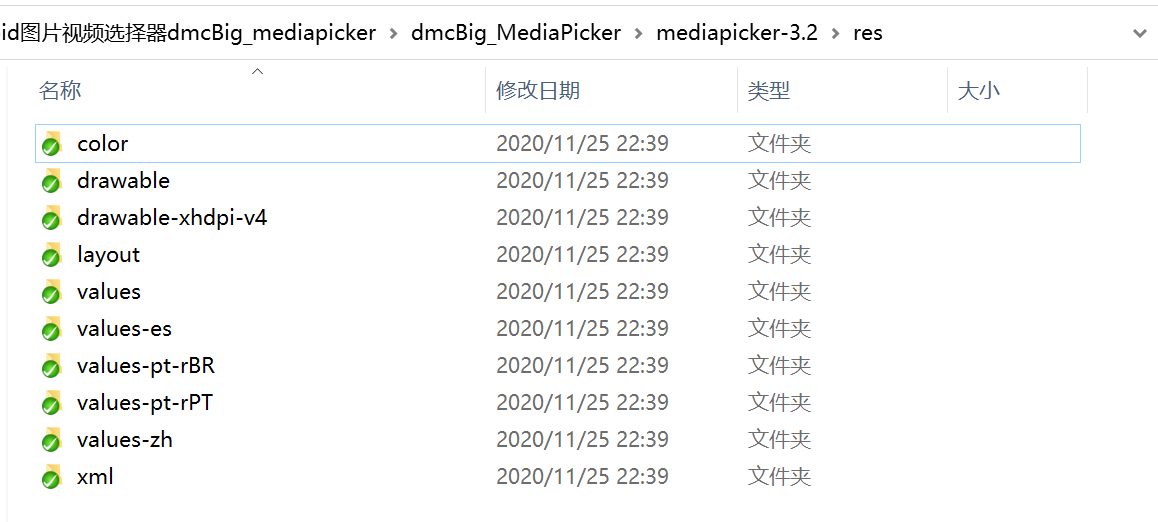
**三方SDK智能布署工具的功能介绍（支持中英文）**

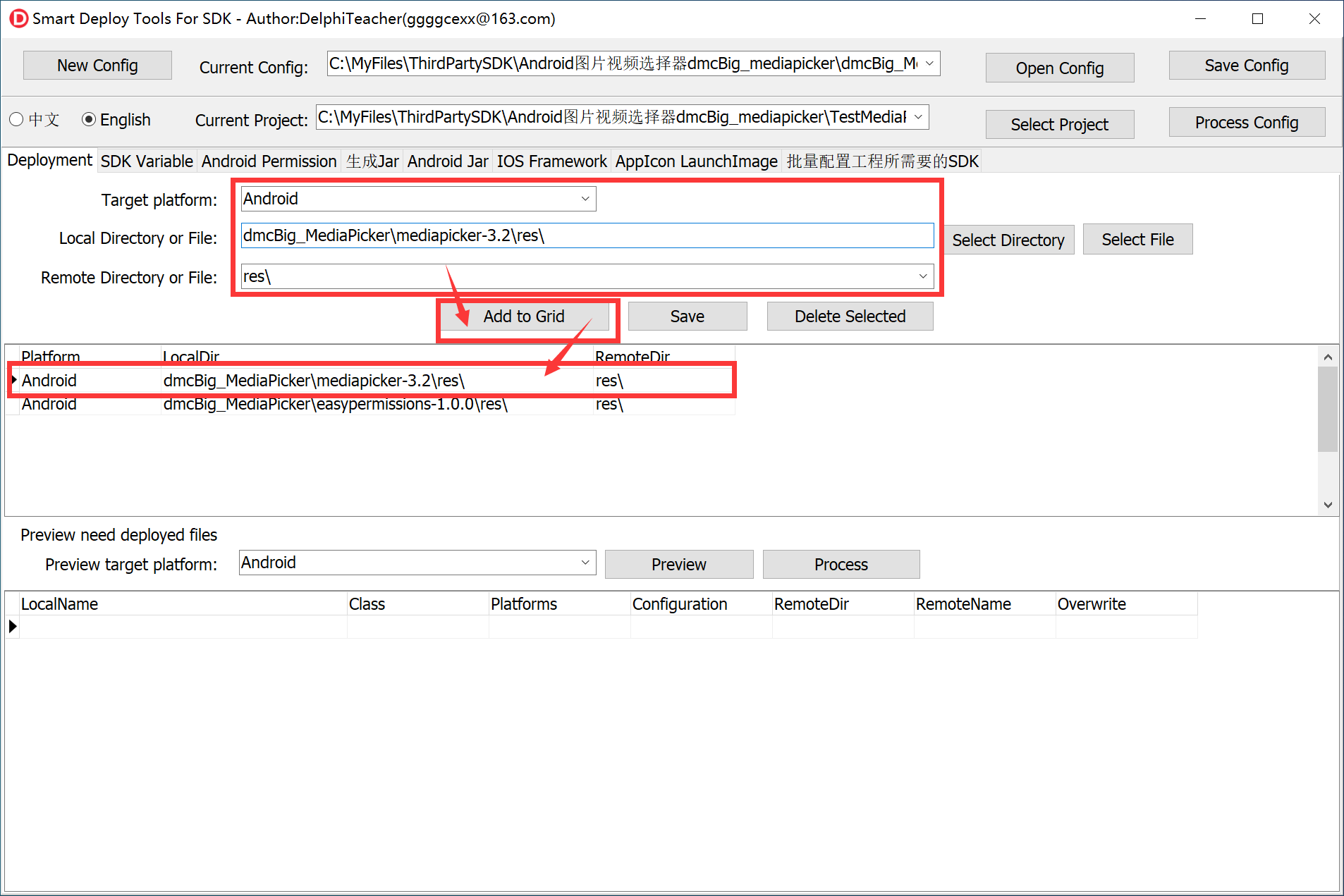
因为我在开发APP的过程中经常需要对接一些三方的SDK,为了提升工作的效率，我开发了这个三方SDK智能布署工具。

1. 布署文件更加方便
   1. 能直接布署文件夹，而不用一个个选择里面的子文件夹和文件。

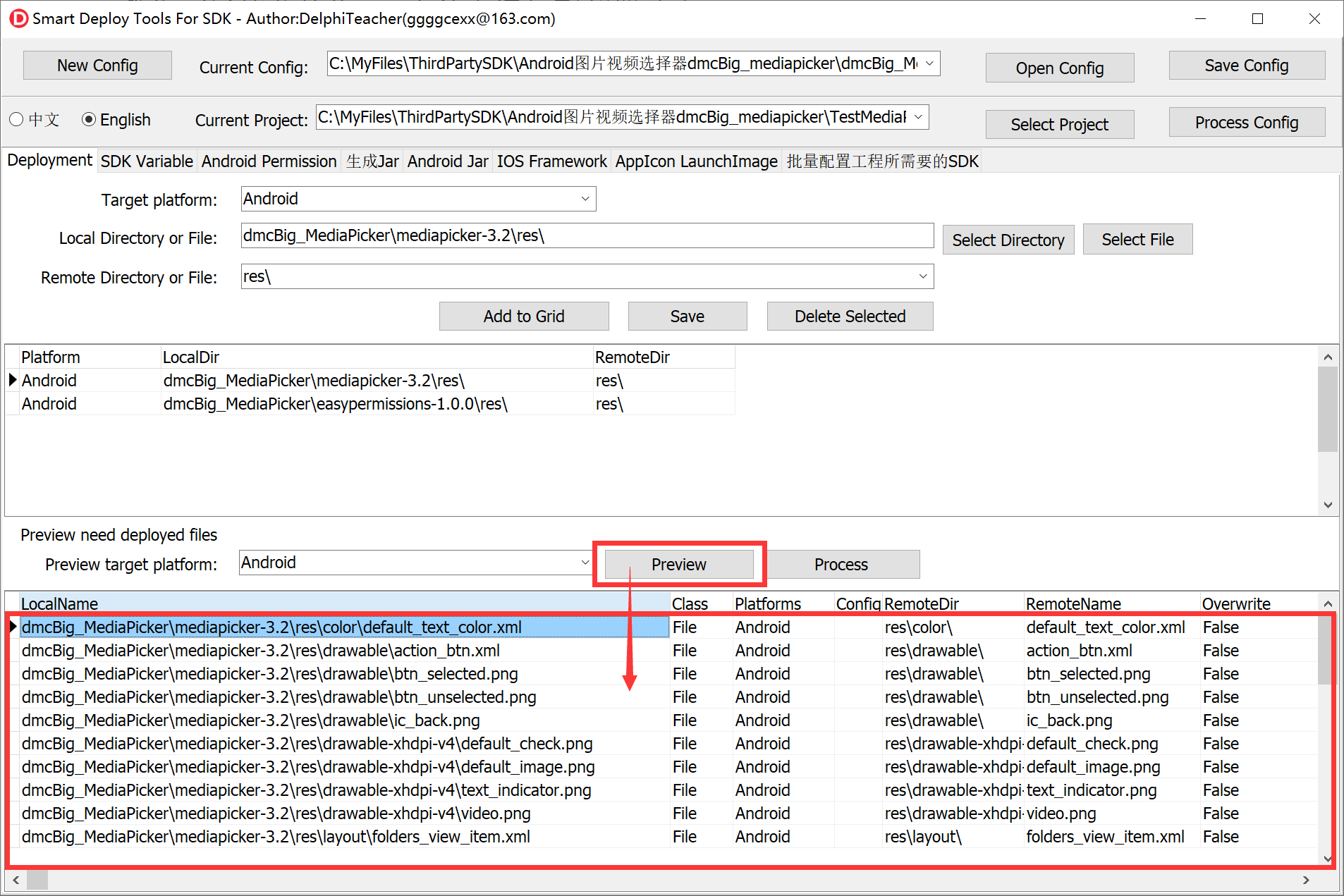
比如我使用的多选图片的Android三方SDK，它里面有一个res文件夹，这个文件夹里面的所有子文件都要布署。



