

Installing CGAL and related programs on Windows operating system

The following section explains how to install CGAL 3.4 with Boost 1.38 with QT4.5 on Windows XP/Vista with Visual Studio 2008/2005

Note that during the entire setup you need internet connection!

Instructions on adding Enviroment Variables in Windows are at the end.

1. Downloading stuff

1. First download CGAL (3.4) windows installer from here. (it is at the bottom of the page)
2. Get the latest version of CMake (currently 2.6.3) from here.
3. Download the latest version of QT (currently 4.5) from here.
4. Download the latest version of Boost (currently 1.38) from here OR the automated windows installer from here (I installed it by compiling myself but other people installed it with the installer with no problems).

2. Install CMake

1. Use the default configuration and don't forget to add C:\Program Files\CMake 2.6\bin to the PATH (if it wasn't added for you)

3. Install Boost

1. If you donwloaded the installer just install boost and continue to the next section otherwise continue to step 2.
2. Create a directory c:\boost and copy boost_1_38_0.7z into it.
3. Extract the Boost files to the C:\boost\boost_1_38_0 directory (right-click the file, go to 7zip and press extract here. If you don't have 7zip download and install from here)
4. **Don't** delete boost_1_38_0.7z file after you are done extracting!!!
5. open cmd (Strart->Run->cmd) and do : cd c:\boost\boost_1_38_0
6. in the cmd type cmake-gui . (<-- this dot is important!!!)
7. Create a new Visual Studio 2008 Generator (or the generator that fits your version of VS) by pressing Configure
8. Press "Add Entry" and create a boolean item called CMAKE_IS_EXPERIMENTAL and set the value to true
9. Press "configure"
10. Select "Build Static" if it not yet selected (infact you should have all the BUILD_XXXXX flags set to true except from BUILD_TESTING)
11. Click Configure until the Generate button becomes available
12. Click Generate
13. Close CMake. (here the dot is not important)
14. Open the newly generated solution in the boost directory and press Build -> Clean Solution when this is done press Build -> Rebuild Solution
15. Wait a long time.... If it asks you to save and restart visual studio using some sort of macro agree to it and let the macro finish.
16. add C:\boost\boost_1_38_0\bin directory to the PATH
17. Remeber I told you not to delete the 7zip file?
18. Extract the Boost files to the C:\boost\boost_1_38_0 directory (right-click the file, go to 7zip and press extract here) tell it to overwrite all existing files. (In case you are

having a de-ja-vu from step 2. You are right. For some unknown reason CMake destroys some of the files so you need to extract them all over again)

19. That's it Boost is installed.

4. Install QT

1. Be warned! This installation took me 5 hours to complete so you might want to start it at the evening and leave it for the night.
2. Run the QT installer executable (no, this is not the 5 hour part).
3. Yes, you might need MinGW (not the source code) so download and install that to C:\MinGW. It is easiest to install directly through the QT installer. If you run into problems (with trolltech mirror), you can download the minGW installer separately from sourceforge.net (currently version 5.1.4), install it, and then continue the QT installation, disregarding the warning message about version compatibility.
4. once the installation of QT is finished Add C:\MinGW\bin to the path (if it wasn't done for you)
5. If you open the lib directory of QT you will see it is filled with a bunch of files but none of them are lib files. So now let's create the lib files.
6. Open Visual Studio Command Prompt (Start->Programs->Visual Studio->Visual Studio Tools->Visual Studio Command Prompt) and write "cd c:\Qt\4.5.0"
7. in the black window write "configure"
8. agree to the license and wait a while (still not 7 hours but I would go eat dinner about now)
9. in the cmd window write "nmake" and go to sleep. This will take about 5 hours (at least on my old slow computer)
10. Add C:\Qt\4.5.0\bin to the path
11. That's it QT is installed
12. Restart your computer (don't forget to bookmark this page or something)

5. Install CGAL

1. Finally we install the crown jewel. Run the installer and select your compiler and **all** the variants, next, next.
2. I suggest installing CGAL to C:\CGAL\CGAL-3.4 ; in any case **never install CGAL to a path with spaces**
3. Tell the setup to create all the environment variables and wait for the install to complete. Including the downloads at the end.
4. Add the environment variable QTDIR = C:\Qt\4.5.0
5. Add the environment variable BOOST_ROOT = C:\boost\boost_1_38
6. Open the command prompt (start->run->cmd)
7. Type cd C:\CGAL\CGAL-3.4
8. run cmake-gui . (<-- this dot is important again!)
9. press configure and select your compiler
10. If you have Cygwin installed (like me) the configure will fail because it was looking for the GMP and the MPFR code in the wrong place. Edit the GMP_INCLUDE_DIR and MPFR_INCLUDE_DIR to be C:/CGAL/CGAL-3.4/auxiliary/gmp/include
11. Edit the CMAKE_BUILD_TYPE to Debug
12. Select WITH_demos and WITH_examples if you wish the demos and the examples to be installed (although I don't see any reason to include those. You can just compile what you need when you need it).
13. You may need to add the variable Boost_INCLUDE_DIR (=C:/boost/boost_1_38_0), and edit the variables Boost_THREAD_LIBRARY_DEBUG (=C:/boost/boost_1_38_0/lib/libboost_thread-vc90-mt-gd-1_38.lib) and Boost_THREAD_LIBRARY_RELEASE (= C:/boost/boost_1_38_0/lib/libboost_thread-vc90-mt-1_38.lib)
14. Press Configure until you can press Generate and then press Generate
15. A solution file will be created here : C:\CGAL\CGAL-3.4. Open it with your visual studio.
16. Close CMAKE
17. Press Build->Clean Solution, Press Build->Rebuild Solution. If you did everything I wrote before, including step 17 in the boost install the compile should work without

problems (maybe some examples will not compile because you are missing 3rd party components but that isn't important).

