

# Android NDK开发之旅16--NDK-- Android Studio2.3下NDK开发流程



小楠总 (/u/70c12759d4fe) [+ 关注](#)

2017.03.21 15:44\* 字数 1734 阅读 2638 评论 54 喜欢 28 赞赏 3

(/u/70c12759d4fe)

本篇文章已授权微信公众号 guolin\_blog（郭霖）独家发布  
本人小楠——一位励志的Android开发者。

欢迎关注我的公众号：



公众号：Android开发进阶

## Android Studio下NDK开发流程

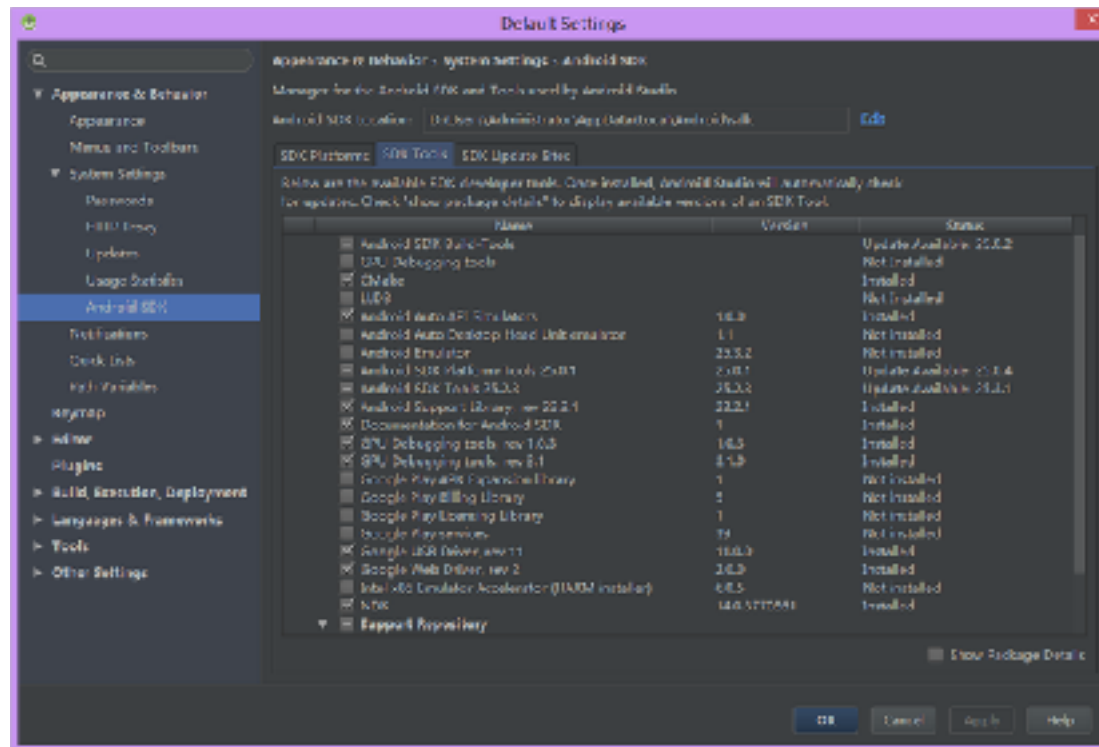
Android Studio目前最新的版本是2.3稳定版，从2.2开始就默认使用CMake的方式来构建NDK项目，当然我们也可以使用ndk-build的方式，这里我们主要介绍CMake的方式。



