1 模拟调试

如果开发者没有折叠屏手机，可以通过修改屏幕分辨率来模拟调试：

#### 1. 主屏（折叠）切全屏（展开）模拟方法：

- 预先将手机设置主屏分辨率：

```
adb shell wm size 1136x2480
```

- 通过修改手机分辨率为全屏分辨率模拟状态切换：

```
adb shell wm size 2200x2480
```

#### 2. 全屏（展开）切主屏（折叠）模拟方法：

- 预先将手机设置全屏分辨率：

```
adb shell wm size 2200x2480
```

- 通过修改手机分辨率为主屏分辨率模拟状态切换：

```
adb shell wm size 1136x2480
```

#### 3. 分辨率恢复方法：adb shell wm size reset