Icon Design Guidelines

Creating a unified look and feel throughout a user interface adds value to your product. Streamlining the graphic style will also make the UI seem more professional to users.

This document provides information to help you create icons for various parts of your application's user interface that match the general styles used by the Android 2.x framework. Following these guidelines will help you to create a polished and unified experience for the user.

The following documents discuss detailed guidelines for the common types of icons used throughout Android applications:

Launcher Icons
: A Launcher icon is a graphic that represents your application on the device's Home screen and in the Launcher window.

Menu Icons
: Menu icons are graphical elements placed in the options menu shown to users when they press the Menu button.

Status Bar Icons
: Status bar icons are used to represent notifications from your application in the status bar.

Tab Icons
: Tab icons are graphical elements used to represent individual tabs in a multi-tab interface.

Dialog Icons
: Dialog icons are shown in pop-up dialog boxes that prompt the user for interaction.

List View Icons
: List view icons are used with `ListView` to graphically represent list items. An example is the Settings application.

To get started creating your icons more quickly, you can download the Android Icon Templates Pack.

Quickview

- You can use several types of icons in an Android application.
- Your icons should follow the general specification in this document.
- You should create separate icon sets for high-, medium-, and low-density screens.

In this document

Topics

Launcher Icons
Menu Icons
Status Bar Icons
Tab Icons
Dialog Icons
List View Icons

Downloads

Android Icon Templates Pack, v2.3 »
Android Icon Templates Pack, v2.0 »
Android Icon Templates Pack, v1.0 »

See also

Supporting Multiple Screens

Using the Android Icon Templates Pack

The Android Icon Templates Pack is a collection of template designs, textures, and layer styles that make it easier for you to create icons that conform to the guidelines given in this document. We recommend downloading the template pack archive before you start designing your icons.

The icon templates are provided in the Adobe Photoshop file format (.psd), which preserves the layers and design treatments we used when creating the standard icons for the Android platform. You can load the template files into any compatible image-editing program, although your ability to work directly with the layers and treatments may vary based on the program you are using.

You can obtain the latest Icon Templates Pack archive using the link below:

Download the Icon Templates Pack for Android 2.3 »

For previous versions of the Icon Templates Pack, see the *Downloads* section in the box at the top-right corner of this page.

Providing Density-Specific Icon Sets

Android is designed to run on a variety of devices that offer a range of screen sizes and resolutions. When you design the icons for your application, it's important keep in mind that your application may be installed on any of those devices. As described in the Supporting Multiple Screens document, the Android platform makes it straightforward for you to provide icons in such a way that they will be displayed properly on any device, regardless of the device's screen size or resolution.

In general, the recommended approach is to create a separate set of icons for each of the three generalized screen densities listed in Table 1. Then, store them in density-specific resource directories in your application. When your application runs, the Android platform will check the characteristics of the device screen and load icons from the appropriate density-specific resources. For more information about how to store density-specific resources in your application, see Resource directory qualifiers for screen size and density.

For tips on how to create and manage icon sets for multiple densities, see Tips for Designers.

Table 1. Summary of finished icon dimensions for each of the three generalized screen densities, by icon type.

Icon Type	**Standard Asset Sizes (in Pixels), for Generalized Screen Densities**		
	Low density screen *(ldpi)*	Medium density screen *(mdpi)*	High density screen *(hdpi)*
Launcher	36 x 36 px	48 x 48 px	72 x 72 px
Menu	36 x 36 px	48 x 48 px	72 x 72 px
Status Bar (Android 2.3 and later)	12w x 19h px (preferred, width may vary)	16w x 25h px (preferred, width may vary)	24w x 38h px (preferred, width may vary)
Status Bar (Android 2.2 and below)	19 x 19 px	25 x 25 px	38 x 38 px
Tab	24 x 24 px	32 x 32 px	48 x 48 px
Dialog	24 x 24 px	32 x 32 px	48 x 48 px
List View	24 x 24 px	32 x 32 px	48 x 48 px

Tips for Designers

Here are some tips that you might find useful as you develop icons or other drawable assets for your application. The tips assume that you are using Adobe Photoshop or a similar raster and vector image-editing program.