18. You are done! CGAL is installed.

At the end of the installation, your PATH should look something like that (this is my PATH) :

```
PATH=C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;
C:\Program Files\ATI Technologies\ATI.ACE\Core-Static;c:\Program Files\Microsoft
SQL Server\90\Tools\bin\;C:\cygwin\bin;C:\cygwin\usr\bin;C:\cygwin\usr\local\b
in;C:\cygwin\usr\X11R6\bin;C:\gs\gs8.63\bin;C:\gs\gs8.63\lib;C:\boost\boost_1_38
_0\bin;C:\cygwin\usr\include;C:\MinGW\bin;C:\Qt\4.5.0\bin;C:\Program Files\CMake
2.6\bin;C:\CGAL\CGAL-3.4\auxiliary\gmp\lib
```

1. Creating a Visual Studio Project that uses CGAL and QT

1. Now that you have installed CGAL you need to configure your visual studio to work with it.
2. Open an empty C++ project using the Win32 Console Application wizard (select empty project and click finish) File->New->Project ... Other Languages -> C++ -> Win32 -> Win32 Console Application
3. When the solution loads up go to Tools->Options ... Projects and Solutions -> VC++ Directories
 1. Make sure that Executable Files (Combobox at the top right) contains the \$(PATH) variable (at the end there)
 2. Select Include Files in the combobox and add :
 1. C:\CGAL\CGAL-3.4\auxiliary\gmp\include
 2. C:\Qt\4.5.0\include
 3. C:\boost\boost_1_38_0
 4. C:\CGAL\CGAL-3.4\include
 3. They should appear at that order at the topmost part of the list (use the arrows to move items up and down)
 4. Select Library Files from the combobox and add :
 1. C:\CGAL\CGAL-3.4\auxiliary\gmp\lib
 2. C:\CGAL\CGAL-3.4\lib
 3. C:\boost\boost_1_38_0\lib
 4. C:\Qt\4.5.0\lib
 5. Press OK until you are back at the main window
 6. Right click your project and select Properties
 7. Go to Configuration Properties -> C/C++ -> General. In there you will see Additional Include Directories. Copy this line there : "C:\Qt\4.5.0\include\QtCore", "C:\Qt\4.5.0\include\QtCore", "C:\Qt\4.5.0\include\QtGui", "C:\Qt\4.5.0\include\QtGui", "C:\Qt\4.5.0\include", "C:\Qt\4.5.0\include\ActiveQt", "debug", ".", "C:\Qt\4.5.0\mkspecs\win32-msvc2005
 8. QT has a bunch of directories in their include so you need to add each directory individually. The ones I write there should be enough for a beginner
 9. Go to Configuration Properties -> Linker -> Input and copy the following line to the Additional Dependencies line : C:\Qt\4.5.0\lib\qtmaind.lib C:\Qt\4.5.0\lib\QtGuid4.lib C:\Qt\4.5.0\lib\QtCored4.lib
 10. You will generally need some of the libs as dependant files. The ones I gave above should be enough for a beginner program.
 11. That's it, you can write your code and compile it here is an example of a main.cpp that should compile now :

----- CUT HERE -----

```
#include <iostream>
#include <boost/format.hpp>
```

```

#include <QtGui>
#include <CGAL/Qt/GraphicsViewNavigation.h>
#include <QLineF>
#include <QRectF>
int main(int argc, char **argv)
{
  QApplication app(argc, argv);
  QGraphicsScene scene;
  scene.setSceneRect(0,0, 100, 100);
  scene.addRect(QRectF(0,0, 100, 100), QPen(QColor(255,0,0)));
  scene.addLine(QLineF(0,0, 100, 100));
  scene.addLine(QLineF(0,100, 100, 0));
  QGraphicsView* view = new QGraphicsView(&scene);
  CGAL::Qt::GraphicsViewNavigation navigation;
  view->installEventFilter(&navigation);
  view->viewport()->installEventFilter(&navigation);
  view->setRenderHint(QPainter::Antialiasing);
  view->show();
  return app.exec();
}

```

----- CUT HERE -----

Good Luck and feel free to ask us questions.

Setting up PATH variable or other Environment variables on windows systems

1. From the desktop, right-click My Computer and click properties.
2. (on Vista click Advanced system settings on the left side)
3. In the System Properties window, click on the Advanced tab.
4. In the Advanced section, click the Environment Variables button.
5. Finally, in the Environment Variables window, highlight the path variable in the Systems Variable section and click edit. 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 below.

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

The following section explains how to install older versions of CGAL:

1. **Install Microsoft Compiler**
 - a. CGAL needs at least Visual C++ 7.1 (or, of course, gcc running under Cygwin – but this is not the subject of this document).
 - b. The best **free** options is probably to download the Express version of Visual C++.
2. **Installing Boost** - installing boost is quite easy using the BoostPro Computing installer (at the time this document was written the updated version of boost was 1.36.0).
3. **Installing CGAL**
 - a. The Windows installer is pretty convenient: <http://www.cgal.org/download.html> (Notice that you should tell the installer where Boost is installed).
 - b. **Visual Studio 2008 users:**
The CGAL installation contains no binaries for Visual Studio 2008 (VC9.0). You

need to compile GMP and MPFR or download this archive file that contains the compiled binaries for CGAL, GMP and MPFR (x86) and also updated auto_link.h to enable auto-linking of projects. Just open the archive to the CGAL installation directory. If encountered in any problems contact me.