

Compiling BCI2000 on Windows 8.1

April 7, 2014 3:24 PM

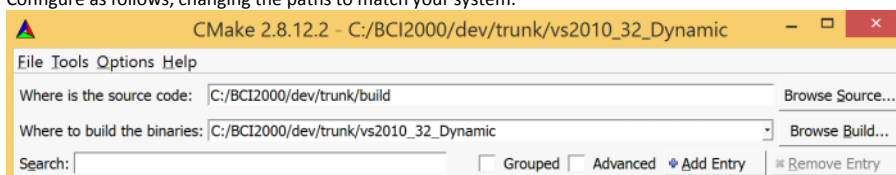
I started with a MacBook Pro Retina 15" Late 2013 model.

The OS is Windows 8.1 Pro in Bootcamp.

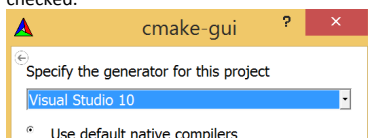
I installed the following software prior to starting this guide. i.e., nothing relevant.

Organize ▾				
Name	Publisher	Installed On	Size	Version
Apple Software Update	Apple Inc.	2014-04-04	2.38 MB	2.1.3.127
Boot Camp Services	Apple Inc.	2014-04-04	11.2 MB	5.1.5640
Google Chrome	Google Inc.	2014-04-07		33.0.1750.154
Intel® Management Engine Components	Intel Corporation	2014-04-04	20.4 MB	9.5.0.1428
Microsoft OneNote 2013 - en-us	Microsoft Corporation	2014-04-07		15.0.4569.1508
Microsoft OneDrive	Microsoft Corporation	2014-04-07	26.7 MB	17.0.4023.1211
Microsoft Visual C++ 2012 Redistributable (x64) - 11.0.50727	Microsoft Corporation	2014-04-04	20.4 MB	11.0.50727.1
Microsoft Visual C++ 2010 x86 Redistributable - 10.0.30319	Microsoft Corporation	2014-04-04	11.0 MB	10.0.30319
Notepad++	Notepad++ Team	2014-04-07		6.5.5
NVIDIA 3D Vision Driver 332.28	NVIDIA Corporation	2014-04-04		332.28
NVIDIA Graphics Driver 332.28	NVIDIA Corporation	2014-04-04		332.28
NVIDIA Update 1.15.2	NVIDIA Corporation	2014-04-04		1.15.2
NVIDIA HD Audio Driver 1.3.30.1	NVIDIA Corporation	2014-04-04		1.3.30.1
Realtek High Definition Audio Driver	Realtek Semiconductor Corp.	2014-04-04		6.0.1.5936

- Download and install MSVC2010
 - <http://www.visualstudio.com/downloads/download-visual-studio-vs>
 - Scroll down to get to Visual Studio 2010 Express - Visual C++ 2010 Express
- Download and install Cmake
 - <http://www.cmake.org/cmake/resources/software.html> Windows (Win32 Installer). I ended up with version 2.8.12.2
 - During install, "Add Cmake to the system PATH for all users."
- Download and install an SVN Client, like TortoiseSVN
<http://sourceforge.net/projects/tortoisesvn/files/latest/download>
 - During TortoiseSVN install, set the option to install "command line client tools".
- Create a BCI2000 account (NOT THE SAME AS AN ACCOUNT FOR THE MESSAGE BOARD!).
http://www.bci2000.org/wiki/index.php/Creating_a_User_Account
- Download BCI2000 source.
 - Open a command prompt.
 - Add svn to the path
 - path %path%;C:\Program Files\TortoiseSVN\bin
 - Change to the directory where you want to download BCI2000 (I used C:\BCI2000\dev)
 - Get BCI2000 (I used revision 4710 for this guide).
 - svn checkout <http://www.bci2000.org/svn/trunk> --username <your BCI2000 account name>
- Create a build directory.
 - cd C:\BCI2000\dev\trunk\
 - md vs2010_32_Dynamic
- Run Cmake
 - Configure as follows, changing the paths to match your system.



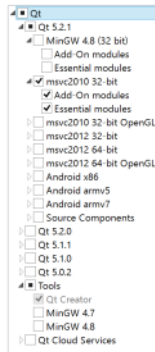
- Click Configure in the bottom left.
- Set the Generator to Visual Studio 10 and make sure "Use default native compilers" is checked.



- Download and install Qt if you have to.
 - When I did the above step, CMake was able to automatically download the pre-compiled Qt static libraries from BCI2000. However, that might not work for you (e.g., proxy), or you may want to use the newest Qt. So, then download and install Qt. This will only allow for dynamic linking to the Qt libraries and therefore your programs will not be portable to different machines.
 - Close cmake-gui
 - <http://qt-project.org/downloads>
 - Configure it to download the precompiled libraries for your development environment. If you have some apps that you want to build 64-bit (e.g., MatlabFilter to use with 64-bit Matlab), then check that box as well.

Select Components

Please select the components you want to install.



9. Finish configuring with CMake

- Open CMake again. It should have remembered your folders.
- Set some of the options. For example, I checked "BUILD_COMMANDLINE_FILTERS" and "USE_EXTERNAL_QT", and I changed "BUILD_MODULES" from "core" to "core;contrib;SignalSource;BCPy2000".
- Click Configure.
- Check your options. e.g., click BUILD_BCPY2000 if desired.
- Tell CMake where to find QT_QMAKE_EXECUTABLE if necessary.

