

区长的专栏

业精于勤，荒于嬉；行成于思，毁于随

目录视图

摘要视图

RSS 订阅

个人资料



区长

访问：415527次

积分：6516

等级：BLOG 6

排名：第2466名

原创：157篇

转载：5篇

译文：0篇

评论：526条

博客专栏



Android React Native

文章：9篇

阅读：31914



Android中的设计模式

文章：5篇

阅读：11654



Android开发最佳实践

文章：8篇

阅读：49798



Android ORM系列

文章：5篇

阅读：14814



Android插件化开发OpenAtlas

文章：8篇

阅读：22145

相关信息

我的Github

我的知乎

【公告】博客系统优化升级

Unity3D学习，离VR开发还有一步

博乐招募开始啦

虚拟现实，一探究竟

Android Studio使用新的Gradle构建工具配置NDK环境

标签：Android NDK experiment Gradle JNI

2015-09-15 16:33 11657人阅读 评论(1) 收藏 举报

分类：Android NDK开发 (5) Android 基础 (142)

版权声明：本文为博主原创文章，未经博主允许不得转载。

从Android Studio 1.3 Beta1开始，就支持了NDK。不过使用的是一个全新的实验性的gradle构建工具。官方地址<http://tools.android.com/tech-docs/new-build-system/gradle-experimental>，英文好的可以直接看原文。

在进行配置前请先将Android Studio更新至最新版本，目前最新版本是1.4 Bata4



并且使用Android Studio内置的SDK管理器下载NDK。

http://blog.csdn.net/sbsujjbcy/article/details/48469569

1/8

我的微博

微博



—区长 浙江 杭州

加关注

转需[嘻嘻]//@ImportNew://@不买饼干的饼干:我还是赶紧先把我那个服务器的用户给弄成非root用户吧,就算我能保证我自己不乱来,我走了之后接手的人我估计绝对会乱来.....

伯乐在线官方微博

: 《一次心惊肉跳的服

TA 的粉丝 (520)

全部»



MikeZhar



蔡宇clyb



Kevin\_Cu



傻子说将

文章搜索

文章分类

【React Native Android】 (9)

【Android 插件化开发】 (19)

【Android 新技术预研】 (49)

【Android 数据持久化】 (7)

【Android 网络请求】 (19)

【Android 设计模式】 (6)

【Android NDK开发】 (6)

【Android 源码解析】 (6)

【Android 常见错误】 (9)

【Android 系统编译】 (10)

【Android UI开发】 (34)

【Android 框架】 (8)

【Android 基础】 (143)

【Linux 系统运维】 (17)

【Java 开发】 (8)

【wamp 常见处理】 (3)

【Gradle】 (1)

【Docker】 (1)

阅读排行

Android Studio使用新的 (11654)

Android OkHttp文件上传 (11538)

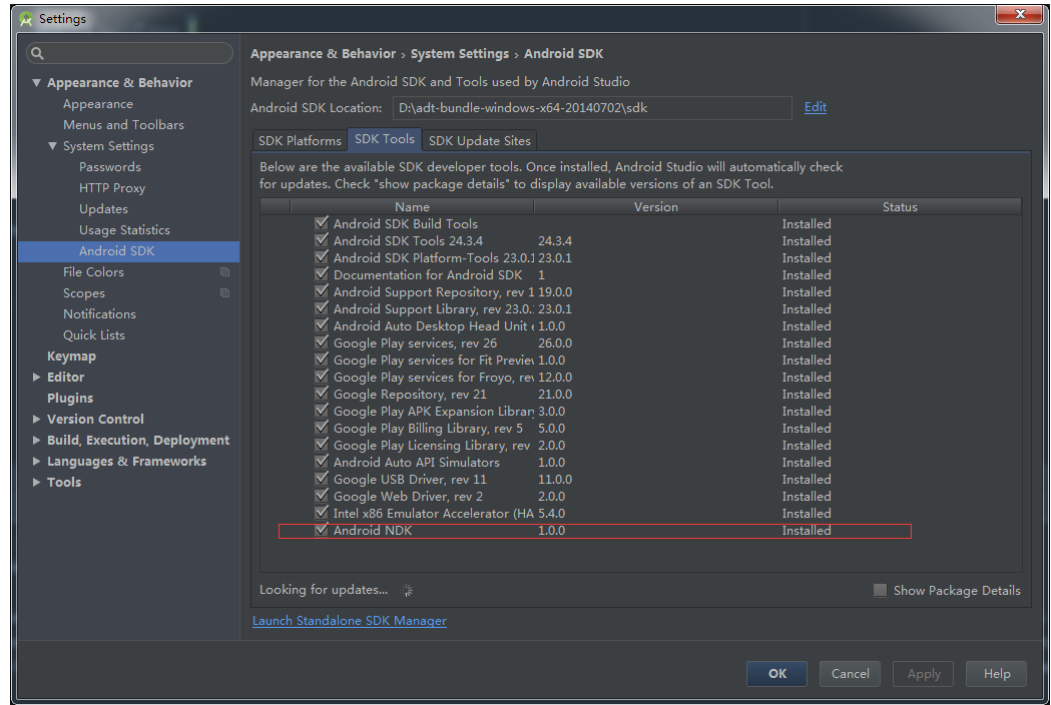
Android 各大网络请求库 (10866)

