



ApkFactory

使用手册 1.1

目录

一、概述.....	2
1. 关于 ApkFactory	2
2. 功能介绍.....	2
3. 安装要求.....	3
二、开始使用.....	3
1. 两种方式都必须设置的信息.....	3
2. 选择性配置的信息.....	4
3. 增加配置信息.....	5
4. 打包方式.....	10
5. 拆包工具.....	12
三、注意事项.....	12
四、问题反馈.....	13
捐赠支持.....	13

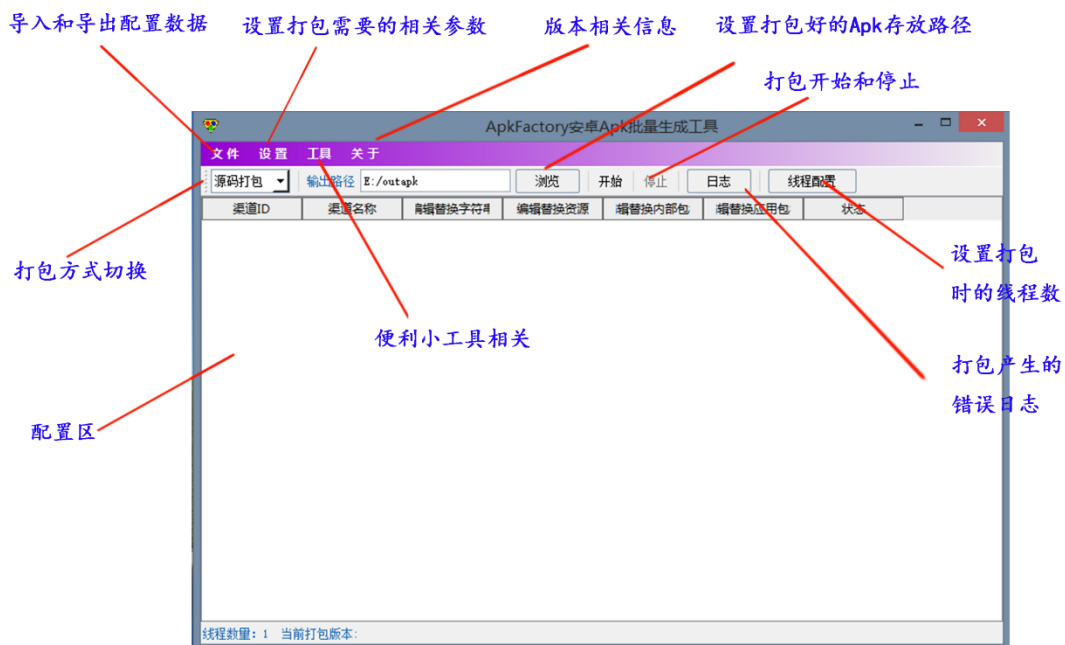
一、概述

1. 关于 ApkFactory

ApkFactory 是一款专注于 android 的批量打包工具，主要定位于一次配置，重复、快速的批量生成 Android 的 Apk 安装包。原名称 AutoPackingAndroid 废弃使用，采用名字：ApkFactory。感谢使用！

2. 功能介绍

ApkFactory功能图解



特别说明：

线程配置是配置同时开多少线程去打包，建议开启两个线程，经过测试过多的线程打包，cpu 已经处理不过来，不得不在不同线程间切换消耗更多时间。

日志是指错误日志，只显示打包过程中出现的错误，如果错误日志为空，就是打包过程中未产生错误，是正确的打包。

配置区是配置打包需要修改的各种参数，采用右键菜单的方式添加和删除配置信息。

注意：打包开始时一定注意看现在的打包方式是源码打包还是反编译打包。避免造成打包错误的问题。

3. 安装要求

安装前请保证机器已经安装好并配置好 Android 开发环境，其中包括：java 环境运行时环境的配置、JAVA_HOME 环境变量的配置。JAVA_HOME 的环境变量一定要配置，否则在源码打包时会报错误。

