1 Installation Guide

1.1 Prerequisites

The installation guide is written for systems with Windows 10 64x with at least one USB 3.0 port. It also assumes that the user will not change any of the default installation directories.

1.2 Installing dependencies

1. Install latest Qt Creator along with the Qt Library 5.7 MSVC 2015.

2. Download and install MSVC2015 compiler from:
   `http://landinghub.visualstudio.com/visual-cpp-build-tools`

3. Download and install Qt5.7 x64 compiled with MSVC2015:
   `http://download.qt.io/official_releases/qt/5.7/5.7.0/qt-opensource-windows-x86-msvc2015_64-5.7.0.exe`

   Remember to add Qt5.7-MSVC installation to QtCreator.

4. Make a new kit for compilation that includes the MSVC 2015 compiler and Qt5.7-MSVC.

5. Adjust the system environment variables:

   - Add ”C:\Qt\Qt5.7.0\5.7\msvc2015_64\bin” to path.
   - Add new variable QT_QPA_PLATFORM_PLUGIN_PATH: ”C:\Qt\Qt5.7.0\5.7\msvc2015_64\plugins\platforms\”

6. Download PCL 1.8 all-in-one installer x64 and the .pdb files:

   - PCL-1.8.0-AllInOne-msvc2015-win64.exe:
     `https://1drv.ms/u/s!ApoY_0Ymu57sg5Qig1RUpLNNF1Z1Lg`
   - pcl-1.8.0-pdb-msvc2015-win64.zip:
     `https://1drv.ms/u/s!ApoY_0Ymu57sg5QhsoFqdG_QMeCVng`

7. Install the PCL all-in-one executable remembering to tick the option to add path for all users, then extract the pdb files in the .zip files and copy them inside the bin folder at C:\Program Files\PCL 1.8.0\bin\.

   Make sure to update your user environment variables to include:

   - PCL_ROOT: ”C:\Program Files\PCL 1.8.0”

Add the following inside the Path variable in your System Variables.

- %PCL_ROOT%\bin;
- %PCL_ROOT%\3rdParty\FLANN\bin;
- %PCL_ROOT%\3rdParty\VTK\bin;
- %OPENNI2_REDIST64%;

Finally restart the computer for the new variables in the path to update.

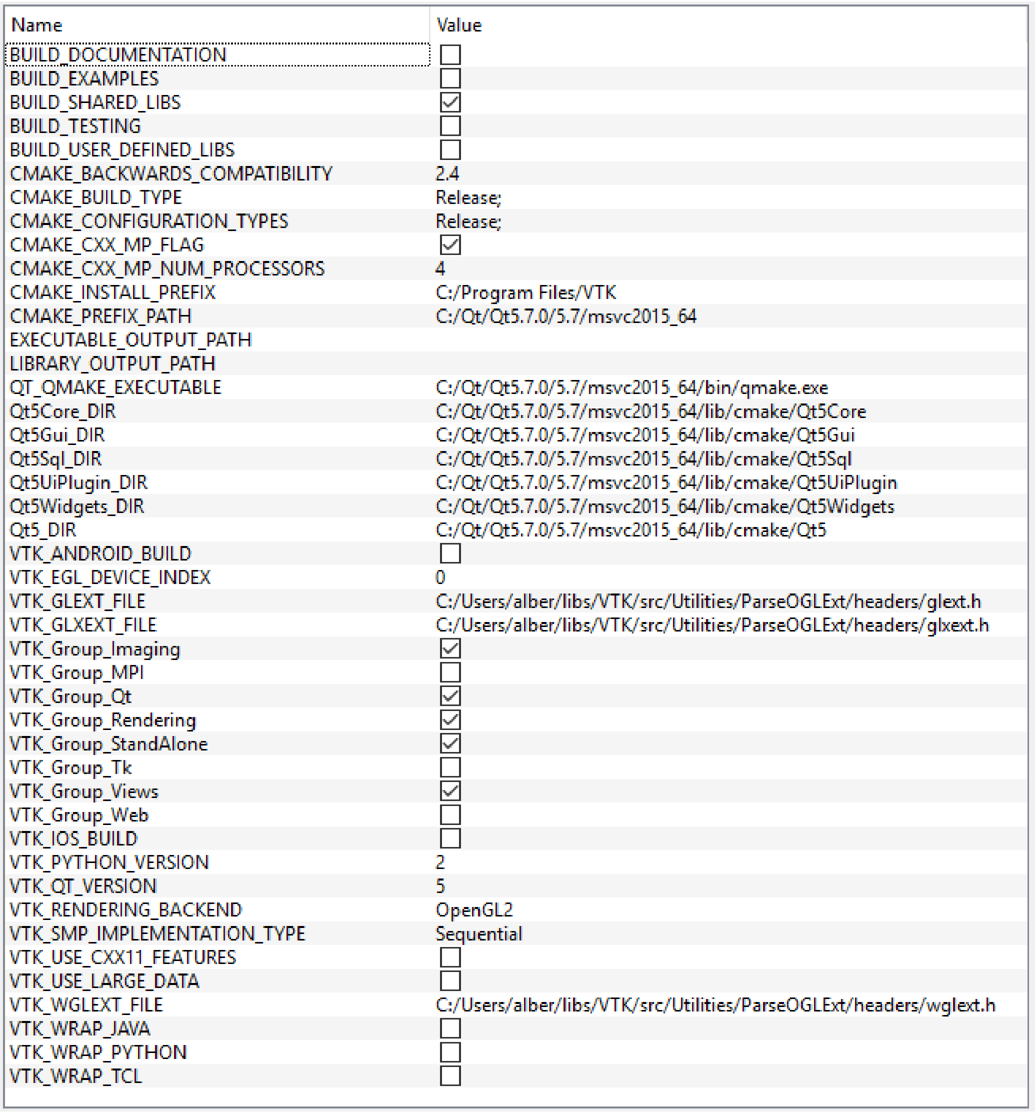
8. Since QVTK is not included by default on the PCL 1.8.0 all-in-one installer it is to be compiled from source with the Qt options enabled.

