# 资源处理

# 本地或网络获取.9图片进行显示

NinePatch从外部获取时需要通过android自带工具aapt进行处理。

NinePatch预编译

使用Android sdk build-tools目录下的aapt对制作出来的NinePatch图进行转化，切换到具体资源的目录下：

D:\NinePatch

运行以下命令：

D:\android studio\sdk\build-tools\31.0.0\aapt s -i ic\_chat.9.png –o ic\_chat\_out.9.png

上述命令应该是android studio存在控制，会存在问题。

所以比上面的地址配置到系统的path路径中。

D:\android studio\sdk\build-tools\31.0.0\

重新输入一下命令

aapt s -i ic\_chat.9.png -o ic\_chat\_out.9.png

注意上面的命令需要定位到具体资源的目录，否则会提示文件找不到：

Crunching single PNG file: ic\_chat.9.png  
 Output file: ic\_chat\_out.9.png  
 ic\_chat.9.png ERROR: Unable to open PNG file

批量图片输出

使用sdk自带的工具aapt对.9图重新输出

aapt c -v -S point\_9 -C point\_10

以上的是批量输出的指令。point\_9是要转换的.9图的文件夹，point\_10是转换输出的文件。在运行指令是point\_10要提前创建，不然会报错。详细的aapt指令自行去查找，这里不做详细介绍。

aapt c -v -S test1 -C test2

上面这个方式有问题：

转换成NinePatchDrawable的时候需要设置一下padding，不设置的话内容区域的信息会丢失，但是拉伸区域不会。

具体可以看demo

有两种方式实现：

1. 使用aapt编译.9图，使用工具类设置一下padding
2. 使用原始的.9图，使用工具类重新生成.9图片

# 问题

使用本地加载.9的方案，有一部分手机会失效，经过排查，是因为使用autosize的一个框架，修改下面的设置页可以实现

<meta-data  
 android:name="design\_width\_in\_dp"  
 android:value="375" />

# 参考资料

NinePatchDrawable详解

<https://modun.blog.csdn.net/article/details/50513187?spm=1001.2101.3001.6661.1&utm_medium=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-1-50513187-blog-51303439.235%5Ev29%5Epc_relevant_default_base3&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-1-50513187-blog-51303439.235%5Ev29%5Epc_relevant_default_base3&utm_relevant_index=1>

在demo中有一个NinePatchChunk的工具类就是从上面的连接中介绍的

<https://stackoverflow.com/questions/11065996/ninepatchdrawable-does-not-get-padding-from-chunk>

[Android][动态使用点9图NinePatchDrawable]

<https://www.jianshu.com/p/b3f5c1a45656>

demo中的ninepatch这个包中的就是上面的连接介绍的，它是一个第三方开源框架

<https://github.com/Anatolii/NinePatchChunk>

项目中使用的是 [Android][动态使用点9图NinePatchDrawable]里面的测试demo

<https://download.csdn.net/download/lgywsdy/11200737>

式样

微软雅黑 Light 5号

段落 固定值 23磅

# 目录1

## 目录2

### 目录3

#### 目录4

##### 目录5