Download CGAL for Windows

To build CGAL from source on Windows you have to download an installer or tarball (/download/last) or you can clone the current development version of CGAL from GitHub (https://github.com/CGAL/cgal). Below are the installation instructions for the installer.

Please note that the directory layout and build process of the development version on GitHub are different from the release version.

General Prerequisites

- Download and install CMake (https://cmake.org)
- Download CGAL installer (/download/last)
- Read the appendix if you don't know how to set/modify an environment variable
- Qt 5 is only needed if you want to run CGAL demos. libQGLViewer is only needed for 3D CGAL demos.
- Identify what is the *Visual Studio Command Prompt* corresponding to the correct version (32- or 64-bit) of the compiler. Look in the start menu under the *Visual Studio* or *Microsoft Windows SDK* folders. At the command prompt, type `cl.exe` to check the version.

Prerequisites for Building the *32-bit* CGAL Library using Microsoft Visual Studio 2010/2012/2013

Boost

The 32-bit and 64-bits binary versions of Boost are available on SourceForge (http://sourceforge.net/projects/boost/files/boost-binaries/).

The installers install both boost headers and precompiled libraries. Please note that the CGAL project is not responsible for the files provided on this web site. At the moment these lines were written, the latest version was 1.59.

```
I download and run the file boost_1_59_0-msvc-xx.0-32.exe (where xx = 10 for MSVC 2010, 11 for 2012, 12 for 2013)
```

```
I choose to extract the files to a new directory c:\dev\libboost_1_59_0
```

You then need to set the following two environment variables to point respectively to the path of the libraries and the headers

```
BOOST_LIBRARYDIR=C:\dev\libboost_1_59_0\lib32-msvc-xx.0
BOOST_INCLUDEDIR=C:\dev\libboost_1_59_0
```

You also need to add the path to the Boost dll files in your PATH environment variable:

```
I add to my PATH environment variable C:\dev\libboost_1_59_0\lib32-msvc-xx.0
```

Qt 5

- Download and install the *Qt library for open source development package for 32-bit Visual Studio 2010/2012/2013* at http://www.qt.io/download/ (http://www.qt.io/download/) (here is the direct link to the offline installers (http://www.qt.io/download-open-source/#section-5)).
- Add the environment variable QTDIR pointing to the place you installed Qt.

```
I selected C:\Qt\Qt5.5.1 so QTDIR is set to C:\Qt\Qt5.5.1
```

- Add to the path the bin directory of Qt

```
I add C:\Qt\Qt5.5.1\5.5\msvcXXXX_XX\bin to my PATH
```

To avoid any conflict with another dll with the same name from another folder, add this path as the first in the list.

libQGLViewer

- Download libQGLViewer sources (not binaries) from http://www.libqglviewer.com (http://www.libqglviewer.com) and extract the archive.

```
I extract the sources in C:\dev, a new directory is created C:\dev\libQGLViewer-2.6.3
```

- Start the ***x86*** *Visual Studio Command Prompt*
- Go to the QGLviewer directory

```
I do cd C:\dev\libQGLViewer-2.6.3\QGLViewer
```

- Run qmake and then nmake

```
 I do:
qmake.exe
nmake
```

- Add the directory of the generated DLL files to the PATH

```
I add C:\dev\libQGLViewer-2.6.3\QGLViewer to my PATH
```

- Define the following environment variables:

```
I set the environment variable QGLVIEWERROOT to C:\dev\libQGLViewer-2.6.3
```

- Type exit to leave the command prompt

Prerequisites for Building the *64-bit* CGAL Library Using Microsoft Visual Studio 2010/2012/2013

Boost

The procedure for getting Boost is the same as described for the 32-bit version, but all "32" must be replaced by "64".

Qt 5

- For the 64-bit version of Qt 5, the only available installer is for Visual Studio 2013, and the instructions are the same as described for 32-bit version. For other versions of Visual Studio, please refer to these instructions: http://doc.qt.io/qt-5/build-sources.html (http://doc.qt.io/qt-5/build-sources.html).

libQGLViewer

The procedure is the same as for the 32-bit version, except that you should use the ***x64*** *Visual Studio Command Prompt*.

Building CGAL

Run the installer and select the version you want to install (this is only used to download the precompiled GMP and MPFR).

```
I choose C:\dev\CGAL-4.7 as installation directory
```

You now need to compile CGAL.

- Open CMake-GUI
- For both "Where is the source code" and "Where to build the binaries", specify the CGAL Installation folder

```
I set C:\dev\CGAL-4.7 for source and C:\dev\CGAL-4.7\build as binaries
```

- Click on Configure
- Choose "Visual Studio xx xxxx" for the 32-bit version and "Visual Studio xx xxxx Win64" for the 64-bit version
- Click on Generate

```
In the directory C:\dev\CGAL-4.7\build a solution file named CGAL.sln has been created
```

- Run Visual Studio and compile ALL_BUILD project both in Debug and Release.
- Add to your path the bin directory of the build

```
I add to my PATH C:\dev\CGAL-4.7\build\bin
```

Building an Example

Examples need neither Qt 5, nor libQGLViewer

- Open CMake-GUI
- Choose "Where is the source code:" to be the Convex_hull_2 examples directory under the CGAL installation.

```
I set it to C:\dev\CGAL-4.7\example\Convex_hull_2
```

- Choose "Where to build the binaries:" to be a build directory

```
I set it to "C:\dev\CGAL-4.7\build-example\Convex_hull_2"
```

- Click Configure and then Generate
- A solution file has been created in the build directory. Open it can compile and run it.

```
I open the file C:\dev\CGAL-4.7\build-examples\Convex_hull_2\Convex_hull_2_example.sln with Visual Studio
```

- Once the compilation is complete, you can run the executable (directly in Visual or by running the .exe created in the Debug or Release folder).

```
I open the C:\dev\CGAL-4.7\build-examples\Convex_hull_2\ch_timing.exe (the window automatically closes if not run from the console).
```

Building a Demo

All demos need Qt 5, and the 3D demos need libQGLViewer.

- Open CMake
- Choose "Where is the source code:" to be the Triangulation_3 demo directory under the CGAL installation.

```
I set it to C:\dev\CGAL-4.7\demo\Triangulation_3
```

- Choose "Where to build the binaries:" to be a build directory

```
I set it to "C:\dev\CGAL-4.7\build-demo\Triangulation_3"
```

- Click Configure and then Generate
- A solution file has been created in the build directory. Open it, build and run it.

```
I open the file C:\dev\CGAL-4.7\build-demo\Triangulation_3\Triangulation_3.sln with Visual Studio
```

- Once the compilation is complete, you can run the executable (directly in Visual or by running the .exe created in the Debug or Release folder).

Appendix: How to Set the PATH Variable or Other Environment Variables on Windows Systems

From the desktop, right-click My Computer and click properties (on Vista/Win7 click Advanced system settings on the left side).

- In the System Properties window, click on the Advanced tab.
- In the Advanced section, click the Environment Variables button.
- Finally, in the *User Variables* area, you can add or edit variables.
- To add a new one click on New, the first line is the name of the variable (PATH for example), the second the value. If several values are added, they must be separated by a *;*.
- To modify one, select it and click on *Edit...*

```
C:\Program Files;C:\Winnt;C:\Winnt\System32
```

- Add or modify the path lines with the paths you wish the computer to access. Each different directory is separated with a semicolon as shown above.