

CSDN新首页上线啦，邀请你来立即体验！(http://feed.csdn.net/)

立即体

CSDN

博客 (http://feed.csdn.net/?ref=toolbar)

学院 (http://edu.csdn.net?ref=toolbar)

下载 (http://download.csdn.net?ref=toolbar)

更多 ▾

搜索



登录 (https://passport.csdn.net/account/login?ref=toolbar)

注册 (http://passport.csdn.net/account/mobileregister?ref=toolbar&action=mobileRegister)

(http://passport.csdn.net/account/mobileregister?ref=toolbar&action=mobileRegister)

ref=toolbar\_utm\_source=csdnblog1)

7



## android studio 使用CMAKE

转载

2016年11月11日 23:50:07



标签：android studio (http://so.csdn.net/so/search/s.do?q=android studio&t=blog) /



cmake (http://so.csdn.net/so/search/s.do?q=cmake&t=blog) /



c语言 (http://so.csdn.net/so/search/s.do?q=c语言&t=blog)

2745

原文地址 (http://blog.csdn.net/xiaoyu\_93/article/details/53082088)

### 前言

之前，每次需要边写C++代码的时候，我的内心都是拒绝的。

1. 它没有代码提示！！这意味着我们必须自己手动敲出所有的代码，对于一个新手来说，要一个字母都不错且大小写也要正确，甚至要记得住所有的jni函数等，真是太折磨人了...平时写Java (http://lib.csdn.net/base/javaee)代码的时候都是写几个字母会出来一大堆提示然后选择的，这样还有一个好处就是很多时候我们不知道有那些函数，但是我们可以通过 obj. ，然后就可以看到它有哪些方法函数了。

2. 很多地方会显示为红色，就像是错误提示的那种，当然，如果没错的话还是能编译运行的，但是如果像编写java代码一样，写错的地方才给我红色显示不是更好吗？

这两个问题可折磨死我这个强迫症了...在网上百度了很久如何能够使用Android (http://lib.csdn.net/base/android) Studio编



Vasil (http://blog.csdn.ne...)

+ 关注

(http://blog.csdn.net/qq\_25817651)

码云

原创

粉丝

喜欢

未开通  
(https://gite  
utm\_sourc

13

8

0

### 他的最新文章

更多文章 (http://blog.csdn.net/qq\_25817651)

Android Studio 3.0新建工程问题

(/qq\_25817651/article/details/73437162)

android ImageLoader实现图像的三级缓存和代码结构优化

(/qq\_25817651/article/details/53611228)

android Camera2 API使用详解

(/qq\_25817651/article/details/53573340)

BottomNavigationView 实现

(/qq\_25817651/article/details/53147938)

写C++代码时有代码提示自动补全功能，但是一直没有找到有效的（不使用CMake的情况下，若有人知道，麻烦告知）。好在一次偶然的机会，在网上看到在 Android Studio 2.2 中愉快地使用 C/C++ (http://blog.csdn.net/wl9739/article/details/52607010)这篇文章，看到上面说Android Studio2.2版本完善了对C/C++的支持，还提及到CMAKE，赶紧把Android Studio从2.0版本升级到了2.2版本，按照介绍新建了一个NDK工程，发现里面居然有我梦寐以求的C++代码自动补全提示！！！喜大普奔！！！

赶紧了解学习下Android Studio2.2版本提供的CMAKE方式。打算以后都使用这种方式了，别的不说，就冲可以有C++代码自动补全提示这个功能我就爱上它了。



## 7 工具安装



### 在线课程



MNO/M在美国点评酒旅移动端的最佳实践  
(http://blog.csdn.net/huiyi/course/detail/603?utm\_source=blog9)  
(http://blog.csdn.net/huiyi/course/detail/603?utm\_source=blog9)



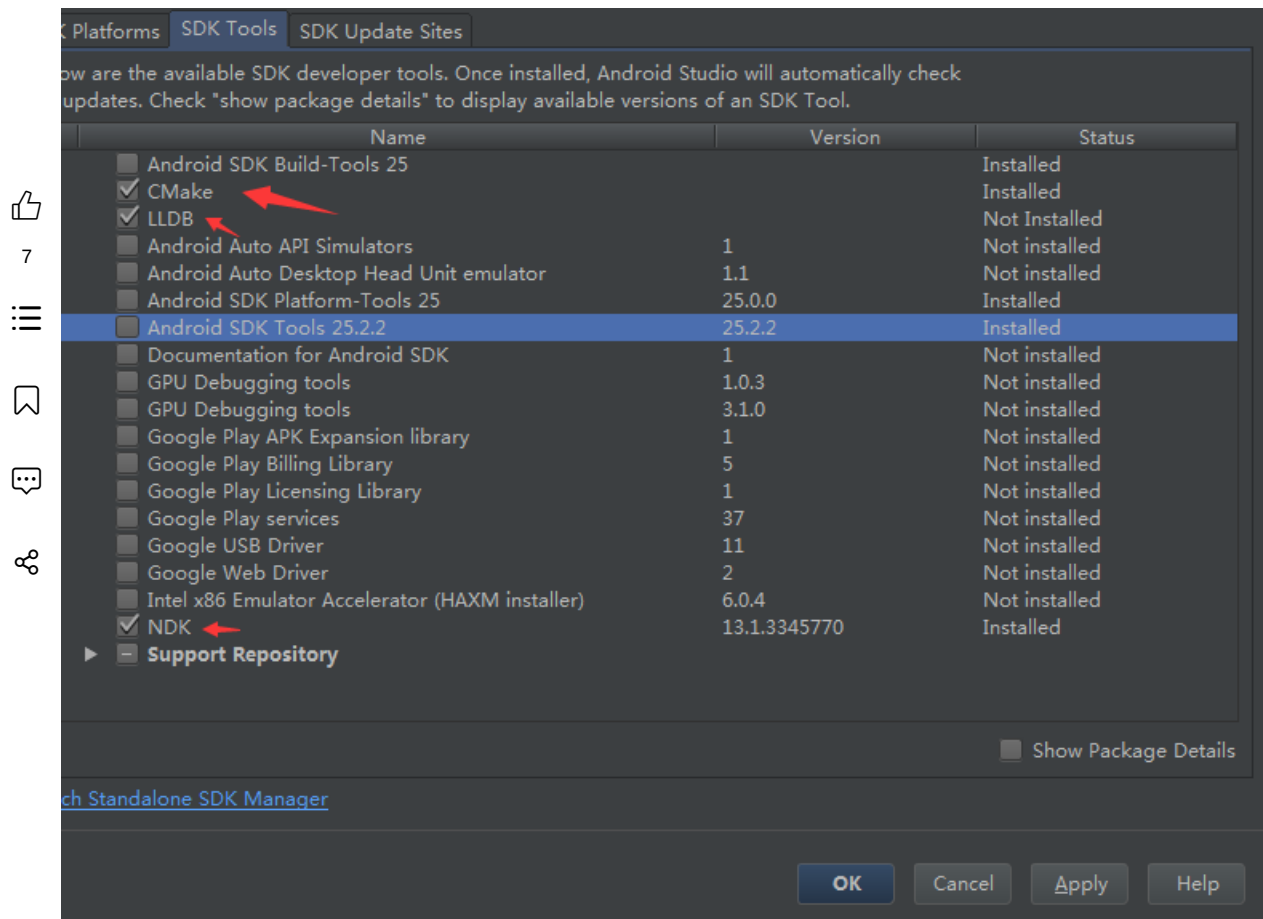
C语言大型软件设计的面向对象  
(http://blog.csdn.net/huiyi/course/detail/594?utm\_source=blog9)  
(http://blog.csdn.net/huiyi/course/detail/594?utm\_source=blog9)

