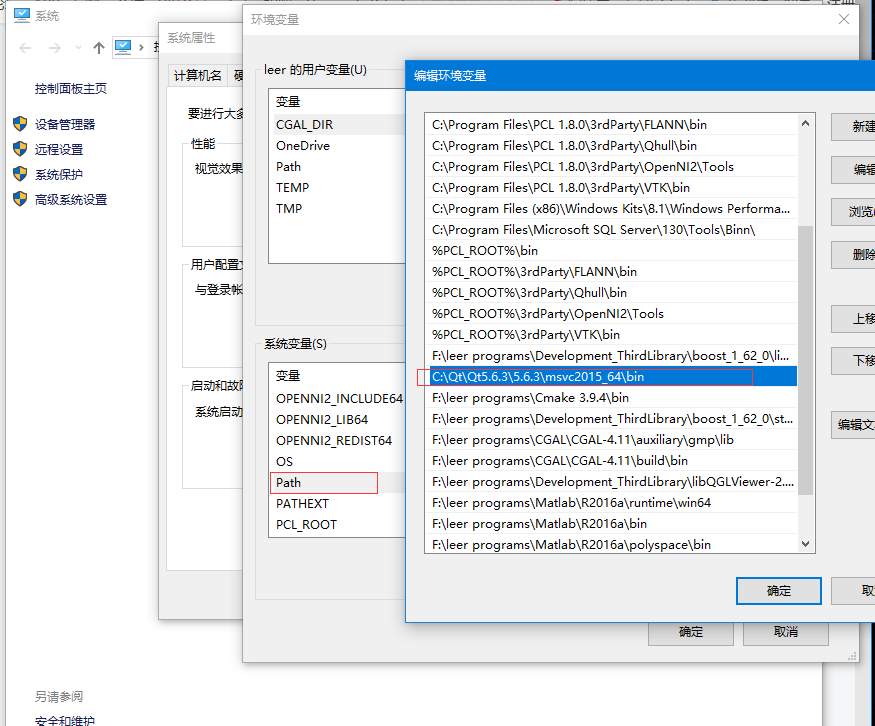
编译cc之前需要安装Qt5.6.3，vs2015，和cmake。

Qt: <http://download.qt.io/archive/qt/> qt-opensource-windows-x86-msvc2015\_64-5.6.3.exe

Vs2015:现在在官网上我只能下载到vs2017，vs2015不知道从哪些渠道进去。请自行百度。

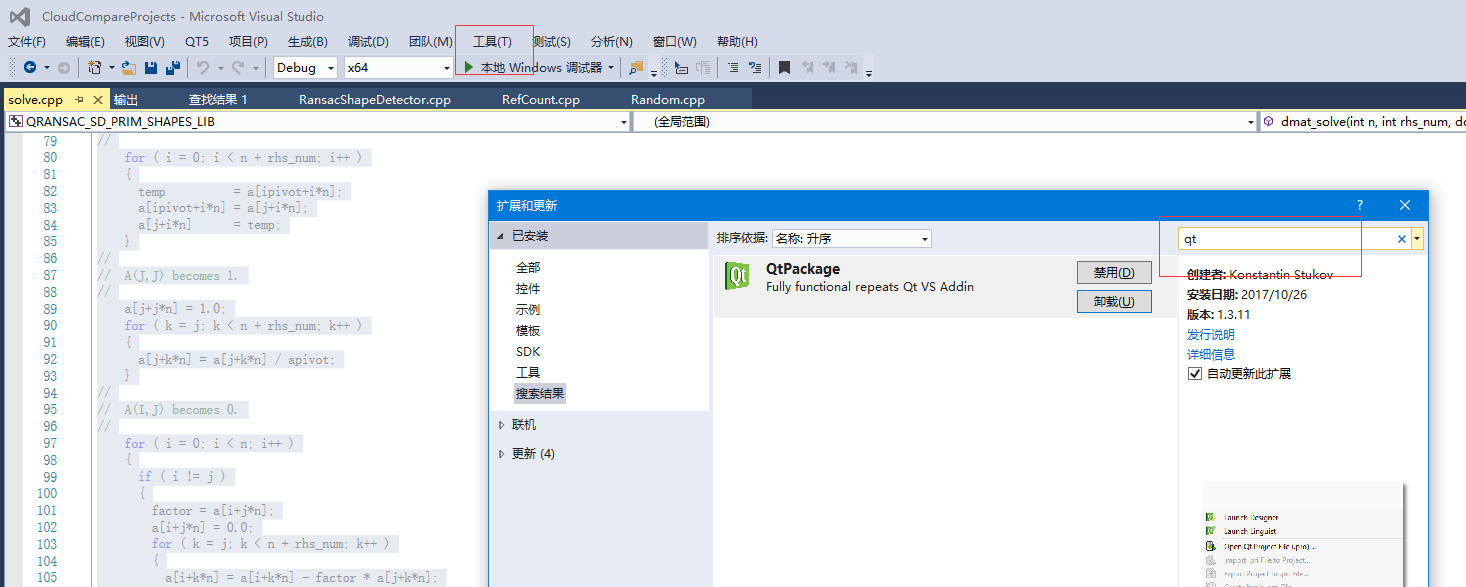
Cmake3.9.4: <https://cmake.org/files/v3.9/>,，下载cmake-3.9.6.zip

