CGAL Easy installation guide

## Introduction

This article describes how to install CGAL 3.5 and create a simple triangulation program using MS Visual Studio 2005 environment. CGAL installation-guidelines [already exist and detailed](http://www.cgal.org/Manual/3.5/doc_html/installation_manual/Chapter_installation_manual.html), but it seems that it’s still not fairly easy to bring all this knowledge into a piece of code that runs. This article will try to specify the easiest way to have CGAL working on your machine. Hopefully, this will help others to use CGAL more easily. Trying to avoid multiple installation workflows (platform dependent etc…) I will describe a single and relatively easy workflow.

## Index

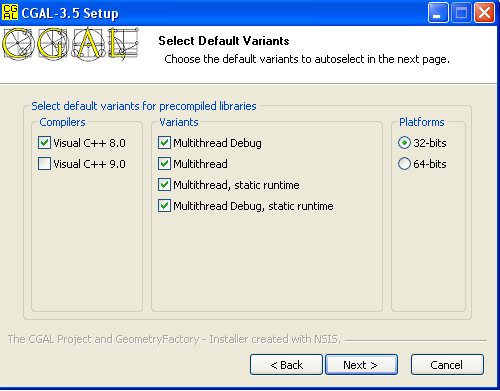
1. Prerequisite development environment
2. Downloading and installing CGAL 3.5
3. Downloading and installing CMAKE 2.8
4. Building/Installing 3rd party libraries
   1. boost
   2. zlib
   3. QT
5. Building CGAL solution and libraries using CMake
6. Building and running CGAL triangulation\_2 example
7. Creating your own CGAL based triangulation program

### Prerequisite development environment

Microsoft Visual Studio 8.0 (2005) development environment

### Downloading and installing CGAL

In order to create CGAL based application you need CGAL source code and run-time libraries. The following describes how to install CGAL as well as CGAL examples and demos

* Download and run CGAL 3.5 installation setup from <http://gforge.inria.fr/frs/download.php/24196/CGAL-3.5-Setup.exe>. it can also be reached from <http://www.cgal.org/cgi-bin/cgal_download.pl>
* Select the following variants to be installed (compiler, md/mt run-time libraries)
* Select defaults in all other setup screens. **I assume that CGAL installation is under c:\Program Files\CGAL-3.5**

1. Downloading and installing CMake

CMake is a cross-platform tool, but we will use it to create the CGAL Visual Studio projects. First, we will install CMAKE 2.8:

* Download CMake 2.8 Windows Installer from <http://www.cmake.org/files/v2.8/cmake-2.8.0-win32-x86.exe>
* Select to add CMake to PATH (second installation dialog)
* Select defaults for all other dialogs

1. Building 3rd party libraries

CGAL uses 3rd party libraries that need to be installed before running the CMake utility.

* 1. Boost

Download and extract boost 1\_41\_0 version from <http://sourceforge.net/projects/boost/files/boost/1.41.0/boost_1_41_0.zip/download>

I will describe the easiest way to build the boost libraries:

* Open VS command prompt using : Start->Programs->Microsoft Visual Studio 2005->Visual Studio Tools->Visual Studio 2005 Command Prompt
* Navigate to boost root directory, for example **D:\boost\boost\_1\_41\_0**.
* Run boostrap command (just write boostrap and Enter)
* Run .\bjam command

Expected results - **D:\boost\boost\_1\_41\_0\stage\lib** contains all boost libraries.

**You will probably need to make sure that %TMP% and %TEMP% environment variables point to an address without spaces**

* 1. zlib

Download and extract zlib 1.2.3 libraries from <http://gnuwin32.sourceforge.net/packages/zlib.htm>

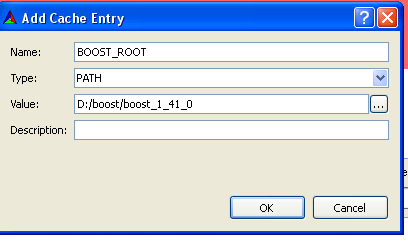
* 1. QT

This article will not describe QT implementation with CGAL. We will build CGAL without QT, but this extension is necessary for GUI development.

1. Building CGAL with Cmake

This is probably the most important part of the process. It will create the VS CGAL environment  
**note: You will probably need to make sure that %TMP% and %TEMP% environment variables point to an address without spaces before doing the bellow operations**

* Open CMake GUI using Start->Programs->CMake 2.8->CMake (cmake gui)
* Press File->Delete Cache in order to verify working ‘from scratch’
* Enter the CGAL root directory (should be C:/Program Files/CGAL-3.5) both in ‘Where is the source code’ and ‘Where to build the binaries’ input panels
* Press ‘Configure’
* Select ‘Visual Studio 8 2005’
* Press ‘Finish’
* The lower window will now contain all missing information. It will probably contain Boost errors (in red), Zlib errors, QT errors, etc... we will now fix the errors
* Add the following entry using ‘Add Entry’



* Pres ‘Configure’ again.
* You should now see the acctual Boost\_dirs and Boost\_Thread\_Library
* I will skip handling zlib and QT make because this is optional
* Press ‘Generate’

Expected results: C:\Program Files\CGAL-3.5 contains a CGAL.sln

* Open the solution in VS and press Build->Build Solution (it is recommended to build both Debug and Release)

Expected Results: CGAL libraries are built under C:\Program Files\CGAL-3.5\lib

1. Building and running CGAL triangulation\_2 example

In order to be able to run CGAL example you need to cmake the example and create the VS project. You can do this easily by:

* Open VS command prompt and perform the following commands:

cd c:/Progran Files/CGAL-3.5/examples/Triangulation\_2

cmake -DCGAL\_DIR=”C:/Program Files/CGAL-3.5” .

make

Expected result: a VS solution was created under c:/Progran Files/CGAL-3.5/examples/Triangulation\_2

And you can open this solution, build and run !

1. Building your own simple CGAL based application

In order to create your own CGAL application you will need to do the following:

* Create a new directory for the solution : d:\my\_triang
* Create a cpp file that contains main function
* Copy C:\Program Files\CGAL-3.5\examples\Triangulation\_2\CMakeLists.txt to the new directory
* Edit CMakeLists.txt:
  + Change the project name
  + Change create\_single\_source\_cgal\_program parameter to contain the new cpp file you just create or use add\_executable and specify all sources (search CMake commands documentation for further information)
* Create the VS studio solution as described in section 6.