

Sizing Android Visual Elements Correctly

Right-size icons, tables, and buttons for all form factors and resolutions

June 14, 2011

URL: <http://www.drdobbs.com/mobile/sizing-android-visual-elements-correctly/230600075>

Developers are now increasingly porting iPhone apps to the Android platform. The differences in platform are many, and the principles and elements that work well for the iPhone don't work as well on Android platforms.

With iOS, from the moment you start designing an app, you know how Apple wants you to build it. The design plans are exhaustive and accessible. Unfortunately, there is no comparable resource for Android that gives consideration to asset creation or creative direction.

In the Android world, we need to define a boundary between buttons and icons and reassess what needs to be designed versus what is better left programmed — and it's important to remain open-minded, patient, and exploratory. Android is a curious frontier and while there are definitely rules and standards, they are not always obvious.

There are only a limited handful of resolutions and sizes within the iOS landscape, but within Android, the number of different devices means that form factors vary significantly. Designing an app that looks good across the spectrum of Android devices is no small task, but it is not impossible. Platform fragmentation on Android is indeed a threat, but a lot of the alarmist noise about the issue is really coming from designers who are intimidated by the operating system and the lack of design guidelines.

Borne out of its own struggles developing and designing for the Android platform, our company set out to address this need by creating a set of [Android Design Guidelines](#). The final document is an exhaustively researched resource that sheds light on making apps work from device to device. Since publishing the original, we've added a supplement that addresses the recent Honeycomb releases.

This article summarizes key portions the guidelines and is intended to help developers and designers size Android assets for optimal resolution.

General Resolution

Ninety percent of Android design woes stem from accommodating multiple device resolutions and screen sizes. Due to the widely varying array of devices, it is difficult to pinpoint a specific pixel resolution. With this in mind, Android has developed four generalized resolutions and four generalized densities for thinking about device screens.

There are four generalized screen sizes:

- Small (2-3 inches)
- Normal (3-5 inches)
- Large (4-7 inches)
- X-Large (7-10 inches) Tablets only

And four generalized resolutions:

- LDPI (100-120 dpi)
- MDPI (120-160 dpi)
- HDPI (160-240 dpi)
- XHDPI (240-320 dpi)

These are represented visually in Figure 1.

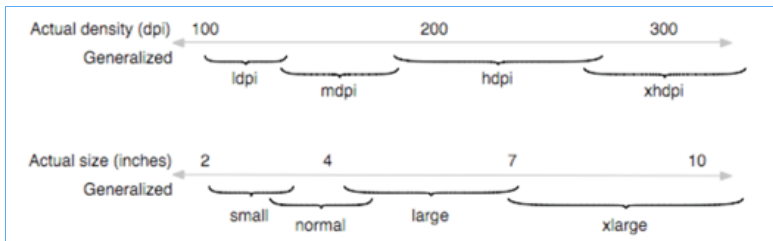


Figure 1: The range of screen sizes and resolutions on current Android devices.

These numbers should enter into your choice of:

- What your buttons look like
- What sort of gradients you use
- How complex your icons are
- What sort of backgrounds you make.

When creating wireframes for an Android app, it is probably wise to work your layout into multiple sizes to make sure that your application will work across as many platforms as possible. Obviously, this will make for a longer project, but due diligence will make for a better application. It may not always be the best course to base your design exclusively on the top devices, depending on who your audience is.

You have two choices: Keep the design simple and squeaky clean (for example, don't use gradients). Or employ a fancy background, and thus be prepared to make custom assets for each of the various resolutions and screen sizes. The variants you'll need to take care of are shown in Table 1.

	Low Density (129), <i>ldpi</i>	Medium Density (160), <i>mdpi</i>	High Density (240), <i>hdpi</i>	Extra High Density (320), <i>xhdpi</i>
<i>Small Screen</i>	QVGA (240x320)			
<i>Normal Screen</i>	WQVGA400 (240x400) WQVGA432 (240x432)	HVGA (320x480)	WVGA800 (480x800)	640x490
<i>Large Screen</i>		WVGA800* (480x800)	WVGA854* (480x854)	
<i>Extra Large Screen</i>	1024x600	1024x768 1280x768	1536x1152, 1920x1152, 1920x1200	2048x1536, 2560x1536, 2560x1600

Table 1: Regardless of the actual screen size or density, applications are programmed into these four categories.

**For the most part, screen sizes and densities correlate.*

The bottom line is: When designing for Android, not taking the complexity of density and screen size into consideration will make the project more difficult.

