使用ant编译之签名问题

DionysosLai([906391500@qq.com](mailto:906391500@qq.com)) 2015/1/5

使用ant编译，可以很大提高项目开发效率。目前，coco2dx官方所提倡的提高几十倍的编译速度，其实就是使用了ant技术。原理提前将引擎库编译成一个so文件，游戏编译时，只需要编译自己的代码即可，而不需要同时编译引擎代码。相比自己的代码与引擎代码量，二者相差了几十倍，因此速度会提高很多。同时另一方面，很多问题，在Eclipse上只会报错，而不能指出具体位置（相信，大家在写mk文件时，就遇到这个问题）。然而，使用了ant编译，问题具体位置将完完全全的显示出来。

当然使用ant编译也有不好的地方，就是每次都会重新编译代码，而不是改动过的代码，这样导致每次生成apk文件都需要1到2分钟，这取决于游戏代码量。

在cocos2dx引擎中，使用ant编译工程，是一项比较复杂的操作，因此打算分成几个部分，详细的将问题阐述清楚。作为第一篇关于ant编译问题，先阐述一个比较简单的话题：签名。这个话题的延伸，就是自动生成apk，自动给apk打签名了。

给apk签名，有两种，一种是手动签名，另一种是自动签名。

**手动签名:**

1. 生成apk

使用ant编译，可以参考这篇pdf文件（<http://download.csdn.net/detail/dionysos_lai/8333741> ）或者网上其他资料。大概命令类似如下：

call cd 工程文件夹\proj.android

call android update project -p .

call ant clean

call ant debug

pause

这是，我写的一个自动编译脚本，双击运行即可，这样会自动生成一个debug版的apk（注意，debgu版apk，系统会自动给apk打上bebug签名，只能将”call ant debug”改成”call ant release”才能生成一个未签名过的release版apk）。

1. Apk签名

如何Apk签名，可以再网上找到相关资料，这里可以参考我自己写的一个脚本命令行：

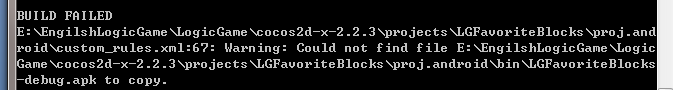
call jarsigner -verbose -keystore key文件名字 -signedjar ok/代签名apk名字 生成apk名字 key文件名字

pause

创建一个.bat文件，将文件放在待签名apk同一级目录下即可，双击运行，之后输入签名key文件的密码，就会在ok文件夹生成签好名的apk了。

这里签名会有一个问题出现：”java.util.zip.ZipException: invalid entry compressed size (expected 4308 but got 4432 bytes)”。这个问题很多人在网上遇到过，在ant编译时，我也曾遇到过，网上的说法有很多，解决也一样。后来，分析问题，其实很简单，就是说apk大小出错，本来应该是4308字节，现在确实4432字节了，然后就报压缩出错问题。其实，之所以出现这个问题，是因为我们尝试使用debug版apk签名了，而debug版apk已经签过名了（这也是apk大小增大的原因）。因此我们只能输出release版apk，再次签名。

直接将”call ant debug”改成”call ant release”，双击运行脚本文件，会报这个如下错误;



（ps，这就是使用ant编译好处，问题出现位置很清楚），这是由于文件” custom\_rules.xml”67行清清楚楚这样写着” <copy file="${source.absolute.dir}/../bin/${ant.project.name}-debug.apk" tofile="${customapk.out}/${ant.project.name}.apk" />”我们要做的就是改成这样即可” <copy file="${source.absolute.dir}/../bin/${ant.project.name}-release.apk" tofile="${customapk.out}/${ant.project.name}.apk" />”

1. 验证签名

Apk签名验证，可使用以下命令行，相关资料可以参考：[**http://stackoverflow.com/questions/7104624/how-do-i-verify-that-an-android-apk-is-signed-with-a-release-certificate/7104680#7104680**](http://stackoverflow.com/questions/7104624/how-do-i-verify-that-an-android-apk-is-signed-with-a-release-certificate/7104680#7104680)

call jarsigner -verify -verbose -certs 待验证apk名字

**自动签名：**

相对于手动签名，自动签名可以省去手动给apk打上签名这一步骤，不过自动签名是只能是release版，debug版不存在这一步骤。

如何自动签名，其实在文件”ant.properties”中，已经写得很详细了：

# This file is used to override default values used by the Ant build system.

#

# This file must be checked into Version Control Systems, as it is

# integral to the build system of your project.

# This file is only used by the Ant script.

# You can use this to override default values such as

# 'source.dir' for the location of your java source folder and

# 'out.dir' for the location of your output folder.

# You can also use it define how the release builds are signed by declaring

# the following properties:

# 'key.store' for the location of your keystore and

# 'key.alias' for the name of the key to use.

# The password will be asked during the build when you use the 'release' target.

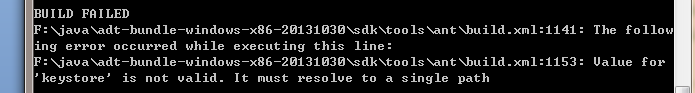
根据其提示，在这个文件中添加如下几句即可：

key.store=E:/LogicGame/cocos2d-x-2.2.3/apk/key

key.alias=key

key.store.password=123456

key.alias.password=123456

注意，这里”key”是签名key，123456，是其密码。这里可能会有一个这样一个错误，

引起这个错误的原因，就是unix与window的格式问题，也就是将”/”，写错了”\”。

ok，关于ant签名差不多就到这里了，希望对大家有用。