在以下环境中测试通过

测试环境：

平台：Windows8.1 64 位

Adt 版本：adt-bundle-windows-x86_64-20140702

Java 版本：jdk1.8.0_11

二、开始使用

ApkFactory 支持两种打包方式，一种是通过源码的打包，与 Eclipse 生成 Apk 的方式相同。在这种模式下，支持应用包名替换、内部包名替换（支持不完全）、渠道号替换、资源文件替换、支持混淆。是最安全的批量打包方式。另一种是通过反编译打包，反编译打包是把原来的包反编译之后进行一些渠道号的替换、资源的替换、应用包名替换（支持不完全）、内部包名替换（支持不完全）之后，重新生成 Apk，签名，zip4 字节优化后生成最终 Apk。这种模式下，支持渠道号替换、资源文件替换。但支持的不是很完全，在全包名路径引用时存在一定的风险，所以建议使用源码打包方式。所以请选择合适的打包方式。下面将分别介绍用法。

无论采用哪种打包，开始之前都必须设置相关信息，如果配置信息不全则不能开始打包。

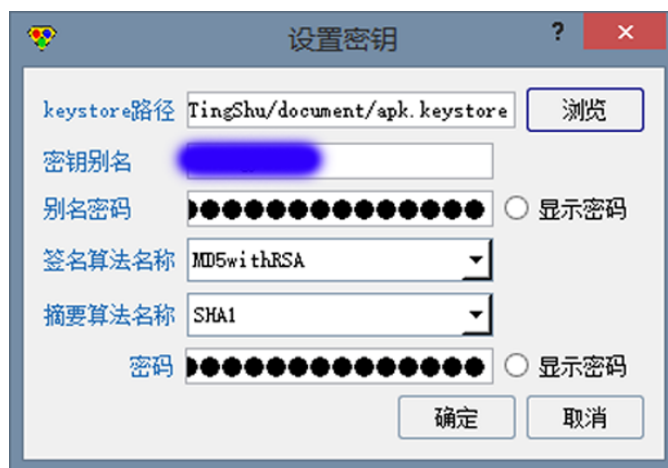
1. 两种方式都必须设置的信息

1) Jdk 路径



无论采用源码打包还是反编译打包，都必须设置 jdk 路径。正常情况下，软件会自动识别你的 java 环境变量并配置 jdk 路径，如果软件识别的不准确或未识别到的，请手动修改到正确位置。