Use common naming conventions for icon assets

Try to name files so that related assets will group together inside a directory when they are sorted alphabetically. In particular, it helps to use a common prefix for each icon type. For example:

Asset Type	**Prefix**	**Example**
Icons	`ic_`	`ic_star.png`

Launcher icons	`ic_launcher`	`ic_launcher_calendar.png`
Menu icons	`ic_menu`	`ic_menu_archive.png`
Status bar icons	`ic_stat_notify`	`ic_stat_notify_msg.png`
Tab icons	`ic_tab`	`ic_tab_recent.png`
Dialog icons	`ic_dialog`	`ic_dialog_info.png`

Note that you are not required to use a shared prefix of any type — doing so is for your convenience only.

Set up a working space that organizes files for multiple densities

Supporting multiple screen densities means you must create multiple versions of the same icon. To help keep the multiple copies of files safe and easier to find, we recommend creating a directory structure in your working space that organizes asset files per resolution. For example:

```
assets/...
    ldpi/...
        _pre_production/...
            working_file.psd
        finished_asset.png
    mdpi/...
        _pre_production/...
            working_file.psd
        finished_asset.png
    hdpi/...
        _pre_production/...
            working_file.psd
        finished_asset.png
```

This structure parallels the density-specific structure in which you will ultimately store the finished assets in your application's resources. Because the structure in your working space is similar to that of the application, you can quickly determine which assets should be copied to each application resources directory. Separating assets by density also helps you detect any variances in filenames across densities, which is important because corresponding assets for different densities must share the same filename.

For comparison, here's the resources directory structure of a typical application:

```
res/...
    drawable-ldpi/...
        finished_asset.png
    drawable-mdpi/...
        finished_asset.png
    drawable-hdpi/...
        finished_asset.png
```

Use vector shapes where possible

Many image-editing programs such as Adobe Photoshop allow you to use a combination of vector shapes and raster layers and effects. When possible, use vector shapes so that if the need arises, assets can be scaled up without loss of detail and edge crispness.

Using vectors also makes it easy to align edges and corners to pixel boundaries at smaller resolutions.

Start with large artboards

Because you will need to create assets for different screen densities, as shown in Table 1, it is best to start your icon designs on large artboards with dimensions that are multiples of the target icon sizes. For example, launcher icons are 72, 48, or 36 pixels wide, depending on screen density. If you initially draw launcher icons on an 864x864 artboard, it will be easier and cleaner to tweak the icons when you scale the artboard down to the target sizes for final asset creation.

It's also beneficial to add guide lines (also known as guides) to your large artboard for the recommended safe margins at the highest target density. Continuing with the example above, per the guidelines, launcher icon content should be 60x60 pixels (56x56 for square icons) within the full 72x72 asset, or a safe margin of 6 pixels on each side. On an 864x864 artboard, this corresponds to horizontal and vertical guide lines 72 pixels from each side of the artboard.

When scaling, redraw bitmap layers as needed

If you scaled an image up from a bitmap layer, rather than from a vector layer, those layers will need to be redrawn manually to appear crisp at higher densities. For example if a 60x60 circle was painted as a bitmap for `mdpi` it will need to be repainted as a 90x90 circle for `hdpi`.

When saving image assets, remove unnecessary metadata

To help keep each image asset as small as possible, make sure to remove any unnecessary headers from the file, such as Adobe Fireworks metadata or Adobe Photoshop headers. To remove the Photoshop header, follow these steps:

1. Under the **File** menu, choose the **Save for Web & Devices** command
2. On the "Save for Web & Devices" panel, set the Preset pop-up to "PNG-24," set the pop-up under Presets to "PNG-24" as well, and select the Transparency box (if the image uses transparency)
3. Select **Save**.

It is also useful to use PNG file size optimization tools such as OptiPNG or Pngcrush.

Make sure that corresponding assets for different densities use the same filenames

Corresponding icon asset files for each density **must use the same filename**, but be stored in density-specific resource directories. This allows the system to look up and load the proper resource according to the screen characteristics of the device. For this reason, make sure that the set of assets in each directory is consistent and that the files do not use density-specific suffixes.

For more information about density-specific resources and how the system uses them to meet the needs of different devices, see Supporting Multiple Screens.

← Back to UI Guidelines ↑ Go to top