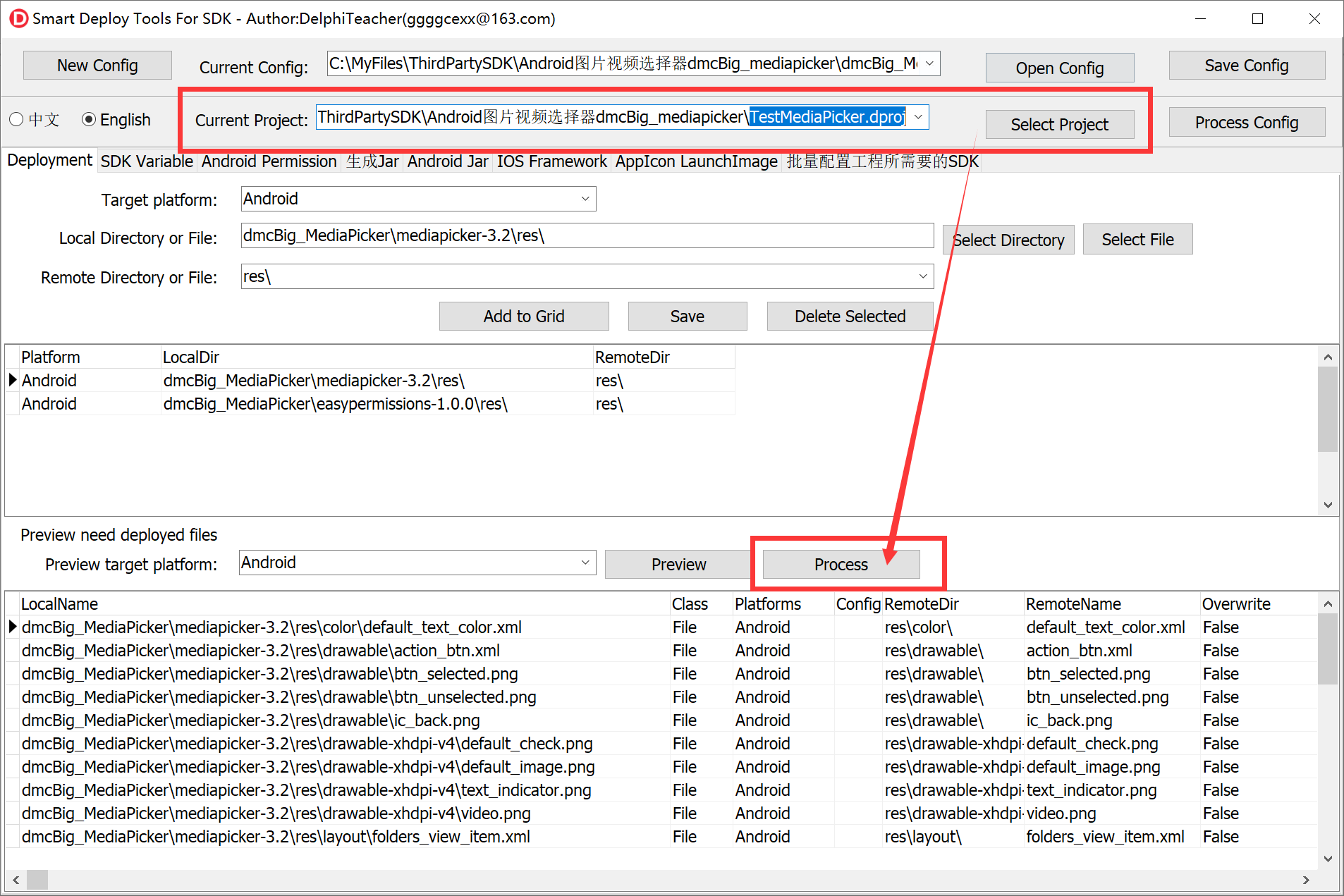
这个工具可以简单的将res文件夹添加到部署项中，



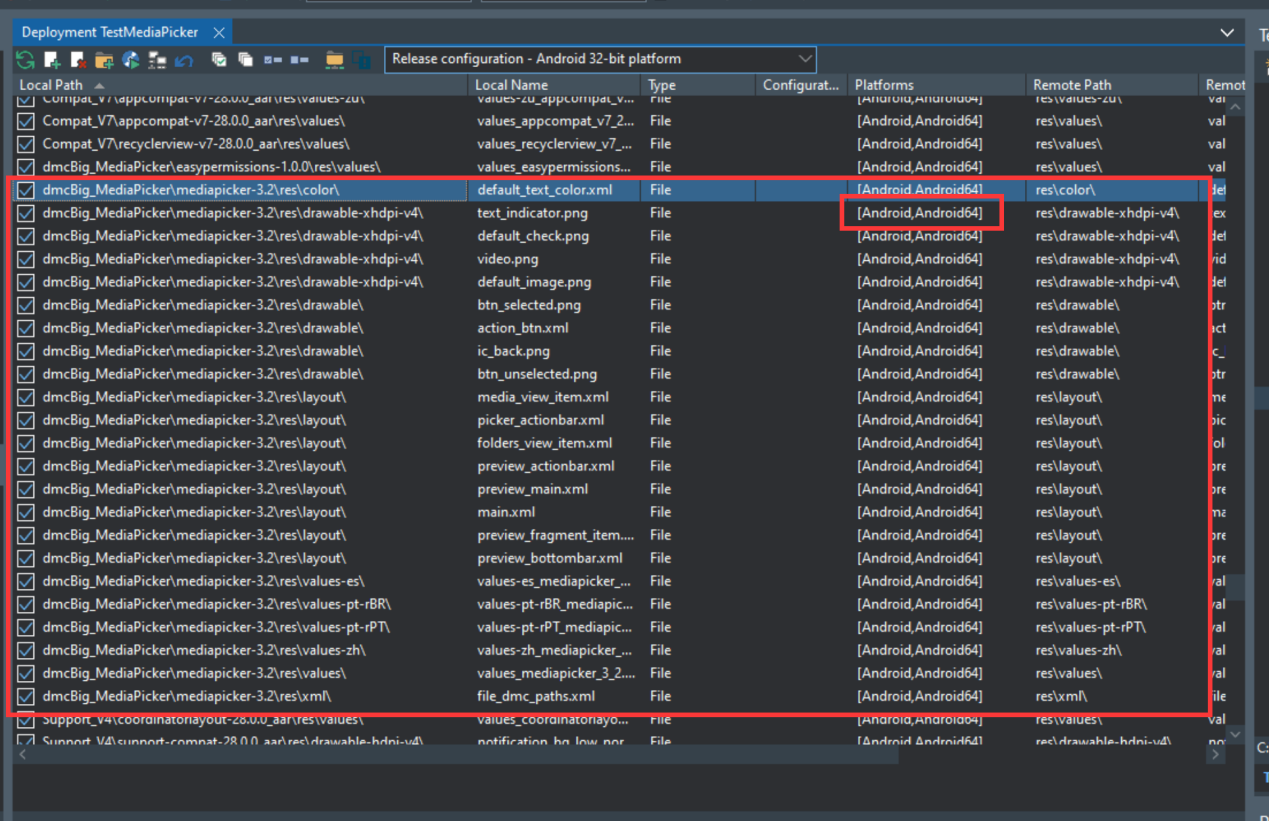
之后点击“Preview”按钮即可列出需要布署的所有文件



选择要布署到的工程，再点击“Process”即可完成布署。

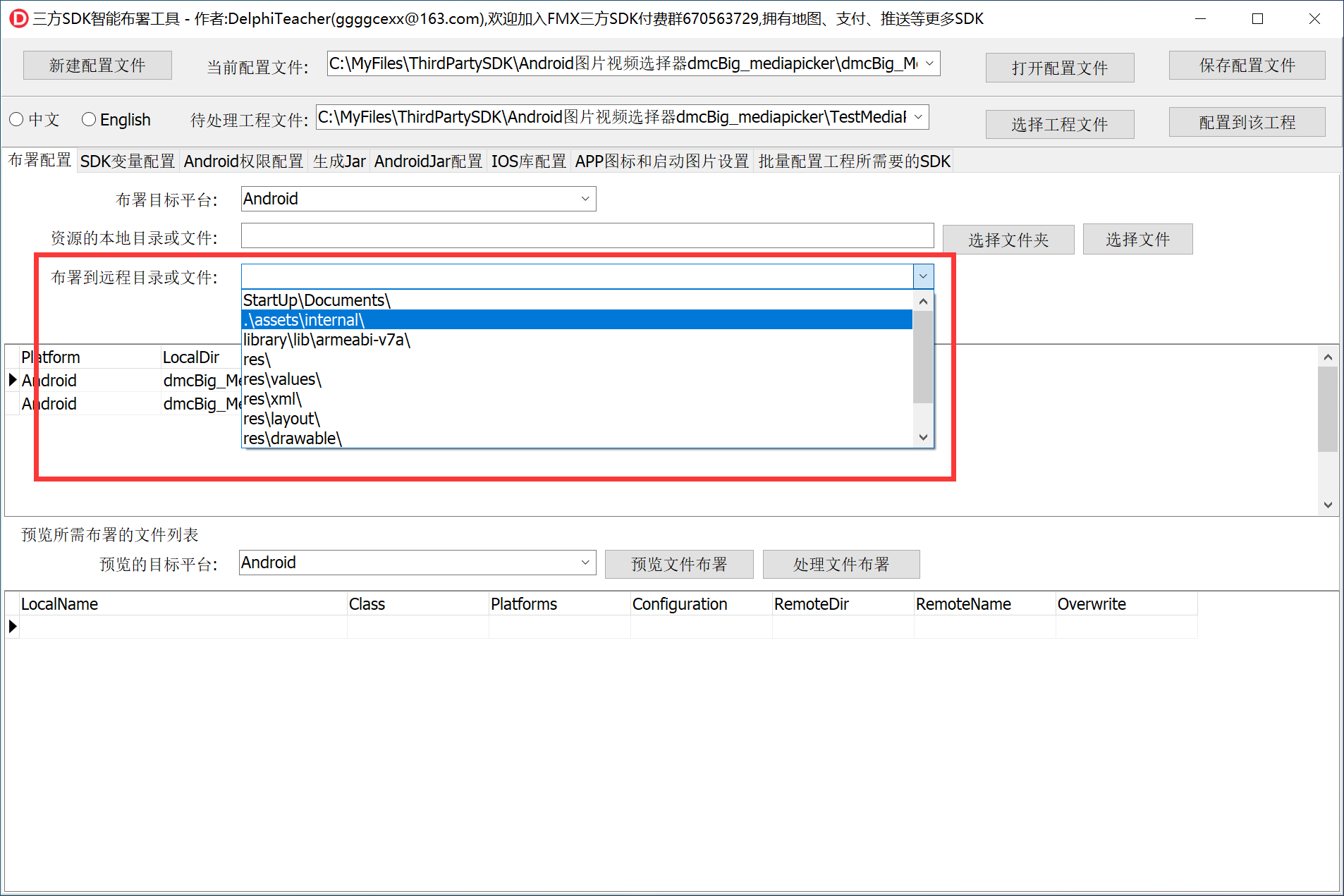