2) 密钥信息



如图，

keystore 路径就是签名密钥的文件路径。

密钥别名就是申请密钥时候设置的别名。

别名密码就是申请密钥时设置的别名密码。

签名算法有两种主流的可供选择，MD5withRSA 和 SHA1withDSA，如果选项中没有您使用的，请手动填入您的签名算法名称。（支持手动编辑）。

摘要算法名称目前只内置了一个，如果不是您采用的，请手动填入您的签名算法。（支持手动编辑）

密码就是您签名包所用的密码。

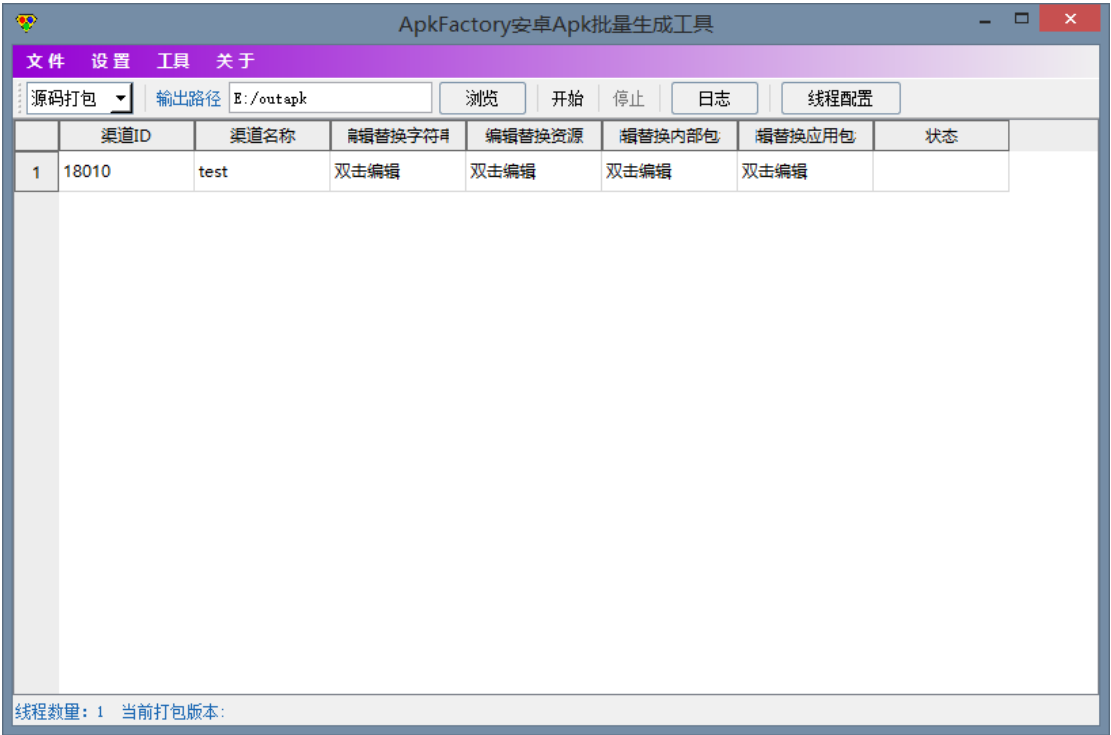
2. 选择性配置的信息

版本号是选择性配置信息，可配置可不配置。配置的目的是为了生成包名的

时候区分版本的。

最后生成包名的命名规则如下：

最后的包名生成规则是渠道名称 + 渠道 ID + 版本号。



如图所示配置，最后生成的包名为：test_18010_V3.0.apk

3. 增加配置信息

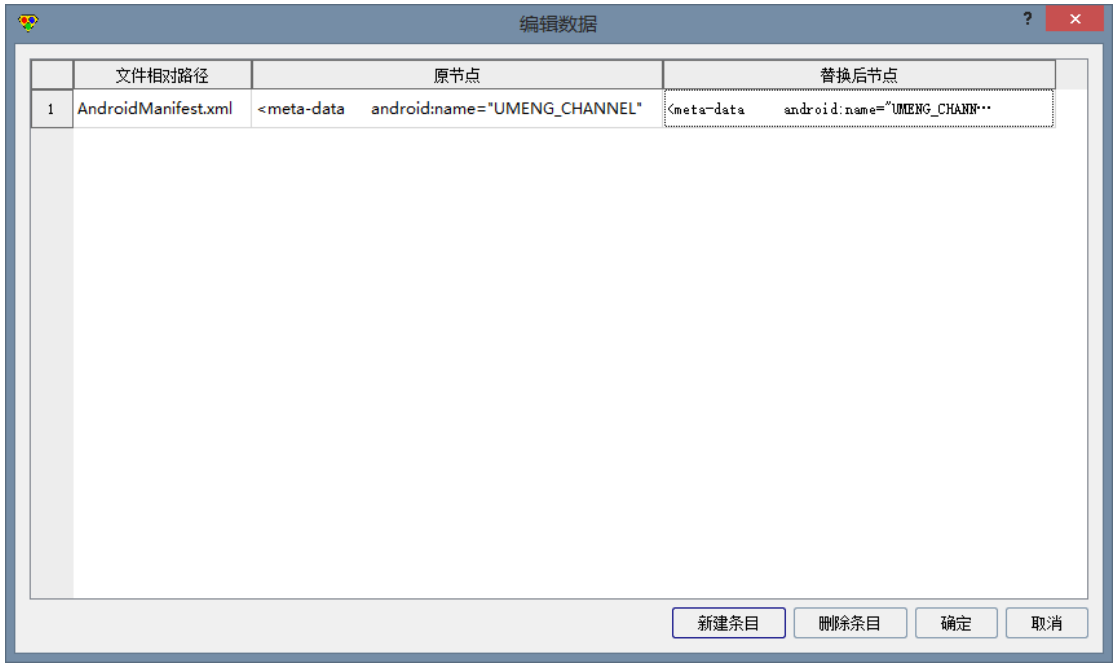
在配置区的第 1 列是配置渠道 ID，这个渠道 ID 除了在生成包名时用于命名没有任何作用。配置区第 2 列是渠道名称，同渠道 ID 一样，渠道名称也是只参与包名的命名，不参与其他。第 3-6 列是参与替换的主要配置区。

