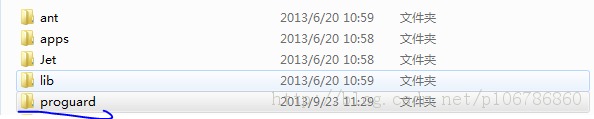
**代码混淆**

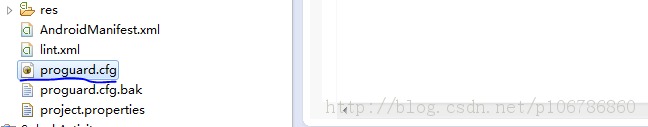
**1.准备工作**

首先，我们得将我们的项目源码、proguard工具（放心这个工具Google已经在sdk中为我们提供了，在sdk\tools目录下可以看到）、 proguard.cfg文件（该文件定义了混淆的规则，创建2.3以上项目自动生成，或者在网上可以很容易下载到）。



**2.在你的项目中添加proguard.cfg文件**

该文件Google公司已经为我们准备好了，创建项目时，在项目的根目录下就有了，但是是从2.3开始的。不过我们基于2.2的项目只需要将该文件拷贝到项目根目录下即可。



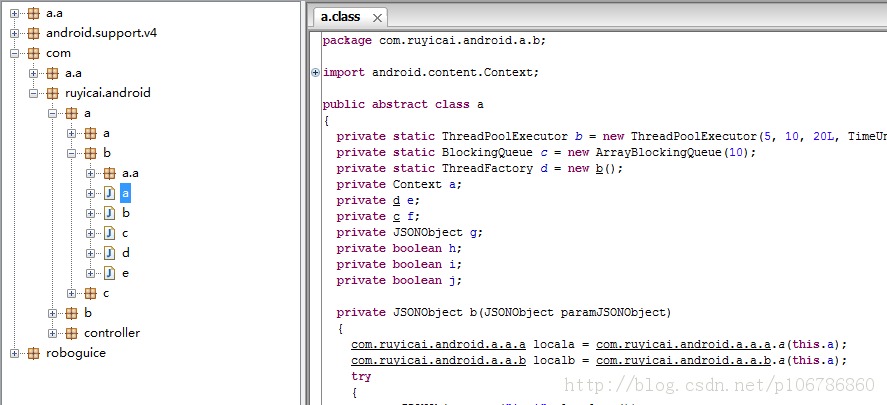
**3.修改项目的default.properties配置文件**

Android项目默认在打包的时候是不混淆代码的，所以我们必须得在default.properties文件中，添加该配置：proguard.config=proguard.cfg。



**4.进行apk打包**

好了，一切就绪，接下来我们就按着平常的打包步骤，进行打包操作。打完包后，我们使用上面讲述的反编译方法，查看现在的jar文件。



会发现，混淆了许多aa、bb、cc的包、aa、bb、cc的类和方法，这样的代码根本无法阅读，也就达到了我们保护源代码的目的了。

**三、produard.cfg文件**

我们在看看produard.cfg文件，该文件中描述了混淆代码的规则，你如果想根据自己的需求来混淆代码，就可以在本文件中编写你的混淆规则。一般情况下，该文件的内容如下：

-optimizationpasses 5

-dontusemixedcaseclassnames//混淆时不会产生形形色色的类名

-dontskipnonpubliclibraryclasses//指定不去忽略非公共的类库

-dontpreverify//不预校验

-verbose

-optimizations !code/simplification/arithmetic,!field/\*,!class/merging/\*//优化

//不进行混淆保持原样

-keep public class \* extends android.app.Activity

-keep public class \* extends android.app.Application

-keep public class \* extends android.app.Service

-keep public class \* extends android.content.BroadcastReceiver

-keep public class \* extends android.content.ContentProvider

-keep public class \* extends android.app.backup.BackupAgentHelper

-keep public class \* extends android.preference.Preference

-keep public class com.android.vending.licensing.ILicensingService

//保护指定的类和类的成员的名称，如果所有指定的类成员出席

-keepclasseswithmembernames class \* {

native <methods>;

}

//保护指定的类和类的成员，但条件是所有指定的类和类成员是要存在

-keepclasseswithmembers class \* {

public <init>(android.content.Context, android.util.AttributeSet);

}

-keepclasseswithmembers class \* {

public <init>(android.content.Context, android.util.AttributeSet, int);

}

//保护指定类的成员，如果此类受到保护他们会保护的更好

-keepclassmembers class \* extends android.app.Activity {

public void \*(android.view.View);

}

-keepclassmembers enum \* {

public static \*\*[] values();

public static \*\* valueOf(java.lang.String);

}

//保护指定的类文件和类成员

-keep class \* implements android.os.Parcelable {

public static final android.os.Parcelable$Creator \*;

}

-keep class \*\*.R$\* {

\*;

