http://www.cnblogs.com/coolbear/p/4429207.html

1. [**VS2010编译Qt4.8.2的64版本库**](http://www.cnblogs.com/coolbear/p/4429207.html)

从http://download.qt.io/archive/qt/4.8/4.8.2/下载，安装qt-win-opensource-4.8.2-vs2010.exe，这个是32位的，里面有编译好的库，可以直接用来开发，里面也有src源文件，现在要用源文件编译VS2010下使用的64位的库确定自己的编译机器是64位。

从开始菜单->Microsoft Visual Studio 2010->Visual Studio Tools->Visual Studio x64 Win64命令提示(2010)进入VS2010的命令行窗口

c:\Qt\4.8.2>configure -prefix c:\qt -debug-and-release -platform win32-msvc2010 -opengldesktop -qt-sql-sqlite -qt-zlib -qt-style-windowsvista -qt-libpng -qt-libjpeg -nomakedemos -nomake examples -nomake tests -mp

如果不想编译某一模块，例如不想使用WebKit模块，在上述命令中添加 -no-webkit

此时会有一个选项 是选择商业版还是开源版，选择开源版输入字母o  
接下来会问你是否接受许可之类的，输入字母y

完成之后，会显示  
Qt is now configured for building. Just run nmake.  
To reconfigure, run nmake confclean and configure.

c:\Qt\4.8.2>nmake

遇到错误  
platform\DefaultLocalizationStrategy.cpp(327) : error C2001: 常量中有换行符  
platform\DefaultLocalizationStrategy.cpp(327) : fatal error C1057: 宏扩展中遇到意外的文件结束  
。。。。。。。。。。  
NMAKE : fatal error U1077: “"C:\Program Files (x86)\Microsoft Visual Studio 10.  
0\VC\BIN\amd64\cl.EXE"”: 返回代码“0x2”  
Stop.  
NMAKE : fatal error U1077: “"C:\Program Files (x86)\Microsoft Visual Studio 10.  
0\VC\BIN\amd64\nmake.exe"”: 返回代码“0x2”  
Stop.  
NMAKE : fatal error U1077: “cd”: 返回代码“0x2”  
Stop.  
NMAKE : fatal error U1077: “cd”: 返回代码“0x2”  
Stop.  
解决办法：是双引号出了问题，一个是中英文问题，另一个是"的个数  
打开C:\Qt\4.8.2\src\3rdparty\webkit\Source\WebCore\platform   
\DefaultLocalizationStrategy.cpp  
定位到325行和327行，将  
"Look Up “%@”" 修改为  
"Look Up ""%@"""  
"Look Up “<selection>”" 修改为  
"Look Up ""<selection>"""  
修改保存后，重新执行nmake命令

大概需要花2~3小时的样子编译完成。我有点奇怪的是，我在配置时，写的输出路径是  
-prefix c:\qt  
我发现生成的DLL给我放在了C:\Qt\build\x64\bin目录下，其中\build\x64\bin是自动创建的，不   
过也有可能是我前几次尝试配置的结果。不管怎样，Qt的DLL是生成完了。

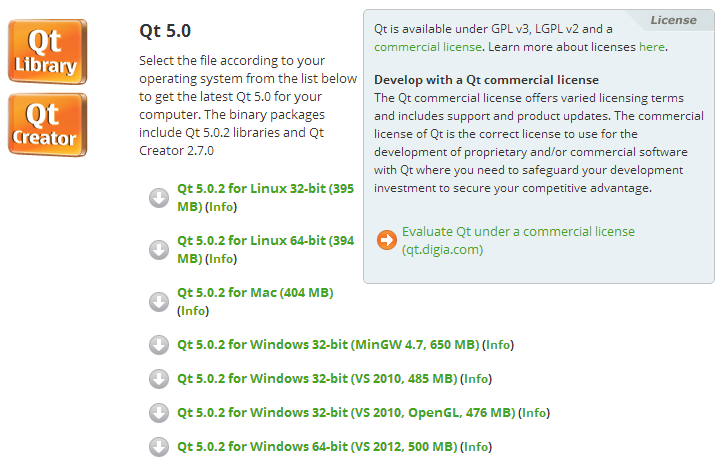
接下来执行安装命令，它负责将include和lib以及其他的东西放到C:\Qt\build\x64下  
c:\Qt\4.8.2>nmake install  
这个也需要执行很久，约半个小时