第 3 列：双击编辑按钮，如图：



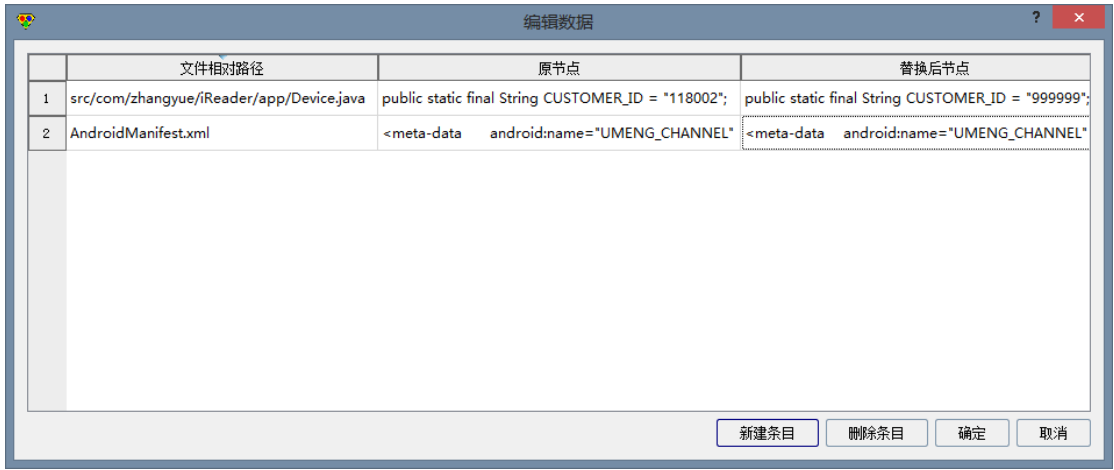
点击，新建条目出现一条空配置，双击表格进行配置。需要说明的是第一列的文件相对路径是指相对于项目源码中的相对路径，如：如果要替换 AndroidManifest.xml 中的内容，则因为 AndroidManifest.xml 在项目目录的根目录，所以在文件相对路径只需要填入 AndroidManifest.xml 即可，第 2 列原节点则填入你要替换的内容，为了保证替换的准确性，原节点填入的信息越能说明节点唯一性越好，因为此处的替换算法只是简单的字符串查找替换，替换后的节点则为要把原节点替换成之后的节点。

如：替换友盟渠道号配置如下：



因为在 AndroidManifest.xml 文件中，<meta-data
android:name="UMENG_CHANNEL"
android:value="118002" />是独一无二的，所以查找替换肯定不会出现错误的替换，是安全的替换。