你们看，res文件夹中的文件布署进去了，并且同时布署到了Android和Android64这两个平台。

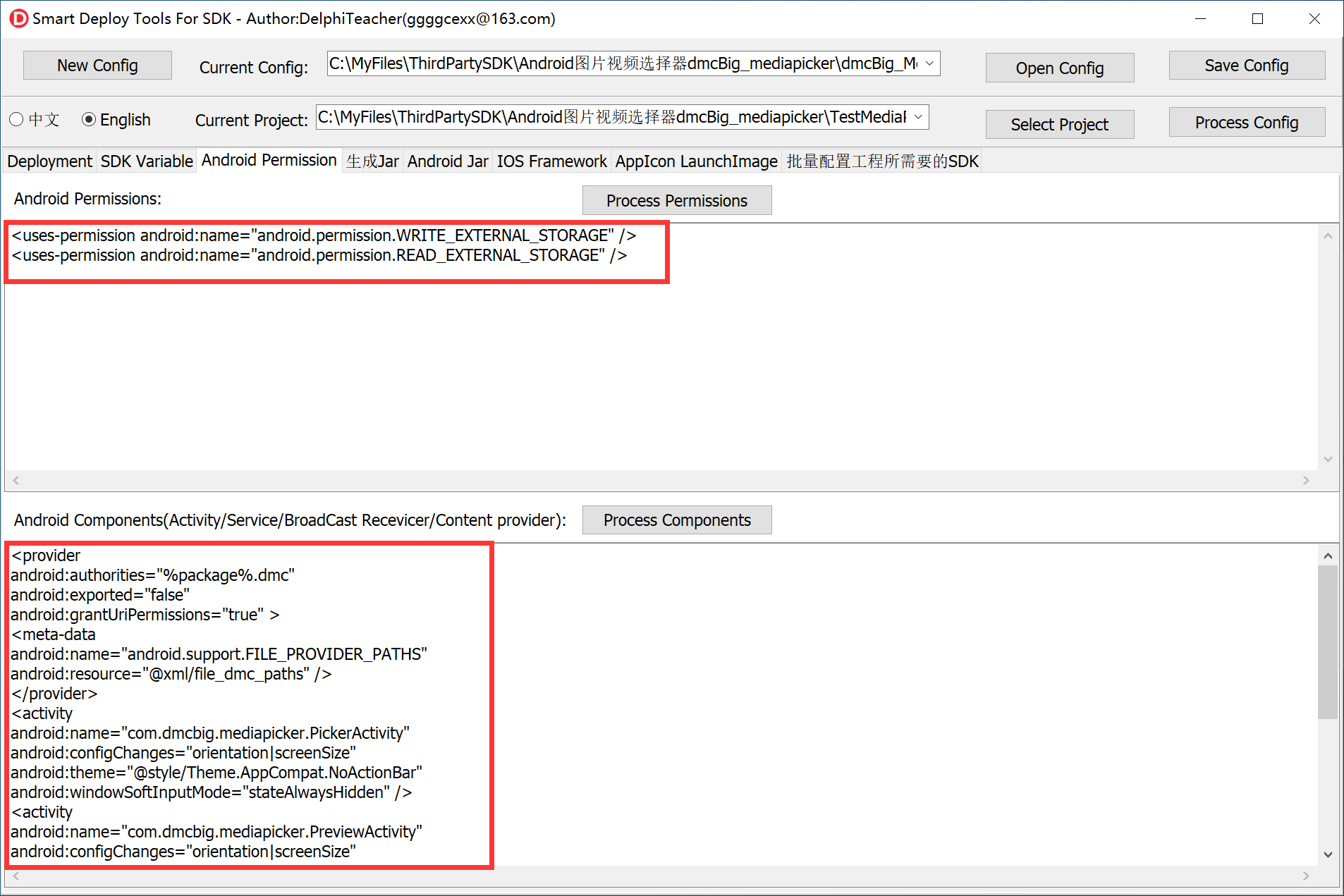


* 1. 布署文件时可以直接选择常用的远程目录RemotePath。

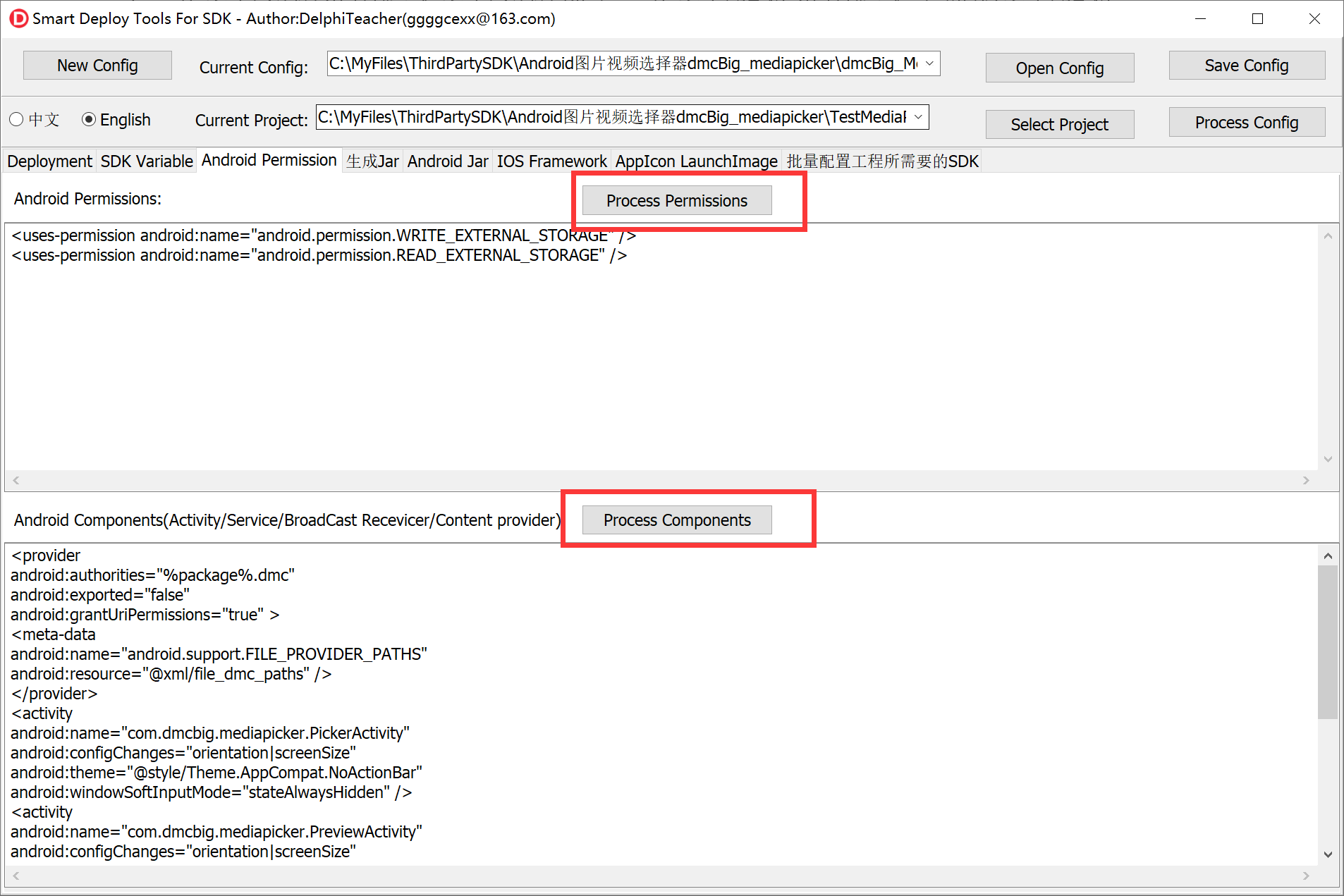
比如StartUp\Documents\、.\assets\internal\、library\lib\armeabi-v7a\等。