1 安装qt5.6.3之后，在path中添加环境变量，记得改成自己的安装路径

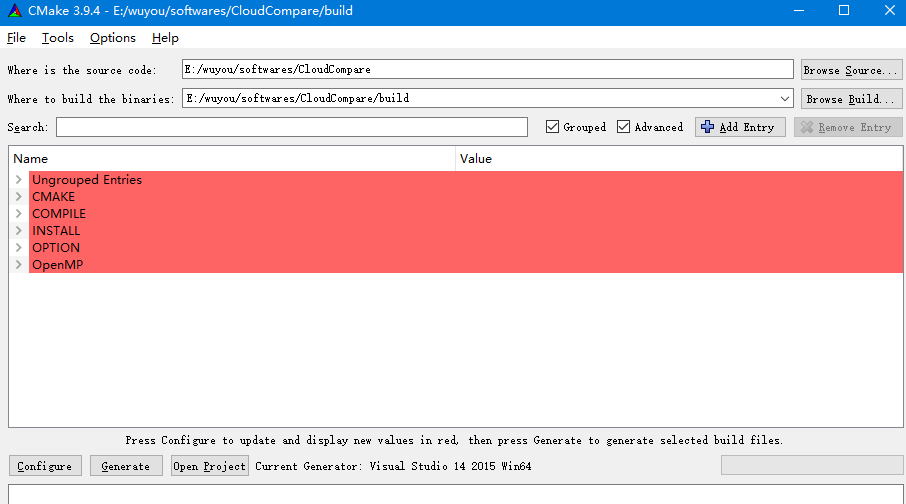


2 在vs2015中安装qt插件，这样你就可以在使用vs2015的同时愉快的和qt玩耍。

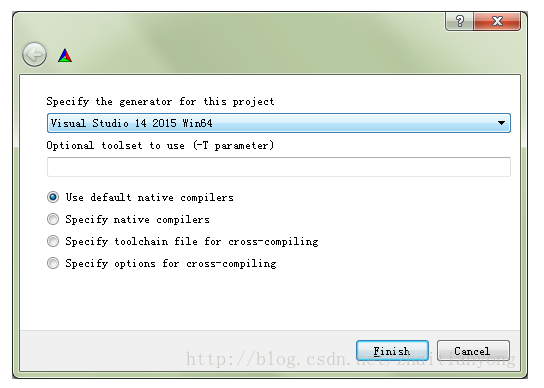
Vs的工具中有 扩展和更新 这个选项，在这里搜索qt，可以找到qt Package，点击安装就好了。