UI Elements

Google has been vague with its sizing rules for UI elements, but considering the densities and screen sizes, a set of loose guideline can be helpful. The following measurements and resolutions come from having measured the navigation elements of a numerous screenshots of top-selling apps. They provide a guideline for preparing icons, tab bars, options menus and context menus.

Icons

Android is fairly rigid about icon creation. Due to the range of screen sizes and densities, it's necessary to design different sets of icons. Also, because most buttons need to be [drawn patched](#), it is important to consider an icon as a separate asset from the button itself. Table 2 shows the icon sizes for different kinds of icons displayed at different screen densities.

Icon Type	Standard Asset Sizes (in Pixels) for Generalized Screen Densities		
	Low Density Screen (<i>ldpi</i>)	Medium Density Screen (<i>mdpi</i>)	High Density Screen (<i>hdpi</i>)
Launcher	36x36	48x48	72x72
Menu	36x36	48x48	72x72
Status Bar (Android 2.3 and later)	12w x 19h (preferred, width may vary)	16w x 25h (preferred, width may vary)	24w x 38h (preferred, width may vary)
Status Bar (Android 2.2 and below)	19x19	25x25	38x38
Tab	24x24	32x32	48x48
Dialog	24x24	32x32	48x48

List View	24x24	32x32	48x48
-----------	-------	-------	-------

Table 2: Standard icon sizes.

Let's examine the various icon categories in more detail.

Launcher Icons

Android icons can be any shape because they sit within a square bounding box. (Figure 2 shows some typical examples.) The reason that Android icons can be practically any shape has to do with the fact that they fall on a grid. The box for square icons is smaller than for other icons. This was done to establish a consistent visual weight across the two types.



Figure 2: Standard Android icons.

For each icon, there are bounding boxes for the full asset (red, in the following diagram), a smaller one for the actual icon (blue), and a smaller one yet (orange), which is the recommended size. To see how these look at the various densities, examine Figure 3.

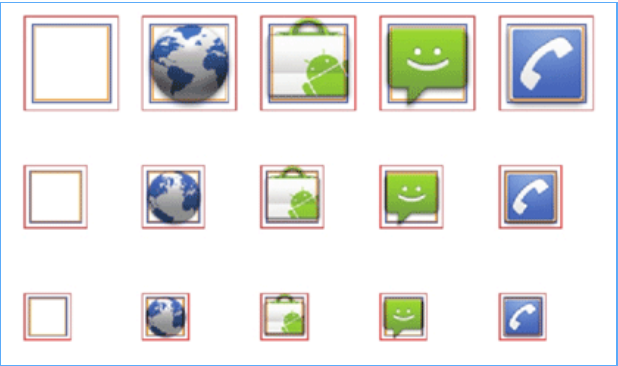


Figure 3: The various bounding boxes and how they fit icons.

The actual icon dimensions for the various bounding boxes are shown in Table 3.

Launcher icon dimensions for high-density (hdpi) screens:	Launcher icon dimensions for medium-density (mdpi) screens:	Launcher icon dimensions for low-density (ldpi) screens:
Full asset with bounding box: 72 x 72 pixels	Full asset with bounding box: 48 x 48 pixels	Full asset with bounding box: 36 x 36 pixels
Icon: 60 x 60 pixels	Icon: 40 x 40 pixels	Icon: 30 x 30 pixels
Square Icon: 56 x 56 pixels	Square Icon: 38 x 38 pixels	Square Icon: 28 x 28 pixels

Table 3: Icon dimensions.

Menu Icons

The menu icons (Figure 4) are used in the option menu, accessible by pushing the menu button. Since the icon will need to be draw9patched, it should be saved as a transparent PNG file.



Figure 4: Typical menu icons.

The recommended sizes for these icons is shown in Table 4.

Menu icon dimensions for high-density (hdpi) screens:	Menu icon dimensions for medium-density (mdpi) screens:	Menu icon dimensions for low-density (ldpi) screens:
Full asset: 72 x 72 pixels	Full asset: 48 x 48 pixels	Full asset: 36 x 36 pixels
Icon: 48 x 48pixels	Icon: 32 x 32 pixels	Icon: 24 x 24 pixels
Square Icon: 30 x 30 pixels	Square Icon: 38 x 38 pixels	Square Icon: 22 x 22 pixels

Table 4: Recommended sizes.