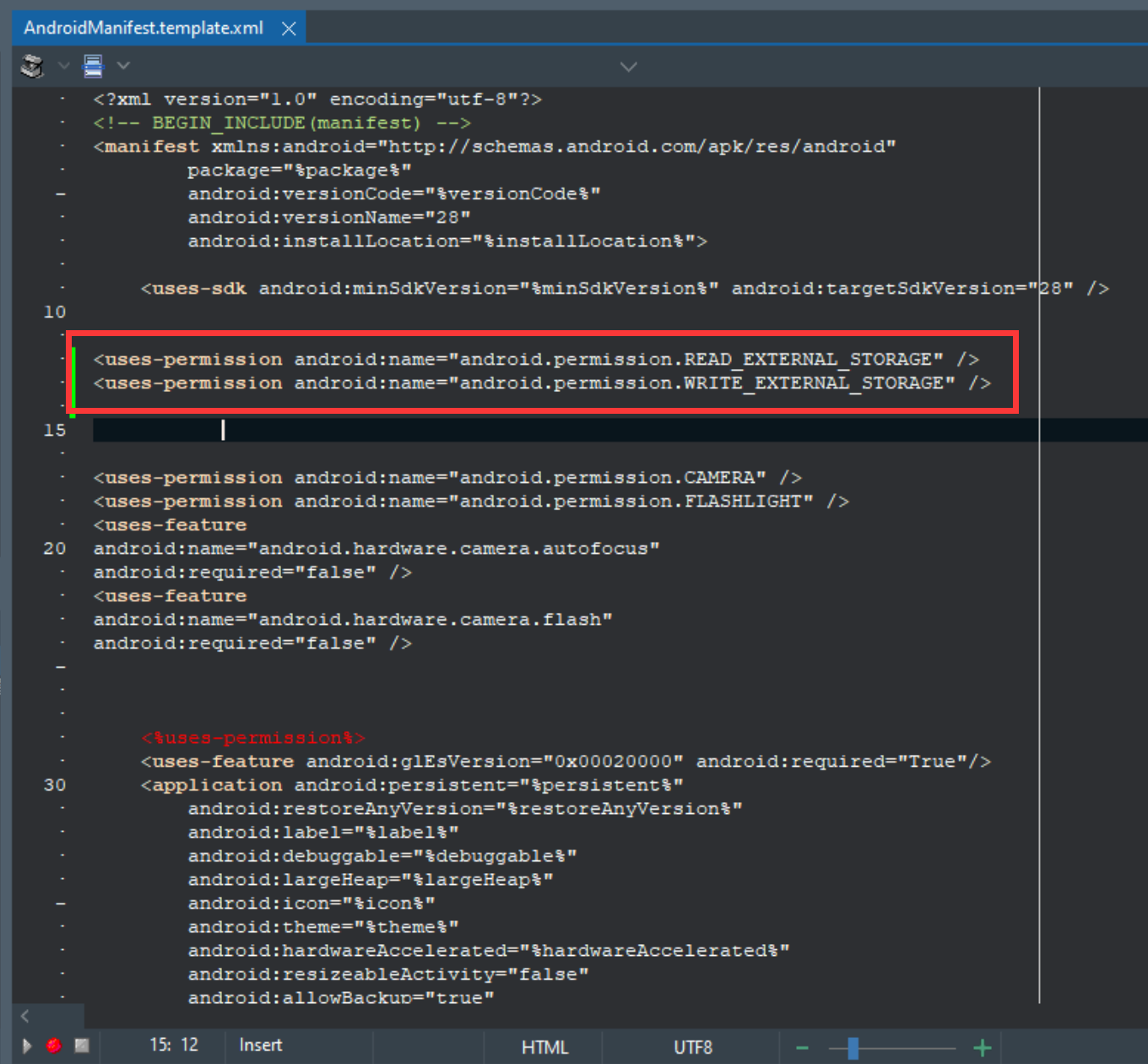
1. 能够直接将SDK所需的权限和Android元件（比如activity、service、receiver、provider等）配置到工程的AndroidManifest.template.xml。
   1. 在安卓权限框中输入安卓权限列表，在安卓组件框中输入需要的安卓组件