### 热门文章

百度地图API解决 android 6.0定位不了的问题 (/qq\_25817651/article/details/50913055)

12684

Android Studio升级到2.2之后，我们可以先配置好NDK开发的一些所需工具，如图，在SDK Tools中勾选安装CMake、LLDB、 NDK。



CMake: 外部构建工具。如果你准备只使用 ndk-build 的话，可以不使用它。  
LLDB: Android Studio上面调试本地代码的工具。

## Android Studio自带DEMO了解CMAKE

[android Camera2 API使用详解 \(/qq\\_25817651/article/details/53573340\)](#)

4838

[android studio下使用TUTK SDK \(/qq\\_25817651/article/details/52252250\)](#)

2879

[android studio 使用CMAKE \(/qq\\_25817651/article/details/53135685\)](#)

2741

[android studio中使用jack编译器 \(/qq\\_25817651/article/details/51627113\)](#)

2531

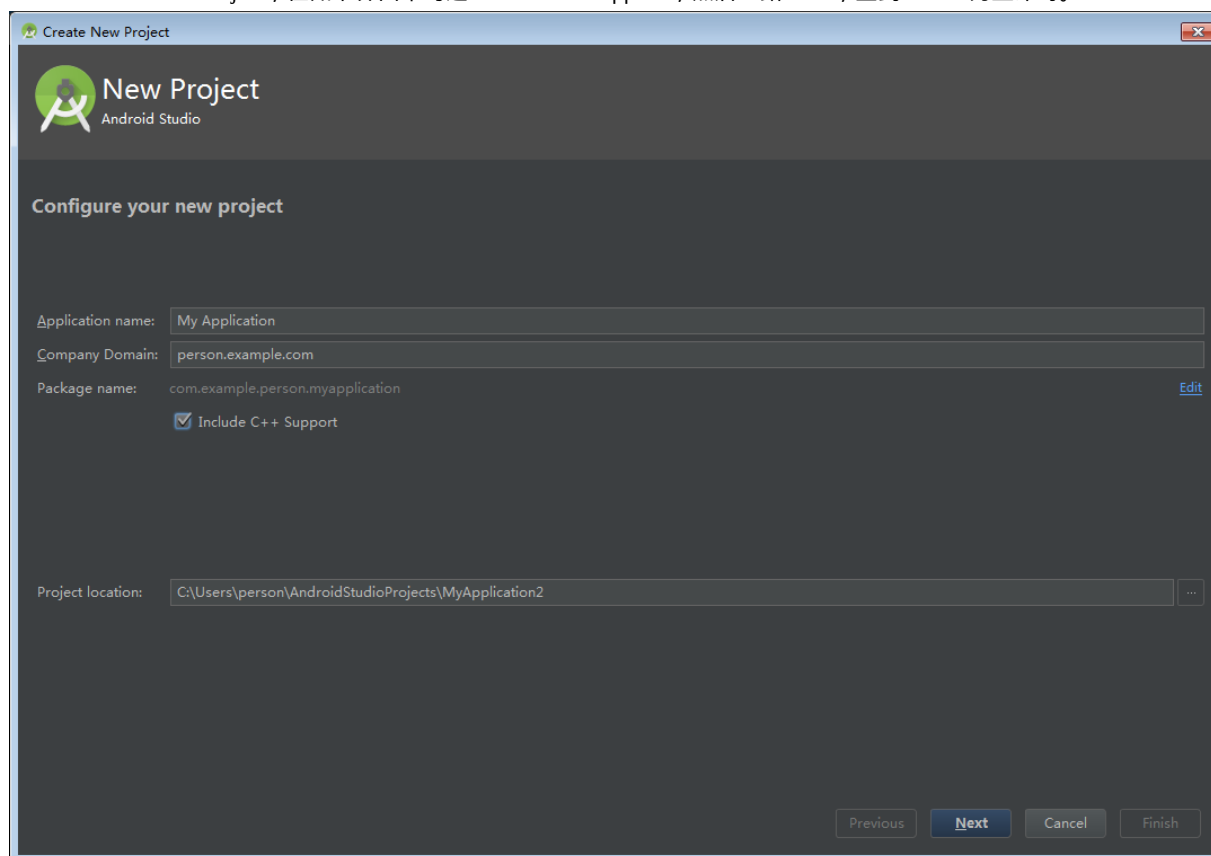
Android Studio升级到2.2版本之后，在创建新的project时，界面上多了一个 Include C++ Support 的选项。勾选它之后将会创建一个默认的C++与JAVA混编的Demo程序。就让我们先来看看这个官方标准Demo吧。

开始之前最好先下载好NDK，见NDK开发 从入门到放弃(一：基本流程入门了解) ([http://blog.csdn.net/xiaoyu\\_93/article/details/52870395](http://blog.csdn.net/xiaoyu_93/article/details/52870395))，即在Project Structure界面Android NDK location处下载或选择正确的路径。或者使用上方提供的工具安装方法来进行下载。否则，创建的新project也会报错，需要配置好后clean。