注意修改环境变量QTDIR的值C:\Qt\build\x64和QT\_BIN的值C:\Qt\build\x64\bin

1. [**Qt5学习笔记——64位 or 32位**](http://www.cnblogs.com/ppffs/p/3147703.html)

<http://www.cnblogs.com/ppffs/p/3147703.html>

   在重新采用Qt开发界面之前，碰到的第一个问题就是操作系统平台的问题。现在项目组已经同意采用了X64平台，于是我将之前用Qt4.8.2编写的程序在X64的环境下进行编译，发现报错***fatal error LNK1112:* 模块计算机类型“x64”与目标计算机类型“X86”冲突**。上网搜寻资料发现，在Windows平台下，Qt只有32位的安装文件，要编译64位的Qt程序，必须将Qt的源码进行64位编译并生成相应的bin、include、lib文件夹。又经过一番资料搜寻后，我终于发现了目前Qt项目的下载页面：<http://qt-project.org/downloads>。



    在最下面一行中，居然有Windows64位的Qt安装程序，一阵大喜之后，后面的几个字母顿时将我的喜悦都浇没了。我嘞个去，居然是VS2012的，上面的安装文件都是在VS2010的环境下，为什么要弄个VS2012的呢？真搞不懂Qt在想什么？开来只有自己下载Qt的源码进行64位编译了。

    我的编译过程主要参照博客<http://blog.csdn.net/ztz0223/article/details/8917182>，只进行了动态编译，主要步骤和参照的那篇博客一样：

1、下载源代码：

[qt-everywhere-opensource-src-5.0.2.zip](http://download.qt-project.org/official_releases/qt/5.0/5.0.2/single/qt-everywhere-opensource-src-5.0.2.zip)

2、解压到E:\Qt\_compl，重命名文件夹为qt-src-5.0.2

3、安装依赖的工具，安装的时候都要注意添加到系统path，会有对话框弹出的时候勾选即可：

1. Perl 5.8 or later

x86 <http://www.activestate.com/activeperl/downloads/thank-you?dl=http://downloads.activestate.com/ActivePerl/releases/5.16.3.1603/ActivePerl-5.16.3.1603-MSWin32-x86-296746.msi>

amd64 <http://www.activestate.com/activeperl/downloads/thank-you?dl=http://downloads.activestate.com/ActivePerl/releases/5.16.3.1603/ActivePerl-5.16.3.1603-MSWin32-x64-296746.msi>

2. Python 2.7 or later

x86 <http://www.activestate.com/activepython/downloads/thank-you?dl=http://downloads.activestate.com/ActivePython/releases/2.7.2.5/ActivePython-2.7.2.5-win32-x86.msi>

amd64 <http://www.activestate.com/activepython/downloads/thank-you?dl=http://downloads.activestate.com/ActivePython/releases/2.7.2.5/ActivePython-2.7.2.5-win64-x64.msi>

3. Ruby

amd64 <http://rubyforge.org/frs/download.php/76806/rubyinstaller-2.0.0-p0-x64.exe>

4、安装ICU，<http://download.qt-project.org/development_releases/prebuilt/icu/src/icu4c-51_1-src.zip>，解压然后，我的路径是：E:\QT Dev\安装包\5.0\icu，然后用vs2012打开文件E:\QT Dev\安装包\5.0\icu\source\allinone\allinone.sln，然后分别以debug和release模式编译一次。然后把E:\QT Dev\安装包\5.0\icu文件夹拷贝到C:\icu下面去，仅仅保留bin;include;lib三个文件夹。其他都删除。然后添加系统环境变量，我这里是添加到administrator的用户环境变量：

