

Supporting Multi-Window

In Android 7.0, users can have multiple apps simultaneously displayed on their device screen with the new platform feature, multi-window. In addition to the default implementation of multi-window, Android 7.0 supports a few varieties of multi-window: split-screen, free-form, and picture-in-picture.

- **Split-screen** is the base implementation of multi-window and provides two activity panes for users to place apps.
- **Freeform** allows users to dynamically resize the activity panes and have more than two apps visible on their screen.
- **Picture-in-picture (PIP)** allows Android devices to continue playing video content in a small window while the user interacts with other applications.

To implement the multi-window feature, device manufacturers set a flag in the config file on their devices to enable or disable multi-window support.

Implementation

Multi-window support is enabled by default in Android 7.0. To disable it, set the `config_supportsMultiWindow` flag to false in the [config.xml](https://android.googlesource.com/platform/frameworks/base/+/master/core/res/res/values/config.xml) file.

For devices that declare `ActivityManager.isLowRam()`, multi-window is disabled regardless of the value of `config_supportsMultiWindow` flag.

Split-screen

The default multi-window experience is split-screen mode, where the System UI is divided directly down the middle of the device in portrait or landscape. Users can resize the window by dragging the dividing line side-to-side or top-to-bottom, depending on the device orientation.

Then device manufacturers can choose if they want to enable freeform or PIP.

Freeform

After enabling standard multi-window mode with the flag `config_supportsMultiWindow`, device manufacturers can optionally allow freeform windowing. This mode is most useful for manufacturers of larger devices, like tablets.

To support freeform mode, enable the `PackageManager#FEATURE_FREEFORM_WINDOW_MANAGEMENT` system feature in [/android/frameworks/base/core/java/android/content/pm/PackageManager.java](https://android.googlesource.com/platform/frameworks/base/core/java/android/content/pm/PackageManager.java) (https://android.googlesource.com/platform/frameworks/base/+/master/core/java/android/content/pm/PackageManager.java) and set `config_freeformWindowManagement` to true in [config.xml](https://android.googlesource.com/platform/frameworks/base/+/master/core/res/res/values/config.xml) (https://android.googlesource.com/platform/frameworks/base/+/master/core/res/res/values/config.xml).

```
<bool name="config_freeformWindowManagement">true</bool>
```

Picture-in-picture

After enabling standard multi-window mode with the flag `config_supportsMultiWindow`, device manufacturers can support [picture-in-picture](https://developer.android.com/training/tv/playback/picture-in-picture.html) to allow users to continue watching video while browsing other activities. While this features is primarily targeted at Android Television devices, other device form factors may support this feature.

To support PIP, enable the `PackageManager#FEATURE_PICTURE_IN_PICTURE` system feature in [/android/frameworks/base/core/java/android/content/pm/PackageManager.java](https://android.googlesource.com/platform/frameworks/base/core/java/android/content/pm/PackageManager.java) (https://android.googlesource.com/platform/frameworks/base/+/master/core/java/android/content/pm/PackageManager.java).

System UI

Support all standard System UIs according to <https://developer.android.com/guide/topics/ui/multi-window.html#testing> (https://developer.android.com/guide/topics/ui/multi-window.html#testing)

Applications

To support multi-window mode for preloaded apps, consult the [Android developer documentation](https://developer.android.com/guide/topics/ui/multi-window.html) (https://developer.android.com/guide/topics/ui/multi-window.html).

Validation

To validate their implementation of multi-window, device manufacturers should run [CTS tests](https://android.googlesource.com/platform/cts/+/master/hostsidetests/services/activitymanager/src/android/server/cts) (https://android.googlesource.com/platform/cts/+/master/hostsidetests/services/activitymanager/src/android/server/cts) and follow the [testing instructions for multi-window](https://developer.android.com/guide/topics/ui/multi-window.html#testing) (https://developer.android.com/guide/topics/ui/multi-window.html#testing).

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](http://creativecommons.org/licenses/by/3.0/) (http://creativecommons.org/licenses/by/3.0/), and code samples are licensed under the [Apache 2.0 License](http://www.apache.org/licenses/LICENSE-2.0) (http://www.apache.org/licenses/LICENSE-2.0). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (https://developers.google.com/terms/site-policies). Java is a registered trademark of Oracle and/or its affiliates.

Last updated March 27, 2017.