Resources Overview

You should always externalize resources such as images and strings from your application code, so that you can maintain them independently. Externalizing your resources also allows you to provide alternative resources that support specific device configurations such as different languages or screen sizes, which becomes increasingly important as more Android-powered devices become available with different configurations. In order to provide compatibility with different configurations, you must organize resources in your project's `res/` directory, using various sub-directories that group resources by type and configuration.

Figure 1. Two different devices, each using the default layout (the app provides no alternative layouts).

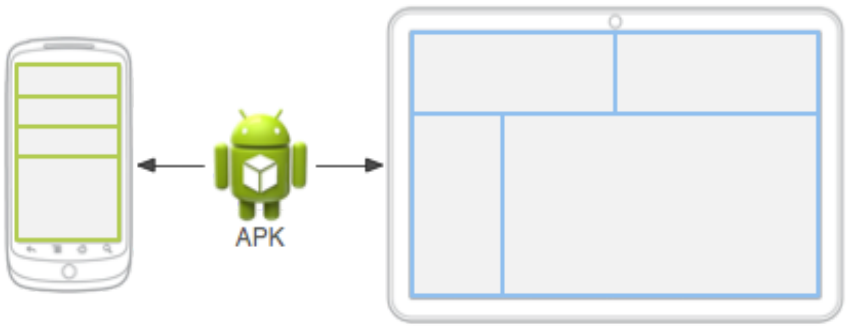

Figure 2. Two different devices, each using a different layout provided for different screen sizes.

For any type of resource, you can specify *default* and multiple *alternative* resources for your application:

- Default resources are those that should be used regardless of the device configuration or when there are no alternative resources that match the current configuration.
- Alternative resources are those that you've designed for use with a specific configuration. To specify that a group of resources are for a specific configuration, append an appropriate configuration qualifier to the directory name.

For example, while your default UI layout is saved in the `res/layout/` directory, you might specify a different layout to be used when the screen is in landscape orientation, by saving it in the `res/layout-land/` directory. Android

automatically applies the appropriate resources by matching the device's current configuration to your resource directory names.

Figure 1 illustrates how the system applies the same layout for two different devices when there are no alternative resources available. Figure 2 shows the same application when it adds an alternative layout resource for larger screens.

The following documents provide a complete guide to how you can organize your application resources, specify alternative resources, access them in your application, and more:

Providing Resources **(https://developer.android.com/guide/topics/resources/providing-resources.html)**

> What kinds of resources you can provide in your app, where to save them, and how to create alternative resources for specific device configurations.

Accessing Resources **(https://developer.android.com/guide/topics/resources/accessing-resources.html)**

> How to use the resources you've provided, either by referencing them from your application code or from other XML resources.

Handling Runtime Changes **(https://developer.android.com/guide/topics/resources/runtime-changes.html)**

> How to manage configuration changes that occur while your Activity is running.

Localization **(https://developer.android.com/guide/topics/resources/localization.html)**

> A bottom-up guide to localizing your application using alternative resources. While this is just one specific use of alternative resources, it is very important in order to reach more users.

Complex XML Resources **(https://developer.android.com/guide/topics/resources/complex-xml-resources.html)**

> An XML format for building complex resources like animated vector drawables in a single XML file.

Resource Types **(https://developer.android.com/guide/topics/resources/available-resources.html)**

> A reference of various resource types you can provide, describing their XML elements, attributes, and syntax. For example, this reference shows you how to create a resource for application menus, drawables, animations, and more.