

批量打包最常用到的地方是进行产品推广时，为每个渠道打一个包。上一篇随笔中，介绍了怎样进行一次完整的打包，批量打包只要在此基础上做一次循环即可。

在打包之前要做两个准备工作，一个是读取渠道，一个是修改存储渠道的文件。

在工程的assets中新建一个channel.cfg文件，专门用了保存渠道，保存方式是 channel=渠道号。

ant的基本库中没有提够循环，需要引入另一个库ant-contrib-1.0b3.jar，以下通过一个例子来了解ant的循环



```
<?xml version="1.0" encoding="UTF-8"?>
<project name="TestAnt" default="deploy">

    <taskdef resource="net/sf/antcontrib/antcontrib.properties">
        <classpath>
            <pathelement location="${ant.home}/exlib/ant-contrib-
1.0b3.jar" />
        </classpath>
    </taskdef>

    <target name="deploy">
        <foreach list="aaa,bbb,ccc,ddd,eee" target="package"
param="channel" delimiter=",">
            </foreach>
        </target>

        <target name="package">
            <echo>${channel}</echo>
        </target>

</project>
```



输出结果：

```

D:\Android\workspace\TestAnt>ant
Buildfile: D:\Android\workspace\TestAnt\build.xml

deploy:

package:
    [echo] aaa

package:
    [echo] bbb

package:
    [echo] ccc

package:
    [echo] ddd

package:
    [echo] eee

BUILD SUCCESSFUL
Total time: 0 seconds

```

cnblogs.com/sw926

在这个例子中aaa,bbb,ccc,ddd,eee以“,”为分隔符，依次取内容赋值给channel，做了五次循环。在target package中，用\${channel}就可以取到渠道。

取到渠道后，通过正则表达式替换channel.cfg的内容

```

<target name="change.channel">
    <replaceregexp flags="s" byline="true">
        <regexp pattern="channel=(.*)" />
        <substitution expression="channel=${channel}" />
        <fileset dir="assets" includes="channel.cfg" />
    </replaceregexp>
</target>

```

flags="s"表示文件是单行，byline="true"表示逐行查找，<regexp pattern="channel=(.\*)" />指定匹配的正则表达式，<substitution expression="channel=\${channel}" />为替换后的内容，<fileset dir="assets" includes="channel.cfg" />指定文件。

我们在工程的根目录新建一个channels.properties的文件，输入channels=aaa,bbb,ccc,ddd,eee完整的批量打包代码为：



```
<?xml version="1.0" encoding="UTF-8"?>
<project name="TestJni" default="package">

    <taskdef resource="net/sf/antcontrib/antcontrib.properties">
        <classpath>
            <pathelement location="${ant.home}/exlib/ant-contrib-1.0b3.jar" />
        </classpath>
    </taskdef>

    <!-- 设置sdk, ndk路径, 证书路径密码 -->
    <loadproperties srcFile="local.properties" />
    <loadproperties srcFile="project.properties" />
    <!-- 所有的渠道 -->
    <loadproperties srcFile="channels.properties" />

    <fail message="sdk.dir is missing." unless="sdk.dir" />
    <fail message="ndk.dir is missing." unless="ndk.dir" />

    <!-- 编译native代码 -->
    <target name="native" unless="native.libraries.are.uptodate">
        <echo message="Building native libraries..." />
        <exec executable="${ndk.dir}/ndk-build.cmd" failonerror="true" />
        <echo message="DONE (Building native libraries)" />
    </target>

    <!-- 引入sdk中的build.xml -->
    <import file="${sdk.dir}/tools/ant/build.xml" />

    <!-- 修改保存渠道的文件 -->
    <target name="change.channel">
        <replaceregexp flags="s" byline="true">
            <regexp pattern="channel=(.*)" />
            <substitution expression="channel=${channel}" />
            <fileset dir="assets" includes="channel.cfg" />
        </replaceregexp>
    </target>

    <!-- 进行一次完整的打包 -->
    <target name="package" depends="change.channel,native,release">
        <!-- 将apk复制的制定的目录 -->
        <copy tofile="sign/${ant.project.name}-${channel}.apk"
file="bin/${ant.project.name}-release.apk" />
    </target>
</project>
```

```

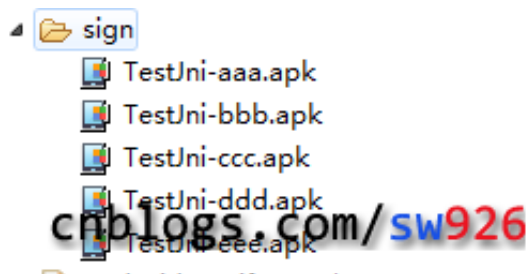
</target>

<!-- 执行批量打包 -->
<target name="deploy">
    <foreach list="${channels}" target="package" param="channel"
delimiter=",">
        </foreach>
    </target>
</project>

```



打开cmd，切换到工程根目录，执行ant deploy，批量打包完成后在制定目录生成了apk



安装apk到手机或模拟器进行测试



在程序内部已经读到渠道。

至此，只能够进行简单的批量打包任务，所用到的也是简单的ant命令，上面的的批量打包代码也有待改进，希望通过以后的学习，能加快打包的速度，实现更复杂的功能。