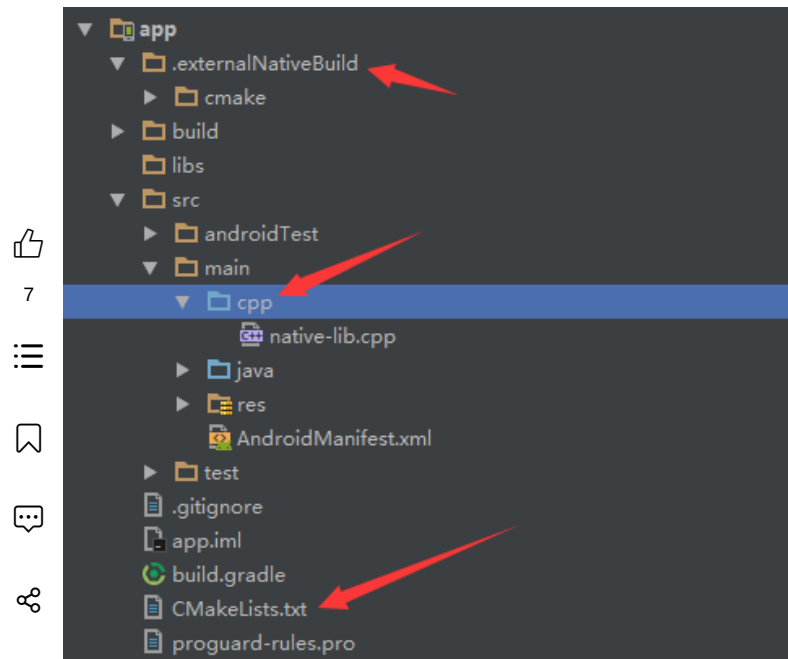
7

File -> New -> New Project，在如下界面中勾选 Include C++ Support ，然后一路 Next ，直到 Finish 为止即可。



项目打开后我们查看目录结构，与常规项目不同的是多了.externalNativeBuild文件夹、cpp文件夹、CMakeLists.txt文件，如

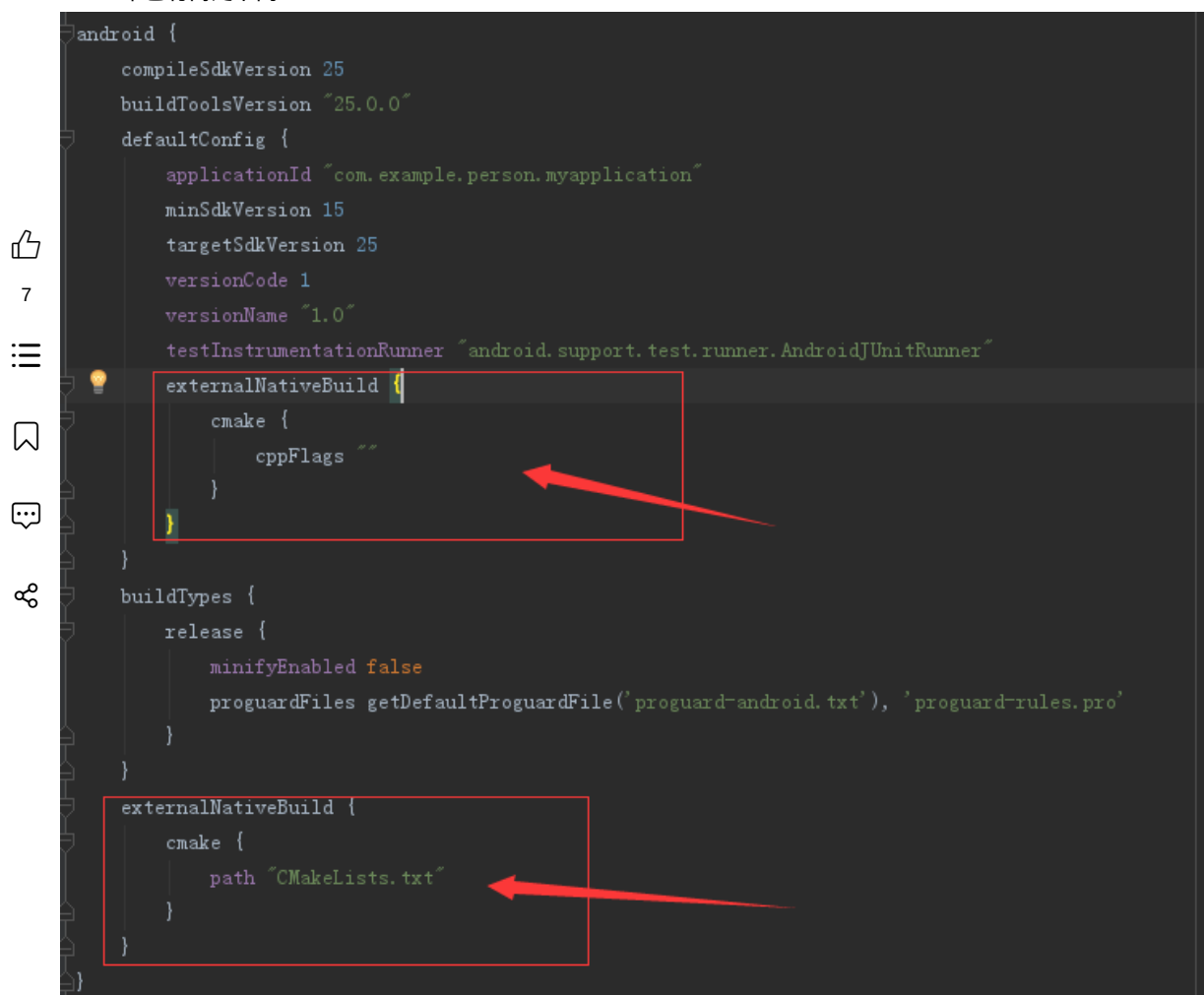
下图：



这三个东西都是NDK部分：

1. .externalNativeBuild文件夹：cmake编译好的文件, 显示支持的各种硬件等信息。系统生成。
2. cpp文件夹：存放C/C++代码文件，native-lib.cpp文件是该Demo中自带的，可更改。需要自己编写。
3. CMakeLists.txt文件：CMake脚本配置的文件。需要自己配置编写。

Gradle中也有两处不同：



java代码：



7



```

• 1 public class MainActivity extends AppCompatActivity {
• 2
• 3     @Override
• 4     protected void onCreate(Bundle savedInstanceState) {
• 5         super.onCreate(savedInstanceState);
• 6         setContentView(R.layout.activity_main);
• 7
• 8         // Example of a call to a native method
• 9         TextView tv = (TextView) findViewById(R.id.sample_text);
• 10        tv.setText(stringFromJNI());
• 11    }
• 12
• 13    /**
• 14     * A native method that is implemented by the 'native-lib' native library,
• 15     * which is packaged with this application.
• 16     */
• 17    public native String stringFromJNI();
• 18
• 19    // Used to load the 'native-lib' library on application startup.
• 20    static {
• 21        System.loadLibrary("native-lib");
• 22    }
• 23}

```

从native-lib.cpp的代码中我们能看到它使用的是静态注册的方式，动态注册的方式代码同传统JNI。

```

• 1 #include <jni.h>
• 2 #include <string>
• 3
• 4 extern "C"
• 5 jstring
• 6 Java_com_example_person_myapplication_MainActivity_stringFromJNI(
• 7     JNIEnv* env,
• 8     jobject /* this */) {
• 9     std::string hello = "Hello from C++";
• 10    return env->NewStringUTF(hello.c_str());
• 11}

