



# 多窗口支持

## 本文内容

- [概览](#)
- [多窗口生命周期](#)
- [针对多窗口模式配置应用](#)
- [在多窗口模式中运行应用](#)
- [测试应用的多窗口支持](#)

## 另请参阅

- [多窗口 Playground 示例应用](#)
- [在 Android N 为多窗口做准备的五条建议](#)

Android N 添加了对同时显示多个应用窗口的支持。在手持设备上，两个应用可以在“分屏”模式中左右并排或上下并排显示。在电视设备上，应用可以使用“画中画”模式，在用户与另一个应用交互的同时继续播放视频。

如果您使用 N Preview SDK 构建应用，则可以配置应用处理多窗口显示的方法。例如，您可以指定 Activity 的最小允许尺寸。您还可以禁用应用的多窗口显示，确保系统仅以全屏模式显示应用。

## 概览

Android N 允许多个应用同时共享屏幕。例如，用户可以分屏显示应用，在左边查看网页，同时在右边写邮件。用户体验取决于设备：

- 运行 Android N 的手持设备具有分屏模式。在此模式中，系统以左右并排或上下并排的方式分屏显示两个应用。用户可以拖动两个应用之间的分界线，放大其中一个应用，同时缩小另一个。
- 在运行 Android N 的 Nexus Player 上，应用能以 [画中画模式](#) 显示，即在用户浏览网页或与其他应用交互的同时继续显示内容。
- 较大设备的制造商可选择启用自由形状模式，在该模式中，用户可以自由调整各 Activity 的尺寸。若制造商启用此功能，设备将同时具有自由形状模式和分屏模式。

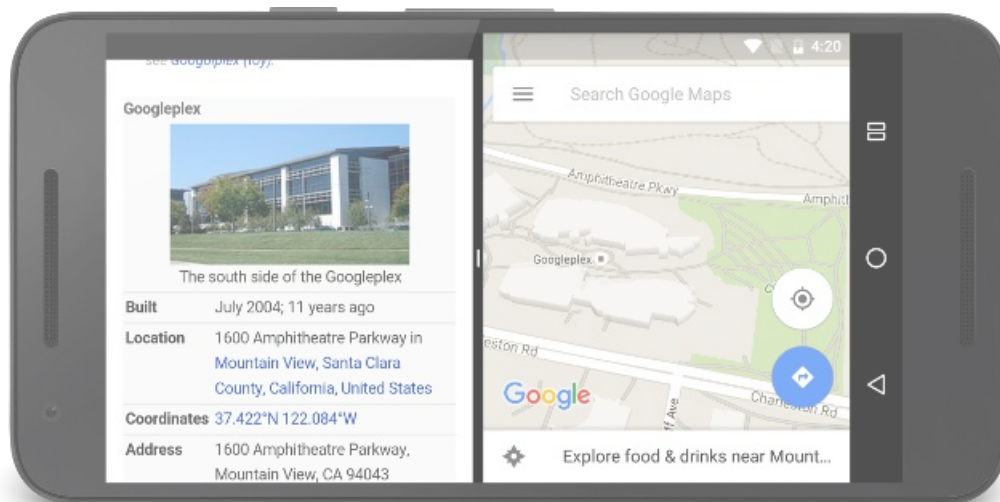


图 1. 两个应用的分屏模式中左右并排显示。

用户可以通过以下方式切换到多窗口模式：

- 若用户打开 [Overview 屏幕](#)并长按 Activity 标题，则可以拖动该 Activity 至屏幕突出显示的区域，使 Activity 进入多窗口模式。
- 若用户长按 Overview 按钮，设备上的当前 Activity 将进入多窗口模式，同时将打开 Overview 屏幕，用户可在该屏幕中选择要共享屏幕的另一个 Activity。