我们先介绍NDK的下载与安装，然后介绍由Android Studio默认创建带C/C++支持的项目开始，然后介绍如何为已有的项目添加C/C++支持。

## NDK工具的下载与安装

如下图所示，在SDK设置那个页面，选择SDK Tools面板，就可以下载NDK。



NDK安装.png

要为应用编译和调试原生代码需要以下组件：

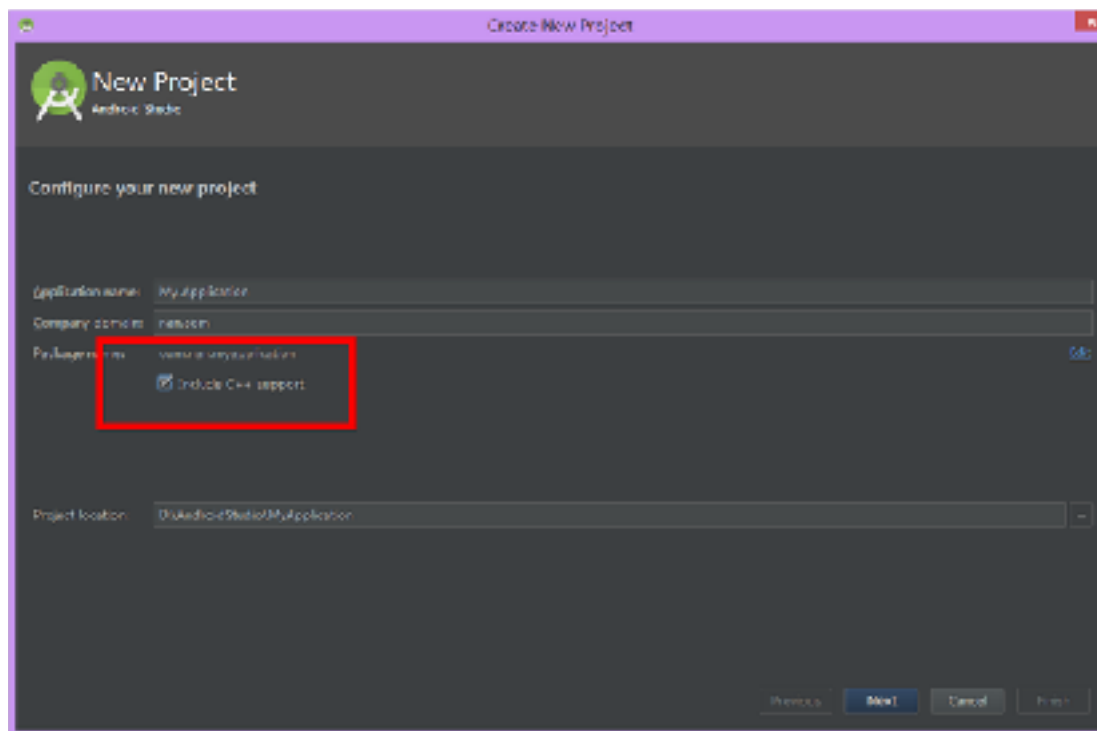
1. **Android 原生开发工具包 (NDK)**：这套工具集允许您为 Android 使用 C 和 C++ 代码，并提供众多平台库，让您可以管理原生 Activity 和访问物理设备组件，例如传感器和触摸输入。