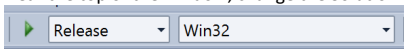
- If your system could not download Qt from BCI2000, or if you checked "USE_EXTERNAL_QT", you may get an error that it could not find Qt.
 - Next to "QT_QMAKE_EXECUTABLE", click on the word "NOTFOUND". This should change it to an editable box with a [...] button at the end (where the red circle is in the below image).
 - Click on the button and navigate to the location of qmake.exe and select it.
 - Click Configure again.
 - Keep resolving errors.
 - Qt5_DIR --> C:/Qt/5.2.1/msvc2010/lib/cmake/Qt5
 - Click Configure again and again until there are no more red boxes.

QT_QMAKE_EXECUTABLE	C:/Qt/5.2.1/msvc2010/bin/qmake.exe
Qt5Core_DIR	C:/Qt/5.2.1/msvc2010/lib/cmake/Qt5Core
Qt5Gui_DIR	C:/Qt/5.2.1/msvc2010/lib/cmake/Qt5Gui
Qt5Gui_LIB	C:/Qt/5.2.1/msvc2010/lib/cmake/Qt5Gui
Qt5MultimediaWidgets_DIR	C:/Qt/5.2.1/msvc2010/lib/cmake/Qt5MultimediaWidgets
Qt5Multimedia_LIB	C:/Qt/5.2.1/msvc2010/lib/cmake/Qt5Multimedia
Qt5Network_DIR	C:/Qt/5.2.1/msvc2010/lib/cmake/Qt5Network
Qt5OpenGL_DIR	C:/Qt/5.2.1/msvc2010/lib/cmake/Qt5OpenGL
Qt5Widgets_DIR	C:/Qt/5.2.1/msvc2010/lib/cmake/Qt5Widgets
Qt5_LIB	C:/Qt/5.2.1/msvc2010/lib/cmake/Qt5

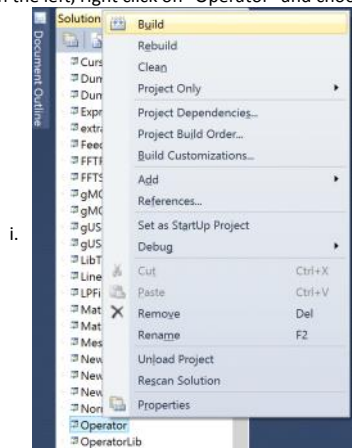
- Click Generate.
- Close CMake.

10. Build BCI2000 Operator.

- With Windows Explorer, navigate to C:\BCI2000\dev\trunk\vs2010_32_Dynamic and double click on BCI2000.sln
- Near the top of the window, change the Solution Configurations to Release.



- On the left, right click on "Operator" and choose "Build".



- Wait. This build will take longer than usual because it has to build ALL the libraries. I hope when it is finished you will see something like this in the Output pane:

```
5> Operator.vcxproj -> C:\BCI2000\dev\trunk\vs2010_32_Dynamic\CMakeFiles\core\Operator\OperatorQt\Release\Operator.exe
===== Build: 5 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

11. Build modules you will use.

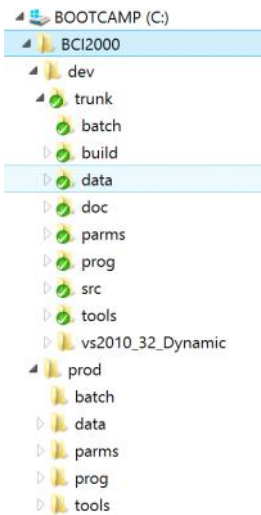
- Using the above technique, build "BCI2000Shell", "SignalGenerator", "SpectralSignalProcessing", and "CursorTask".
 - Hint: You can hold Ctrl to highlight them all and build them all at once.
- Repeat for PythonSignalSource, PythonSignalProcessing, and PythonApplication if you plan to use those.

12. Copy some dlls

- From C:\Qt\5.2.1\msvc2010\bin, copy the following dll files to C:\BCI2000\dev\trunk

- \prog
 - i. Qt5Core.dll, Qt5Widgets.dll, Qt5Gui.dll, icuin51.dll, icuuc51.dll, libGLSv2.dll, icudt51.dll, libEGL.dll, Qt5OpenGL.dll
 - ii. Alternatively, you could copy those dll files to C:\Windows\SysWOW64. If you also copy the 64-bit versions of the above dll's to C:\Windows\System32, then you can also build 64-bit BCI2000 modules and they will not conflict with the 32-bit modules.

13. Copy the important BCI2000 folders to a directory away from your development directory.
 - a. Important directories are batch, data, parms, prog, and tools. Like the following:



The green check indicates a folder managed by TortoiseSVN

14. Try it out!
 - a. Using Windows Explorer, navigate to C:\BCI2000\prod\batch and double-click on "CursorTask_SignalGenerator.bat".
 - b. Click on SetConfig
 - i. I had to click on "Config", then set my RenderingQuality to "low", before I could "SetConfig"
 - c. Click on Start.
 - d. Move the cursor around with your mouse. Notice how the mouse changes Ch1 and Ch2 signals, and how the gain on the cursor trajectory seems to be updated after every trial.