Android 实现变色状态栏 (9185)

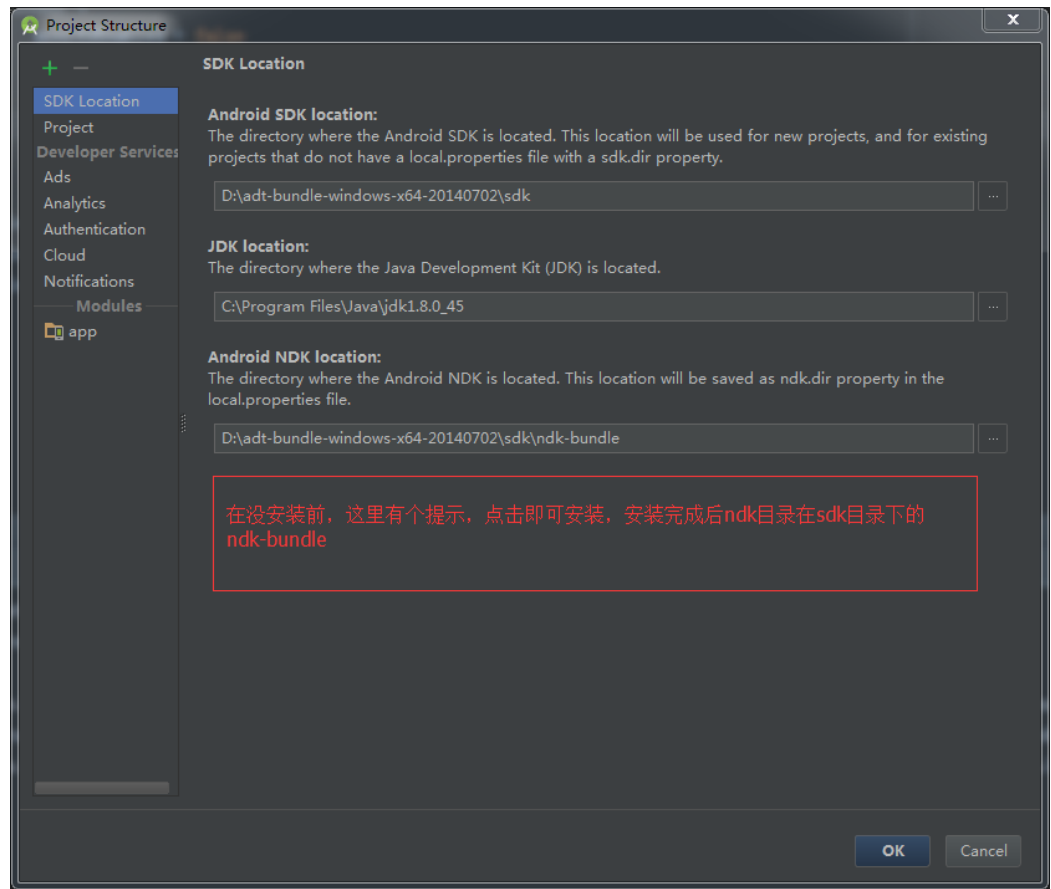
android 项目实战——打 (8616)

Android FastJson与不规 (8510)

Android 实现沉浸式状态 (8349)



