

<supports-screens>

SYNTAX:

```
<supports-screens android:resizeable=["true" | "false"]
                  android:smallScreens=["true" | "false"]
                  android:normalScreens=["true" | "false"]
                  android:largeScreens=["true" | "false"]
                  android:xlargeScreens=["true" | "false"]
                  android:anyDensity=["true" | "false"] />
```

CONTAINED IN:

[<manifest>](#)

DESCRIPTION:

Lets you specify the screen dimensions the application supports. By default, a modern application (using API Level 4 or higher) supports all screen sizes; older applications are assumed to support only the "normal" screen size. Screen size is determined as the available pixels to an application after density scaling has been applied. (Note that screen size is a separate axis from screen density.)

An application "supports" a given screen size if it fills the entire screen and works as expected. By default, the system will resize your application to fill the screen, if you have set either [minSdkVersion](#) or [targetSdkVersion](#) to "4" or higher. Resizing works well for most applications and you don't have to do any extra work to make your application work on larger screens.

In addition to allowing the system to resize your application, you can add additional support for different screen sizes by providing [alternative layout resources](#) for different sizes. For instance, you might want to modify the layout of an activity when it is on a tablet or similar device that has an *xlarge* screen.

If your application does not support *large* or *xlarge* screens, then you should declare that it is not resizeable by setting [android:resizeable](#) to "false", so that the system will not resize your application on larger screens.

If your application does not support *small* screens, then there isn't much the system can do to make the application work well on a smaller screen, so external services (such as Android Market) should not allow users to install the application on such screens.

For more information, see [Supporting Multiple Screens](#).

ATTRIBUTES:

[android:resizeable](#)

Indicates whether the application is resizeable for different screen sizes. This attribute is true, by default, if you have set either [minSdkVersion](#) or [targetSdkVersion](#) to "4" or higher. Otherwise, it is false by default. If set false, the system will not resize your application when run on *large* or *xlarge* screens. Instead, the application appears in a "postage stamp" that equals the *normal* screen size that your application does support. This is less than an ideal experience for users, because the application appears smaller than the available screen, but it might help your application run normally if it were designed only for the *normal* screen size and some behaviors do not work when resized.

To provide the best experience on all screen sizes, you should allow resizing and, if your application does not work well on larger screens, follow the guide to [Supporting Multiple Screens](#) to enable additional screen support.

[android:smallScreens](#)

Indicates whether the application supports smaller screen form-factors. A small screen is defined as one with a smaller aspect ratio than the "normal" (traditional HVGA) screen. An application that does not support small

screens *will not be available* for small screen devices from external services (such as Android Market), because there is little the platform can do to make such an application work on a smaller screen. If the application has set either [minSdkVersion](#) or [targetSdkVersion](#) to "4" or higher, the default value for this is "true", any value less than "4" results in this set to "false".

`android:normalScreens`

Indicates whether an application supports the "normal" screen form-factors. Traditionally this is an HVGA medium density screen, but WQVGA low density and WVGA high density are also considered to be normal. This attribute is "true" by default, and applications currently should leave it that way.

`android:largeScreens`

Indicates whether the application supports larger screen form-factors. A large screen is defined as a screen that is significantly larger than a "normal" phone screen, and thus might require some special care on the application's part to make good use of it, though it may rely on resizing by the system to fill the screen. If the application has set either [minSdkVersion](#) or [targetSdkVersion](#) to "4" or higher, the default value for this is "true", any value less than "4" results in this set to "false".

`android:xlargeScreens`

Indicates whether the application supports extra large screen form-factors. An xlarge screen is defined as a screen that is significantly larger than a "large" screen, such as a tablet (or something larger) and may require special care on the application's part to make good use of it, though it may rely on resizing by the system to fill the screen. If the application has set either [minSdkVersion](#) or [targetSdkVersion](#) to "4" or higher, the default value for this is "true", any value less than "4" results in this set to "false".

This attribute was introduced in API Level 9.

`android:anyDensity`

Indicates whether the application includes resources to accommodate any screen density. Older applications (before API Level 4) are assumed unable to accomodate all densities and this is "false" by default. If the application has set either [minSdkVersion](#) or [targetSdkVersion](#) to "4" or higher, the default value for this is "true". Otherwise, it is "false". You can explicitly supply your abilities here.

Based on the "standard" device screen density (medium dpi), the Android framework will scale down application assets by a factor of 0.75 (low dpi screens) or scale them up by a factor of 1.5 (high dpi screens), when you don't provide alternative resources for a specific screen density. The screen density is expressed as dots-per-inch (dpi).

INTRODUCED IN:
API Level 4

SEE ALSO:

- [Supporting Multiple Screens](#)
- [DisplayMetrics](#)

[← Back to The AndroidManifest.xml File](#)

[↑ Go to top](#)

Except as noted, this content is licensed under [Apache 2.0](#). For details and restrictions, see the [Content License](#).

Android 3.1 r1 - 17 Jun 2011 10:58

[Site Terms of Service](#) - [Privacy Policy](#) - [Brand Guidelines](#)