2. CMake：一款外部构建工具，可与 Gradle 搭配使用来构建原生库。如果您只计划使用 ndk-build，则不需要此组件。
3. LLDB：一种调试程序，Android Studio 使用它来调试原生代码。

## 由Android Studio默认创建带C/C++支持的项目

我们在创建项目的时候，在向导的 Configure your new project 部分，选中 Include C++ Support 复选框如下图所示：



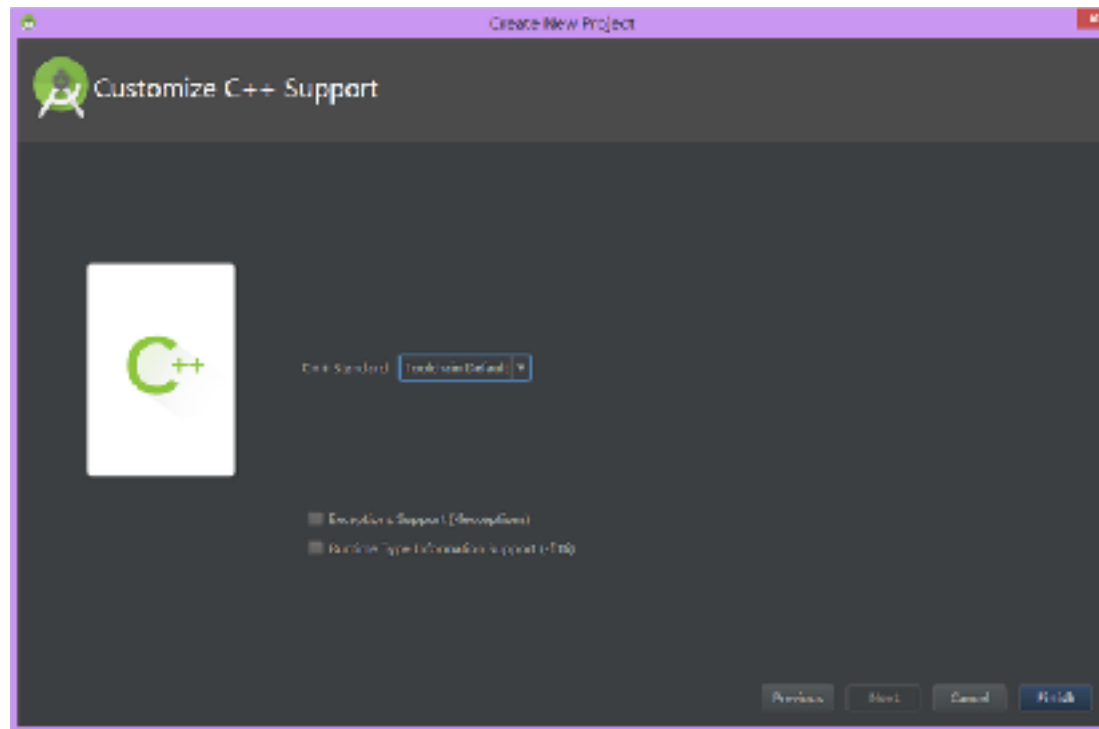
创建项目.png

在向导的 Customize C++ Support 部分，您可以使用下列选项自定义项目：

1. C++ Standard：使用下拉列表选择您希望使用哪种 C++ 标准。选择 Toolchain Default 会使用默认的 CMake 设置。



2. Exceptions Support : 如果您希望启用对 C++ 异常处理的支持, 请选中此复选框。如果启用此复选框, Android Studio 会将 -fexceptions 标志添加到模块级 build.gradle 文件的 cppFlags 中, Gradle 会将其传递到 CMake。
3. Runtime Type Information Support : 如果您希望支持 RTTI, 请选中此复选框。如果启用此复选框, Android Studio 会将 -frtti 标志添加到模块级 build.gradle 文件的 cppFlags 中, Gradle 会将其传递到 CMake。

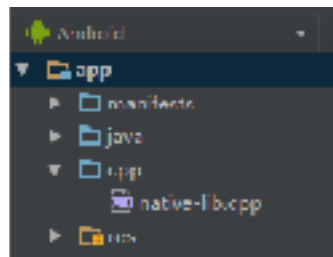


创建项目.png

这里我们不选择, 因为暂时用不到, 直接点击完成即可。

创建好的项目如下图所示:





项目.png

其中：

1. cpp目录存放C/C++的头文件或者源文件
2. External Build Files存放我们的CMake脚本文件，这是通过Gradle来进行配置的。

下面我们来瞧一瞧app的build.gradle文件：



```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 24
    buildToolsVersion "25.0.1"
    defaultConfig {
        applicationId "com.nan.testndk"
        minSdkVersion 16
        targetSdkVersion 24
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"

        externalNativeBuild {
            cmake {
                cppFlags ""
            }
        }

    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-
        }
    }

    externalNativeBuild {
        cmake {
            path "CMakeLists.txt"
        }
    }
}

dependencies {
    //...一些库
}
```

介绍一下新增的externalNativeBuild标签：

配置另一个 externalNativeBuild {} 块，为 CMake 或 ndk-build 指定可选参数和标志、以及配置CMakeLists的文件路径。



