

1. 以 HelloWorld 为例

AndroidManifest.xml: 描述 Android 应用的信息, 包括类名, 组件名等

assets: 资源目录

lib: so 文件存放路径

java: 源代码存放目录

- example: 类名

- hello: 类名

- helloStr.java: 类实现文件

- MainActivity.java: 类实现文件

res: 资源文件

- drawable: 图片文件存放路径

- layout: 布局文件存放路径(窗体)

- activity_main.xml: 布局文件

- content_main.xml: 布局文件

- menu: 菜单文件存放路径

- menu_main.xml: 菜单文件

- mipmap-hdpi: hdpi 的贴图

- ic_launcher.png

- mipmap-mdpi: mdpi 的贴图

- ic_launcher.png

- mipmap-xhdpi: xhdpi 的贴图

- ic_launcher.png

- mipmap-xxhdpi: xxhdpi 的贴图

- ic_launcher.png

- mipmap-xxxhdpi: xxxhdpi 的贴图, 对应清晰度不一样的手机, 调用的贴图不一样

- ic_launcher.png

values: 属性资源

- colors.xml: 颜色资源

- dimens.xml: 尺寸资源

- strings.xml: 字符串资源

- styles.xml: 样式资源

values-v21: 同上, 对应 api21

- styles.xml

values-w820dp: 同上, 对应 w820dp 的资源

- dimens.xml

Android 程序在搜索资源的时候, 是按照手机的类型来进行一级一级搜索的。例如, 如果是中文系统, 会使用 values-ch 目录下的 strings, 如果找不到, 会找上一级 values 目录下的 strings。

2. R 类

在程序代码中, 并不能直接地引用资源文件, 需要使用 R 类充当连接代码和资源的桥梁, R 类可以在 build 下找到。

3. 反编译的目录

与编译前是高度相似的

AndroidManifest.xml

apktool.yml

t.log

original

-AndroidManifest.xml

-META-INF

--CERT.RSA

--CERT.SF

--MANIFEST.MF

res

-anim

--abc_fade_in.xml

--abc_fade_out.xml

--abc_grow_fade_in_from_bottom.xml

...

Public.xml

Aapt 在编译资源的时候，会根据这个 xml 文件来规定每个资源的 id 值