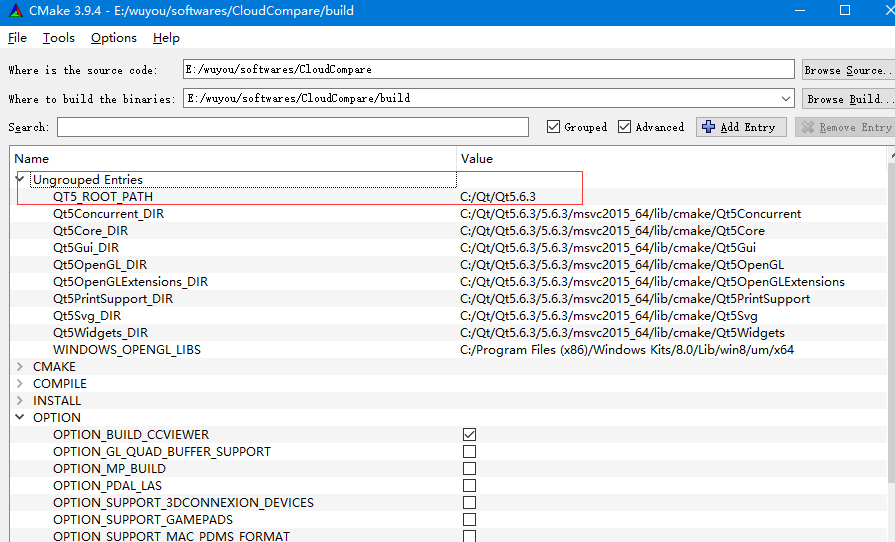
3 cmake3.9.4 解压后点击cmake-gui



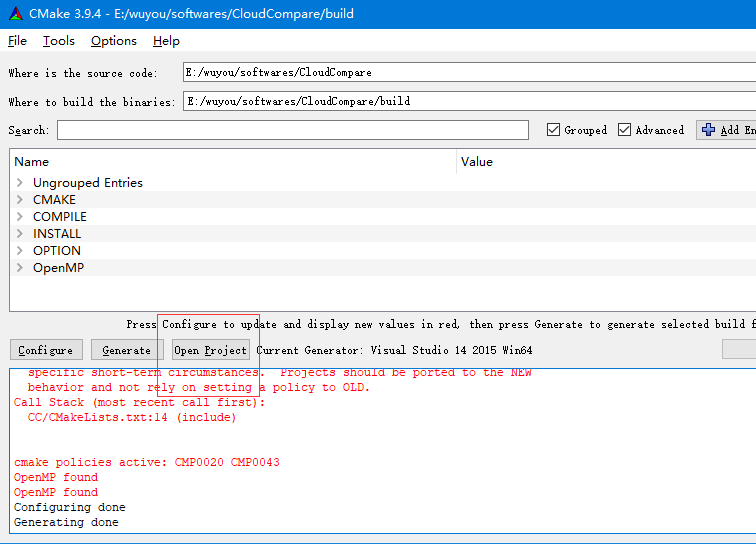
把下载下来的源文件路径输到source code那一栏，在同一个目录下创建一个build的文件夹，点击configure，编译器选择如下。



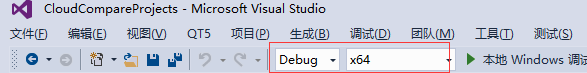
Configure之后，QT5\_ROOT\_PATH的路径为空，补上路径。



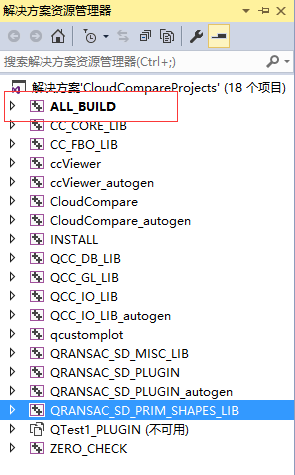
点击generate。完成之后点击Open Project.



