**Setting up a 64 bit development environment for the TEDDI project**

The following documents the requirements for installing a 64 bit Qt development environment as used for the TEDDI project on a Windows platform.

1. Visual studio – VS2010

As required for using the latest Galil libraries, Matlab integration and executable performance we are currently using the Visual Studio compile and build environment.

1. QtCreator

A minimum version of 5.0.2 is required for QtCreator. Download Qt SDK for Open Source C++ development on Windows. This should equate to something like Qt\_SDK\_Win\_offline\_v1\_2\_en.exe if using the offline version. Amongst other things this includes QtLibraries, Qt Creator IDE and Qt development tools. Qt currently installs in C:\QtSDK by default.

As installed this way the SDK has VS2010 support built in, under C:\QtSDK\Desktop\Qt\nnn\msvc2010, where nnn specifies the QT version.

1. Windows SDK

Debugging of MSVC2010 C++ from QtCreator requires additional Windows debugger installation. To perform this download the widows SDK (e.g. winsdk\_web.exe), execute and install only these items as follows (an excerpt from Microsoft documentation):

* Select the **Debugging Tools** option under the **Common Utilities** if you want the x64 version of the Debugging Tools and you are installing on an x64 computer. This option automatically detects the CPU architecture of the computer on which you are installing the tools and it is the fastest method of installing the tools.
* Select the **Debugging Tools** option under the **Redistributable Packages** to download all three versions of the Debugging Tools (x86, x64, Itanium).

1. Build Windows VS2010 Qt Libraries

The standard Qt SDK is supplied with x86 (32 bit) versions of the Qt libraries. The 64 bit version of the libraries must be rebuilt for 64 bit 2Easy linking. Download the Qt everywhere version of Qt (e.g. qt-everywhere-opensource-src-5.0.2.zip). Unpack this in a suitable location (e.g. c:\Qt…). Using a VS2010 x64 command prompt window (under the menu system), change to the top level (just extracted) and execute “configure –debug-and-release –opensource –platform win32-msvc2010”. This configures Qt for building. Staying at the top level in the VS2010 x64 command prompt window execute “nmake”. The build process can take a number of hours! Execute “nmake install”.

Ensure the generated Qt x64 library folder is added to the PATH environment variable e.g. C:\Qt\5.0.2\_vs2010\_x64\lib.

1. Add 64 bit Qt build configurations

Building a 64 bit version of 2Easy requires Qt Creator debug and release build configurations to be set up. The following process will need repeating for each.

**Create the build configuration.** Select “Projects” from the left hand Qt Creator tool bar to bring up the build settings window (see figure 1). Select the relevant project from the tabbed list. From the “Edit build configuration:” pull down menu select a pre-existing configuration of the debug or release variety. From the adjacent “Add” pull down menu select “Clone Selected” and enter a suitable name when prompted (use the name to identify the QT/compiler versions to be used and whether debug or release).