或者在项目上右键打开Project Structure，切到的SDK Location页进行安装



注意这个NDK安装好后其目录在SDK目录下的ndk-bundle目录下，与以往的不同。

要修改的文件见下图

- Android 识别二代身份证 (7339)
- Android 使用OpenCV的 (7205)
- Android 热修复Nuwa的 (6750)

推荐文章

\* 致JavaScript也将征服的物联网世界

\* 从苏宁电器到卡巴斯基：难忘的三年硕士时光

\* 作为一名基层管理者如何利用情商管理自己和团队（一）

\* Android CircleImageView圆形ImageView

\* 高质量代码的命名法则

最新评论

Android开发中无处不在的设计模  
jujishou521: 好

Android 识别二代身份证证件信  
q59200182: 为什么有些手机上  
面照相时黑的 打不开呢

在Android中使用Lombok减少编  
福临天下: 个人感觉确实挺好的，  
你在实际项目中用了吗？为什  
么？谢谢！

Android 实现沉浸式状态栏  
offbye: 总结的不错， 但  
Android5又有了些变化

Android开发中无处不在的设计模  
singsong: 楼主写的很好，学习  
了。

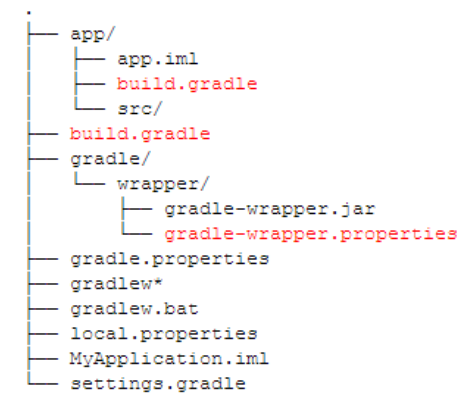
Android 使用OpenCV的三种方式  
寄意兰州: 好像.so结尾的都是动  
态库，第一种方法说是静态库，  
有点疑问

Android React Native组件的生  
于连林520wcf: 不错

Android 实现变色状态栏  
ApplyDevelop: 博主你好，我是  
新大二在校学生，初学android。  
请问您说：但是我们并不想它移  
动到上面，只是想它状态...

android 项目实战——打造超级  
suilie5112: 楼主，我也是改好后  
总是提示账号或密码错误，调试  
发现验证登录成功时传的值不对  
没有xhxm

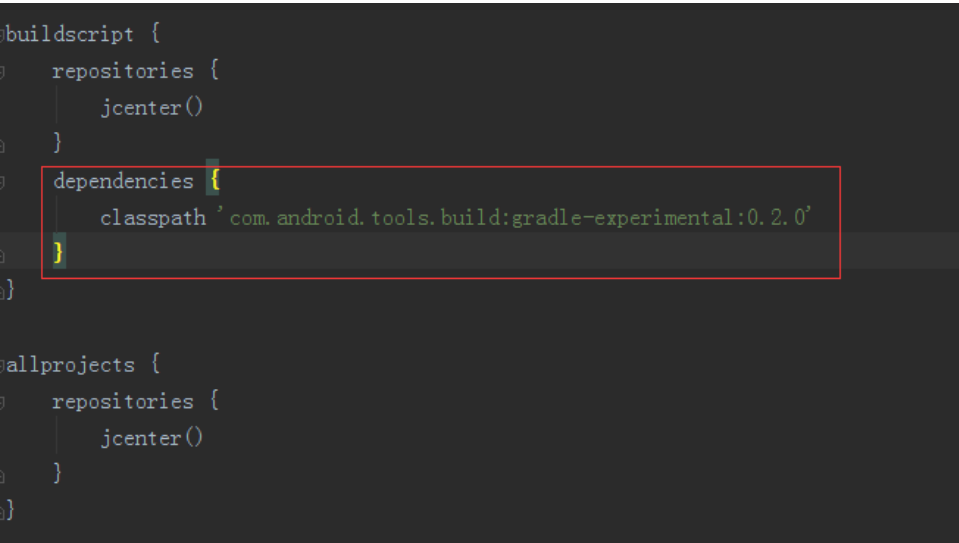
android 项目实战——打造超级  
suilie5112: @u012605830:你解  
决了吗？什么原因？



首先需要将根目录下的gradle修改为实验性的gradle

```
1 classpath 'com.android.tools.build:gradle-experimental:0.2.0'
```