```

CMakeLists.txt文件中我们需要注意的下面这三个地方。两个library的名字（需一致）以及一个cpp文件的路径，彼此需要对应一致，当我们自己定义library以及自己创建cpp文件时需要对应修改。

```
# Creates and names a library, sets it as either STATIC
# or SHARED, and provides the relative paths to its source code.
# You can define multiple libraries, and CMake builds it for you.
# Gradle automatically packages shared libraries with your APK.

add_library( # Sets the name of the library.
             native-lib

             # Sets the library as a shared library.
             SHARED

             # Provides a relative path to your source file(s).
             # Associated headers in the same location as their source
             # file are automatically included.
             src/main/cpp/native-lib.cpp )

# Searches for a specified prebuilt library and stores the path as a
# variable. Because system libraries are included in the search path by
# default, you only need to specify the name of the public NDK library
# you want to add. CMake verifies that the library exists before
# completing its build.

find_library( # Sets the name of the path variable.
             log-lib

             # Specifies the name of the NDK library that
             # you want CMake to locate.
             log )

# Specifies libraries CMake should link to your target library. You
# can link multiple libraries, such as libraries you define in the
# build script, prebuilt third-party libraries, or system libraries.
```



```
target_link_libraries( # Specifies the target library.
    native-lib

    # Links the target library to the log library
    # included in the NDK.
    ${log-lib} )
```



运行代码，就能看到效果，调用了C++方法在界面上显示了 Hello from C++ 字符串。这就是CMake方式进行NDK开发的Demo。

7

还记得传统JNI方式中吗？我们使用了ndk-build来编译C/C++文件为so文件，对于使用过第三方库的开发者来说，对so肯定不陌生。我们只能使用别人给的so文件，而无法看到C/C++源码，更别说是修改了。但是在这里我们好像一直没看到so文件的影子，那么，我们安装运行的apk中，有对应的so文件吗？如果想验证一下apk是否有so文件，我们可以使用 APK



Analyzer:



1. 选择 Build > Analyze APK。



2. 选择 apk，并点击 OK。

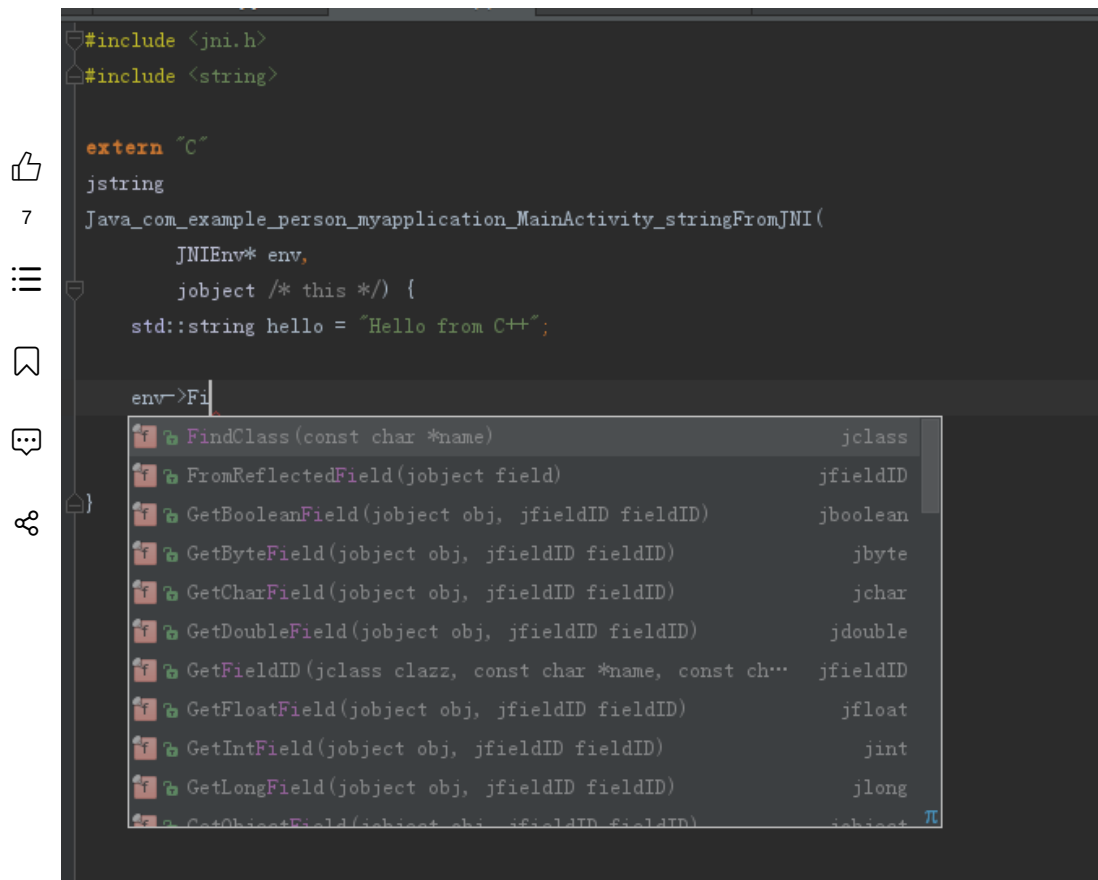
3. 如下图，在 APK Analyzer 窗口中，选择 lib/x86/，可以看见 libnative-lib.so 。



File	Raw File Size	Download Size	% of Total Download
classes.dex	2.5 MB	786.9 KB	54.9%
lib	1.2 MB	461.3 KB	32.2%
mips64	236 KB	72.3 KB	5%
mips	218.3 KB	74.3 KB	5.2%
arm64-v8a	178.1 KB	67 KB	4.7%
x86_64	174.5 KB	69 KB	4.8%
x86	173.7 KB	71.7 KB	5%
libnative-lib.so	173.7 KB	71.7 KB	5%
armeabi	113.7 KB	54.2 KB	3.8%
armeabi-v7a	109.7 KB	52.8 KB	3.7%
res	220.9 KB	121.5 KB	8.5%
layout	55.2 KB	23.1 KB	1.6%
layout-v17	36.6 KB	13.5 KB	0.9%
drawable	30.9 KB	12 KB	0.8%
drawable-xhdpi-v4	30.2 KB	28.4 KB	2%
color	11.4 KB	5 KB	0.4%
mipmap-xxhdpi-v4	9.3 KB	9.2 KB	0.6%
layout-v21	7.5 KB	3 KB	0.2%
mipmap-xxhdpi-v4	6.8 KB	6.8 KB	0.5%
drawable-v21	6.1 KB	2.3 KB	0.2%
color-v23	5.8 KB	2.5 KB	0.2%

Tips：Instant Run 并不兼容使用了 native code 的项目。Android Studio 会自动禁止 Instant Run 功能。

但是我个人目前更关注的问题是C++代码自动补全提示的功能，当在cpp文件中写代码时，里面不再是一大片的红色，输入若干个字母的时候，也能给出提示了，简直叼渣天，至于原因...我就不清楚了，为什么在这里就有这个强大的功能了呢？哪位大神能解释下...



