# **基于已有项目的NDK环境配置**

以前的项目，没有勾选Include C++ Support的项目，怎样配置，才能支持NDK开发 ，特写一篇文章予以说明 。这个问题被我搁置了几天，一来是因为最近在赶项目没时间，而来我也没这样做过，所以得自己做一下试验 。

对于这类问题 ， 其实可以做一下简要的分析：

勾选了Include C++ Support为我们做了什么 ？

① 为我们配置好的NDK环境  
② 创建了CPP目录  
③ 创建CMakeLists.txt文件，主要用于cmake工程构建配置  
④ 在gradle中做了如下配置

// 配置cmake build 参数

externalNativeBuild {

cmake {

arguments "-DANDROID\_TOOLCHAIN=clang"

abiFilters "armeabi-v7a", "armeabi","x86"

}

}

// cmake构建的配置文件

externalNativeBuild {

cmake {

path "CMakeLists.txt"

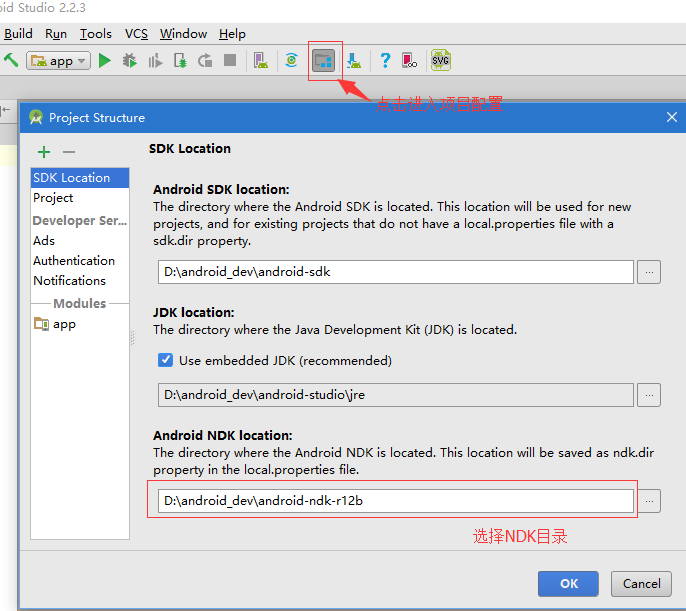
}

}

分析完之后 ， 我们照着配置就OK了 ， 下面为Demo流程 。

## **已有项目配置NDK开发环境**

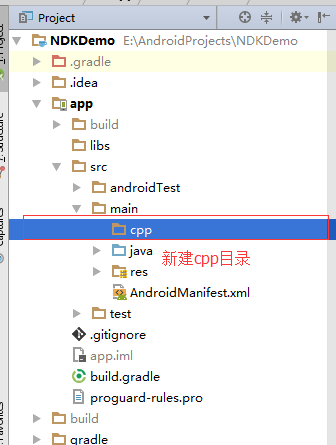
