CMAKE\_GENERATOR

MinGW Makefiles

OPENSSL\_INCLUDE\_DIR

set PATH=%qtdir%;%mingw dir%;%PATH%;

like:

set PATH=C:/QT/Qt5.7.0/5.7/mingw53\_32\bin;C:/QT/Qt5.7.0/Tools/mingw530\_32\bin;C:\ProgramFiles\CMake\bin;%PATH%;

cmake -G "MinGW Makefiles" -DCMAKE\_BUILD\_TYPE=Debug "../qca-2.1.0"

cmake -G "MinGW Makefiles" -DCMAKE\_BUILD\_TYPE=Release "../qca-2.1.0"

-DCMAKE\_BUILD\_TYPE=Debug/Release

**五、      内部变量**

**CMAKE\_C\_COMPILER**：指定C编译器

**CMAKE\_CXX\_COMPILER**：

**CMAKE\_C\_FLAGS**：编译C文件时的选项，如-g；也可以通过add\_definitions添加编译选项

**EXECUTABLE\_OUTPUT\_PATH**：可执行文件的存放路径

**LIBRARY\_OUTPUT\_PATH**：库文件路径

**CMAKE\_BUILD\_TYPE**:：build 类型(Debug, Release, ...)，CMAKE\_BUILD\_TYPE=Debug

**BUILD\_SHARED\_LIBS**：Switch between shared and static libraries

# 参考

## QCA 2.1.0在Windows上用MinGW的编译过程

QCA是Qt Cryptographic Architecture的缩写，简而言之就是安全组件（如OpenSSL）的Qt封装，方便Qt程序调用，而不用写一大串C代码。QCA自然是开源的，而且好像是KDE的一个project（反正代码是在KDE的git服务器上）。

时隔四年，QCA总算发布了新版本，虽然是小版本号的变化（2.0.3 -> 2.1.0），但却是非常重要的更新！！！首先支持使用Qt5了，其次使用CMake而不是qmake来管理了。如果不支持Qt5的话，我也不会关心QCA的…………

但是目前主流Linux发行版的官方仓库还是2.0.3版的QCA，而且这次也没有Windows builds下载，只有源码包，好在编译不太麻烦，下面就记录了本次编译的过程（环境：Windows 8.1 x64 Pro, MinGW64 4.9.1 x64, MSYS, Qt 5.3.2 x64 static, OpenSSL 1.0.1j x64 static, CMake 3.1.0）。这次要编译的就是静态链接的QCA，如果你要编译动态链接的QCA，上面的工具链必须也是动态链接的！

首先下载[QCA-2.1.0源码包](http://delta.affinix.com/download/qca/2.0/qca-2.1.0.tar.gz)，随便解压到哪里。打开MSYS，先把qmake加入到PATH中：

|  |  |
| --- | --- |
| 1 | export PATH=$PATH:/d/Qt/Qt5.3.2/5.3/mingw491\_64\_static/bin |

*注意：如果你要编译静态链接库，你得先自己去编译一个静态链接的Qt*

然后cd进入qca的源码目录，执行

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | mkdir build && cd build  cmake -G "MSYS Makefiles" \  -DOPENSSL\_INCLUDE\_DIR=/usr/local64/include/ \  -DBUILD\_TESTS=OFF \  -DBUILD\_SHARED\_LIBS=OFF \  -DUSE\_RELATIVE\_PATHS=ON \  -DCMAKE\_INSTALL\_PREFIX=/usr/local64/qca \  .. |

cmake的第一个参数是指定使用MSYS的Makefile，不然在Windows下CMake会用NMake（Visual Studio的那个），下面依次是指定OpenSSL的include目录（该include下应有openssl文件夹）；关闭测试；编译静态链接库；使用相对位置（编译为relocatable package）；指定安装位置。

不知道是不是之前Qt编译的问题，我必须修改D:\Qt\Qt5.3.2\5.3\mingw491\_64\_static\lib\cmake\Qt5Network下Qt5Network\_QGenericEnginePlugin.cmake和Qt5Network\_QNativeWifiEnginePlugin.cmake文件，把所有像bearer/qgenericbearer.lib的字串改成bearer/libqgenericbearer.a（.lib改成.a，前面加个lib）。不然Qt自己宏检查文件会提示文件不存在而报错。

没问题就可以执行make了，静等几分钟然后make install就好了！

P.S. 不知道为何这样编译出来的QCA在linking阶段会失败，一大堆的undefined reference，原因未知。正在尝试换用MSVC编译

P.P.S. QCA 2.1.0不能静态链接编译，目前只能动态链接，静态就出错。[Bugtracker](https://bugs.kde.org/show_bug.cgi?id=341472)