## CMake编译so文件

在Android Studio的Terminal中输入cmake，会提示：

- 1 'cmake' 不是内部或外部命令，也不是可运行的程序或批处理文件。

需要在下载配置好CMake的环境变量，若没这个问题可跳过这个步骤。在如下网址中下载好对应的文件：

<https://cmake.org/download/> (<https://cmake.org/download/>)

### Release Candidate (3.7.0-rc3)

The release was packaged with CPack which is included as part of the release. The .sh files are self extracting gzipped tar files. To install a .sh file, run it with /bin/sh and follow the directions. The OS-machine.tar.gz files are gzipped tar files of the install tree. The OS-machine.tar.Z files are compressed tar files of the install tree. The tar file distributions can be untared in any directory. They are prefixed by the version of CMake. For example, the Linux-x86\_64 tar file is all under the directory cmake-Linux-x86\_64. This prefix can be removed as long as the share, bin, man and doc directories are moved relative to each other. To build the source distributions, unpack them with zip or tar and follow the instructions in Readme.txt at the top of the source tree. See also the [CMake 3.7 Release Notes](#). Source distributions:



7



Platform	Files
Unix/Linux Source (has \n line feeds)	<a href="#">cmake-3.7.0-rc3.tar.gz</a>
	<a href="#">cmake-3.7.0-rc3.tar.Z</a>
Windows Source (has \r\n line feeds)	<a href="#">cmake-3.7.0-rc3.zip</a>

Binary distributions:

Platform	Files
Windows win64-x64 Installer: <b>Installer tool has changed. Uninstall CMake 3.4 or lower first!</b>	<a href="#">cmake-3.7.0-rc3-win64-x64.msi</a>
Windows win64-x64 ZIP	<a href="#">cmake-3.7.0-rc3-win64-x64.zip</a>
Windows win32-x86 Installer: <b>Installer tool has changed. Uninstall CMake 3.4 or lower first!</b>	<a href="#">cmake-3.7.0-rc3-win32-x86.msi</a>
Windows win32-x86 ZIP	<a href="#">cmake-3.7.0-rc3-win32-x86.zip</a>
Mac OSX 10.6 or later	<a href="#">cmake-3.7.0-rc3-Darwin-x86_64.dmg</a>
	<a href="#">cmake-3.7.0-rc3-Darwin-x86_64.tar.gz</a>
Linux x86_64	<a href="#">cmake-3.7.0-rc3-Linux-x86_64.sh</a>
	<a href="#">cmake-3.7.0-rc3-Linux-x86_64.tar.gz</a>

下载zip包后，解压缩。然后将cmake.exe所在的路径，添加到path环境变量中。cmake就成为有效命令了。

在CMakeLists.txt文件上级目录下执行 cmake . 命令，报错，暂未解决：



7



- 1 -- Building for: NMake Makefiles
- 2 -- The C compiler identification is unknown
- 3 -- The CXX compiler identification is unknown
- 4 CMake Error in CMakeLists.txt:
- 5 The CMAKE\_C\_COMPILER:
- 6
- 7 cl
- 8
- 9 is not a full path and was not found in the PATH.
- 10
- 11 To use the NMake generator with Visual C++, cmake must be run from a shell
- 12 that can use the compiler cl from the command line. This environment is
- 13 unable to invoke the cl compiler. To fix this problem, run cmake from the
- 14 Visual Studio Command Prompt (vcvarsall.bat).
- 15
- 16 Tell CMake where to find the compiler by setting either the environment
- 17 variable "CC" or the CMake cache entry CMAKE\_C\_COMPILER to the full path to
- 18 the compiler, or to the compiler name if it is in the PATH.
- 19
- 20
- 21 CMake Error in CMakeLists.txt:
- 22 The CMAKE\_CXX\_COMPILER:
- 23
- 24 cl
- 25
- 26 is not a full path and was not found in the PATH.
- 27
- 28 To use the NMake generator with Visual C++, cmake must be run from a shell
- 29 that can use the compiler cl from the command line. This environment is
- 30 unable to invoke the cl compiler. To fix this problem, run cmake from the
- 31 Visual Studio Command Prompt (vcvarsall.bat).
- 32
- 33 Tell CMake where to find the compiler by setting either the environment
- 34 variable "CXX" or the CMake cache entry CMAKE\_CXX\_COMPILER to the full path
- 35 to the compiler, or to the compiler name if it is in the PATH.
- 36
- 37
- 38-- Configuring incomplete, errors occurred!
- 39

- 40See also "E:/AndroidDemo\_XY/app/CMakeFiles/CMakeOutput.log".  
See also "E:/AndroidDemo\_XY/app/CMakeFiles/CMakeError.log".