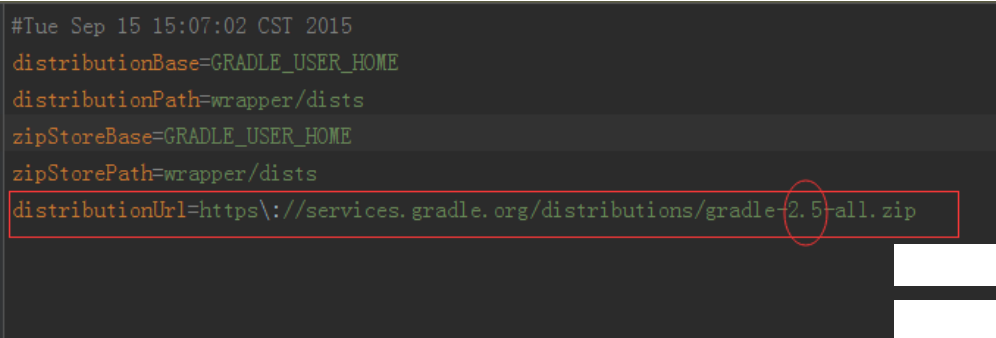
最终是这样的



并且需要gradle 2.5的支持，将gradle/gradle-wrapper.properties中的版本修改为2.5

```
1 distributionUrl=https\://services.gradle.org/distributions/gradle-2.5-all.zip
```

最终是这样的



然后修改原来的app/build.gradle，注意现在各项配置都是有等号的，并且使用的是com.android.model.application而不再是原来的com.android.application，如果是Library，则是com.android.model.library。然后android外包了一层model。原来在android里面的buildTypes 移到了外面。等等，看下面内容，注意变化。

```
1 apply plugin: 'com.android.model.application'
2
3
4 model{
5     android {
```