1. ICU\_DIR =C:\icu
2. include= %ICU\_DIR%\include
3. lib=%ICU\_DIR%\lib
4. Path=%ICU\_DIR%\bin

**注意在第四步中，Path中添加的是%ICU\_DIR%\bin，而不是新建一个Path，因为原本就已经有了一个Path（我就犯了这个错误^\_^），同时将这些变量添加到用户变量中，而不是环境变量中！**

5、开始编译：

1、开启msvc 2012的cmd进入E:\Qt\_compl\qt-src-5.0.2，开始配置，需要指定-opengl desktop，因为设计到一个把opengl转化为d3d的库，这么配置就是不使用这块了：

1. E:\Qt\_compl\qt-src-5.0.2>configure -prefix c:\qt -debug-and-release -platform win32-msvc2012 -c++11  -icu -opengl desktop -qt-sql-sqlite -qt-zlib -qt-style-windowsvista -qt-libpng -qt-libjpeg -nomake demos -nomake examples -nomake tests -mp

    置于*configure -prefix c:\qt -debug-and-release -platform win32-msvc2012 -c++11  -icu -opengl desktop -qt-sql-sqlite -qt-zlib -qt-style-windowsvista -qt-libpng -qt-libjpeg -nomake demos -nomake examples -nomake tests -mp* 这句话代表的含义，可以输入configure -help查看每个模块的定义，这句话主要是用来设置需要编译哪些模块。其中c:\qt为编译之后的目录，可以自定义。我是采用VS2010进行64位的编译，因此上面一句话应该改为：

**configure -prefix D:\Qt\5.0.2 -debug-and-release -platform win32-msvc2010 -c++11  -icu -opengl desktop -qt-sql-sqlite -qt-zlib -qt-style-windowsvista -qt-libpng -qt-libjpeg -nomake demos -nomake examples -nomake tests -mp**

**【更多关于configure的设置，可以参考博客：**[**http://www.cnblogs.com/liff-engineer/archive/2013/05/20/3088740.html**](http://www.cnblogs.com/liff-engineer/archive/2013/05/20/3088740.html)**】**

    之后继续进行下面的步骤……

2、编译结束的时候，输入nmake install，就会讲安装的东西复制到c:\qt，但是会提示报错：

1. 已复制         1 个文件。
2. copy /y E:\Qt\_compl\qt-src-5.0.2\qttools\src\linguist\phrasebooks\russian.qph c:\qt\phrasebooks
3. 已复制         1 个文件。
4. copy /y E:\Qt\_compl\qt-src-5.0.2\qttools\src\linguist\phrasebooks\spanish.qph c:\qt\phrasebooks
5. 已复制         1 个文件。
6. copy /y E:\Qt\_compl\qt-src-5.0.2\qttools\src\linguist\phrasebooks\swedish.qph c:\qt\phrasebooks
7. 已复制         1 个文件。
8. copy /y E:\Qt\_compl\qt-src-5.0.2\qttools\lib\cmake\Qt5LinguistTools\Qt5LinguistToolsConfig.cmake c:\qt\lib\cmake\Qt5LinguistTools
9. 已复制         1 个文件。
10. copy /y E:\Qt\_compl\qt-src-5.0.2\qttools\lib\cmake\Qt5LinguistTools\Qt5LinguistToolsMacros.cmake c:\qt\lib\cmake\Qt5LinguistTools
11. 已复制         1 个文件。
12. cd qtwebkit-examples-and-demos\ && ( if not exist Makefile E:\Qt\_compl\qt-src-5.0.2\qtbase\bin\qmake E:\Qt\_compl\qt-src-5.0.2\qtwebkit-examples-and-demos\qtwebkit-examples-and-demos.pro -o Makefile ) && "D:\Program Files (x86)\Microsoft Visual Studio 11.0\VC\BIN\nmake.exe
13. " -f Makefile install
14. Microsoft (R) Program Maintenance Utility Version 11.00.50727.1
15. Copyright (C) Microsoft Corporation.  All rights reserved.
16. NMAKE : fatal error U1073: don't know how to make 'install'
17. Stop.
18. NMAKE : fatal error U1077: 'cd' : return code '0x2'
19. Stop.
20. E:\Qt\_compl\qt-src-5.0.2>

