Screen Compatibility Mode

In this document

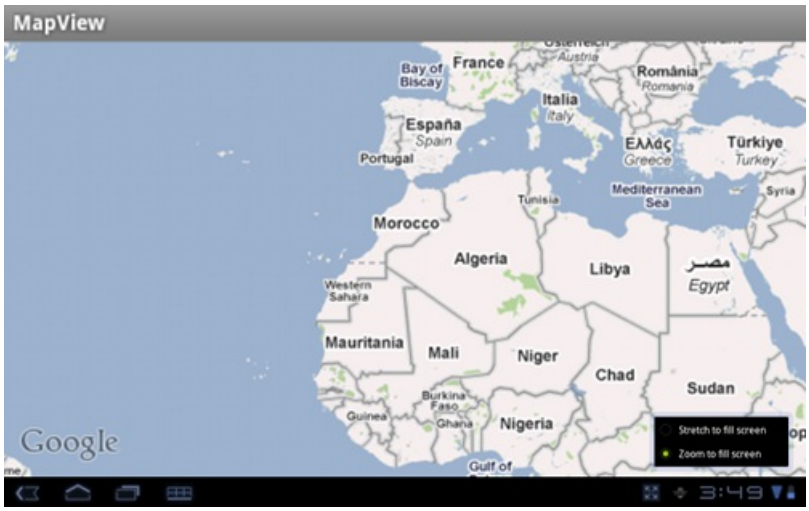

Figure 1. An application running in compatibility mode on an Android 3.2 tablet.

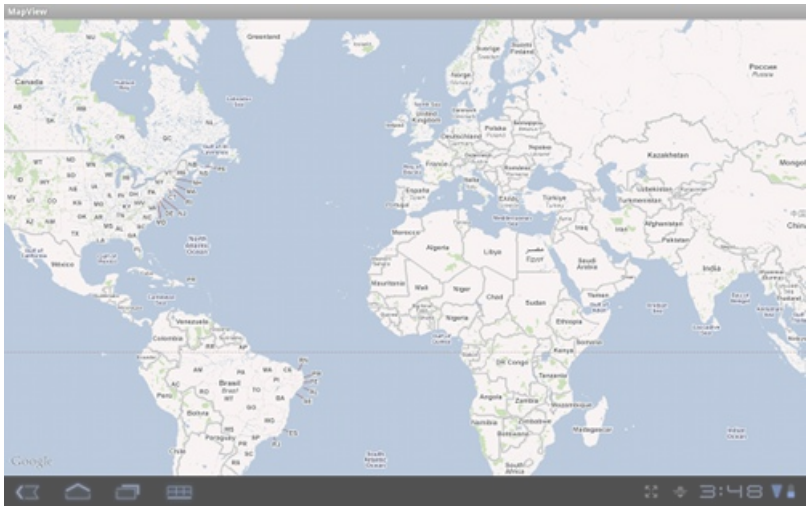

Figure 2. The same application from figure 1, with compatibility mode disabled.

> **Notice:** If you've developed an application for a version of Android lower than Android 3.0, but it does resize properly for larger screens such as tablets, you should disable screen compatibility mode in order to maintain the best user experience. To learn how to quickly disable the user option, jump to Disabling Screen Compatibility Mode.

Screen compatibility mode is an escape hatch for applications that are not properly designed to resize for larger screens such as tablets. Since Android 1.6, Android has supported a variety of screen sizes and does most of the work to resize application layouts so that they properly fit each screen. However, if your application does not successfully follow the guide to Supporting Multiple Screens, then it might encounter some rendering issues on larger screens. For applications with this problem, screen compatibility mode can make the application a little more usable on larger screens.

There are two versions of screen compatibility mode with slightly different behaviors:

Version 1 (Android 1.6 - 3.1)

> The system draws the application's UI in a "postage stamp" window. That is, the system draws the application's layout the same as it would on a normal size handset (emulating a 320dp x 480dp screen), with a black border that fills the remaining area of the screen.
>
> This was introduced with Android 1.6 to handle apps that were designed only for the original screen size of 320dp x 480dp. Because there are so few active devices remaining that run Android 1.5, almost all applications should be developed against Android 1.6 or greater and should not have version 1 of screen compatibility mode enabled for larger screens. This version is considered obsolete.
>
> To disable this version of screen compatibility mode, you simply need to set `android:minSdkVersion` or `android:targetSdkVersion` to `"4"` or higher, or set `android:resizeable` to `"true"`.

Version 2 (Android 3.2 and greater)

> The system draws the application's layout the same as it would on a normal size handset (approximately emulating a 320dp x 480dp screen), then scales it up to fill the screen. This essentially "zooms" in on your layout to make it bigger, which will usually cause artifacts such as blurring and pixelation in your UI.
>
> This was introduced with Android 3.2 to further assist applications on the latest tablet devices when the applications have not yet implemented techniques for Supporting Multiple Screens.
>
> In general, large screen devices running Android 3.2 or higher allow users to enable screen compatibility mode when the application does not **explicitly declare that it supports large screens** in the manifest file. When this is the case, an icon (with outward-pointing arrows) appears next to the clock in the system bar, which allows the user to toggle screen compatibility mode on and off (figure 3). An application can also explicitly declare that it *does not* support large screens such that screen compatibility mode is always enabled and the user cannot disable it. (How to declare your application's support for large screens is discussed in the following sections.)