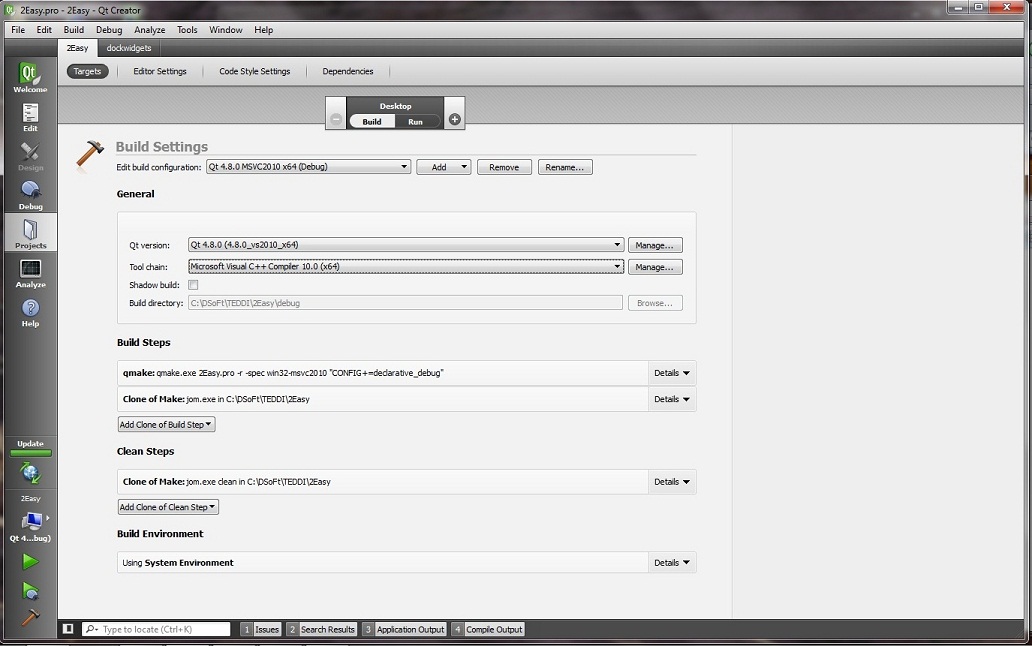


Figure 1: Qt Creator build settings

**Add new Qt version to Qt Creator and set this for the relevant build.** In the “Build Settings” windows (see figure 1) ensure the just added configuration and project are selected. Following the Qt Version line (see figure 2) press the “Manage…” button, bringing up the “Build & Run” window. Press the “Add” button to add a new Qt version. Browse to and select the required Qt version’s copy of qmake.exe (e.g. C:\Qt\5.0.2\bin\qmake.exe) and enter this using the “Open” button. Select this Qt version in the Qt Version pull down of the “Build Settings” window.

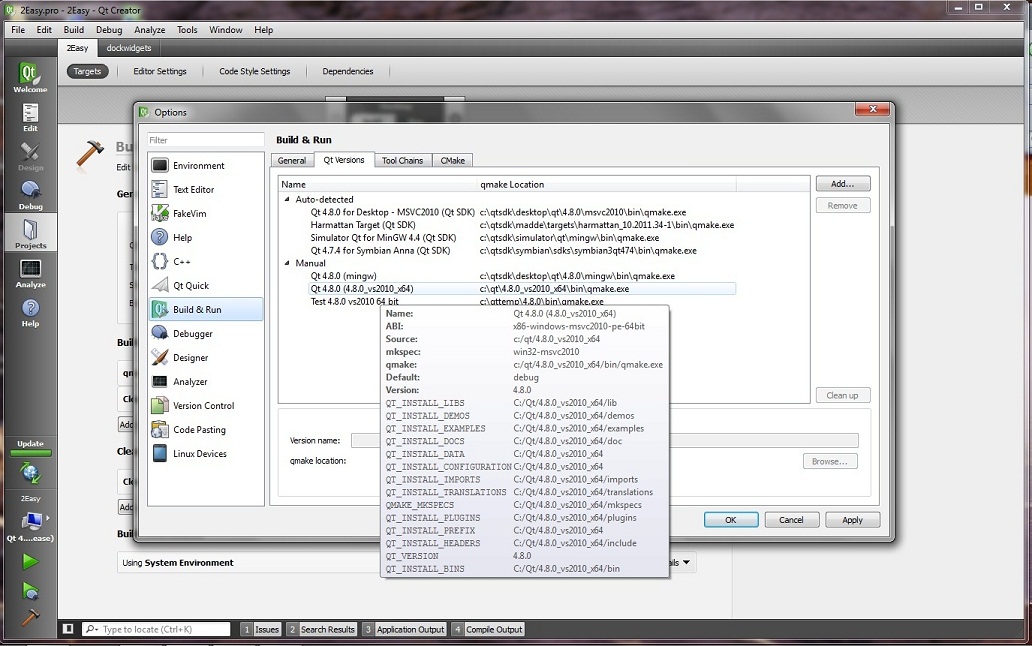


Figure : QT Creator Build & Run settings

**Set the “Tool Chain” for the relevant build.** The “Tool Chain” pull down menu (see figure 1) should have suitable possible compilers available. Select the x64 Microsoft Visual C++ 10.0 compiler.

