<https://blog.csdn.net/bruce_0712/article/details/53574510>

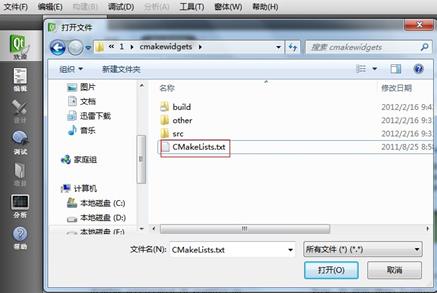
用CMake管理项目非常方便。忘记怎么用的话，看看另外一篇笔记。之前用cmake做了几个wxWidgets和c++的小练习。之前是用Editplus或UltraEdit作为文本编辑器的。最近一段时间一直在学习Qt。Qt也可以方便的集成MS visual studio中，但除了C#，其他语言的项目在vs中并不能体现出vs的优势。所以再挖掘qtcreator的潜力，以后电脑上就只需要Editplus，vs和qtcreator这几个工具了，这样很方便。

1、在Qt Creator中设置CMake

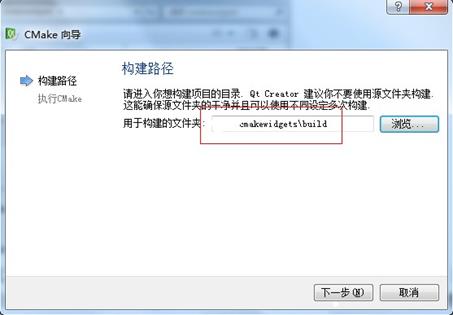


工具-----选项----构建和运行，到CMake的tab页，选择cmake的安装路径。

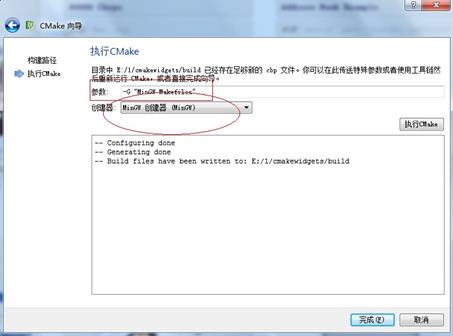
2、打开CMake项目



文件----打开文件或工程，选择CMake项目中根目录下的CMakeLists.txt文件，这时会运行CMake向导，需要选择CMake项目构建的目录。为了与CMake项目的一般处理形式一致，我选择在项目的根目录中的build作为qmake的影子模式构建（Shadow build）目录。



下一步：设置CMake参数并执行CMake：



-G参数为CMake指定要生成什么编译器的Makefile，我这里使用MinGW。如果机器中有多个不同的MinGW，创建器也记得选择一下。

设置完参数后，开始"执行CMake"，很快会生成Makefile。到这里，已经在qtcreator IDE成功打开CMake项目。

4、构建项目

到这里，项目已经成功打开。单击qtcreator左下方的小榔头，开始构建：



happy了。这次构建的是一个wxWidgets的小工程，非常顺利，与CMake、MinGW直接用命令行一样方便。

总结一下：用CMake就是为了方便生成不同编译器的Makefile，包括vs的SLN等等。这些都可以在CMake的项目中进行指定，然后在qt中，项目属性中设置工具链。例如-G 的时候指定生成vs的Makefile，在工具链中再指定nmake为编译器，那么，vs都不用打开，项目已经编译成功，而且其他同事用VS一样可以很方便的打开项目，真正做到无缝生产了。当然，这些其实都是CMake的优点，与qtcreator没有什么关系。qtcreator在这里只是个文本编辑器。