待完成



## CMake和传统 JNI在目录结构和配置文件上的区别

7

现在我们有了CMake和传统JNI两种开发NDK的方法，它们在目录结构和Gradle上有所区别，下面我们将分别介绍目录区别和Gradle配置的区别。

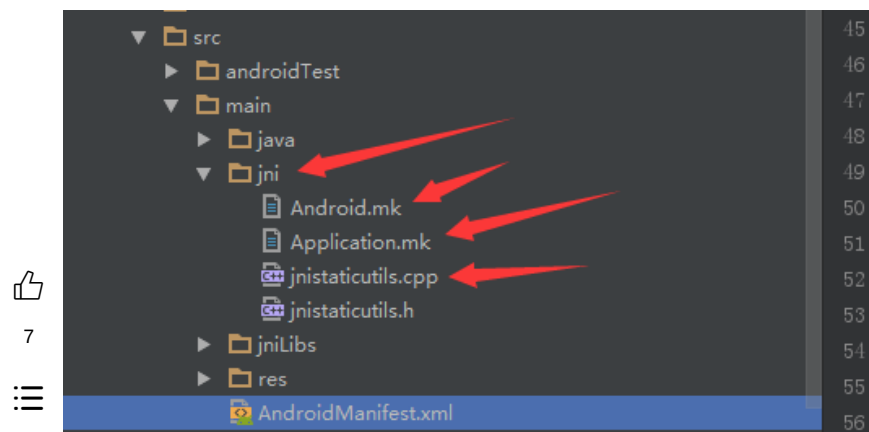


一、目录结构

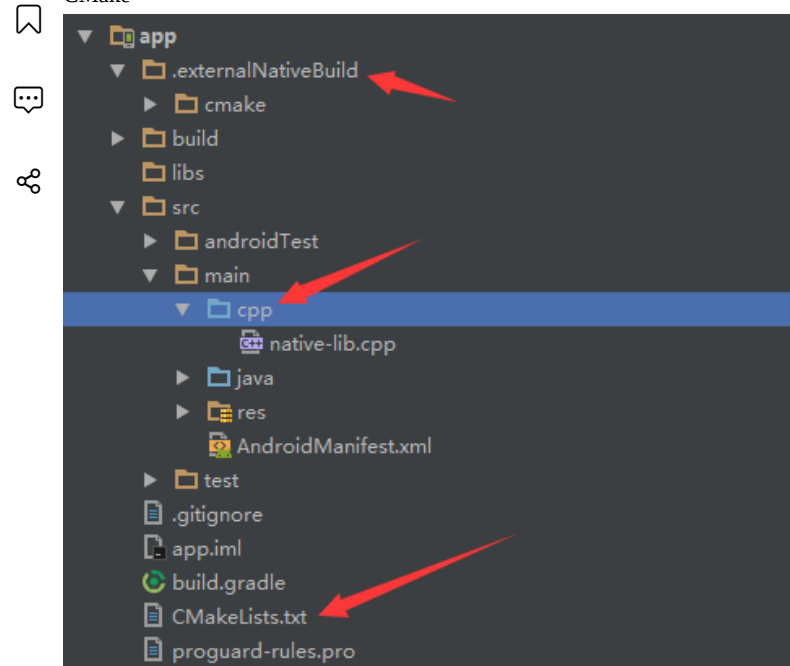


传统JNI





CMake



这两种方式在目录上的区别就是两点：

1. 以前的jni目录改成cpp，名字更换了，下面还是存放C/C++文件。
2. 之前对C/C++文件的编译配置Android.mk、Application.mk文件放在jni目录下，现在改成CMakeLists.txt文件。（事实上这

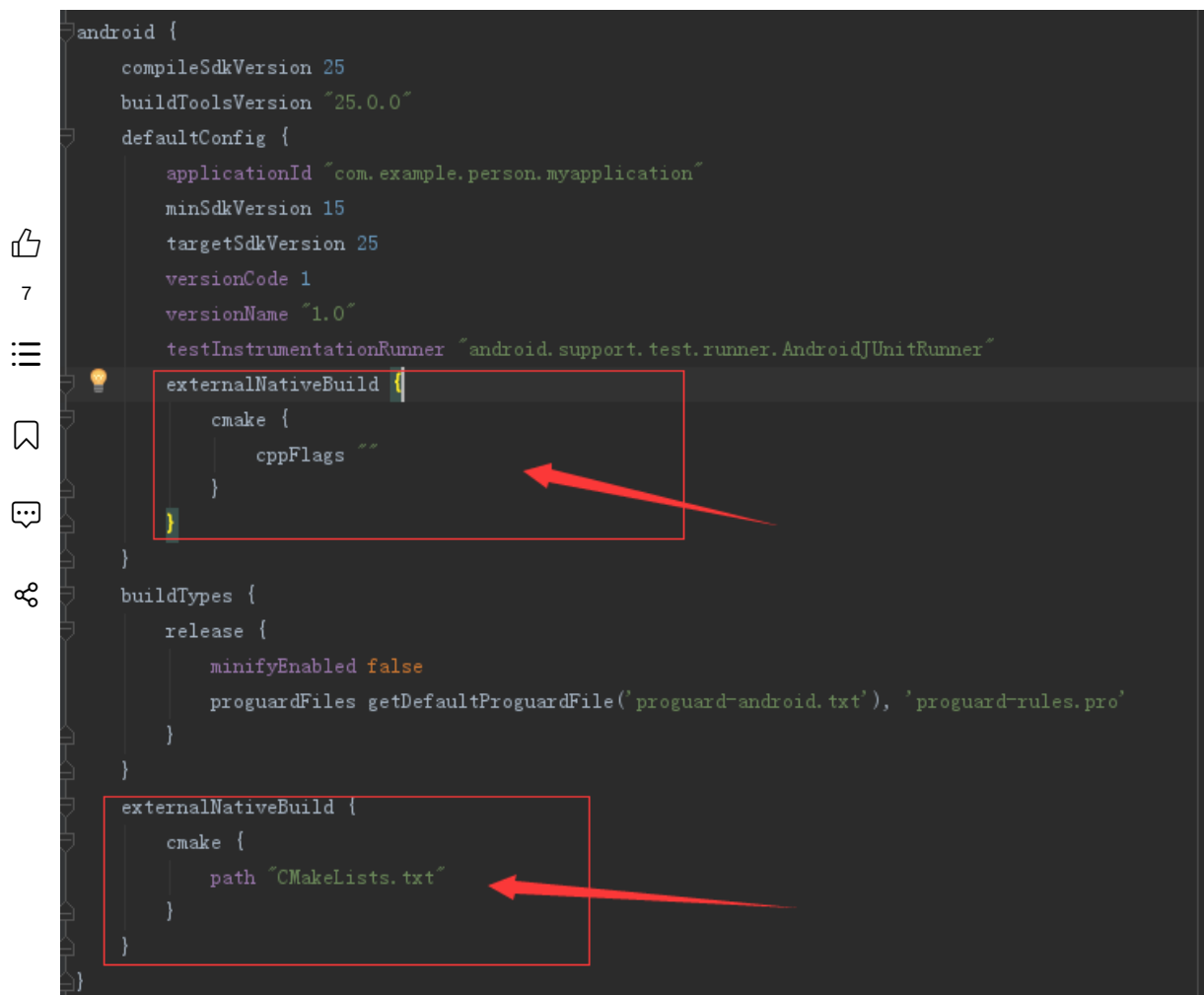
些文件的位置是可任意存放的，只需要配置好就行。但最好还是按照默认习惯放置。)

## 二、Gradle

### 传统JNI

```
defaultConfig {  
    applicationId "com.example.person.myapplication"  
    minSdkVersion 15  
    targetSdkVersion 25  
    versionCode 1  
    versionName "1.0"  
  
    // NDK  
    ndk {  
        moduleName "jnistaticutils"  
        abiFilters "armeabi", "armeabi-v7a", "x86" //输出指定三种CPU体系结构下的so库  
    }  
}  
  
buildTypes {  
    release {  
        minifyEnabled false  
        proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'  
    }  
}  
  
externalNativeBuild {  
    ndkBuild {  
        path "src/main/jni/Android.mk"  
    }  
}
```

## CMake



事实上，我们在使用传统JNI方式的时候，上面的两处地方我们都省略了，也不会造成什么异常（若Android.mk存放位置在其他地方则需要配置）。CMake方式中第一处也可省略，但是第二处不能省略（同样，位置路径要写对），因为没有它将无法Build生成.externalNativeBuild文件夹。