下面来看一下CMake的构建脚本文件：CMakeLists.txt

```
#指定CMake的最小版本
cmake_minimum_required(VERSION 3.4.1)

#添加我们自己的模块，名字是native-lib，SHARED可分享的，以及配置源文件或者头文件的路径
add_library(
    native-lib
    SHARED
    src/main/cpp/native-lib.cpp )

#找到log这个模块
find_library(
    log-lib
    log )

#把我们自己的模块和log模块关联起来
target_link_libraries(
    native-lib
    ${log-lib} )
```

为了方便查阅，我把一些注释删掉了。

下面来看看native-lib.cpp，就是一些简单的JNI交互，返回一个字符串给Java层（我们的MainActivity）

```
#include <jni.h>
#include <string>

extern "C"
JNIEXPORT jstring JNICALL
Java_com_nan_testndk_MainActivity_stringFromJNI(
    JNIEnv *env,
    jobject /* this */) {
    std::string hello = "Hello from C++";
    return env->NewStringUTF(hello.c_str());
}
```

在MainActivity里面，主要就是加载这个动态库，然后调用JNI方法，把获取到的字符串显示到TextView上面：



```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        TextView tv = (TextView) findViewById(R.id.sample_text);

        //调用JNI方法，把获取到的字符串显示到TextView上面
        tv.setText(stringFromJNI());
    }

    //一个native方法的例子
    public native String stringFromJNI();

    public native int getIntFromJni();

    //加载native的so文件，注意不用加lib前缀
    static {
        System.loadLibrary("native-lib");
    }

}
```

## 为已有的项目添加C/C++支持

上面介绍的是用Android Studio创建带C/C++支持的默认项目，下面我们介绍如何为已有的项目添加C/C++支持。

为了给出例子，我们随便创建一个空的项目。

创建一个类，专门用于文件加密解密，加载so文件，然后写完native方法以后，alt+enter一下自动创建jni目录和cpp的源文件





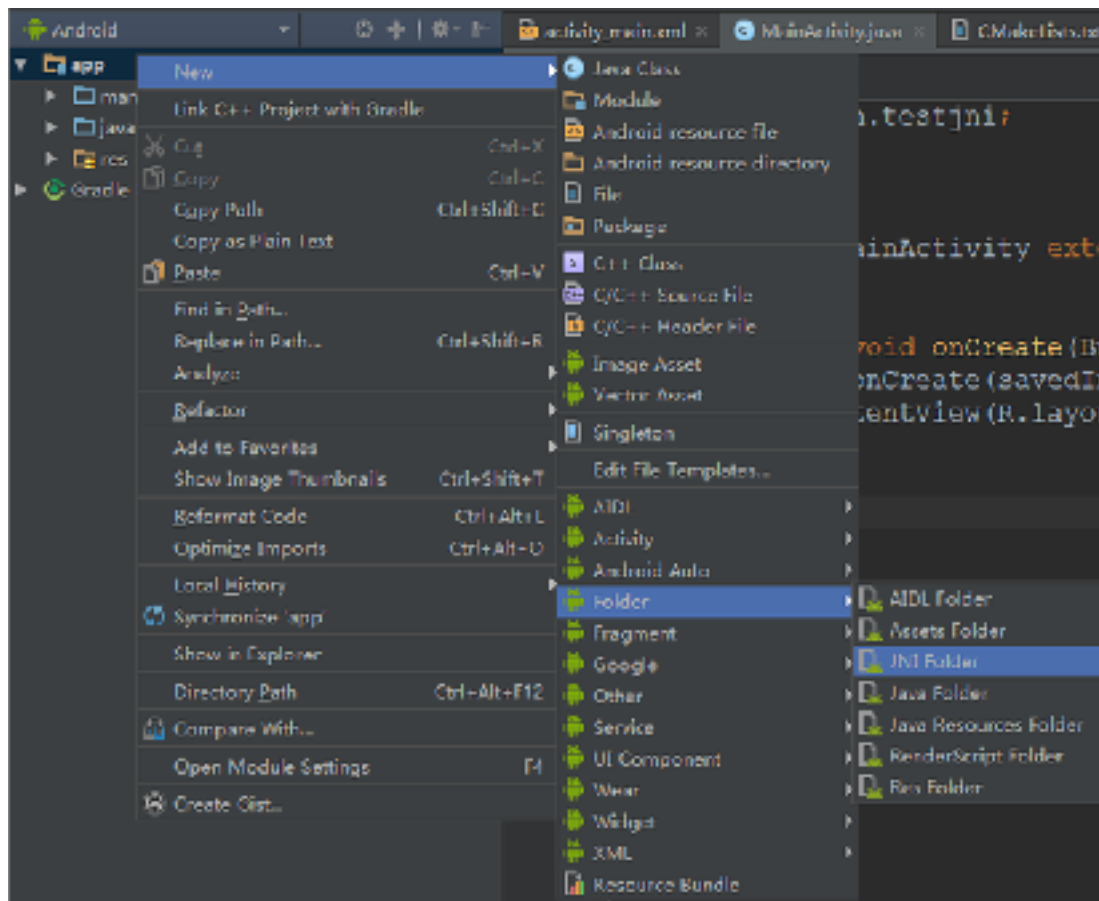
```
public class Cryptor {  
  
    static {  
        //加载动态库.so文件，注意不用写lib前缀，系统会默认添加  
        System.loadLibrary("crypt");  
    }  
  
    public static native void cryptFile(String src, String dest);  
  
    public static native void decryptFile(String src, String dest);  
  
}
```