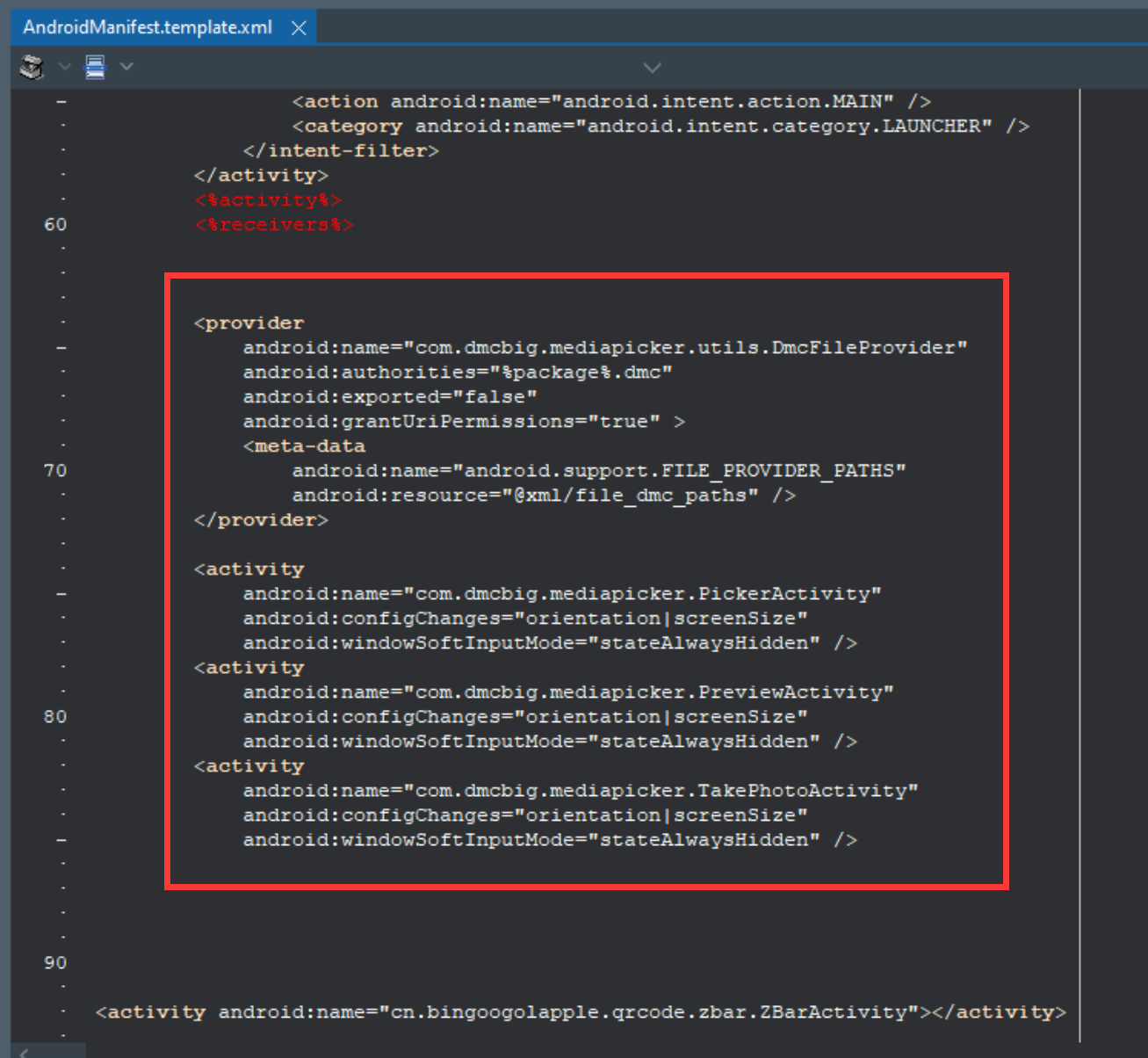


点击右边的布署按钮，即可配置到工程中



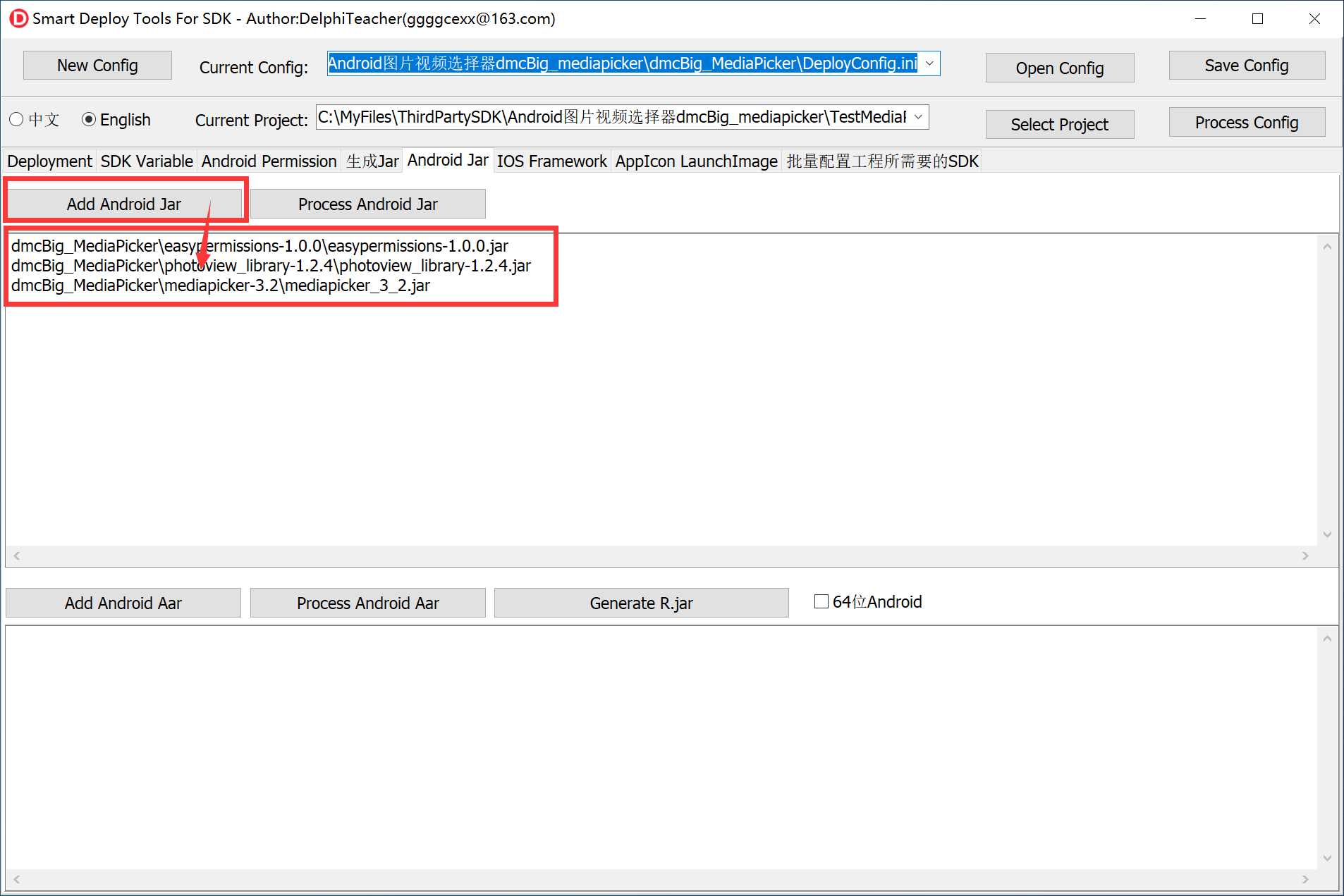
我们可以打开工程目录下面的AndroidManifest.template.xml查看一下：