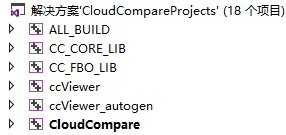
打开之后，设置调试模式：



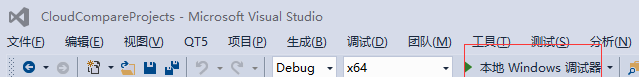
右键ALL\_BUILD，点击生成。



全部生成之后，点CloudCompare项目，右键选择设为启动项目，该项目的名字变成加粗。



点击调试：



等待一个漫长的寻找dll过程就编译完成了。

**如果无法直接使用我编译好的代码，可以参考以下教程编译初始版本。**

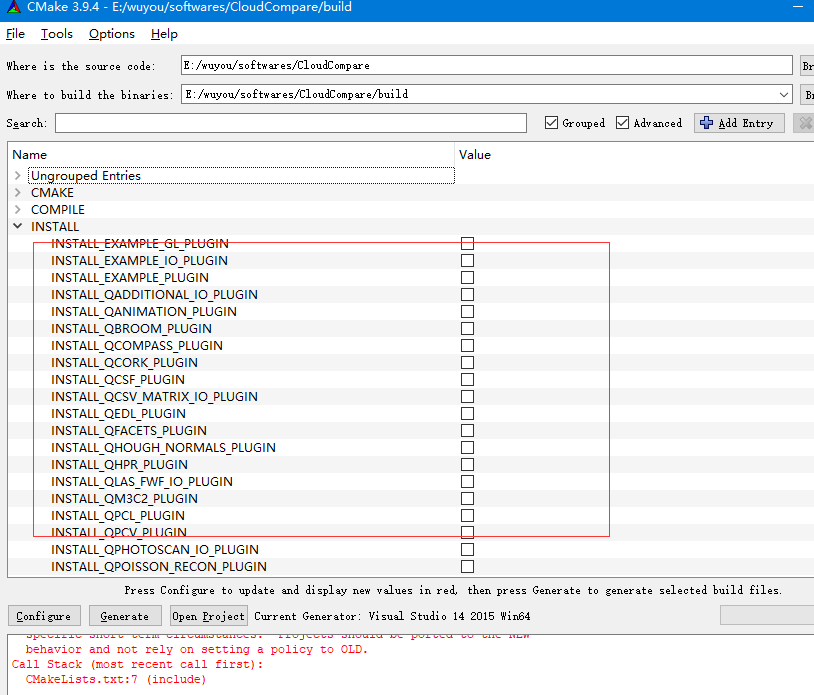
https://blog.csdn.net/zhaitianyong/article/details/78195513?utm\_source=debugrun&utm\_medium=referral

建议操作基本参照该教程。

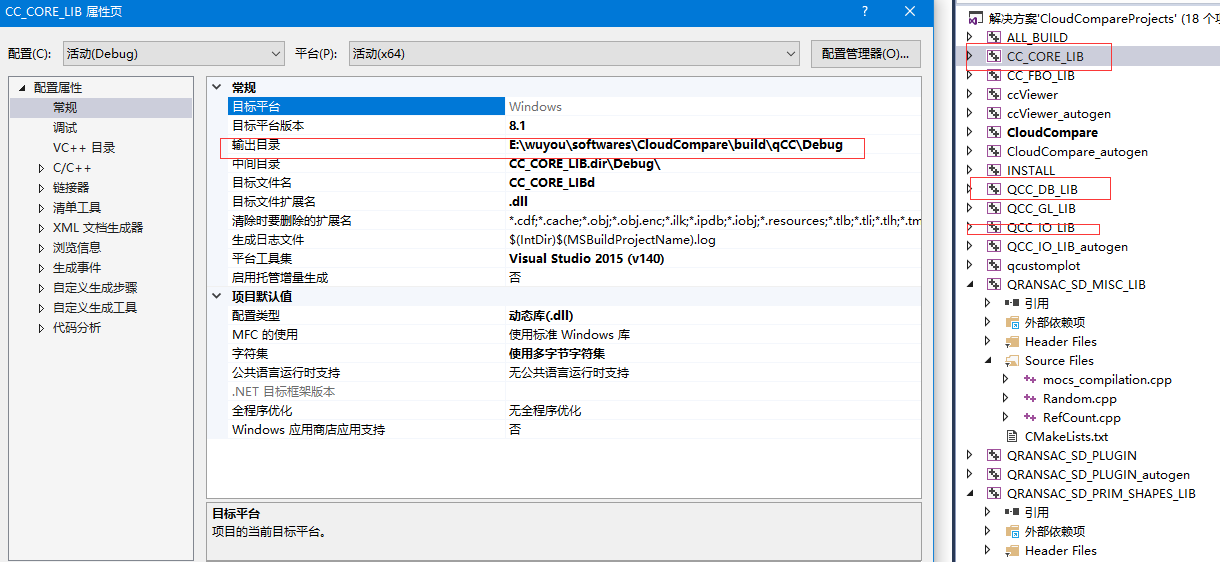
教程中使用的是qt5.6.3，Qt Package，cmake3.5.2(3.9.4我用这个版本也可以编译)，cc的代码从github上下载，win10（我）vs2015（尽量选这个版本，功能比前面几个版本全，2017版本还不成熟，有一些不可预知的错误）。