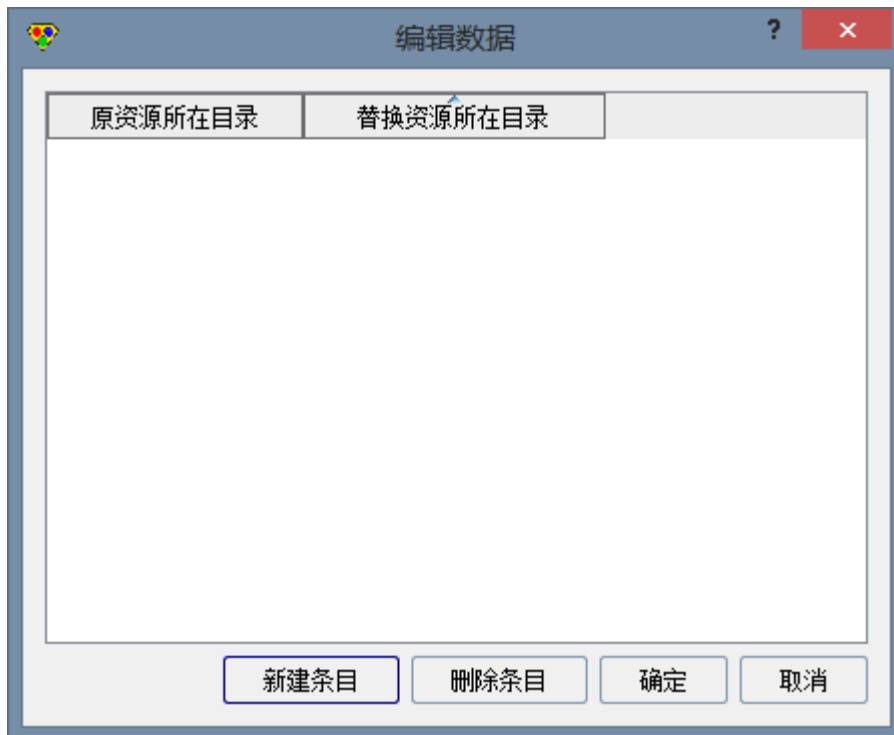
再例如需要替换某 java 文件中的某个变量，可进行如下配置，如图：



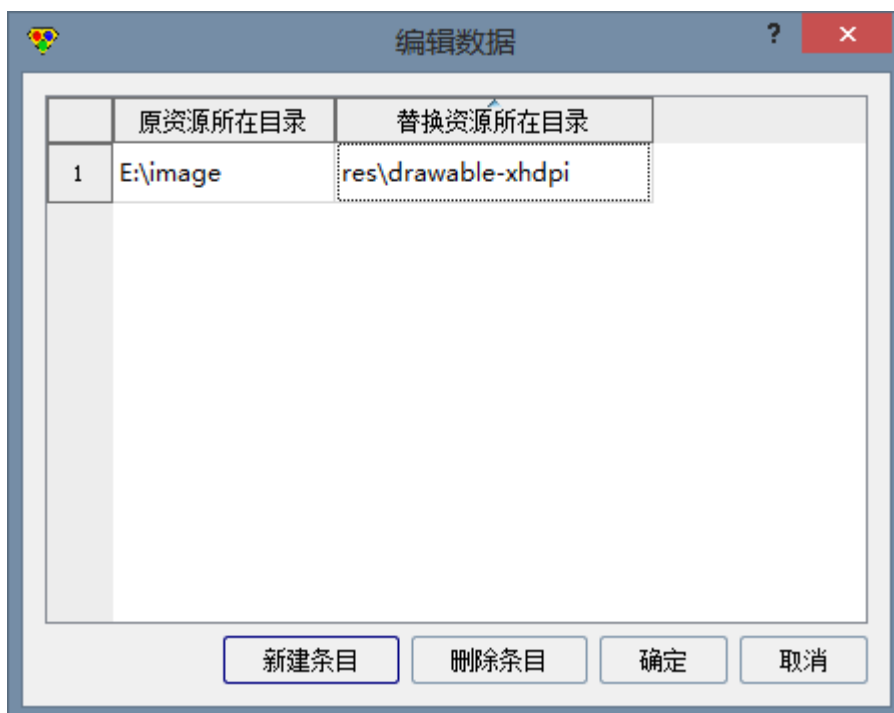
因为在 src\com\zhangyue\ting\samples.java 文件中，原节点 public static final String CUSTOMER_ID = "118002";是唯一的，所以替换也是安全的。不会出现错误。同样需要替换多个文件中多个变量时，只要根据要求增加配置条目即可。