另外，传统JNI开发还需要在项目根目录下的gradle.properties文件中配置



- 1 android.useDeprecatedNdk=true

否则Build项目的时候会报错。

## CMake的优势



7



1. 可以直接的在C/C++代码中加入断点，进行调试
2. java引用的C/C++中的方法，可以直接ctrl+左键进入
3. 对于include的头文件或者库，也可以直接进入
4. 不需要配置命令行操作，手动的生成头文件，不需要配置android.useDeprecatedNdk=true属性



## 普通Android项目转NDK开发项目



我们之前说了，在创建新project时若勾选了 Include C++ Support 选项（需Android Studio版本不低于2.2），则该项目就已经是一个NDK开发项目了。那么，若项目已经存在，该如何再转为NDK开发项目呢？



Tips: 在配置好NDK的前提下。且各种地方的配置均不再赘述。如Android.mk中如何配置、CMakeLists.txt文件中的配置、各个地方的统一等。

传统JNI方式，在NDK开发 从入门到放弃(一：基本流程入门了解) ([http://blog.csdn.net/xiaoyu\\_93/article/details/52870395](http://blog.csdn.net/xiaoyu_93/article/details/52870395))中我们了解过：

1. 新建jni目录，写好C/C++代码。静态注册JNI时我们使用了javah -jni对JAVA类进行操作自动生成了jni目录以及对应的头文件（事实上，当我们有一定经验后可以自己写，而不再需要使用该辅助命令来保证不写错，另外动态注册也是一个很值得提倡的方式），然后根据头文件写了C/C++代码。但在动态注册JNI时我们可以自己先创建好jni目录且写好C/C++代码。
2. 在jni目录下创建且配置好Android.mk和Application.mk两个文件。
3. build.gradle文件中根据情况进行配置，可不进行配置使用默认值。
4. 通过ndk-build操作，我们能得到对应的so文件，放置在相应位置，java代码中即可调用C/C++代码，运行程序。

回顾前面提到的CMake方式的NDK开发，我们得到如下步骤：

1. 新建cpp目录，写好C/C++代码。

2. 创建且配置CMakeLists.txt文件。
3. build.gradle文件中根据情况进行配置，CMakeLists.txt文件的路径必须配置。
4. java代码中即可调用C/C++代码，运行程序。
5. project的build.gradle文件中，gradle版本不能低于2.2，否则会报错。

Error: Could not find method externalNativeBuild() for arguments  
[build\_e8k5d0w74qu1ajxjr9snog\$\_run\_closure1\$\_closure6@1398219d] on  
project ':app' of type org.gradle.api.Project.。



7



qq\_31800771 (/qq\_31800771) 2017-10-10 11:37

1楼


(/qq\_31800771)我也遇到相同的问题，在CMakeLists.txt文件上级目录下执行cmake .命令，报错，未解决：不知道你之前遇到的这个问题解决了没？

回复

## 相关文章推荐

### android studio 2.2 C++ 支持（CMAKE方法）(/yangzhengqui/article/details/51544687)

上次学到了约束布局，现在继续学习android studio 2.2使用CMAKE编译C++，并支持在C++打断点，写C++代码时也支持打断点。学习的时候是使用了国外的一篇博客的代码：<http://b...>


 yangzhengqui (<http://blog.csdn.net/yangzhengqui>) 2016-05-31 12:47 16273

## android studio 2.2 使用cmake编译NDK (/lmw21848/article/details/52183584)

android studio 2.2 已经进入beta版本，新功能添加众多，NDK编程也得到了简化。官方博客介绍。本文介绍如何使用新版android studio调用 c++代码，为了超级通俗易懂，例...



7

 lmw21848 (<http://blog.csdn.net/lmw21848>) 2016-08-11 17:51 11364




### 深入浅出 Docker 技术栈实践课

本课程的目标在于为企业级的开发、运维环境中使用 Docker 提供实战型指导，为你深入浅出地讲解 Docker 容器技术，以便大家课后可以在线下自己实践 Docker 技术。

([http://www.baidu.com/cb.php?c=lgF\\_pyfqHmknjcvPWR0IZ0qnfK9ujYzP1fvn1Ds0Aw-581c4nHb3rjD0TAq15HfLPWRznjb0T1Ydm1fsrHuBuAFBn1FhrHn40AwY5HDdnH0YrjTsnjT0lgF\\_5y9YIZ0IQzqMpgwBUVqoQhP8QvIGIAPCmgfEmvq\\_lyd8Q1R4uHfvPjTzPWbdR4HcsP1qdIAdxTvqdThP-5HDknHK9mHkEusKzujYk0AFV5H00TZcqN0KdpyfqHRLPjnvnfKEpyfqHnsnj0YnsKWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqnWbLPj6](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjcvPWR0IZ0qnfK9ujYzP1fvn1Ds0Aw-581c4nHb3rjD0TAq15HfLPWRznjb0T1Ydm1fsrHuBuAFBn1FhrHn40AwY5HDdnH0YrjTsnjT0lgF_5y9YIZ0IQzqMpgwBUVqoQhP8QvIGIAPCmgfEmvq_lyd8Q1R4uHfvPjTzPWbdR4HcsP1qdIAdxTvqdThP-5HDknHK9mHkEusKzujYk0AFV5H00TZcqN0KdpyfqHRLPjnvnfKEpyfqHnsnj0YnsKWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqnWbLPj6))


## NDK开发 从入门到放弃(七：Android Studio 2.2 CMAKE 高效NDK开发) (/cuiyufeng2/article...)

更深入的了解如何开发NDK

 cuiyufeng2 (<http://blog.csdn.net/cuiyufeng2>) 2017-03-20 14:45 2916

## Android Studio Cmake配置项 (/qq\_35100676/article/details/69385173)

Cmake的配置 Android Studio 2.2以上支持了Cmake的配置JNI的相关参数，简化了通过Android.mk配置。并很好的继承了C++的编辑方式。以下是对应的引入第三方so和第三...

 qq\_35100676 ([http://blog.csdn.net/qq\\_35100676](http://blog.csdn.net/qq_35100676)) 2017-04-06 11:22 2094



## Webpack 前端工程化入门

如果你了解在公司级别的大型项目中是如何做工程化的？如果你想深入了解 Webpack，并且想用它的各种特性来提升构建效率？那么你就有理由来了解一下这门课。

([## Android开发学习之路--Android Studio cmake编译ffmpeg \(/eastmoon502136/article/detail...](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjcvPjR0IZ0qnfK9ujYzP1ndnHfY0Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1Y3uHczuhcLnhN-nywbn19W0AwY5HDdnH0YrjTsnjT0lgF_5y9YIZ0IQzqMpgwBUvqoQhP8QvIGIAPCmgfEmvq_lyd8Q1R4uH0vPymvPAmLuhF-PzHRduHfLrHcsPjqdIAdxTvqdThP-5HDknHK3mhkEusKzujYk0AFV5H00TZcq0KdpyfqHRLPjnvnfKEpyfqHnsnj0YnsKWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqPWT3Pf) :三</a></p></div><div data-bbox=)