    (a) Install cmake-gui and download the VTK source from: `http://www.vtk.org/files/release/7.1/VTK-7.1.0.zip`

    (b) Extract the source code, launch cmake-gui and iteratively modify the settings and hit configure button until you have a screen similar to the one in Figure 1. Then press ”generate” to compile.

9. Download and install Kinect SDK 2.0 from: `https://www.microsoft.com/en-us/download/details.aspx?id=44561`

10. Download and install Intel RealSense SDK from: `https://registrationcenter.intel.com/en/forms/?productid=2797`



Name	Value
BUILD_DOCUMENTATION	☐
BUILD_EXAMPLES	☐
BUILD_SHARED_LIBS	☑
BUILD_TESTING	☐
BUILD_USER_DEFINED_LIBS	☐
CMAKE_BACKWARDS_COMPATIBILITY	2.4
CMAKE_BUILD_TYPE	Release;
CMAKE_CONFIGURATION_TYPES	Release;
CMAKE_CXX_MP_FLAG	☑
CMAKE_CXX_MP_NUM_PROCESSORS	4
CMAKE_INSTALL_PREFIX	C:/Program Files/VTK
CMAKE_PREFIX_PATH	C:/Qt/Qt5.7.0/5.7/msvc2015_64
EXECUTABLE_OUTPUT_PATH	
LIBRARY_OUTPUT_PATH	
QT_QMAKE_EXECUTABLE	C:/Qt/Qt5.7.0/5.7/msvc2015_64/bin/qmake.exe
Qt5Core_DIR	C:/Qt/Qt5.7.0/5.7/msvc2015_64/lib/cmake/Qt5Core
Qt5Gui_DIR	C:/Qt/Qt5.7.0/5.7/msvc2015_64/lib/cmake/Qt5Gui
Qt5Sql_DIR	C:/Qt/Qt5.7.0/5.7/msvc2015_64/lib/cmake/Qt5Sql
Qt5UiPlugin_DIR	C:/Qt/Qt5.7.0/5.7/msvc2015_64/lib/cmake/Qt5UiPlugin
Qt5Widgets_DIR	C:/Qt/Qt5.7.0/5.7/msvc2015_64/lib/cmake/Qt5Widgets
Qt5_DIR	C:/Qt/Qt5.7.0/5.7/msvc2015_64/lib/cmake/Qt5
VTK_ANDROID_BUILD	☐
VTK_EGL_DEVICE_INDEX	0
VTK_GLEXT_FILE	C:/Users/alber/libs/VTK/src/Utilities/ParseOGLExt/headers/glext.h
VTK_GLXEXT_FILE	C:/Users/alber/libs/VTK/src/Utilities/ParseOGLExt/headers/glxext.h
VTK_Group_Imaging	☑
VTK_Group_MPI	☐
VTK_Group_Qt	☑
VTK_Group_Rendering	☑
VTK_Group_StandAlone	☑
VTK_Group_Tk	☐
VTK_Group_Views	☑
VTK_Group_Web	☐
VTK_IOS_BUILD	☐
VTK_PYTHON_VERSION	2
VTK_QT_VERSION	5
VTK_RENDERING_BACKEND	OpenGL2
VTK_SMP_IMPLEMENTATION_TYPE	Sequential
VTK_USE_CXX11_FEATURES	☐
VTK_USE_LARGE_DATA	☐
VTK_WGLEXT_FILE	C:/Users/alber/libs/VTK/src/Utilities/ParseOGLExt/headers/wglext.h
VTK_WRAP_JAVA	☐
VTK_WRAP_PYTHON	☐
VTK_WRAP_TCL	☐

Figure 1: Suggested options for VTK with Qt support compilation