

## 打包平台的脚本构建规约

### 规则说明

修改工程

工程名

国际支付SDK-Android 4.2发布版本

请输入工程名称.

工程描述

http://svnhz.alipay.net/svn/android\_phone\_cashier/branches/ali01516027\_C

请输入工程描述信息.

工程SVN地址

http://svnhz.alipay.net/svn/android\_phone\_cashier/branches/ali01516027\_C

请输入工程SVN地址.

SVN用户名

ruiqi.li

请输入工程SVN用户名.

SVN密码

\*\*\*\*\*

请输入工程SVN密码.

pack.jar相对路径

autopack/

请输入pack.jar的SVN工程内的相对地址.

删除工程

取消

修改

上述是一个工程创建的截图，在录入上述打包工程的时候，需要写入相对路径，这个相对路径是对应你配置的svn地址而言的。

比如你配置的svn是[http://svnhz.alipay.net/svn/android\\_phone\\_cashier/branches/ali01516027 CP 20141128 android\\_phone\\_cashier/international sdk](http://svnhz.alipay.net/svn/android_phone_cashier/branches/ali01516027_CP_20141128_android_phone_cashier/international_sdk)

系统会在打包的时候，checkout这个svn，将代码checkout到一个目录，这个目录是打包平台创建的，假定是A,假定你在这里配置的pack.jar相对路径是autopack/，那么打包平台将调用你的脚本按照：

`java -jar A/autopack/pack.jar`。现在知道了这个相对路径的含义了吧。

同样，为了让打包平台知道目前你的工程支持哪些打包模式，你需要在上述指定的pack.jar相对路径下面，创建一个文件叫：functions.json，这个文件和pack.jar是在同一个目录下面。

这里的functions.json 非常重要，他描述了打包平台调用你的打包脚本的参数以及打包形式。用户在点击那些打包按钮的时候，需要传递的参数也是这里描述的。

如下就是一个functions.json的格式：

```
[
  {
    "args": {
      "dir.spec": "$SVN_PATH",
      "signals": "4",
      "info": ""
    },
    "name": "【开发环境包】打包framework和Demo，不混淆代码包含日志"
  },
  {
    "args": {
      "dir.spec": "$SVN_PATH",
      "signals": "5",
      "info": ""
    },
    "name": "【开发环境包】打包framework和Demo，并混淆代码不删除日志"
  }
]
```

这个文件对应的是一个json数组，里面的每个元素对应一种打包模式，上图对应两种打包模式，其中一个打包名称是**【开发环境包】打包framework和Demo，不混淆代码包含日志**，这个打包需要入参是：

```
"dir.spec": "$SVN_PATH",
"signals": "4",
"info": ""
```

打包平台在执行上述打包的时候，将会执行如下的代码：

```
java -jar -Ddir.spec=/home/admin/pack/4544545/ -Dsignals=4 -Dinfo="" -jar pack.jar
```

这里有个重点没有描述，就是打包平台提供了一些系统变量，比如这里的**SVN\_PATH**，用户在args中描述这个值，打包平台，会将当前用户svn流下载到的目录A（上文已经提到了A代表啥意思），替换这个SVN\_PATH

**到现在已经描述了，怎么样提供打包功能列表，打包平台怎么样去根据用户提供的参数去打包。**

那么问题来了，打包平台怎么知道打包的结果是正确还是错误呢？这是个很好的问题，这里我们对pack.jar的功能提出了要求，我们要求业务方在实现打包pack.jar的时候，需要将本次的打包结果输出，也就是打包平台在调用pack.jar的时候，你在内部的执行过程结束后，要生成打包报告，这个报告很简单，格式如下：

```
{
  "result": true,
  "pkgs": [
    "/autopack/internation_sdk/pack/framework.jar",
    "/autopack/internation_sdk/pack/demo.apk",
    "/autopack/internation_sdk/pack/ui-resource.jar",
    "/autopack/internation_sdk/pack/ui-engine.jar"
  ]
}
```

这里的result是告诉打包平台本次的打包结果，pkgs是告诉打包平台，本次你希望输出的打包内容。这些打包内容，都是相对路径，他们的实际路径是SVN\_PATH/\*\*,比如上述的framework.jar，其对应SVN\_PATH/autopack/internation\_sdk/pack/framework.jar.(你这里只管输入相对路径，打包平台他知道怎么去识别实际的绝对路径)。

用户生成的这个结果，我们将其写入一个文件，这个文件叫：result.json，他落地是跟pack.jar同一个目录（嘿嘿）

打包平台在调用pack.jar执行完毕后，将会去读取result.json这个文件作为本地的打包结果给用户展示，如果没有发现这个文件，将认为本地打包失败，如果发现这个文件，但是result结果为false，也会认为打包结果是失败的。

那么打包平台怎么知道pack.jar执行结束呢？这个。。我就不说了，只要你的方法执行结束，相信我，我们一定知道你执行完毕了。