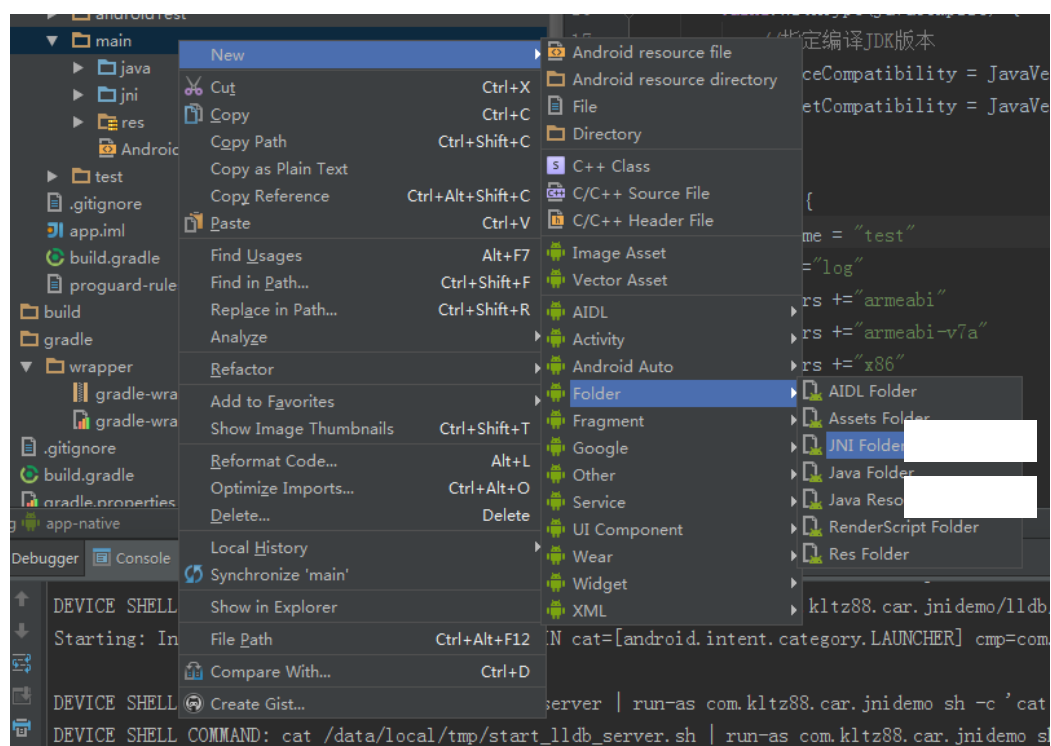
```

6      compileSdkVersion = 22
7      buildToolsVersion = "23.0.1"
8
9      defaultConfig.with {
10         applicationId = "com.kltz88.car.jnidemo"
11         minSdkVersion.apiLevel = 14
12         targetSdkVersion.apiLevel = 22
13         versionCode = 1
14         versionName = "1.0"
15     }
16     tasks.withType(JavaCompile) {
17         //指定编译JDK版本
18         sourceCompatibility = JavaVersion.VERSION_1_7
19         targetCompatibility = JavaVersion.VERSION_1_7
20     }
21 }
22 android.ndk {
23     moduleName = "test"
24     ldLibs += "log"
25     abiFilters += "armeabi"
26     abiFilters += "armeabi-v7a"
27     abiFilters += "x86"
28 }
29 android.buildTypes {
30     release {
31         minifyEnabled = false
32         proguardFiles += file('proguard-rules.pro')
33     }
34 }
35 }
36
37 dependencies {
38     compile fileTree(dir: 'libs', include: ['*.jar'])
39     compile 'com.android.support:appcompat-v7:22.2.1'
40 }

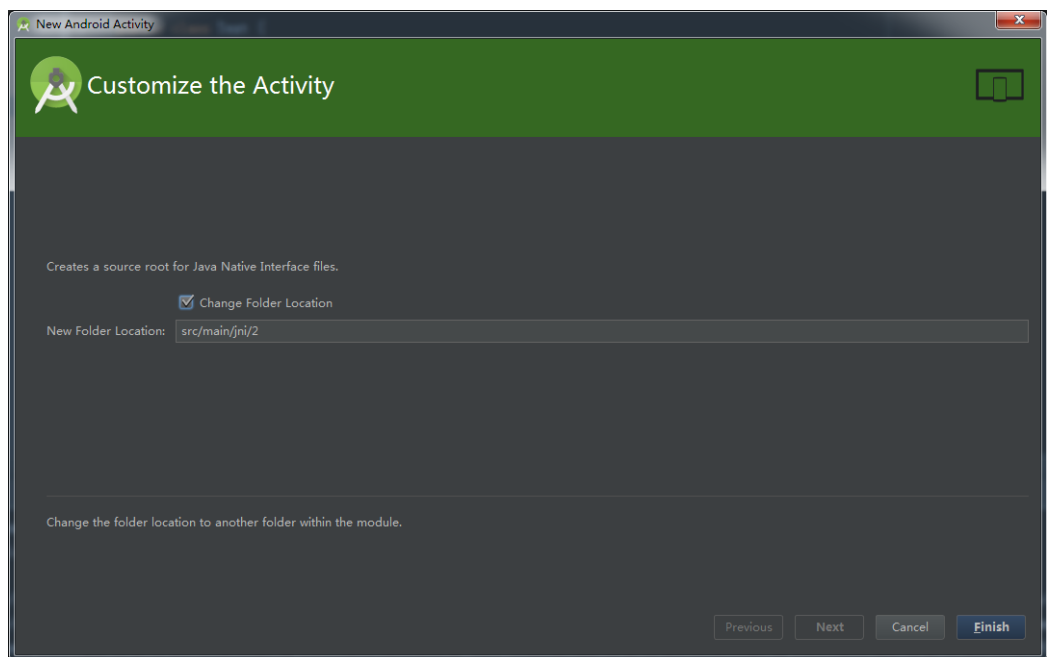
```