然后会自动产生C++代码，crypt.c，可以看到Android Studio自动帮我们获取了C字符串，以及在方法的末尾进行了释放，十分贴心，省略了我们每次使用javah命令去生成头文件的过程：

```
#include <jni.h>  
  
JNIEXPORT void JNICALL  
Java_com_nan_testjni_Cryptor_cryptFile(JNIEnv *env, jclass type, jstring src_, jstri  
    const char *src = (*env)->GetStringUTFChars(env, src_, 0);  
    const char *dest = (*env)->GetStringUTFChars(env, dest_, 0);  
  
    // TODO  
  
    (*env)->ReleaseStringUTFChars(env, src_, src);  
    (*env)->ReleaseStringUTFChars(env, dest_, dest);  
}  
  
JNIEXPORT void JNICALL  
Java_com_nan_testjni_Cryptor_decryptFile(JNIEnv *env, jclass type, jstring src_, jst  
    const char *src = (*env)->GetStringUTFChars(env, src_, 0);  
    const char *dest = (*env)->GetStringUTFChars(env, dest_, 0);  
  
    // TODO  
  
    (*env)->ReleaseStringUTFChars(env, src_, src);  
    (*env)->ReleaseStringUTFChars(env, dest_, dest);  
}
```

也可以手动创建一个JNI目录，如下图所示：





创建JNI的目录.png

创建出来的目录是cpp，但是我们切换到Project视图发现还是叫jni。然后在这个目录可以手动创建我们的C/C++源文件：

```
fileCrypt.c
```

转到Project视图，在app的目录下面创建一个File，名字为CMakeLists.txt，推荐使用这个名字和文件路径。

**注意：必须先创建源文件，否则下面创建CMake脚本同步的时候不会通过。**

创建CMake脚本.png

脚本中文件创建我们自己的NDK模块，叫做fileCrypt，专门用来做文件加密解密的：

```
#指定CMake的最小版本
cmake_minimum_required(VERSION 3.4.1)

#添加我们自己的模块，名字是native-lib，SHARED可分享的，以及配置源文件或者头文件的路径
add_library(
    fileCrypt
    SHARED
    src/main/jni/native-lib.c )
```