最新的android studio2.2引入了cmake可以很好地实现ndk的编写。这里使用最新的方式，对于以前的android下的ndk编译什

么的可以参考之前的文章:Android开发学习之路-ND...



eastmoon502136 (<http://blog.csdn.net/eastmoon502136>) 2016-10-20 22:33 11215



## Android Studio快捷键、配置 Android Studio、cmake配置Android原生项目、cmake命令 (/...

本文件来自官方文档 <https://developer.android.google.cn/studio/intro/keyboard-shortcuts.html> 另有：配置 Android...



feixiangsmile (<http://blog.csdn.net/feixiangsmile>) 2017-04-25 22:26 294

## AndroidStudio2.3 NDK Cmake配置 (/pkandroid/article/details/67074346)



最近使用移植一个项目，里面用到了NDK开发，然后把过程记录一下，1、创建NativeUtils.java，里面写了一些本地方法2、锤子一下，找到app目录下的classes文件夹，逐级打开...




pkandroid (<http://blog.csdn.net/pkandroid>) 2017-03-30 10:53 1012

## Android Studio CMake (二) (/hbgk/article/details/53352204)

前面 Android studio CMake (一), 针对简单的项目已经足够。然而, 现实中却总不是那么简单, 例如可能会包含子目录。 1. 同样, 创建JNI目录和CMakeLists.tx...

 hbgk (<http://blog.csdn.net/hhbgk>) 2016-11-26 16:03  1031




## Android Studio Cmake配置项 (/qq\_15807167/article/details/67634741)

 Cmake的配置Android Studio 2.2以上支持了Cmake的配置JNI的相关参数, 简化了通过Android.mk配置。并很好的继承了C++的编辑方式。以下是对应的引入第三方so和第三方....

  qq\_15807167 ([http://blog.csdn.net/qq\\_15807167](http://blog.csdn.net/qq_15807167)) 2017-03-28 13:18  866



## AndroidStudio2.2下利用CMake编译方式的NDK opencv开发 (/ddjll8877/article/details/526...

 AndroidStudio2.2对C++开发做了很好的支持, 默认使用CMake构建编译, 以前做Andorid下的opencv开发都是基于Android.mk方式是编译和配置opencv依赖的加载, 本文...

  ddjll8877 (<http://blog.csdn.net/ddjll8877>) 2016-09-26 14:41  5044



## Android Studio2.2配置MakeList使用cmake编译c文件 (/qq\_35599978/article/details/563152...

初次使用cmake来编译c文件在Android项目中,那个快感能传递到骨髓里,欲罢不能.只能说谁用谁知道.如果还没有使用,请赶快用一次,初次体验一共三部: 第一:studio升级到2.2以后的版本,安...

 qq\_35599978 ([http://blog.csdn.net/qq\\_35599978](http://blog.csdn.net/qq_35599978)) 2017-02-21 21:59  404

## Android中CMake的使用之二具体的使用细节 (/fpcc/article/details/69942534)

Android中CMake的使用之二具体的使用细节 在上一节中提供了一个最简单的使用说明, 但是在实际情况中, 一般来说, 应用程序中对C++的控制代码会安排在一个或几个专门的JAVA类中。同样, 接口函数...

 fpcc (<http://blog.csdn.net/fpcc>) 2017-04-10 09:45  575

## cmake android (/jsa158/article/details/52217754)

对于多平台管理生成native库，cmake的使用是非常方便的，下面就构建的cmake工程怎么生成android可用的native库或者可执行程序做一下记录 因为是开源的东西，在linux使用会省去很...



jsa158 (<http://blog.csdn.net/jsa158>) 2016-08-16 10:33 1462



7

## Android Studio 2.3使用CMake方式NDK入门 (/aidesudi/article/details/68946918)



参考官网：<https://developer.android.com/studio/projects/add-native-code.html#new-project> 使用CMake需要搭配使用an...



aidesudi (<http://blog.csdn.net/aidesudi>) 2017-04-01 18:34 3353



## android studio 下的cmake使用 (/hhh901119/article/details/54583913)



一种java调用c++的方式是使用javah生成.class文件,ndk-build生成.so文件（ubuntu操作系统下），会生成一个动态库。另一种方式是用cmake下的CMakeLists.txt...



hhh901119 (<http://blog.csdn.net/hhh901119>) 2017-01-17 16:10 998

## Android studio 使用Cmake完成C/C++ 的使用以及生成so文件 (/u014800493/article/details/5...


Android studio 2.2版本以后对C/C++的支持可以说很方便了，当然官方推荐使用Cmake完成对C/C++的支持 2.2版本以上的同学新建一个项目就知道了，步骤如下：File -> N...



u014800493 (<http://blog.csdn.net/u014800493>) 2017-02-09 16:04 4323

## AndroidStudio用Cmake方式编译NDK代码 (/joe544351900/article/details/53637549)

1.cmake是什么？CMake是一个跨平台的安装（编译）工具，可以用简单的语句来描述所有平台的安装(编译过程)。他能够输出各种各样的makefile或者project文件，能测试编译器所支持的C+...

 joe544351900 (<http://blog.csdn.net/joe544351900>) 2016-12-14 12:57 6726

## AndroidStudio使用CMake编译jni的C/C++文件 (/binglumeng/article/details/54667659)

Android开发主流工具已变成AndroidStudio，新版AS已经支持CMake编译工具，可以用于编译C/C++文件，增强了Android调用jni代码的便捷性。相比于之前繁杂的ndk配置方式，...



7



binglumeng (<http://blog.csdn.net/binglumeng>) 2017-01-22 11:52 1997



## NDK学习(二)，在NDK开发中引入第三方库 ( AndroidStudio Cmake ) (/mxw3755/article/de...



在NDK中可能需要复用之前已经编译好的so文件，所以本章的目标是给一个之前编译好的so文件外加一个需要调用接口的头文件，在现在项目中复用。在本次实践过程中，已经有编译完成的各种cpu架构的...



mxw3755 (<http://blog.csdn.net/mxw3755>) 2017-02-23 17:41 1718



## sdk tools没有cmake下载项的解决办法。 (/u014290233/article/details/53418371)

sdk tools没有cmake下载项的解决办法。



u014290233 (<http://blog.csdn.net/u014290233>) 2016-12-01 09:43 2562