回到主配置区

第 4 列：双击编辑按钮，如图：



其中操作方式与第三列类似，不做介绍，主要介绍各个参数作用。第一列，如图，是用作原资源目录的配置，替换资源则是相对于项目目录的路径配置。如现需要替换某个渠道下的启动画面，则只需指定启动画面在 PC 中的路径位置，如图



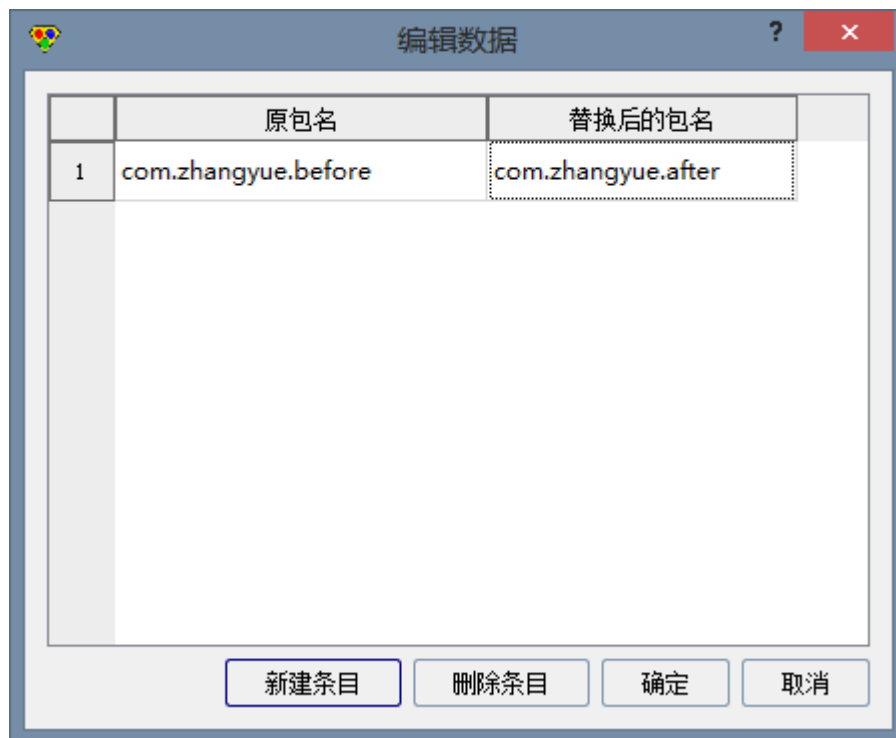
进行该配置后，则在打包开始后，软件会把 E:\image 中的所有文件打入到该渠道包中的 res\drawable-xhdpi 目录中。如果要替换的资源比较多，则分别增加配置条目即可。

回到主配置区

第 5 列

与前两列的操作方式类似。不做概述，只说明一下参数作用，第 5 列是替换内部包名，即指 App 应用内部某个子包的名字，配置的原包名即指未替换之前的包名，如：com.zhangyue.before。替换后的包名即指进行替换之后的包名，如：com.zhangyue.after