用户可以在两个 Activity 共享屏幕的同时在这两个 Activity 之间[拖放](#)数据（在此之前，用户只能在一个 Activity 内部拖放数据）。

## 多窗口生命周期

多窗口模式不会更改 [Activity 生命周期](#)。

在多窗口模式中，在指定时间只有最近与用户交互过的 Activity 为活动状态。该 Activity 将被视为顶级 Activity。所有其他 Activity 虽然可见，但均处于暂停状态。但是，这些已暂停但可见的 Activity 在系统中享有比不可见 Activity 更高的优先级。如果用户与其中一个暂停的 Activity 交互，该 Activity 将恢复，而之前的顶级 Activity 将暂停。

**注：**在多窗口模式中，用户仍可以看到处于暂停状态的应用。应用在暂停状态下可能仍需要继续其操作。例如，处于暂停模式但可见的视频播放应用应继续显示视频。因此，我们建议播放视频的活动不要暂停其 `onPause()` 处理程序中的视频。应暂停 `onStop()` 中的视频，并恢复 `onStart()` 中的视频播放。

如[处理运行时变更](#)中所述，用户使用多窗口模式显示应用时，系统将通知 Activity 发生配置变更。这也会发生在当用户调整应用大小，或将应用恢复到全屏模式时。该变更与系统通知应用设备从纵向模式切换到横向模式时的 Activity 生命周期影响基本相同，但设备不仅仅是交换尺寸，而是会变更尺寸。如[处理运行时变更](#)中所述，您的 Activity 可以自行处理配置变更，或允许系统销毁 Activity，并以新的尺寸重新创建该 Activity。

如果用户调整窗口大小，并在任意维度放大窗口尺寸，系统将调整 Activity 以匹配用户操作，同时根据需要发布[运行时变更](#)。如果应用在新公开区域的绘制滞后，系统将使用 `windowBackground` 属性或默认 `windowBackgroundFallback` 样式属性指定的颜色暂时填充该区域。

## 针对多窗口模式配置应用

如果您的应用面向 Android N，您可以对应用的 Activity 是否支持多窗口显示以及显示方式进行配置。您可以在清单文件中设置属性，以控制大小和布局。根 Activity 的属性设置适用于其任务栈中的所有 Activity。例如，如果根 Activity 已 `android:resizeableActivity` 设定为 `true`，则任务栈中的所有 Activity 都将可以调整大小。

**注：**如果您使用低于 Android N 版本的 SDK 构建多向应用，则用户在多窗口模式中使用应用时，系统将强制调整应用大小。系统将显示对话框，提醒用户应用可能会发生异常。系统不会调整定向应用的大小；如果用户尝试在多窗口模式下打开定向应用，应用将全屏显示。

### `android:resizeableActivity`

在清单的 `<activity>` 或 `<application>` 节点中设置该属性，启用或禁用多窗口显示：

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

如果该属性设置为 `true`，Activity 将能以分屏和自由形状模式启动。如果此属性设置为 `false`，Activity 将不支持多窗口模式。如果该值为 `false`，且用户尝试在多窗口模式下启动 Activity，该 Activity 将全屏显示。

如果您的应用面向 Android N，但未对该属性指定值，则该属性的值默认设为 `true`。

### `android:supportsPictureInPicture`

在清单文件的 `<activity>` 节点中设置该属性，指明 Activity 是否支持画中画显示。如果 `android:resizeableActivity` 为 `false`，将忽略该属性。

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

## 布局属性

对于 Android N，`<layout>` 清单元素支持以下几种属性，这些属性影响 Activity 在多窗口模式中的行为：

`android:defaultWidth`

以自由形状模式启动时 Activity 的默认宽度。

`android:defaultHeight`

以自由形状模式启动时 Activity 的默认高度。

`android:gravity`

以自由形状模式启动时 Activity 的初始位置。请参阅 [Gravity](#) 参考资料，了解合适的值设置。

`android:minimalHeight`、`android:minimalWidth`

分屏和自由形状模式中 Activity 的最小高度和最小宽度。如果用户在分屏模式中移动分界线，使 Activity 尺寸低于指定的最小值，系统会将 Activity 裁剪为用户请求的尺寸。