3、进入E:\Qt\_compl\qt-src-5.0.2，把这个文件夹下面的makefile给修改了：

1. install\_subtargets: module-qtbase-install\_subtargets module-qtsvg-install\_subtargets module-qtxmlpatterns-install\_subtargets module-qtjsbackend-install\_subtargets module-qtdeclarative-install\_subtargets module-qtmultimedia-install\_subtargets module-qtactiveqt-install\_subtargets module-qtwebkit-install\_subtargets module-qttools-install\_subtargets module-qtwebkit-examples-and-demos-install\_subtargets module-qtimageformats-install\_subtargets module-qtgraphicaleffects-install\_subtargets module-qtscript-install\_subtargets module-qtquick1-install\_subtargets module-qttranslations-install\_subtargets module-qtdoc-install\_subtargets FORCE
3. 去掉module-qtwebkit-examples-and-demos-install\_subtargets

4、再执行nmake install

整个编译过程就OK了。

 6、编译和安装文档，如果不做这一步，那么C:\Qt\bin，目录下面的assistance.exe将是没有文档索引的

1、编译文档：

1. E:\Qt\_compl\qt-src-5.0.2>nmake docs

2、安装文档：

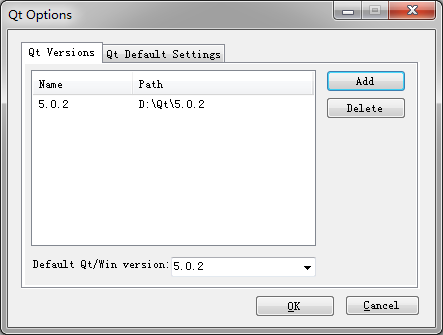
1. E:\Qt\_compl\qt-src-5.0.2>nmake install\_docs

    整个编译过程大概持续两个小时，汗！

    到此，64位的Qt5.0.2动态编译就全部结束了，在我的Qt的安装目录下，生成了938M的文件，下面来试试能不能编译64位的Qt程序……

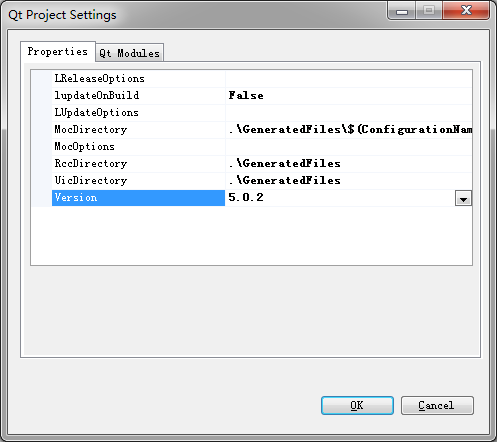
    等一下，还得把qt-vs-addin-1.2.1-opensource.exe安装一下~^\_^

    进入VS2010后，在Qt5中点击Qt Options，在Qt Versions选项卡中添加Qt5.0.2的版本，我的是这样子的：

    新建了一个Qt程序，进行64位编译之后发现，报了如下错误：

There's no Qt version assigned to this project for platform Win32.

Please use the 'change Qt version' feature and choose a valid Qt version for this platform.

    Oh My God！难道我花了两个小时编译的64位Qt不能用！还是不死心，最后又是通过搜寻，发现了解决的方法：还是在Qt5选项中，选择**Qt Project Settings**，在**Version**那一栏中，指定为**5.0.2**,我的页面如下：

    再次编译，发现已经可以成功编译为64位的Debug和Release版本，OK好好享受吧！~^\_^