在android.ndk中进行ndk的配置，模块名字是必须的。然后使用了log，并且支持armeabi，armeabi-v7a，x86三个平台。

然后增加jni目录，在main上右键新建jni目录



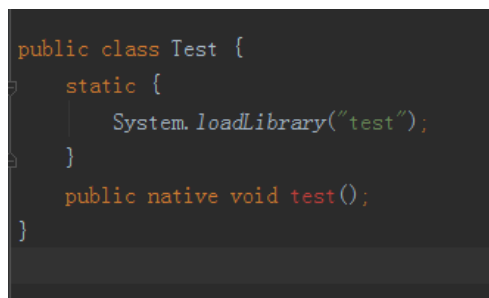
在弹出的窗口中你可以修改jni目录，默认是在main/jni中



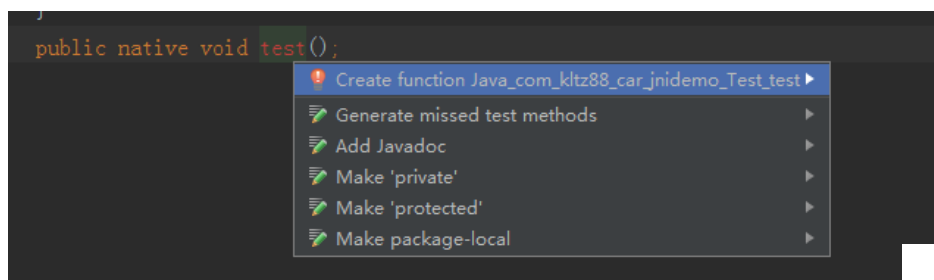
新建一个Test的类，加载jni库，声明native方法

```
1 public class Test {  
2     static {  
3         System.loadLibrary("test");  
4     }  
5     public native void test();  
6 }
```

这时候你会发现test方法是红色的



在上面按alt+enter生成c层的方法



修改为下方的代码

```
1 #include <jni.h>  
2 #include "android/log.h"  
3 JNIEXPORT void JNICALL  
4 Java_com_kltz88_car_jnidemo_Test_test(JNIEnv *env, jobject instance) {  
5     __android_log_write(ANDROID_LOG_ERROR, "TAG", "TEST");  
6 }  
7 }
```

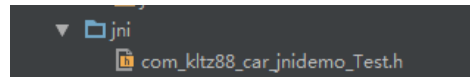
如果你需要生成头文件，使用命令行即可

```
D:\code\JniDemo>cd app/src/main/java

D:\code\JniDemo\app\src\main\java>javah -d ../jni com.kltz88.car.jnidemo.Test

D:\code\JniDemo\app\src\main\java>
```

之后会在jni目录生成对应的头文件



然后在Java层调用

```
1 public class MainActivity extends AppCompatActivity {
2     @Override
3     protected void onCreate(Bundle savedInstanceState) {
4         super.onCreate(savedInstanceState);
5         setContentView(R.layout.activity_main);
6         Test t=new Test();
7         t.test();
8     }
9 }
```

便会看到日志的输出。

```
09-15 16:02:26.515 29012-29012/com.kltz88.car.jnidemo D/dalvikvm: Late-enabling CheckJNI
09-15 16:02:26.754 29012-29012/com.kltz88.car.jnidemo D/dalvikvm: GetMethodID: not returning static method Landroid/os/Process;
09-15 16:02:26.754 29012-29012/com.kltz88.car.jnidemo D/dalvikvm: GetMethodID: not returning static method Landroid/os/Process;
09-15 16:02:26.814 29012-29012/com.kltz88.car.jnidemo D/dalvikvm: Trying to load lib /data/app-lib/com.kltz88.car.jnidemo-2/lib
09-15 16:02:26.816 29012-29012/com.kltz88.car.jnidemo D/dalvikvm: Added shared lib /data/app-lib/com.kltz88.car.jnidemo-2/libte
09-15 16:02:26.816 29012-29012/com.kltz88.car.jnidemo D/dalvikvm: No JNI_OnLoad found in /data/app-lib/com.kltz88.car.jnidemo-2
09-15 16:02:26.816 29012-29012/com.kltz88.car.jnidemo E/TAG: TEST
09-15 16:02:26.840 29012-29012/com.kltz88.car.jnidemo D/libEGL: loaded /system/lib/egl/libEGL_tegra.so
09-15 16:02:26.840 29012-29012/com.kltz88.car.jnidemo I/: Attempting to load EGL implementation /system/lib/egl/libEGL_tegra_im
09-15 16:02:26.846 29012-29012/com.kltz88.car.jnidemo I/: Loaded EGL implementation /system/lib/egl/libEGL_tegra_impl
09-15 16:02:26.847 29012-29012/com.kltz88.car.jnidemo D/libEGL: loaded /system/lib/egl/libGLESv1_CM_tegra.so
09-15 16:02:26.856 29012-29012/com.kltz88.car.jnidemo D/libEGL: loaded /system/lib/egl/libGLESv2_tegra.so
09-15 16:02:26.931 29012-29012/com.kltz88.car.jnidemo D/OpenGLRenderer: Enabling debug mode 0
```