例如，以下节点显示了如何指定 Activity 在自由形状模式中显示时 Activity 的默认大小、位置和最小尺寸：

```
<activity android:name=".MyActivity">
  <layout android:defaultHeight="500dp"
    android:defaultWidth="600dp"
    android:gravity="top|end"
    android:minimalHeight="450dp"
    android:minimalWidth="300dp" />
</activity>
```

## 在多窗口模式中运行应用

Android N 添加了新功能，以支持可在多窗口模式中运行的应用。

### 多窗口模式中被禁用的功能

在设备处于多窗口模式中时，某些功能会被禁用或忽略，因为这些功能对与其他 Activity 或应用共享设备屏幕的 Activity 而言没有意义。此类功能包括：

- 某些[系统 UI](#) 自定义选项将被禁用；例如，在非全屏模式中，应用无法隐藏状态栏。
- 系统将忽略对 `android:screenOrientation` 属性所作的更改。

### 多窗口变更通知和查询

`Activity` 类中添加了以下新方法，以支持多窗口显示。有关各方法的详细信息，请参阅 [N Preview SDK 参考](#)。

`Activity.isInMultiWindowMode()`

调用该方法以确认 Activity 是否处于多窗口模式。

`Activity.isInPictureInPictureMode()`

调用该方法以确认 Activity 是否处于画中画模式。

**注：**画中画模式是多窗口模式的特例。如果 `myActivity.isInPictureInPictureMode()` 返回 `true`，则 `myActivity.isInMultiWindowMode()` 也返回 `true`。

`Activity.onMultiWindowModeChanged()`

Activity 进入或退出多窗口模式时系统将调用此方法。在 Activity 进入多窗口模式时，系统向该方法传递 `true` 值，在退出多窗口模式时，则传递 `false` 值。

`Activity.onPictureInPictureModeChanged()`

Activity 进入或退出画中画模式时系统将调用此方法。在 Activity 进入画中画模式时，系统向该方法传递 `true` 值，在退出画中画模式时，

则传递 `false` 值。

每个方法还有 `Fragment` 版本，例如 `Fragment.isInMultiWindowMode()`。

## 进入画中画模式

如需在画中画模式中启动 Activity，请调用新方法 `Activity.enterPictureInPictureMode()`。如果设备不支持画中画模式，则此方法无效。如需了解详细信息，请参阅[画中画](#)文档。

## 在多窗口模式中启动新 Activity

在启动新 Activity 时，用户可以提示系统如果可能，应将新 Activity 显示在当前 Activity 旁边。要执行此操作，可使用标志 `Intent.FLAG_ACTIVITY_LAUNCH_TO_ADJACENT`。传递此标志将请求以下行为：

- 如果设备处于分屏模式，系统会尝试在启动系统的 Activity 旁创建新 Activity，这样两个 Activity 将共享屏幕。系统并不一定能实现此操作，但如果可以，系统将使两个 Activity 处于相邻的位置。
- 如果设备不处于分屏模式，则该标志无效。

如果设备处于自由形状模式，则在启动新 Activity 时，用户可通过调用 `ActivityOptions.setLaunchBounds()` 指定新 Activity 的尺寸和屏幕位置。如果设备不处于多窗口模式，则该方法无效。

**注：**如果您在任务栈中启动 Activity，该 Activity 将替换屏幕上的 Activity，并继承其所有的多窗口属性。如果要在多窗口模式中以单独的窗口启动新 Activity，则必须新的任务栈中启动此 Activity。

## 支持拖放

用户可以在两个 Activity 共享屏幕的同时在这两个 Activity 之间[拖放](#)数据（在此之前，用户只能在一个 Activity 内部拖放数据）。因此，如果您的应用目前不支持拖放功能，您可以在其中添加此功能。

N Preview SDK 扩展了 `android.view` 软件包，以支持跨应用拖放。有关以下类和方法的详细信息，请参阅[N Preview SDK 参考](#)。

