

# Sodino's Blog

见多食广 与食巨近

[🏠 首页](#)[📁 归档](#)[👤 关于](#)[📡 RSS](#)

## 【Android】Gadle改变应用名称(appName)的两种方法

📅 Sep 1, 2016 | 📁 Android

### 应用场景

- App在debug/release编译环境下，想通过改变appName来给予区分。
- `productFlavors` 增加各种编译参数时，比如更换 `applicationId` 包名，appName也需要跟着变动。

### 方法一：设置manifestPlaceholders

```
android {  
    ...  
    defaultConfig{  
        ...  
        manifestPlaceholders = [k_appName : "哈啰"]    // 设置默认的k_appName  
    }  
  
    // 依据debug/release变动的话设置如下  
    buildTypes {  
        debug {  
            manifestPlaceholders = [k_appName : "Debug哈啰"]  
        }  
    }  
  
    // 依据flavors变动的话设置如下  
    productFlavors {  
        autoTest {  
            manifestPlaceholders = [k_appName : "AT哈啰"]  
        }  
  
        appStore {  
            // do nothing  
        }  
    }  
}
```

```
}  
}
```

对应的 `AndroidManifest.xml` 需要对 `android:label` 修改为：

```
<application  
    android:icon="@mipmap/ic_launcher"  
    android:label="${k_appName}" // 这里取k_appName  
>
```

---

## 方法二：定义resValue

---

```
android {  
    ...  
    defaultConfig {  
        ...  
        resValue "string", "app_name", "哈啰" // 设置默认的app_name  
    }  
  
    // 依据debug/release变动的话设置如下  
    buildTypes {  
        debug {  
            resValue "string", "app_name", "Debug哈啰"  
        }  
    }  
  
    // 依据flavors变动的话设置如下  
    productFlavors {  
        autoTest {  
            resValue "string", "app_name", "AT哈啰"  
        }  
  
        appStore {  
            // do nothing  
        }  
    }  
}
```

因为gradle会根据 `resValue` 的定义自动生成 `<string name="app_name">` 资源。所以需要将 `/res/values/strings.xml` 中默认生成的 `<string name="app_name">HelloWorld</string>` 删除，避免出现重复的string资源定义错误。

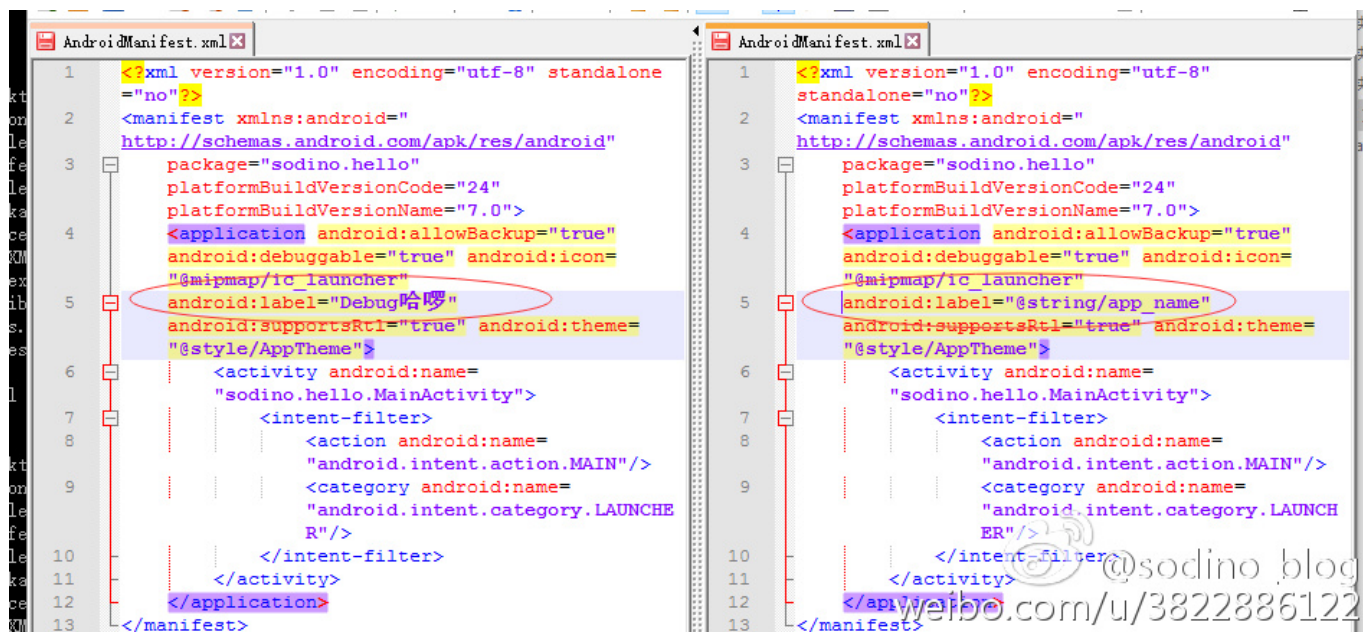
```
Error:Execution failed for task ':app:mergeAutoTestDebugResources'.  
> [string/app_name] D:\github\HelloWorld\app\src\main\res\values\strings.xml
```