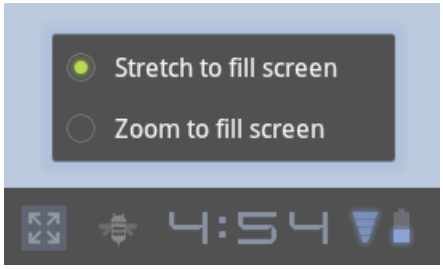

Figure 3. The pop up menu to toggle screen compatibility mode (currently disabled, so normal resizing occurs).

As a developer, you have control over when your application uses screen compatibility mode. The following sections describe how you can choose to disable or enable screen compatibility mode for larger screens when running Android 3.2 or higher.

Disabling Screen Compatibility Mode

If you've developed your application primarily for versions of Android lower than 3.0, but **your application does resize properly** for larger screens such as tablets, **you should disable screen compatibility mode** in order to maintain the best user experience. Otherwise, users may enable screen compatibility mode and experience your application in a less-than-ideal format.

By default, screen compatibility mode for devices running Android 3.2 and higher is offered to users as an optional feature when one of the following is true:

- Your application has set both `android:minSdkVersion` and `android:targetSdkVersion` to `"10"` or lower and **does not explicitly declare support** for large screens using the `<supports-screens>` element.
- Your application has set either `android:minSdkVersion` or `android:targetSdkVersion` to `"11"` or higher and **explicitly declares that it does not support** large screens, using the `<supports-screens>` element.

To completely disable the user option for screen compatibility mode and remove the icon in the system bar, you can do one of the following:

- **Easiest:**

  In your manifest file, add the `<supports-screens>` element and specify the `android:xlargeScreens` attribute to `"true"`:

```
<supports-screens android:xlargeScreens="true" />
```

That's it. This declares that your application supports all larger screen sizes, so the system will always resize your layout to fit the screen. This works regardless of what values you've set in the `<uses-sdk>` attributes.

- **Easy but has other effects:**

  In your manifest's `<uses-sdk>` element, set `android:targetSdkVersion` to `"11"` or higher:

  ```
  <uses-sdk android:minSdkVersion="4" android:targetSdkVersion="11" />
  ```

  This declares that your application supports Android 3.0 and, thus, is designed to work on larger screens such as tablets.

  > **Caution:** When running on Android 3.0 and greater, this also has the effect of enabling the Holographic theme for you UI, adding the Action Bar to your activities, and removing the Options Menu button in the system bar.

  If screen compatibility mode is still enabled after you change this, check your manifest's `<supports-screens>` and be sure that there are no attributes set `"false"`. The best practice is to always explicitly declare your support for different screen sizes using the `<supports-screens>` element, so you should use this element anyway.

  For more information about updating your application to target Android 3.0 devices, read Optimizing Apps for Android 3.0.

Enabling Screen Compatibility Mode

When your application is targeting Android 3.2 (API level 13) or higher, you can affect whether compatibility mode is enabled for certain screens by using the `<supports-screens>` element.

> **Note:** Screen compatibility mode is **not** a mode in which you should want your application to run—it causes pixelation and blurring in your UI, due to zooming. The proper way to make your application work well on large screens is to follow the guide to Supporting Multiple Screens and provide alternative layouts for different screen sizes.

By default, when you've set either `android:minSdkVersion` or `android:targetSdkVersion` to `"11"` or higher, screen compatibility mode is **not** available to users. If either of these are true and your application does not resize properly for larger screens, you can choose to enable screen compatibility mode in one of the following ways:

- In your manifest file, add the `<supports-screens>` element and specify the `android:compatibleWidthLimitDp` attribute to `"320"`:

  ```
  <supports-screens android:compatibleWidthLimitDp="320" />
  ```

  This indicates that the maximum "smallest screen width" for which your application is designed is 320dp. This way, any devices with their smallest side being larger than this value will offer screen compatibility mode as a user-optional feature.

  > **Note:** Currently, screen compatibility mode only emulates handset screens with a 320dp width, so screen compatibility mode is not applied to any device if your value for `android:compatibleWidthLimitDp` is larger than 320.

- If your application is functionally broken when resized for large screens and you want to force users into screen compatibility mode (rather than simply providing the option), you can use the `android:largestWidthLimitDp` attribute:

  ```
  <supports-screens android:largestWidthLimitDp="320" />
  ```

  This works the same as `android:compatibleWidthLimitDp` except it force-enables screen compatibility mode and does not allow users to disable it.