一 ， 配置NDK环境



NDK config project

二 ， 新建CPP

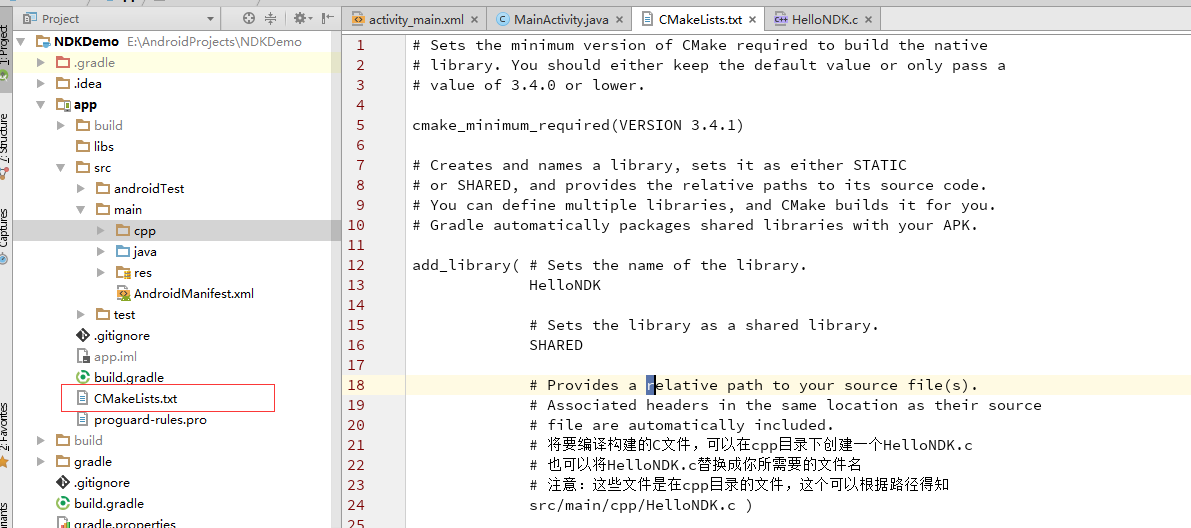
将项目目录视图，切换到project视图，在main目录下，新建cpp目录



new cpp dir

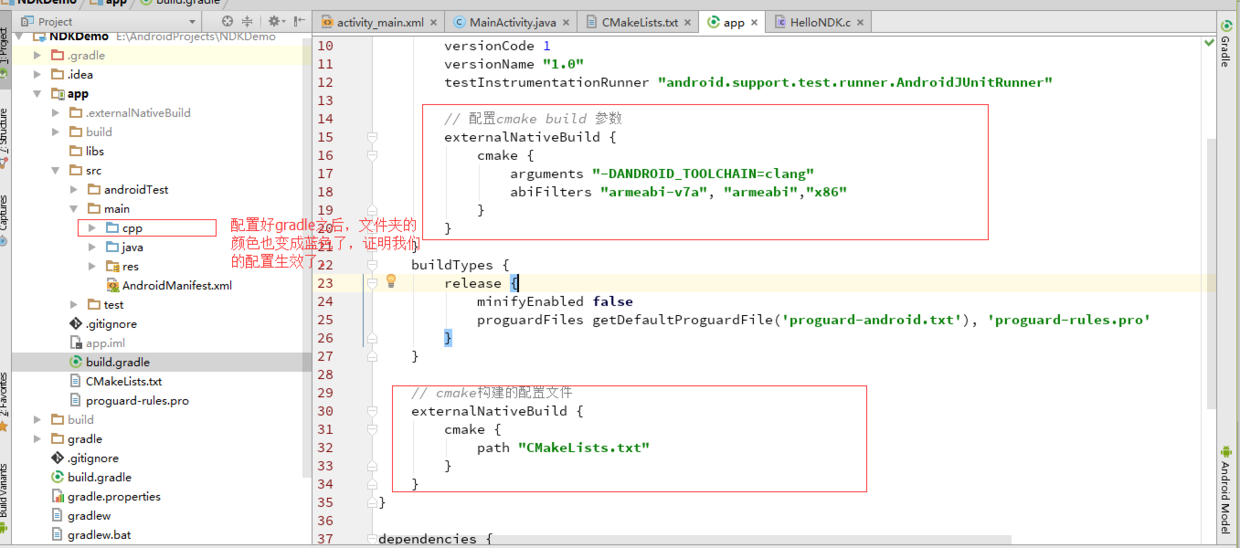
三 ，创建CMakeLists.txt

选中app右键新建文件，文件名为CMakeLists.txt，然后将以前创建好的NDK的CMakeLists.txt复制过来，或者直接复制粘贴文件也是可行的 。



创建CMakeLists.txt文件

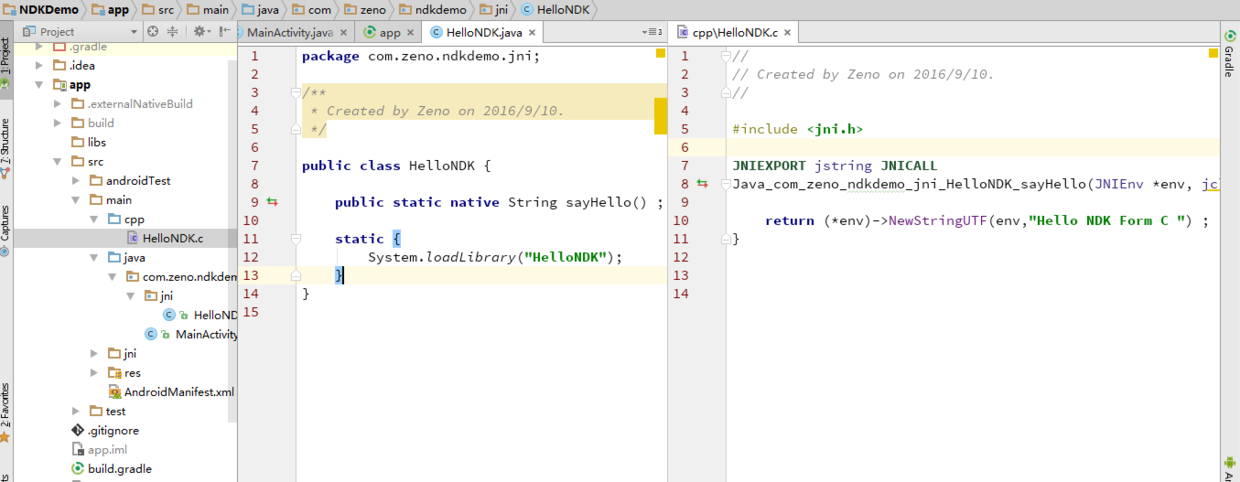
四 ， 配置gradle  
配置好gradle之后，NDK环境才算是真正搭建好 。



config gradle

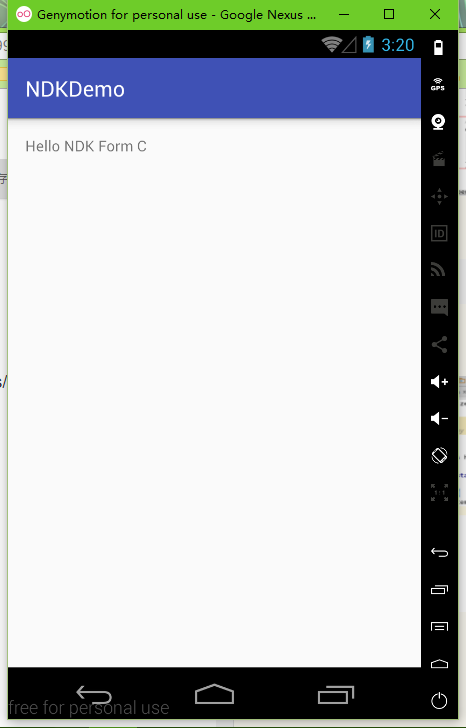
五 ， 编写测试

很简单的代码



test

****显示结果：****



ndk

## **结语**

这种问题的思路，其实很简单 ， 建立在已有的案例基础上，这种问题是最好解决的 ， 首先要观察和分析已有案例的和现有项目的区别，然后，通过手动补齐配置和一些参数变量等等 。