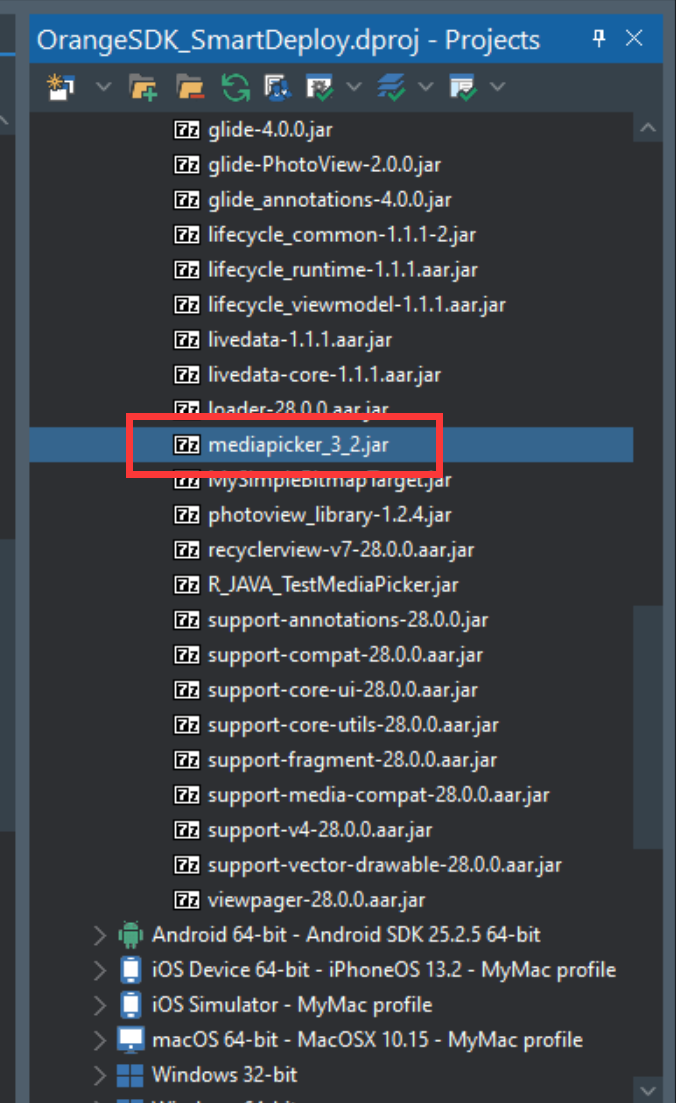
1. 能够直接将SDK所需的JAR添加到工程。

点击“Add Android Jar”,选择所需要添加的Jar文件，



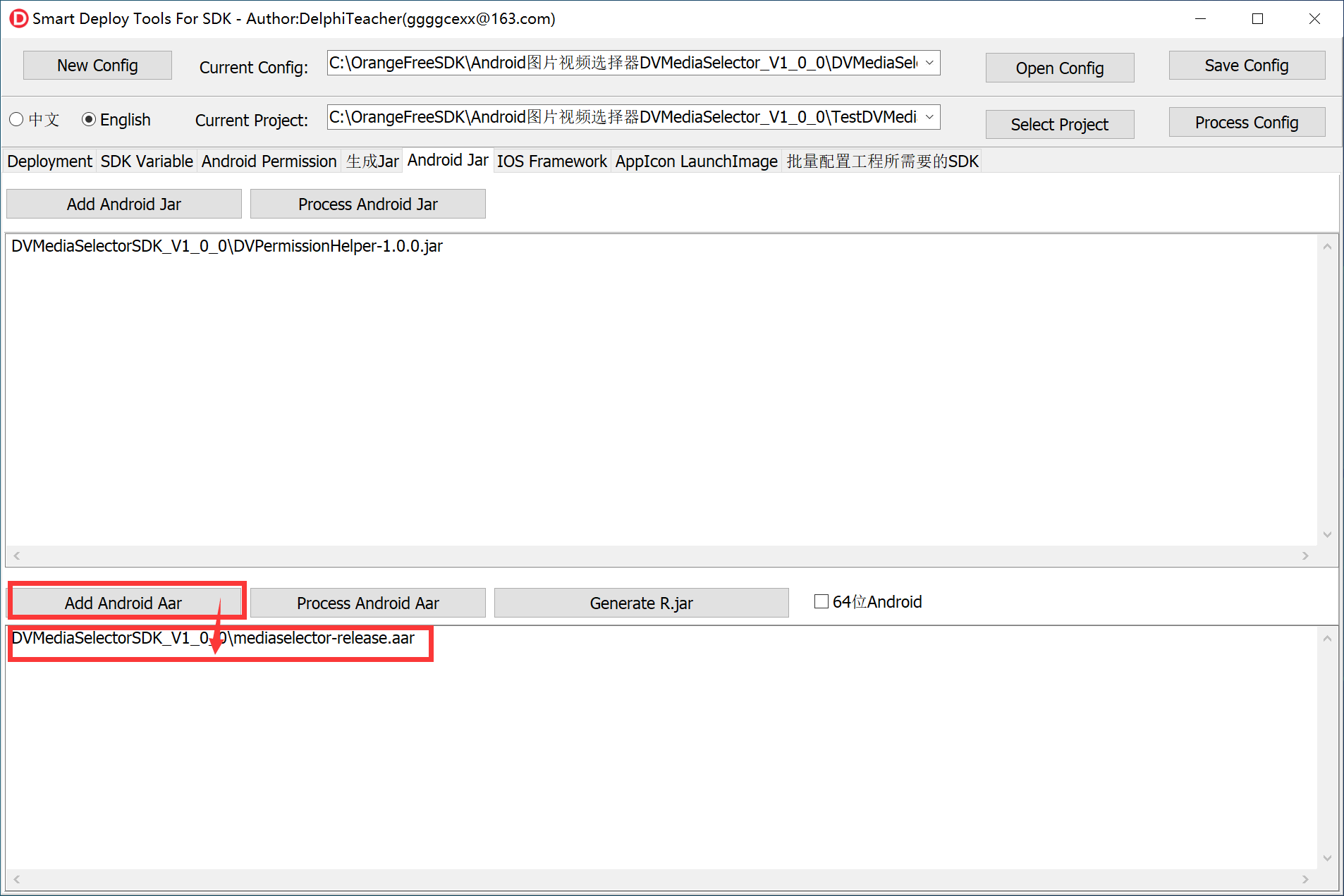
点击右边的“Process Android Jar”按钮，即可将Jar配置到工程中去。

我们展开工程的Android->Libraries节点检查一下：



1. 能够简单的自动配置Android的Aar包

点击“Add Android Aar”按钮来添加aar包，



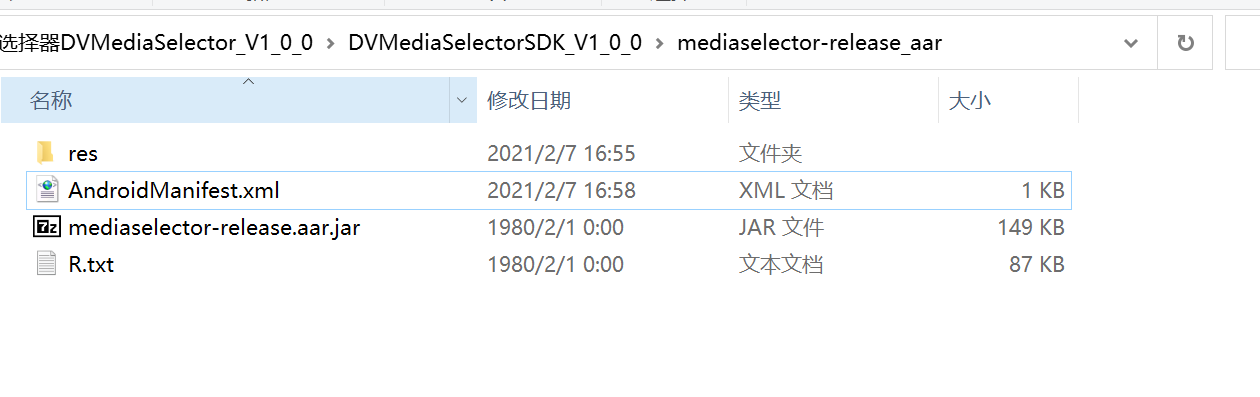
再点击右边的“Process Android Aar”按钮来将aar配置到工程中

Aar是一个zip压缩包，里面包含jar、资源文件、so动态库、AndroidManifest.xml的配置等，我这个工具可以解压aar压缩包，把jar和里面的res资源文件配置到工程中去，但是AndroidManifest.xml中的配置需要手动配置。

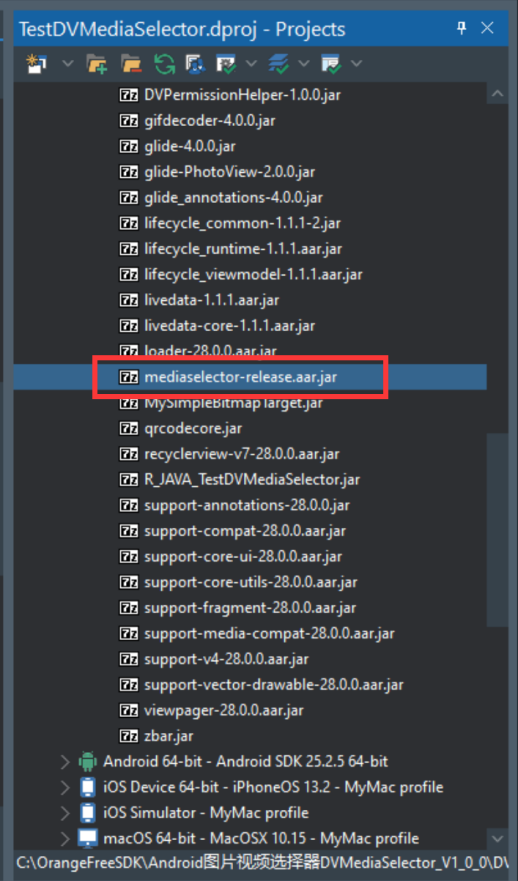
比如示例中的这个mediaselector-release.aar，点击了“Process Android Aar”按钮之后，会将它解压到mediaselector-release\_aar,