**这里有两个地方需要注意**

1. 路径一定要注意跟我们所创建的目录名字一致，注意你创建的是“jni”还是“cpp”目录，否则同步不了。例如我们刚刚通过Android Studio创建的目录实质上是“jni”目录，因此这里写jni。你也可以手动创建cpp目录，然后这里写cpp目录，与标准的项目一样。
2. 我们创建的有可能是C也有可能是C++，所以这里要注意写.c还是.cpp后缀，否则同步会失败。

然后选中app，右击，选择下图中的选项：

关联.png

---

找到我们创建的脚本文件，确认：



关联.png

这时候Android Studio就会自动同步，然后在app模块的build.gradle文件自动添加配置：

```
externalNativeBuild {  
    cmake {  
        path 'CMakeLists.txt'  
    }  
}
```

当然我们也可以手动配置app的build.gradle文件，然后自己手动同步。

为了加快构建速度，我们只输出armabi的动态库文件，在app的build.gradle文件添加一下配置：

```
ndk{  
    abiFilters 'armeabi'  
}
```

同时我们依样画葫芦，也顺便加上C/C++编译的时候需要的参数信息：

```
externalNativeBuild {  
    cmake {  
        cppFlags ""  
    }  
}
```

完整的app的build.gradle文件如下：



```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 24
    buildToolsVersion "25.0.1"
    defaultConfig {
        applicationId "com.nan.testjni"
        minSdkVersion 16
        targetSdkVersion 24
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
        externalNativeBuild {
            cmake {
                cppFlags ""
            }
        }
        ndk {
            //abiFilters 'x86', 'x86_64', 'armeabi', 'armeabi-v7a', 'arm64-v8a'//所有3
            abiFilters 'armeabi'
        }
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-
        }
    }
    externalNativeBuild {
        cmake {
            path 'CMakeLists.txt'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:24.2.1'
    compile 'com.android.support.constraint:constraint-layout:1.0.0-alpha3'
    testCompile 'junit:junit:4.12'

    compile 'com.tbruyelle.rxpermissions2:rxpermissions:0.9.3@aar'
    compile 'io.reactivex.rxjava2:rxandroid:2.0.1'
    compile 'io.reactivex.rxjava2:rxjava:2.0.7'
}
```



等下要操作SD卡，我们不妨把rxpermissions也加上。

下面我们把C代码完善一下，完整的crypt.c文件如下，功能与上一篇博客一样，这里不再赘述：



```
#include <jni.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

//用于加密解密的密码
char password[] = "I AM MI MA";

JNIEXPORT void JNICALL
Java_com_nan_testjni_Cryptor_cryptFile(JNIEnv *env, jclass type, jstring src_, jstri

    const char *c_src = (*env)->GetStringUTFChars(env, src_, NULL);
    const char *c_dest = (*env)->GetStringUTFChars(env, dest_, NULL);

    FILE *f_read = fopen(c_src, "rb");
    FILE *f_write = fopen(c_dest, "wb");

    //判断文件是否正确打开
    if (f_read == NULL || f_write == NULL) {
        printf("file open field");
        return;
    }

    //一次读取一个字符
    int ch;
    int i = 0;
    int pwd_len = strlen(password);
    while ((ch = fgetc(f_read)) != EOF) {
        //通过异或运算进行加密
        fputc(ch ^ password[i % pwd_len], f_write);
        i++;
    }

    //关闭文件
    fclose(f_read);
    fclose(f_write);

    (*env)->ReleaseStringUTFChars(env, src_, c_src);
    (*env)->ReleaseStringUTFChars(env, dest_, c_dest);
}

JNIEXPORT void JNICALL
Java_com_nan_testjni_Cryptor_decryptFile(JNIEnv *env, jclass type, jstring src_, jst

    const char *c_src = (*env)->GetStringUTFChars(env, src_, NULL);
    const char *c_dest = (*env)->GetStringUTFChars(env, dest_, NULL);

    FILE *f_read = fopen(c_src, "rb");
    FILE *f_write = fopen(c_dest, "wb");
```





```
//判断文件是否正确打开
if (f_read == NULL || f_write == NULL) {
    printf("file open field");
    return;
}

//一次读取一个字符
int ch;
int i = 0;
int pwd_len = strlen(password);
while ((ch = fgetc(f_read)) != EOF) {
    //通过异或运算进行加密
    fputc(ch ^ password[i % pwd_len], f_write);
    i++;
}

//关闭文件
fclose(f_read);
fclose(f_write);

(*env)->ReleaseStringUTFChars(env, src_, c_src);
(*env)->ReleaseStringUTFChars(env, dest_, c_dest);
}
```