`android.view.DropPermissions`

令牌对象，负责指定对接收拖放数据的应用授予的权限。

`View.startDragAndDrop()`

`View.startDrag()` 的新别名。要启用跨 Activity 拖放，请传递新标志 `View.DRAG_FLAG_GLOBAL`。如需对接收拖放数据的 Activity 授予 URI 权限，可根据情况传递新标志 `View.DRAG_FLAG_GLOBAL_URI_READ` 或 `View.DRAG_FLAG_GLOBAL_URI_WRITE`。

`View.cancelDragAndDrop()`

取消当前正在进行的拖动操作。只能由发起拖动操作的应用调用。

`View.updateDragShadow()`

替换当前正在进行的拖动操作的拖动阴影。只能由发起拖动操作的应用调用。

`Activity.requestDropPermissions()`

请求使用 `DragEvent` 中包含的 `ClipData` 传递的内容 URI 的权限。

## 测试应用的多窗口支持

无论您是否针对 Android N 更新应用，都应验证应用在多窗口模式下的行为，以防用户尝试在运行 Android N 的设备上以多窗口模式启动应用。

## 配置测试设备

如果在设备上安装 Android N，则将自动支持分屏模式。

## 如果应用并非使用 N Preview SDK 构建

如果您的应用不是使用 N Preview SDK 构建的，则用户尝试在多窗口模式中使用应用时，系统将强制调整应用大小，除非应用进行了定向声明。

如果您的应用没有进行定向声明，则应在运行 Android N 的设备上启动应用，并尝试将应用切换到分屏模式。验证并确保在强制调整应用大小时用户体验可接受。

如果应用进行了定向声明，则应尝试将应用切换到多窗口模式。验证并确保执行此操作后，应用仍保持全屏模式。

## 如果支持多窗口模式

如果您的应用是使用 N Preview SDK 构建的，且未禁用多窗口支持，则分别在分屏和自由形状模式下验证以下行为。

- 在全屏模式下启动应用，然后通过长按 Overview 按钮切换到多窗口模式。验证并确保应用正常切换。
- 直接的多窗口模式中启动应用，验证并确保应用正常启动。您可以按一下 Overview 按钮，再长按应用的标题栏，并将其拖动到屏幕上任一突出显示的区域，从而在多窗口模式中启动应用。
- 拖动分界线，在分屏模式中调整应用的大小。验证并确保应用正常调整大小且未崩溃，并且必要的 UI 元素仍可见。
- 如果您指定了应用的最小尺寸，请尝试将应用尺寸调整到低于最小值。验证并确保无法将应用尺寸调整到低于指定最小值。
- 完成所有测试后，验证并确保应用性能可以接受。例如，验证并确保调整应用大小后更新 UI 没有长时间的滞后。

### 测试检查单

要在多窗口模式中验证应用性能，请执行以下操作。除非另有说明，否则请分别在分屏和多窗口模式中执行以下操作。

- 进入和退出多窗口模式。
- 从您的应用切换到另一个应用，验证并确保应用在非活动但可见的状态下正常运行。例如，如果您的应用在播放视频，则验证并确保在用户与另一个应用交互时视频仍在继续播放。
- 在分屏模式中，尝试移动分界线，放大或缩小应用。分别在左右和上下并排显示模式中尝试这些操作。验证并确保应用不会崩溃，主要功能可见，且调整操作不需要过长时间。
- 快速连续执行几次调整操作。验证并确保应用不会崩溃或出现内存泄漏。有关检查应用内存使用率的信息，请参阅[查看内存使用率](#)。
- 在多个不同窗口配置中正常使用应用，验证并确保应用正常运行。验证并确保文本可读，且 UI 元素大小正常，不影响交互。

## 如果已禁用多窗口支持

如果您通过设置 `android:resizableActivity="false"` 禁用了多窗口支持，则应在运行 Android N 的设备上启动应用，并尝试将应用切换到自由形状和分屏模式。验证并确保执行此操作后，应用仍保持全屏模式。