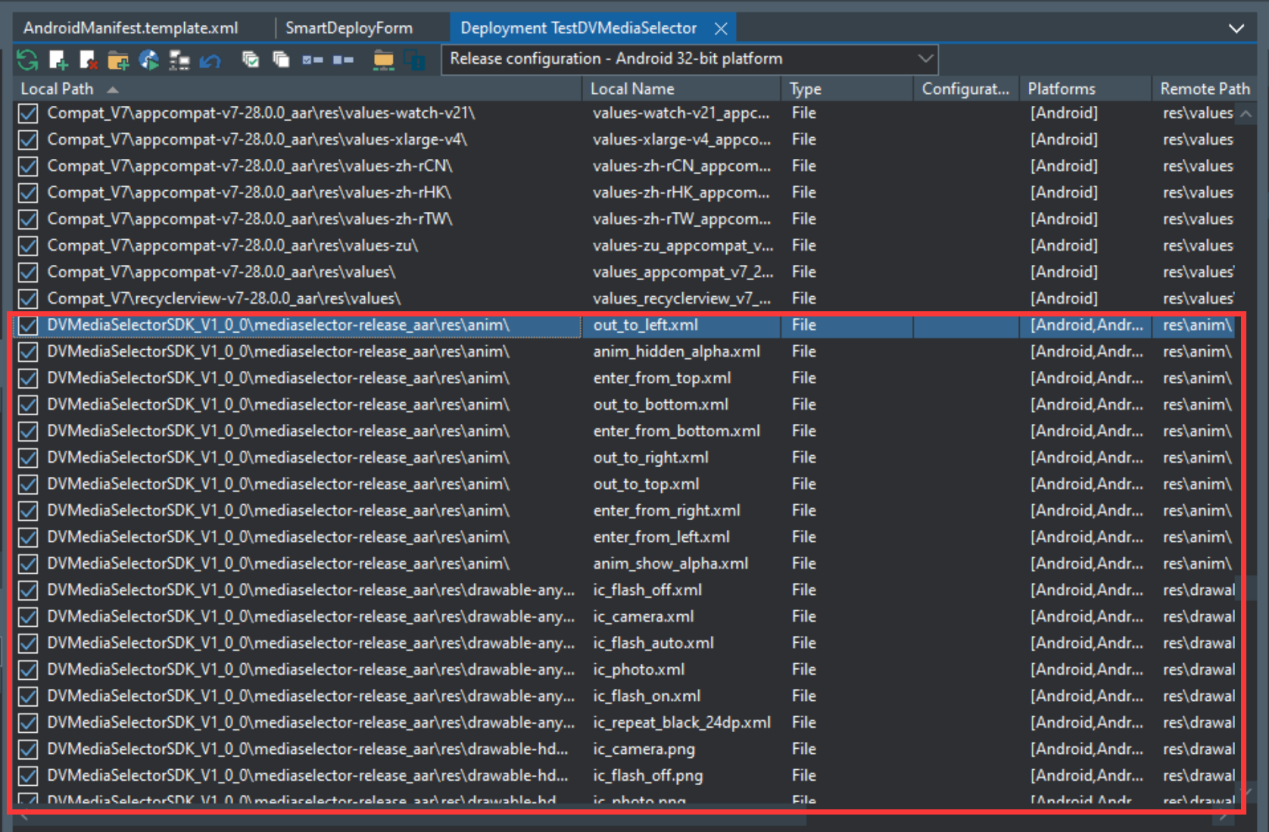
它里面的内容如下：



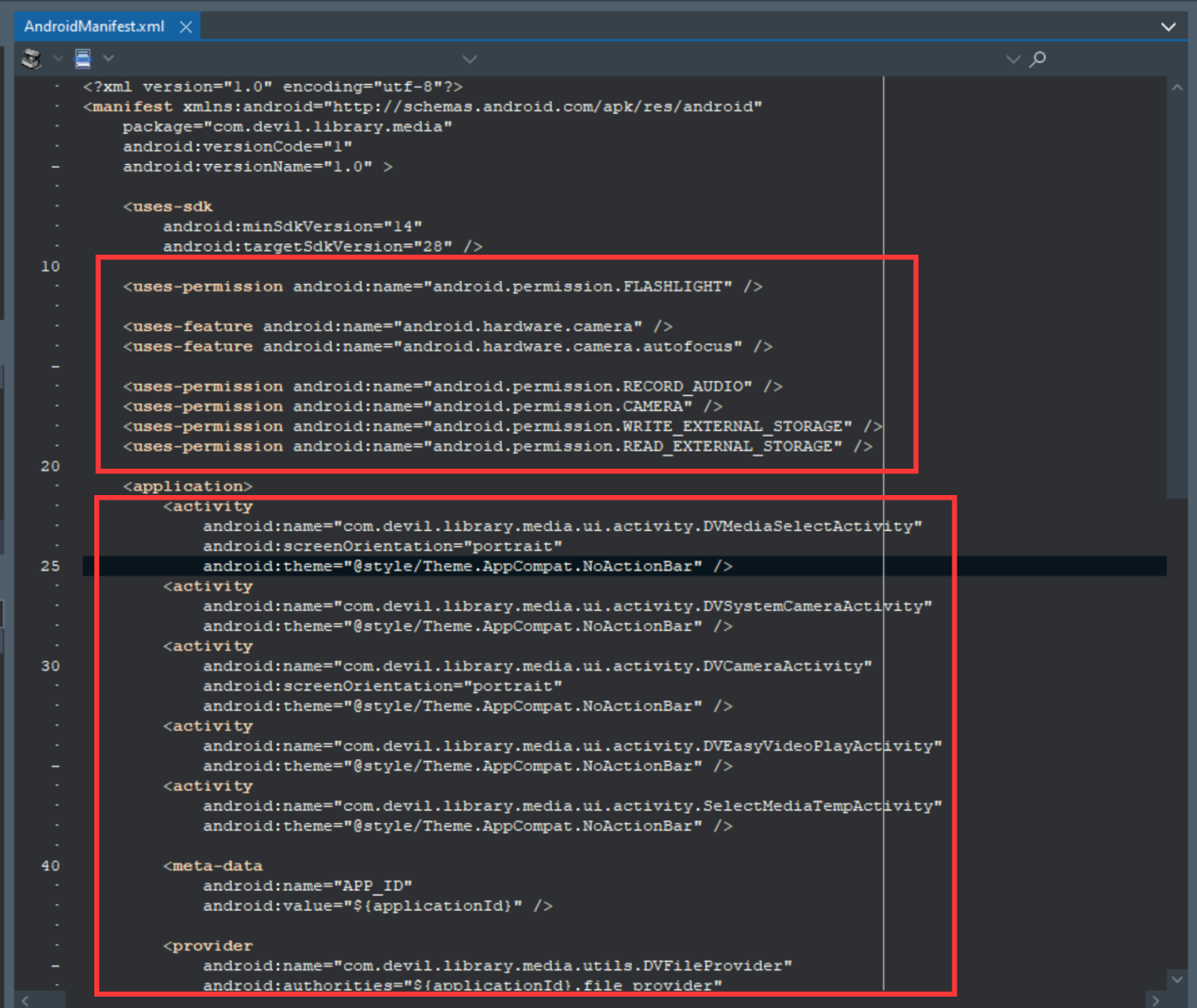
工具会自动将这个jar配置到工程中，



工具还会自动将res目录布署到工程中，



但是aar包中的AndroidManifest.xml中的权限和Android元件需要手动配置到工程目录中的AndroidManifest.template.xml中



