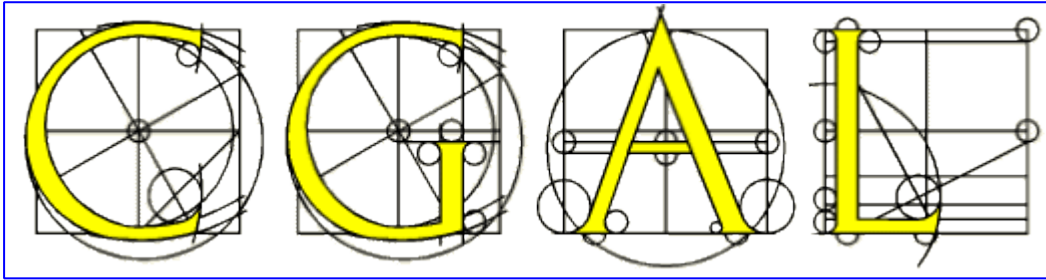


CGAL 的安装编译

eryar@163.com

关键字 Key Word: CGAL, C++, Delaunay Triangulation, Voronoi diagram,

一、引言 Introduction



CGAL, Computational Geometry Algorithms Library, 计算几何算法库。设计目标是以 C++ 库的形式提供方便、高效、可靠的几何算法。CGAL 可用于各种需要几何计算的领域, 如计算机图形学, 科学可视化, 计算机辅助设计与建模, 地理信息系统, 分子生物学, 医学成像, 机器人运动规划, 网格生成, 数值方法等。

The goal of the CGAL Open Source Project is to provide easy access to efficient and reliable geometric algorithms in the form of a C++ library. CGAL is used in various areas needing geometric computation, such as: computer graphics, scientific visualization, computer aided design and modeling, geographic information systems, molecular biology, medical imaging, robotics and motion planning, mesh generation, numerical methods...

计算几何算法库 (CGAL) 提供计算几何相关的数据结构和算法, 如三角剖分 (2D 约束三角剖分及二维和三维 Delaunay 三角剖分), Voronoi 图 (二维和三维的点, 2D 加权 Voronoi 图, 分割 Voronoi 图等), 多边形 (布尔操作, 偏置), 多面体 (布尔运算), 曲线整理及其应用, 网格生成 (二维 Delaunay 网格生成和三维表面和体积网格生成等), 几何处理 (表面网格简化, 细分和参数化等), 凸壳算法 (2D, 3D 和 dD), 搜索结构 (近邻搜索, kd 树等), 插值, 形状分析, 拟合, 距离等。

The Computational Geometry Algorithms Library (CGAL), offers data structures and algorithms like triangulations (2D constrained triangulations and Delaunay triangulations in 2D and 3D, periodic triangulations in 3D), Voronoi diagrams (for 2D and 3D points, 2D additively weighted Voronoi diagrams, and segment Voronoi diagrams), polygons (Boolean operations, offsets, straight skeleton), polyhedra (Boolean operations), arrangements of curves and their applications (2D and 3D envelopes, Minkowski sums), mesh generation (2D Delaunay mesh generation and 3D surface and volume mesh generation, skin surfaces), geometry processing (surface mesh simplification, subdivision and parameterization, as well as estimation of local differential properties, and approximation of ridges and umbilics), alpha shapes, convex hull algorithms (in 2D, 3D and interpolation and placement of streamlines), shape analysis, fitting, and distances (smallest enclosing sphere of points or spheres, smallest enclosing ellipsoid of points, principal component analysis), and kinetic data structures.

More information please see: <http://www.cgal.org/>.

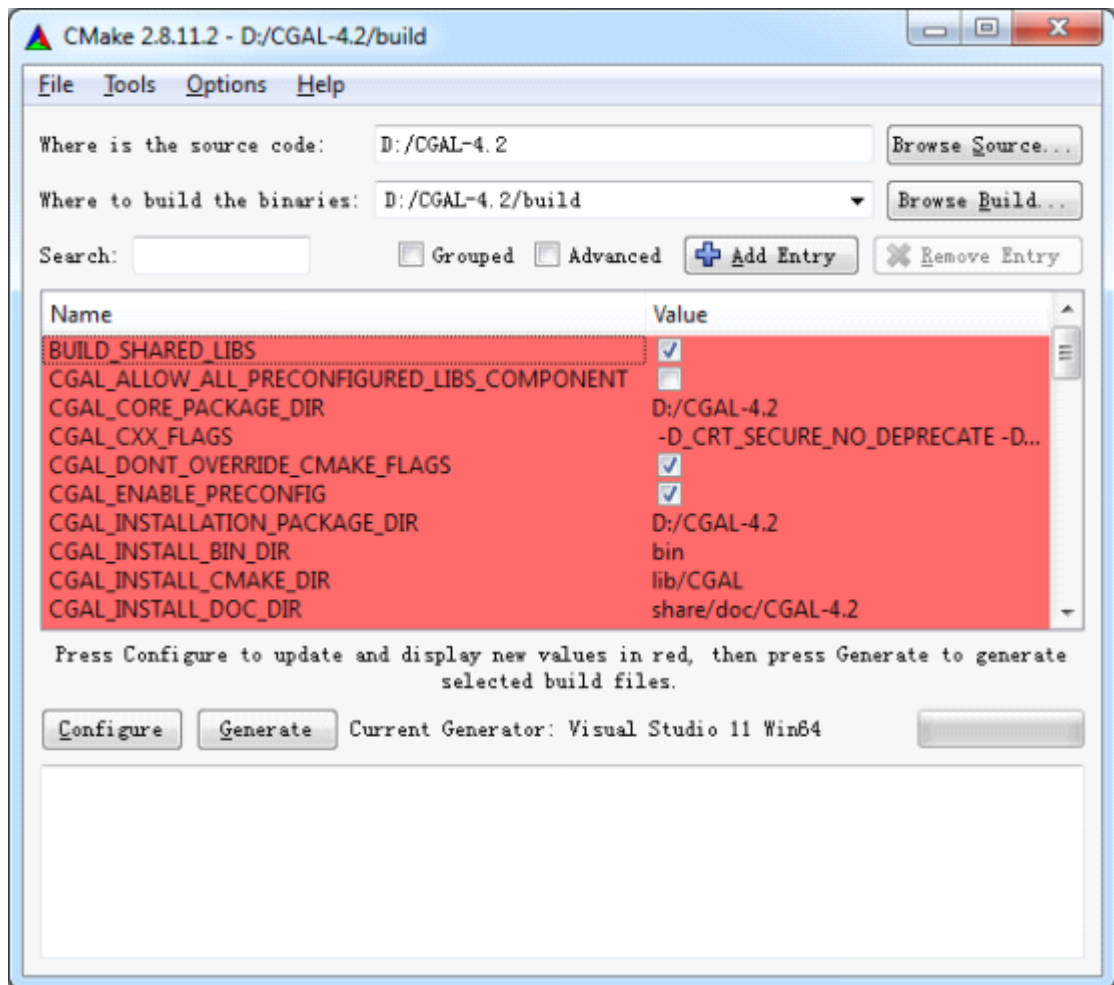
二、安装 Installation

1. 编译器 Compiler: Visual Studio
2. CMake: 下载 CMake: www.cmake.org
3. Boost 库: 根据机器选择安装 32/64 位的 Boost 库, <http://boost.teeks99.com>.
4. Qt: 只有要运行 demo 程序时才需要, 根据需要安装, 最好安装一下;
5. libQGLViewer: 只有要运行 3D CGAL 的 demo 程序时才需要, 最好安装一下;
6. CGAL 安装包: Download CGAL installer (http://gforge.inria.fr/frs/?group_id=52).

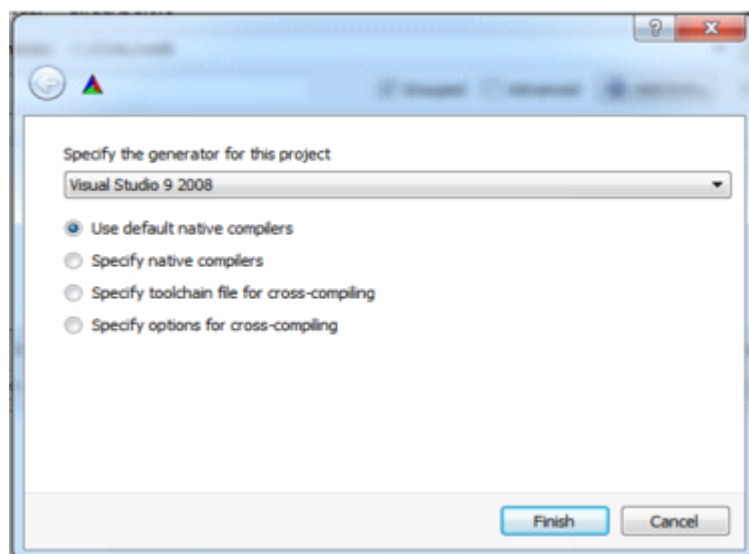
运行 CGAL 安装包, 按提示安装。因为 CGAL 需要 GMP 和 MPFR 第三方库, 主要是用来处理多精度的整数、有理数 (multi precision integers and rational numbers, and for multi precision floating point numbers)。由于 Visual C++ 对 GMP 和 MPFR 的工程支持不是很好, CGAL 提供编译好的 GMP 和 MPFR, 由安装包在安装时从网上下载。所以在安装过程中如果需要下载 GMP 和 MPFR, 一定要下载。

三、编译 Compiling

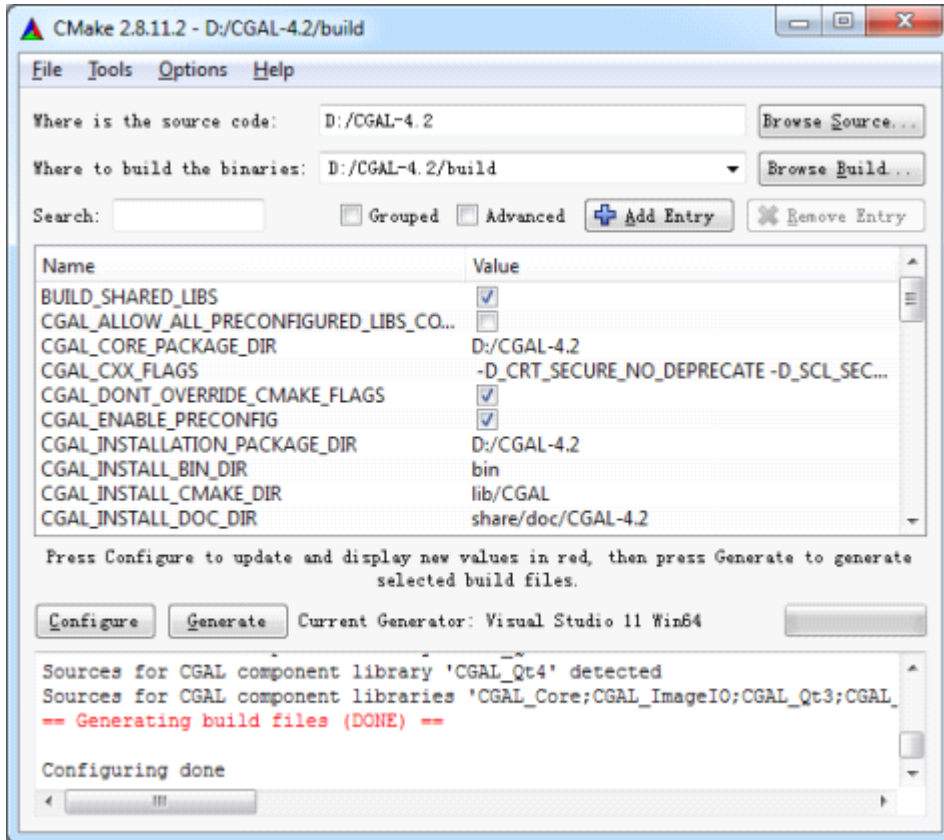
先用 CMake 生成 Visual Studio 的工程文件。打开 CMake，设置 source code 和 build the binaries 的目录，如下图所示：



点击 Configure，选择所需要的编译器，如下图所示：

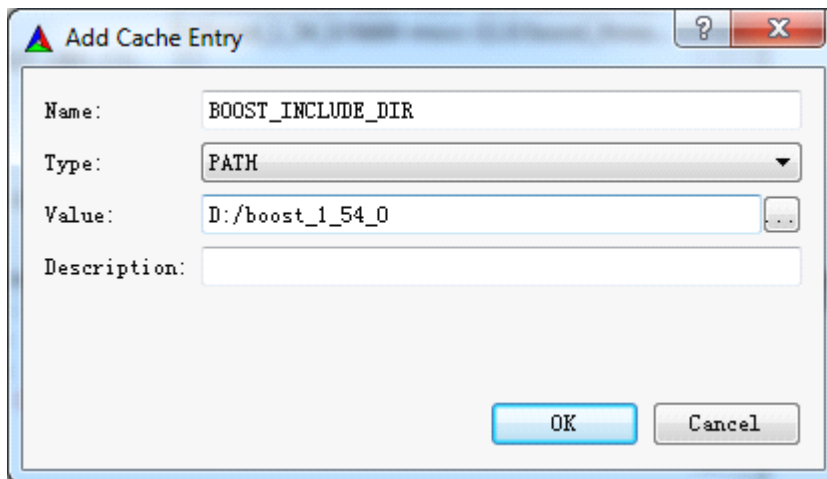


希望一切都顺利，你将会看到“Configuring done”，如下图所示：

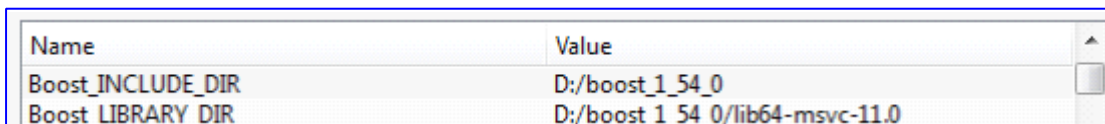


但是，也可能第一次并没有成功，可能会出现下面的错误，如找不到 Boost 库：
Can not find BOOST_INCLUDE_DIR。

这时就需要手动在 CMake 中添加一下，点 Add Entry，添加相应的路径，如下图所示：

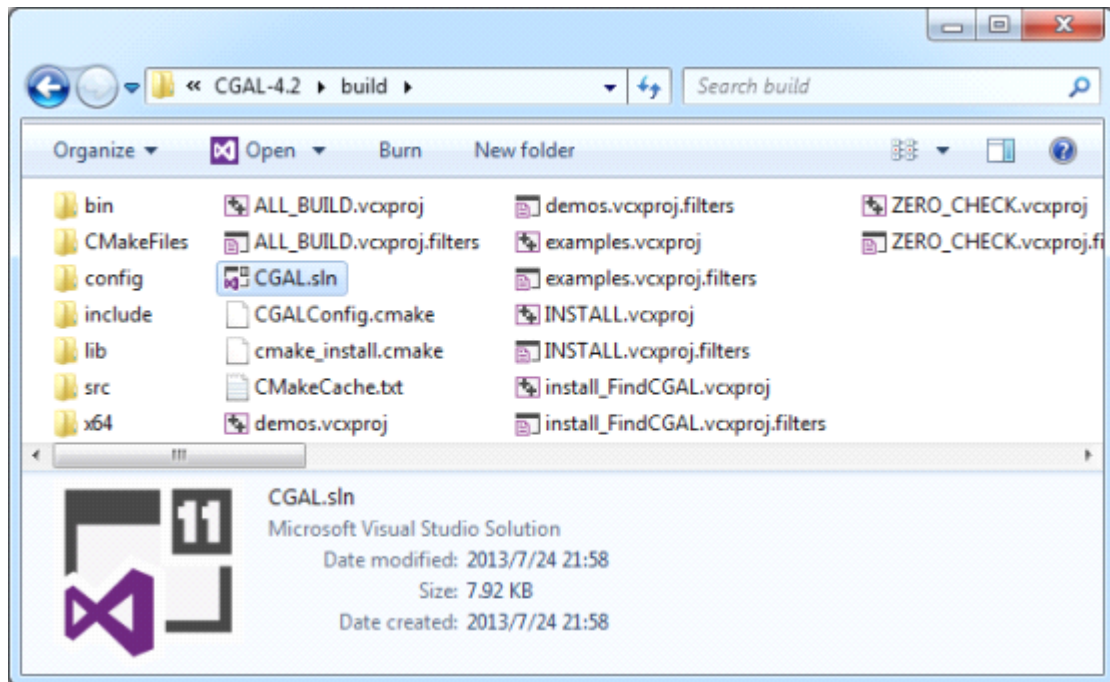


类似地，添加其它报错的 Boost 库，如下图所示：



再点“Configure”，如果“Generate”可以点击，就点击“”生成 Visual Studio 的解决方案。

生成的解决方案如下图所示：



打开解决方案，编译整个解决方案即可。

Congratulations!