1. Host system environment variable

The 2Easy project file determines what settings to apply for a given host from the value of the 2EASY\_SYSTEM environment variable. Set this to determine the correct configuration settings to be applied by the project file.

1. MATLAB

Download and install the 64 bit version of MATLAB. The 2Easy project file determines what settings to apply for a given host from the value of the 2EASY\_SYSTEM environment variable. Modify the 2Easy project file to add to the LIBS and INCLUDEPATH settings for your host, for example as follows:

* LIBS += $$quote(-LC:/Program Files/MATLAB/R2011b/extern/lib/win64/Microsoft/) –llibeng –llibmx -llibmat
* INCLUDEPATH += “C:/Program Files/MATLAB/R2011b/extern/include/”

Something like the following should be added to the PATH environment variable, C:/Program Files/MATLAB/R2011b/bin;C/Program Files/MATLAB/R2011b/bin/win64. The former of the two will probably have been included by the MATLAB installation, but the latter could be missing.

1. Galil

Download the GalilTools standard template (STL) C++ library e.g. LibGalil-2.0.1.447.zip. This is the mechanism currently used to access a Galil controller for the TEDDI C++ code. Extract and copy the relevant folder (e.g. vs2010-x64). Assuming the GalilTools have previously been installed a sensible location may be under the GalilTools home directory e.g. C:/Program Files/Galil/GalilTools.

The 2Easy project file determines what settings to apply for a given host from the value of the 2EASY\_SYSTEM environment variable. Modify the 2Easy project file to add to the LIBS and INCLUDEPATH settings for your host, for example as follows:

* LIBS += $$quote(-LC:/Program Files/Galil/GalilTools/LibGalil-2.0.1.447-vs2010-x64/release/) –lGalil2
* INCLUDEPATH += “C:/Program Files/Galil/GalilTools/LibGalil-2.0.1.447-vs2010-x64”

Ensure the installed Galil x64 library folder is added to the PATH environment variable e.g. C:\Program Files\Galil\GalilTools\LibGalil-2.0.1.447-vs2010-x64\release

1. LabVIEW

Download and install 64 bit NI LabVIEW 2010 development version or runtime engine. Download and install NI Labview IMAQ and DAQmx. Versions 4.6.4 and 9.8 respectively have been tested and used most recenetly. These are required by the aSpect detector dll and the Keithley class.

1. Deployment

Successful deployment of 2Easy to a Windows system requires the installation of a number of supporting libraries. Figure 3 shows the set of executables and libraries that constitute a 2Easy installation. On a system with a development environment installed these may well not all have to be in a single location. For sonsistent deployment an installer is available.

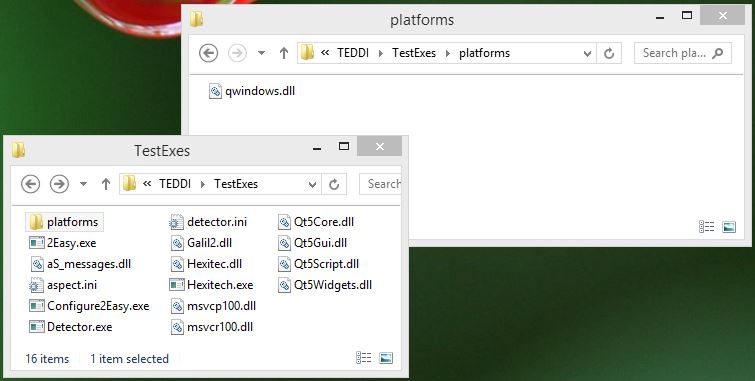


Figure : 2Easy executables and supporting libraries