最后，我们写两个测试按钮，分别调用加密解密的方法：

记得添加权限以及使用动态权限：

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
```

测试的Activity代码如下：



```
public class MainActivity extends AppCompatActivity {

    private RxPermissions mRxPermissions;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mRxPermissions = new RxPermissions(this);

        mRxPermissions.request(Manifest.permission.READ_EXTERNAL_STORAGE, Manifest.permission.WRITE_EXTERNAL_STORAGE)
            .subscribe(new Consumer<Boolean>() {
                @Override
                public void accept(Boolean granted) throws Exception {
                    if (!granted) {
                        finish();
                    }
                }
            });
    }

    public void crypt(View v) {
        String src = Environment.getExternalStorageDirectory()
            .getAbsolutePath() + File.separatorChar + "test_src.txt";
        String dest = Environment.getExternalStorageDirectory()
            .getAbsolutePath() + File.separatorChar + "test_crypt.txt";
        Cryptor.cryptFile(src, dest);
        Toast.makeText(this, "加密完成了", Toast.LENGTH_SHORT).show();
    }

    public void decrypt(View v) {
        String src = Environment.getExternalStorageDirectory()
            .getAbsolutePath() + File.separatorChar + "test_crypt.txt";
        String dest = Environment.getExternalStorageDirectory()
            .getAbsolutePath() + File.separatorChar + "test_decrypt.txt";
        Cryptor.decryptFile(src, dest);
        Toast.makeText(this, "解密完成了", Toast.LENGTH_SHORT).show();
    }
}
```

如果觉得我的文字对你有所帮助的话，欢迎关注我的公众号：



公众号：Android开发进阶


我的群欢迎大家进来探讨各种技术与非技术的话题，有兴趣的朋友们加我私人微信  
**huannan88**，我拉你进群交（♂）流（♀）。

小礼物走一走，来简书关注我

赞赏支持



(/u/6a1e91720980b15942)  
(/u/6a1e91720980b15942)

 Android NDK开发之旅 (/nb/10785521)

举报文章 © 著作权归作者所有



小楠总 (/u/70c12759d4fe)

写了 208018 字，被 1297 人关注，获得了 1019 个喜欢

(/u/70c12759d4fe)

+ 关注

官方微信公众号：小楠总 Tips：点击下方的微信图标即可弹出二维码。我是魅族小楠总，致力于各种IT技...

喜欢 | 28



更多分享

(http://cwb.assets.jianshu.io/notes/images/10420492/weibo/image\_!





(/apps/download?utm\_source=nbc)

被以下专题收入，发现更多相似内容



Android... (/c/be572900783f?utm\_source=desktop&utm\_medium=notes-included-collection)



编程设计之An... (/c/de015ec77144?utm\_source=desktop&utm\_medium=notes-included-collection)



DNK 开发 (/c/6b5d3eac4428?utm\_source=desktop&utm\_medium=notes-included-collection)



NDK (/c/5f80594d645b?utm\_source=desktop&utm\_medium=notes-included-collection)

## Android - 收藏集 - 掘金 (/p/5ad013eb5364?utm\_campaign=maleskine&...

用两张图告诉你，为什么你的 App 会卡顿？ - Android - 掘金 Cover 有什么料？从这篇文章中你能获得这些料：知道setContentView()之后发生了什么？... Android 获取 View 宽高的常用正确方式，避免为零 - 掘金...