如图：



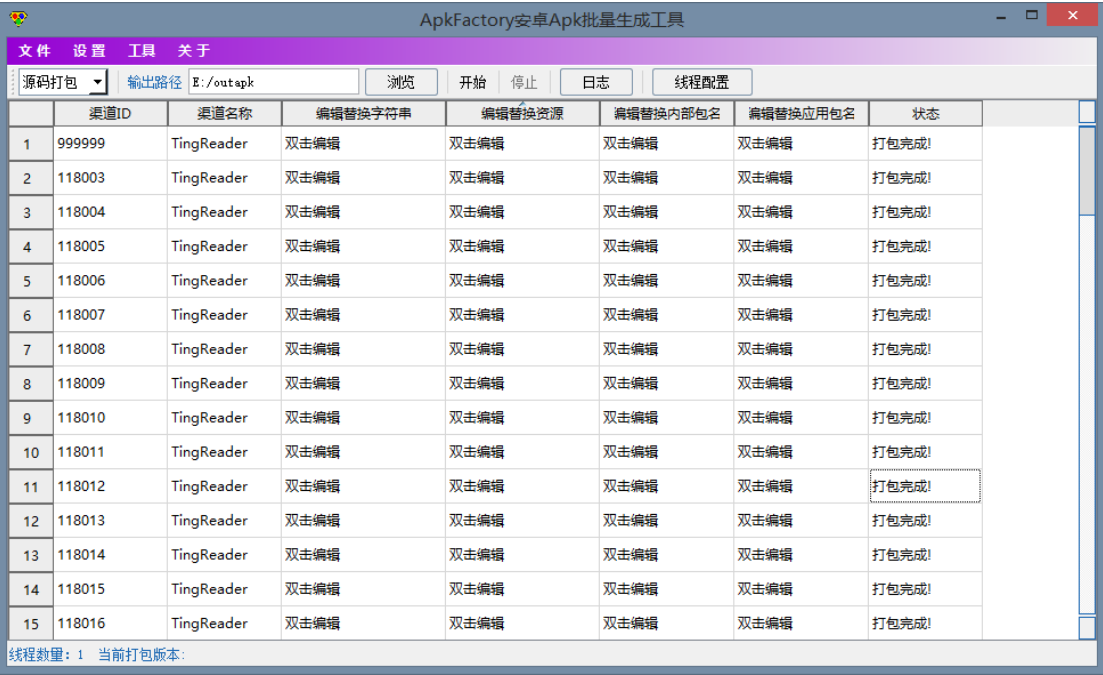
需要修改多个子包名，增加相应配置即可。

回到主配置区

第 6 列

第 6 列配置方式与前面类似，不再概述，主要说明配置参数。第 6 列配置的主要是应用包名，想一个手机同时安装多个同一应用的，则需要修改不同应用包名，则需要在此列中增加相应的配置，配置方式与第 5 列相同。只是应用包名的替换，一个渠道只能替换一个。如果想多个渠道配置多个，请在主配置去增加相同渠道分别配置。

主配置区 3-6 列的配置都是选择性配置，需要修改哪些，则配置哪些。完成 3-6 列的配置后，则完成了一个渠道的配置。需要完成多个渠道则重复增加配置条目，分别配置，配置好如图：



4. 打包方式

a) 源码打包

首先在进行源码打包之前，请先保证你的项目能在 Eclipse 中无错误可打包。如果在 Eclipse 无法进行打包的，在 ApkFactory 中同样无法打包，会报错误。请确保你的项目所有文件都是用 utf-8 无 BOM 方式编码，存在 ANSI 编码或

者 utf-8 有 BOM 方式编码的文件源码打包时将出现错误。

源码打包必须在设置中配置源码打包配置。如图：



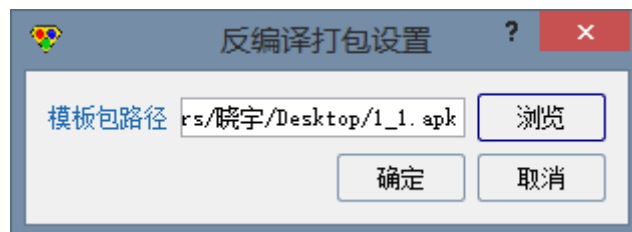
源码路径指需要打包的项目路径。

AndroidSdk 目录则只使用的 sdk 路径，注意路径指向的是 sdk 目录，目录不要设置错，否则会出现找不到 sdk 路径问题。

源码打包配置好，选择源码打包方式，点击开始按钮则会开始进行源码的打包。

2) 反编译打包

反编译打包则必须在配置中设置反编译打包的配置，如图：



反编译打包只需配置一个参数，即模板包路径，模板即指需要反编译的 apk 包，反编译打包中，要替换字符串时，如果是在 smali 文件中，则需要指明相应的 smali 文件，而非 java 文件，因为反编译后不会得到 java 文件。只会得到 smali 文件。所以替换字符串配置时，如果选用反编译方式，请注意。