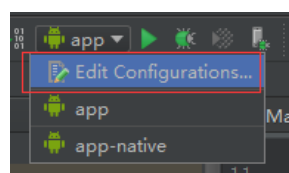
注意如果没有在gradle中指定java的对应版本，运行时可能会报错

```
1 tasks.withType(JavaCompile) {
2     //指定编译JDK版本
3     sourceCompatibility = JavaVersion.VERSION_1_7
4     targetCompatibility = JavaVersion.VERSION_1_7
5 }
```

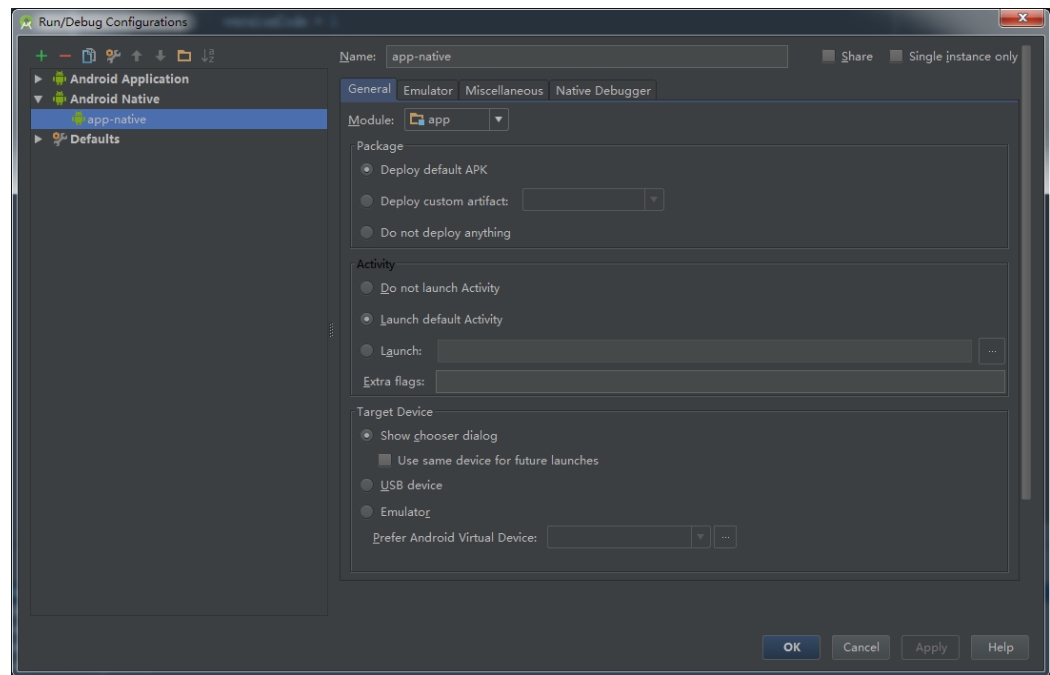
错误信息如下，遇到该错误加上上面的代码即可

```
UNEXPECTED TOP-LEVEL EXCEPTION:
com.android.dx.cf.iface.ParseException: bad class file magic (cafebabe) or version (0034.0000)
at com.android.dx.cf.direct.DirectClassFile.parse0(DirectClassFile.java:472)
at com.android.dx.cf.direct.DirectClassFile.parse(DirectClassFile.java:406)
at com.android.dx.cf.direct.DirectClassFile.parseToInterfacesIfNecessary(DirectClassFile.java:388)
at com.android.dx.cf.direct.DirectClassFile.getMagic(DirectClassFile.java:251)
at com.android.dx.command.dexer.Main.parseClass(Main.java:764)
at com.android.dx.command.dexer.Main.access$1500(Main.java:85)
at com.android.dx.command.dexer.Main$ClassParserTask.call(Main.java:1684)
at com.android.dx.command.dexer.Main.processClass(Main.java:749)
... 21 more
UNEXPECTED TOP-LEVEL EXCEPTION:
```

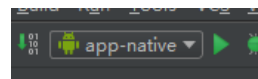
最后就是调试的问题了，理论上来说应该是支持调试的，但是我在windows上并没有试验成功。下面贴出步骤。