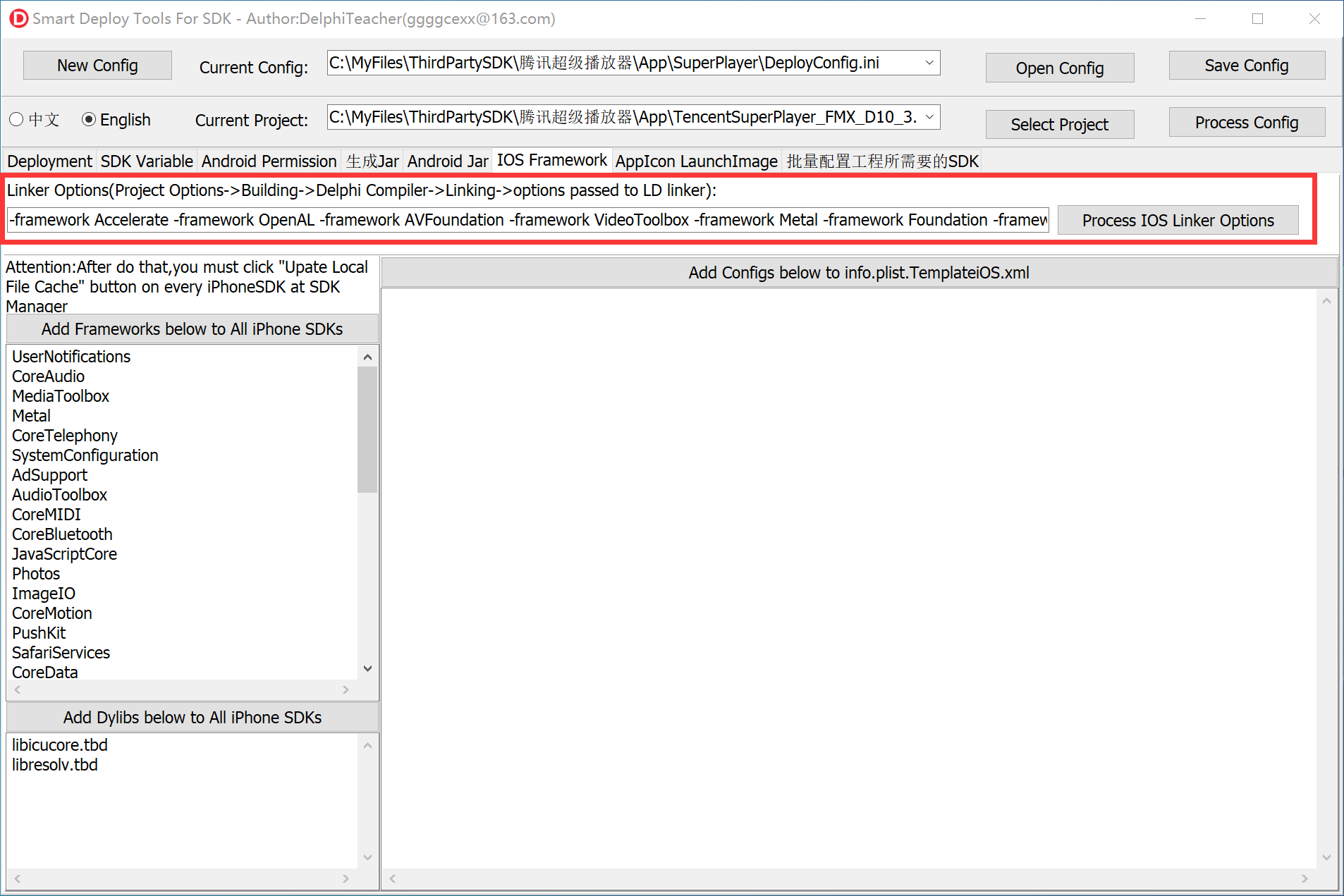
另外一个很重要的按钮是“Generate R.jar”。

因为aar中需要根据ID来访问资源文件，否则会报错，因此这个按钮就是用于生成资源文件的ID，并且打包到成jar，它生成在工程目录中的OrangeSDK\_JarGen目录，您需要手动将该jar添加到工程中去。

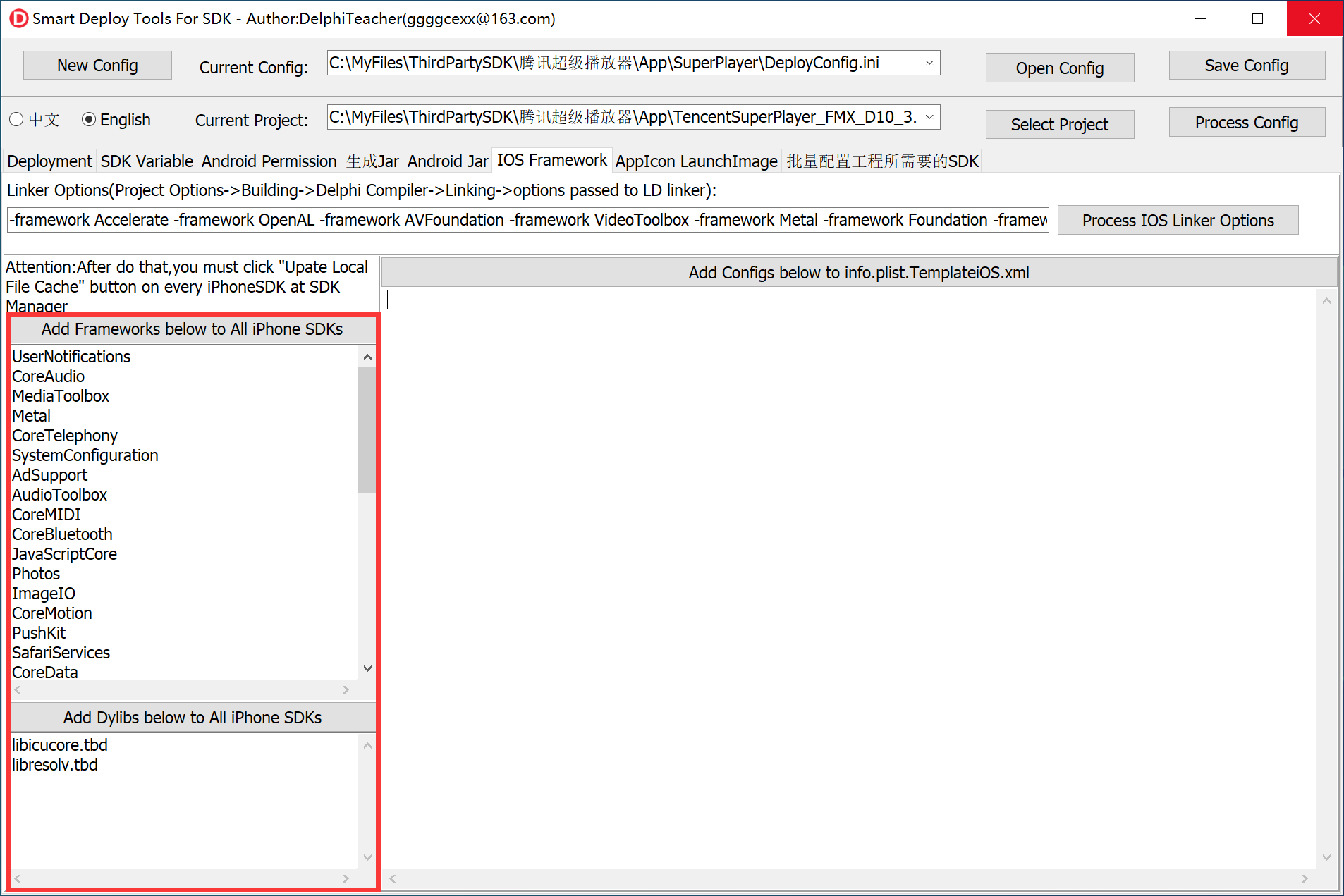


1. 能够直接给工程配置IOS的链接参数Options passed to the LD linker

如果一些三方IOS SDK需要配置链接参数，那么就在“IOS Framework”页面中的”Linker Options”中输入好，点击右边的“Process IOS Linker Options”即可直接配置到工程中去。



1. 能够直接给IOS SDK添加常用的Framework，比如SystemConfiguration、CoreTelephony等。



* 1. “Add Frameworks below to All iPhone SDKS”按钮，点击了这个按钮之后，会将列表框中的framework添加到Delphi的IOS SDK中，因为一个个添加太费时间了。点击了这个之后，需要重启IDE，并且在SDK Manager中点击“Update Local File Cache”才算配置完。
  2. “Add Dylibs below to All iPhone SDKs”按钮，这个功能和上面这个功能类似。

1. 设置工程图标和启动图片

只要选好一个图标即可生成各种尺寸的图标，并且一次性全部设置完毕，不用再一个个去选了。

1. 所有的配置项都能保存到配置文件中，方便下次直接使用。
2. 选择工程文件，勾选所需要的SDK（支付宝支付），进行一次性配置，并且将该工程所需要的SDK保存起来，下次重建工程的时候，再次重新配置即可