Tab Icons

Tab icons (Figure 5) are a lot like menu icons, except that two assets need to be created for the tab bar so as to differentiate between active and inactive icons. The recommended sizes are slightly different as well.



Figure 5: Tab icons.

The recommended sizes for tab icons are shown in Table 5.

Menu icon dimensions for high-density (hdpi) screens:	Menu icon dimensions for medium-density (mdpi) screens:	Menu icon dimensions for low-density (ldpi) screens:
Full asset: 48 x 48 pixels	Full asset: 32 x 32 pixels	Full asset: 24 x 24
Icon: 42 x 42 pixels	Icon: 28 x 28 pixels	Icon: 22 x 22 pixels

Table 5: Recommendations for tab icons.

The Tab Bar

Despite considerable searching, I have not found a pixel size for the tabs in a tab bar. (Figure 6)



Figure 6: Typical tabs for an Android app.

After measuring a sampling of screens, our team created these recommendation:

- HDPI: 480x96
- MDPI: 320x64
- LDPI: 240x48

Status Bar Icons

Status Bar icons (Figure 7) represent notifications from your app.



Figure 7: Status bar icons.

Status bar icons have changed since Android v. 2.3 (Gingerbread), so it is important to create assets, not just for the different densities, but also for older versions of the operating system. Table 6 contains the recommended sizes for these icons.

Menu icon dimensions for high-density (hdpi) screens:	Menu icon dimensions for medium-density (mdpi) screens:	Menu icon dimensions for low-density (ldpi) screens:
Full asset: 24 x 38 pixels	Full asset: 16 x 25 pixels	Full asset: 12 x 19
Icon: 24 x 24 pixels	Icon: 16 x 16 pixels	Icon: 12 x 12 pixels

Table 6: Recommendations for status bar icons.

Dialog Icon

Unlike the iPhone alert, the Android dialog is customizable. Below are the guidelines for building dialog icons. In addition, dialog icon assets use a safe frame to keep them aligned rather than employing bounding boxes. The safe frame is 1-pixel wide. (Note: Status bar icons for API 2.2 and earlier also rely on a simple safe frame, as it is recommended that the icons maintain a square shape.)

- HDPI: 48 x 48 pixels
- MDPI: 32 x 32 pixels
- LDPI: 24 x 24 pixels

The Options Menu

The option menu (Figure 8) stores activities. From it, you do such things as toggle settings, save data, logout, and so forth.

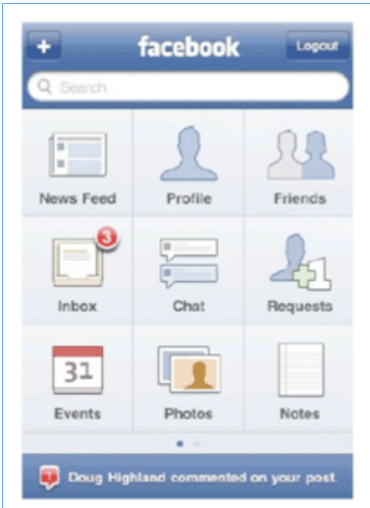


Figure 8: A typical options menu..

The width is adjustable based on the number of buttons and the size of the screen, and although it is customizable with skins, the size does not change.

- HDPI - 100x
- MDPI - 66x
- LDPI - 50x

The Context Menu

The Context menu is similar to a "right click" on a desktop. The user will touch and hold to bring up the menu, which will provide commands that pertain to the selected activity. Context menus are customizable, but design and development should coordinate on the cost/value of customization.

While the width of the cells are adjustable, the standard heights are :

- HDPI: 100 pixels

- MDPI: 66 pixels
- LDPI: 50 pixels

Summary

Designing an Android app requires constant collaboration between design and development. Because many assets can be created with XML, the design responsibility will fall to both parties. With a common set of guidelines, the dialog between design and development is greatly facilitated.

References

[Android Developer Resources](#).

— Adam Beckley works on Android applications at Mutual Mobile, a mobile strategy company based in Austin, Texas. Some figures and some table data for this article were taken from the Android's own developer guidelines.

IT PRO'S GUIDE:
iPhone 5 and iOS 6
What enterprise IT needs to know about the new hardware package and updated OS.

InformationWeek
IT Pro's Guide to
iPhone 5 and iOS 6

InformationWeek
reports

Download Now

[Terms of Service](#) | [Privacy Statement](#) | [Copyright © 2012 UBM Tech. All rights reserved.](#)