因为该教程不允许转载，所以我只是贴个链接。下面是对教程的补充：

1 如果编译的过程要添加插件：还需要把目标插件选上。



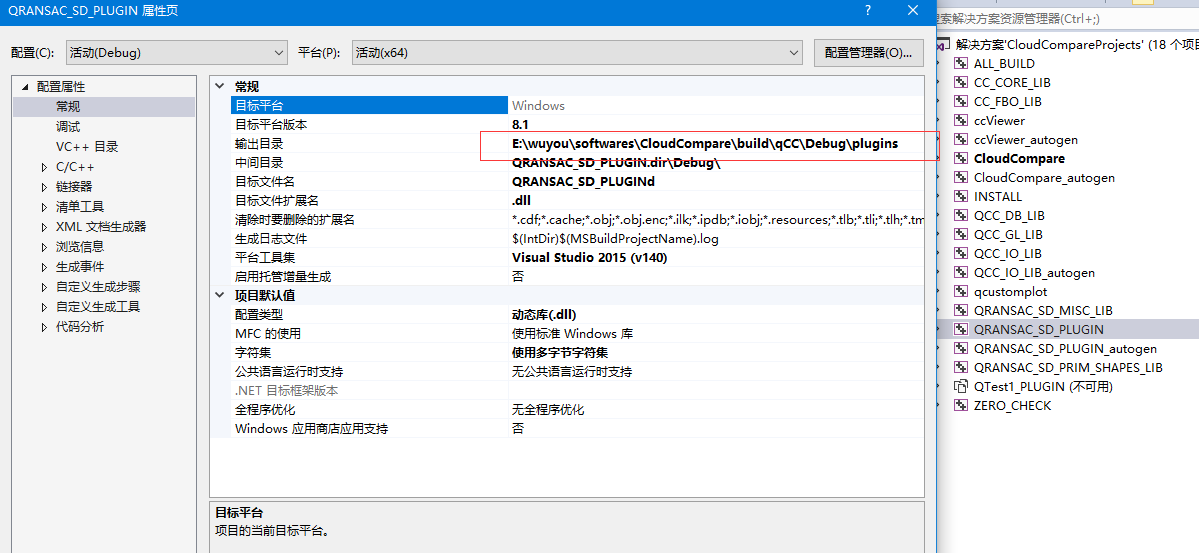
2 教程中copy 一些dll文件的操作比较麻烦，可以直接在项目工程中改路径：



3 此外，插件的生成路径需要做一下修改：



这是插件的搜索路径，所以需要把插件的路径改成上面的某个路径，我是直接选第三个路径，这样就可以把几个主要的dll都放在差不多的位置。



4 如果有遇到extern int dmat\_solve….找不到定义的问题，我是出现在Sphere.cpp和Cone.cpp中。可能是某一步的连接没做好，导致跨越项目无法引用外部函数。在vs中搜索dmat\_mat，会出来这个函数定义的地方，在solve.cpp中有，把整个函数copy出来，放到Sphere.cpp中，在Sphere.h写上函数定义。在Cone.h中也写上函数定义，不过要加上extern。