掘金官方 (/u/5fc9b6410f4f?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

## Android - 收藏集 (/p/dad51f6c9c4d?utm\_campaign=maleskine&utm\_c...

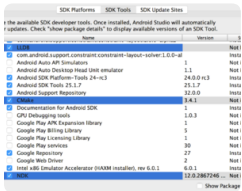
用两张图告诉你，为什么你的 App 会卡顿？ - Android - 掘金 Cover 有什么料？从这篇文章中你能获得这些料：知道setContentView()之后发生了什么？... Android 获取 View 宽高的常用正确方式，避免为零 - 掘金...



passiontim (/u/e946d18f163c?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)




(/p/2a39ed18cfc1?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

3.3、Android Studio 添加 C 和 C++ 项目 (/p/2a39ed18cfc1?utm\_campai...


文章摘要1、 c、c++项目的创建2、添加C、C++项目依赖库4、配置Gradle，编译lib so库 英文文献 搭配使用 Android Studio 2.2 或更高版本与 Android Plugin for Gradle 版本 2.2.0 或更高版本时，您可以将 C ...

 Android那些事儿 (/u/8fc5734c6857?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

JNI：为项目添加C和C++代码库 (/p/6b31c37e9fdc?utm\_campaign=male...


developer.android.com/studio/projects/add-native-code.html 本文内容 下载 NDK 和构建工具 创建支持 C/C++ 的新项目 构建和运行示例应用 向现有项目添加 C/C++ 代码 创建新的原生源文件 创建 CMak...

 四季明 (/u/39ff43a96104?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

Android Studio 2.3.3 图解配置NDK开发环境以及Hello Word To jni~ (/p/6...

LZ-Says：半夜睡觉滚了地上了，无奈之下醒来了，想想最近几天因为一个括号导致JNI迟迟不能开展，心里面无奈又崩溃，索性直接起来整完得了~ 前言 当前毕业的时候，感觉自己掌握了全世界，随着参加工作的...

 静心Study (/u/21890e8770b2?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)



(/p/e667e8e80f5d?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

### Toronto留學生日常 (/p/e667e8e80f5d?utm\_campaign=maleskine&utm\_...

hello大家好我是Denise！今天是加拿大時間2月6號，是我們開學第一天。我就讀於多倫多的一所天主教學校，接下來會為大家介紹一些國外學校的日常。今天先說說日常的住跟吃。首先說說home stay，我住在...



Denise0112 (/u/d78863ff1f26?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/091b3eee3f52?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

### 为了摆脱"制服诱惑",他们到底能做什么? (/p/091b3eee3f52?utm\_camp...

旁友们大嘎好！今天是让大家久等的... 两周前我向大家发起了征集，和小伙伴「同业云」一起寻找在银行上班的老铁们，想帮他们来做一次大改造 ▼ 于是我在报名的 100 多人中随机挑选了 2 位幸运儿，他们是@...



拒绝基本 (/u/60c0fd70bff5?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

### 感赏第50天20170411 (/p/e7b33bb2e0a8?utm\_campaign=maleskine&ut...

感赏自己坚持5点闹铃响就起床，坚持出门到学校操场快走半小时，买好早餐回家再喊儿子起床！感赏儿子起床刷牙洗脸速度很快，今早给儿子带的是糯米饭包鸡柳肉，还吃了一个茶叶蛋。儿子在我提醒下稍微收...



瑾言琤行 (/u/6b387329217c?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

### 1.7 (/p/d1007663d37d?utm\_campaign=maleskine&utm\_content=note&...

诺基亚N1发布了！自以为现在最喜欢的手机品牌，没有之一。有一种情愫叫做怀旧，怀念我的父母都用过的



物品，我居然也能看到它的模样，然后，就爱上了。



Beijing (/u/d31d09d103b9?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

---

## DateTime时间格式转换大全 (/p/15200cce4c4d?utm\_campaign=maleski...

Datetime时间类型转换一览 复制代码 Console.WriteLine(DateTime.Now.ToString());

Console.WriteLine(DateTime.Now.ToShortDateString()); Con...



北风知我意 (/u/5d54a0a035e1?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