[string/app\_name] D:\github\HelloWorld\app\build\generated\res\resValues\autoTes  
Error: Duplicate resources

## 两者比较

直接用APK反编译后的图来对比吧：

左边是 manifestPlaceholders 方法一的实现；右边是 resValue 定义方法二的实现



compare

可以看出， manifestPlaceholders 属于字符串直接替换赋值，当应用有 多语言 切换时，应用名是仍然会是固定的”Debug哈啰”，不会变化。

而 resValue 定义显得委婉一些，只要 res/values-en/strings.xml 中有 <string app\_name> 的外语定义，是会跟随系统的语言变化而切换app名称的。

## 后记

本文同时给出了 buildTypes 和 productFlavors 中的示例，但在实际处理中，请根据实际情况自己选择使用。

如果两者都声明了 app\_name，则最后以 buildTypes 的设定为准。

原因是Gradle的资源合并功能中在权重上：

BuildType > Flavor > main > Dependencies.

见下图，或[点击链接Resource Merging](#)

[Technical docs](#) > [New Build System](#) >

## Resource Merging

The Gradle-based build system uses a new merging mechanism for resources. In previous build system, merging was done by passing a list of resources in the overlays would be automatically added (default behavior is for overlays is to only override existing resources, not create new).

One of the goals of the Gradle-based build system was providing more flexibility, and one frequently asked feature request was the ability to have a merging mechanism that is run ahead of *aapt* and generates a single, merged, resources folder that is fed to *aapt*. This merging has the advantage of being incremental (ie it can rerun the merge by only applying the change in a single file).

The merged resources are coming from 3 types of sources:

- The main resources, associated with the *main* sourceSet, generally located in *src/main/res*
- The variant overlays, coming from the Build Type, and Flavor(s).
- The Library Project dependencies, which contribute resources through the *res* entry in their aar bundle.

### Priority Order

The priority order is the following:

**BuildType -> Flavor -> main -> Dependencies.**

This means that if a resource is declared in both the Build Type and in *main*, the one from Build Type will be selected.

Note that for the scope of the merging, resources of the same (type, name) but different qualifiers are handled separately.

This means that if *src/main/res* has

- *res/layout/foo.xml*
- *res/layout-land/foo.xml*

and *src/debug/res* has

- *res/layout/foo.xml*

 @sodino blog  
weibo.com/u/3822886122

resource.merging

### About Sodino

 Android

◀ [【Android】记录Android MediaPlayer的两个bug及其它特殊表现](#)

[【Linux】启动mongo db后台服务](#) ▶