}

**四、可能出现的问题**

**1.**因为使用了升级后使用android-support-v4而发生混淆错误，无法打包出现如下错误：

Unexpected error while evaluating instruction:      Class       = [android/support/v4/view/AccessibilityDelegateCompat$AccessibilityDelegateJellyBeanImpl]      Method      = [newAccessiblityDelegateBridge(Landroid/support/v4/view/AccessibilityDelegateCompat;)Ljava/lang/Object;]   Instruction = [18] areturn    Exception   = 1 (Can't find any super classes of [android/support/v4/view/AccessibilityDelegateCompatIcs$1] (not even immediate super class [android/view/View$AccessibilityDelegate]))    Unexpected error while performing partial evaluation:      Class       = [android/support/v4/view/AccessibilityDelegateCompat$AccessibilityDelegateJellyBeanImpl]     Method      = [newAccessiblityDelegateBridge(Landroid/support/v4/view/AccessibilityDelegateCompat;)Ljava/lang/Object;]      Exception   = 1 (Can't find any super classes of [android/support/v4/view/AccessibilityDelegateCompatIcs$1] (not even immediate super class [android/view/View$AccessibilityDelegate]))   java.lang.IllegalArgumentException: Can't find any super classes of [android/support/v4/view/AccessibilityDelegateCompatIcs$1] (not even immediate super class [android/view/View$AccessibilityDelegate])      at proguard.evaluation.value.ReferenceValue.generalize(ReferenceValue.java:287)

你需要在proguard.cfg文件中添加如下代码：

-libraryjars   libs/android-support-v4.jar   
-dontwarn android.support.v4.\*\*     //不对错误提出警告。  
-keep class android.support.v4.\*\* { \*; }     
-keep interface android.support.v4.app.\*\* { \*; }     
-keep public class \* extends android.support.v4.\*\*     
-keep public class \* extends android.app.Fragment

**2.**如果你使用了第三方的包，你需要使用一下配置，让ProGuard知道库中的一些类并不是在所有的API中可用：

-libraryjars   libs/roboguice-2.0.jar

-dontwarn roboguice.\*\*

**3.**在打包的过程中，出现如下错误：

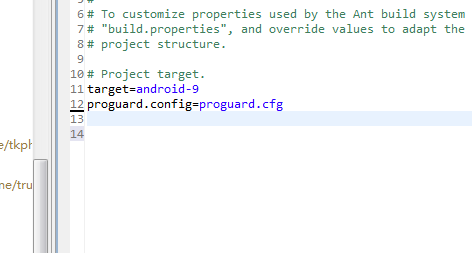
Proguard returned with error code 1. See console   
[2012-09-25 09:50:39 - estore3601] Error: @C:\Users\hp\AppData\Local\Temp\android\_5192919586673521494.pro (文件名、目录名或卷标语法不正确。)

你需要系在ProGuard最新版本，如v4.8。然后解压produard4.8.zip文件，将bin和lib两个文件夹覆盖sdk目录\tool\proguard目录中同名的文件夹。

[ndroid应用安全——代码安全（android代码混淆）](http://blog.csdn.net/xyz_lmn/article/details/8802785)

分类： [android 应用安全](http://blog.csdn.net/xyz_lmn/article/category/1395130)2013-04-15 19:28 4900人阅读 [评论](http://blog.csdn.net/xyz_lmn/article/details/8802785#comments)(1) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/xyz_lmn/article/details/8802785#report)

       android2.3的SDK开始在eclipse中支持代码混淆功能（理论上java都支持混淆，但关键在于如何编写proguard的混淆脚本，2.3的SDK使用简单的配置就可以实现混淆）。使用SDK2.3后，新建的工程下和之前相比，都会多了一个文件“proguard.cfg”。这个文件就是混淆所需的proguard脚本。在工程的"default.properties"中添加这样一句话“proguard.config=proguard.cfg”即可实现混淆（如下图）。

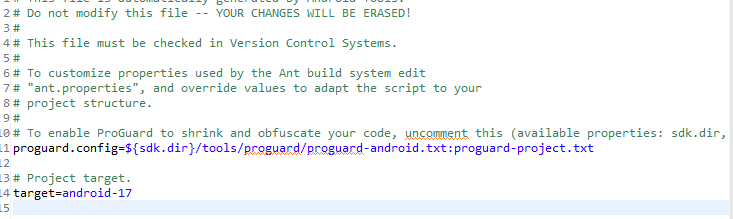


       android4.0SDK，ADT14或者更高版本，项目中没有default.properties文件，只有project.properties文件，找到project.properties文件即可。该文件中有以下两行：

**[html]** [view plaincopyprint?](http://blog.csdn.net/xyz_lmn/article/details/8802785)

1. # To enable ProGuard to shrink and obfuscate your code, uncomment this (available properties: sdk.dir, user.home):
2. # proguard.config=${sdk.dir}/tools/proguard/proguard-android.txt:proguard-project.txt

              根据这段说明，只要将proguard.config前面的#去掉（下图），就可以利用ProGuard来混淆代码了！



      默认的设置是不带优化功能的，下面代码可优化混淆：

**[html]** [view plaincopyprint?](http://blog.csdn.net/xyz_lmn/article/details/8802785)

1. proguard.config=${sdk.dir}/tools/proguard/proguard-android-optimize.txt:proguard-project.txt

proguard 配置

最常用的配置选项  
-dontwarn 缺省proguard 会检查每一个引用是否正确，但是第三方库里面往往有些不会用到的类，没有正确引用。如果不配置的话，系统就会报错。  
-keep 指定的类和类成员被保留作为 入口 。  
-keepclassmembers 指定的类成员被保留。  
-keepclasseswithmembers 指定的类和类成员被保留，假如指定的类成员存在的话。

使用的JNI、jar时更应该考虑混淆的配置。