| Directory | Resource Type |
| --- | --- |
| `animator/` | XML files that define property animations. |
| `anim/` | XML files that define tween animations. (Property animations can also be saved in this directory, but the `animator/` directory is preferred for property animations to distinguish between the two types.) |
| `color/` | XML files that define a state list of colors. See Color State List Resource |
| `drawable/` | Bitmap files (`.png`, `.9.png`, `.jpg`, `.gif`) or XML files that are compiled into the following drawable resource subtypes:<br>• Bitmap files<br>• Nine-Patches (re-sizable bitmaps)<br>• State lists<br>• Shapes<br>• Animation drawables<br>• Other drawables<br>See Drawable Resources. |
| `mipmap/` | Drawable files for different launcher icon densities. For more information on managing launcher icons with `mipmap/` folders, see Managing Projects Overview. |
| `layout/` | XML files that define a user interface layout. See Layout Resource. |
| `menu/` | XML files that define application menus, such as an Options Menu, Context Menu, or Sub Menu. See Menu Resource. |
| `raw/` | Arbitrary files to save in their raw form. To open these resources with a raw `InputStream`, call `Resources.openRawResource()` with the resource ID, which is `R.raw.`*`filename`*.<br>However, if you need access to original file names and file hierarchy, you might consider saving some resources in the `assets/` directory (instead of `res/raw/`). Files in `assets/` are not given a resource ID, so you can read them only using `AssetManager`. |
| `values/` | XML files that contain simple values, such as strings, integers, and colors.<br>Whereas XML resource files in other `res/` subdirectories define a single resource based on the XML filename, files in the `values/` directory describe multiple resources. For a file in this directory, each child of the `<resources>` element defines a single resource. For example, a `<string>` element creates an `R.string` resource and a `<color>` element creates an `R.color` resource.<br>Because each resource is defined with its own XML element, you can name the file whatever you want and place different resource types in one file. However, for clarity, you might want to place unique resource types in different files. For example, here are some filename conventions for resources you can create in this directory: |

- arrays.xml for resource arrays (typed arrays).
- colors.xml for color values
- dimens.xml for dimension values.
- strings.xml for string values.
- styles.xml for styles.

See String Resources, Style Resource, and More Resource Types.

`xml/` Arbitrary XML files that can be read at runtime by calling `Resources.getXML()`. Various XML configuration files must be saved here, such as a searchable configuration.