反编译配置好后，在主界面切换配置到反编译打包，点击开始按钮，则开始反编译打包。

导入导出数据

在文件中有两个功能，一个是导入数据，一个是导出数据。

都是针对配置区所配置的数据的，当配置好所有数据后，建议在导出配置中

导出一份作为备份，避免不小心的丢失或误改。

3) 单渠道打包

在主配置区选中某行时，点击开始打包按钮，此时是单渠道打包，只会打包选中的那个渠道。

4) 多渠道打包

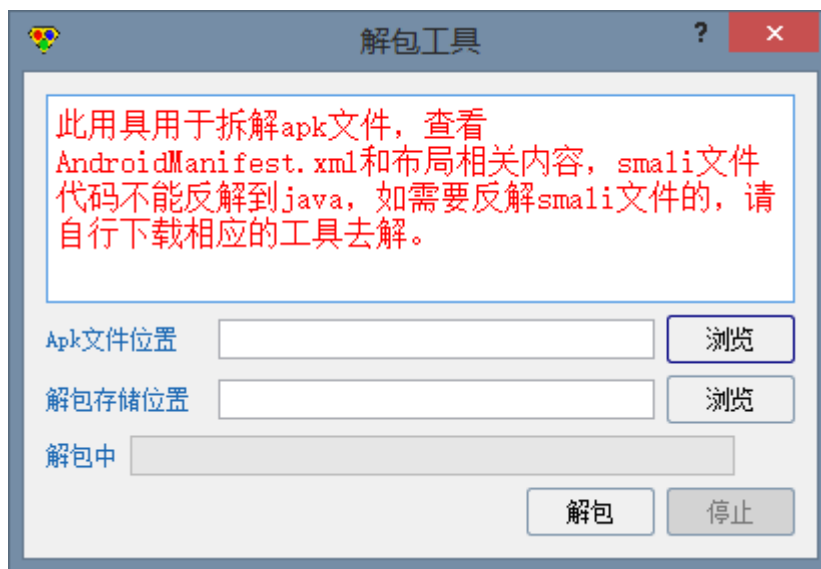
要进行多渠道打包，只需选中需要打包的渠道，支持跳选，选中之后点击开始按钮，则会打包所有选中的渠道。

5) 所有配置打包

要想打包所有配置的渠道，必须不要选中任何行，建议是点击右边空白一次后再点击开始按钮。这样会进行所有配置的打包。

5. 拆包工具

在菜单栏的工具菜单下目前集成了拆包工具，此工具主要用于拆解 Apk 包成 smali 文件，以及拆解出其中的 xml 资源文件。主要为了打包完成后验证渠道替换是否成功使用，也可用于拆解别人 Apk，了解其实现思路。具体根据实际用途使用。



如图：

三、注意事项

打包时出错

打包错误大致会有如下原因：

1) 文件中存在编码不一致问题，具体看打包日志，会看到相应编码有问题文件信息，解决办法，把编码有问题文件编码修改成统一 utf-8 无 BOM 格式

2) 资源出错。在反编译打包时，如果原 apk 中存在非正常格式 png 文件，则会出现资源出错的问题。这种问题就是 png 格式不规范造成，在 eclipse 低标准下是可以通过的，但是在反编译中是不能通过的，解决办法，把存在问题的资源文件用 ps 重新生成正确格式

3) 清除上次缓存出错，这个错误是因为子进程结束不及时，造成。后续版本会修改这个问题，目前的解决办法：关闭软件，重新启动。

四、问题反馈

这是款开源的打包工具，其中还存在一些需要完善的点。希望大家在使用过程中有问题及时反馈给我，我是冰剑。

我的 QQ: 21314601

邮箱: lixiaoyu0123@163.com

捐赠支持

感谢使用 ApkFactory，目前自动检测更新功能还不可用，因为目前没有服务器用来做为检查更新的平台，所以现在想建立检查更新平台，同时想建立服务器打包项目，但是目前个人资金有限，希望能得到大家的支持。

下面是开发者的支付宝账号，期待大家的支持，感谢！

21314601@qq.com 验证名称为：李晓宇

微信收款二维码：