增加native运行方法



修改为



然后下断点，理论上是能调试的，但是我遇到了下面的问题，导致无法继续，也查过了，查不出问题所在，就这样吧，改天在linux下试试是不是正常。

```
DEVICE SHELL COMMAND: run-as com.klts88.car.jnidemo mkdir /data/data/com.klts88.car.jnidemo/11d
DEVICE SHELL COMMAND: run-as com.klts88.car.jnidemo mkdir /data/data/com.klts88.car.jnidemo/11d/bin
DEVICE SHELL COMMAND: cat /data/local/tmp/11d-server | run-as com.klts88.car.jnidemo sh -c 'cat > /data/data/com.klts88.car.jnidemo/11d/bin/11d-server'
DEVICE SHELL COMMAND: cat /data/local/tmp/start_11d_server.sh | run-as com.klts88.car.jnidemo sh -c 'cat > /data/data/com.klts88.car.jnidemo/11d/bin/start_11d_server.sh; chmod 700 /data/data/com.klts88.car.jnidemo/11d/bin
Starting 11d server: run-as com.klts88.car.jnidemo /data/data/com.klts88.car.jnidemo/11d/bin/start_11d_server.sh /data/data/com.klts88.car.jnidemo/11d /data/data/com.klts88.car.jnidemo/11d/tmp/platform.port1442304421145
Now Launching Native Debug Session
Failed to attach native debugger: Device "[81d6a949]" not found
Device "[91d6a949]" not found
```

你会发现整个流程下来，原来的Android.mk和Application.mk都不再需要了，所有配置在gradle中配置即可。

官方的github示例代码  
<https://github.com/googlesamples/android-ndk>

我的示例代码  
<http://download.csdn.net/detail/sbsujjbcy/9108775>

顶 踩  
5 2

上一篇 Android ORM系列之GreenDao自定义类型转换器与代码混淆配置  
下一篇 Android NDK开发之JNI基础知识

我的同类文章

【Android NDK开发】（5）			【Android 基础】（142）		
• Android OpenCV中的几种...	2015-11-02	阅读 1125	• Android 使用OpenCV的三...	2015-10-30	阅读 7158
• Android NDK开发之JNI基础...	2015-09-16	阅读 2015	• android studio下ndk开发	2015-04-15	阅读 1744
• NDK开发环境	2013-07-30	阅读 1079			



参考知识库



**Android**知识库  
12674 关注 | 1190 收录



**Java EE**知识库  
1655 关注 | 618 收录



**Java SE**知识库  
9868 关注 | 454 收录



**Java Web**知识库  
10171 关注 | 1074 收录

猜你在找

- Android APP开发之真机调试环境实现
- Android开发之Android Studio环境搭建视频教程
- 【Android APP开发】Android高级商业布局快速实现
- Android应用的调试
- Android 5.x顶级视频教程
- 深入理解Android之Gradle
- 深入理解Android之Gradle主要是gradle语法相关
- Android应用开发编译框架流程与IDE及Gradle概要
- Android面试必过Android常见的问题
- Android 开源项目分类汇总



查看评论

1楼 邵鸿鑫 2015-11-22 21:27发表



写的真好呢，学习了

您还没有登录,请[登录](#)或[注册](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

- 全部主题
- Hadoop
- AWS
- 移动游戏
- Java
- Android
- iOS
- Swift
- 智能硬件
- Docker
- OpenStack
- VPN
- Spark
- ERP
- IE10
- Eclipse
- CRM
- JavaScript
- 数据库
- Ubuntu
- NFC
- WAP
- jQuery
- BI
- HTML5
- Spring
- Apache
- .NET
- API
- HTML
- SDK
- IIS
- Fedora
- XML
- LBS
- Unity
- Splashtop
- UML
- components
- Windows Mobile
- Rails
- QEMU
- KDE
- Cassandra
- CloudStack
- FTC
- coremail
- OPhone
- CouchBase
- 云计算
- iOS6
- Rackspace
- Web App
- SpringSide
- Maemo
- Compuware
- 大数据
- aptech
- Perl
- Tornado
- Ruby
- Hibernate
- ThinkPHP
- HBase
- Pure
- Solr
- Angular
- Cloud Foundry
- Redis
- Scala
- Django
- Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持  
京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved