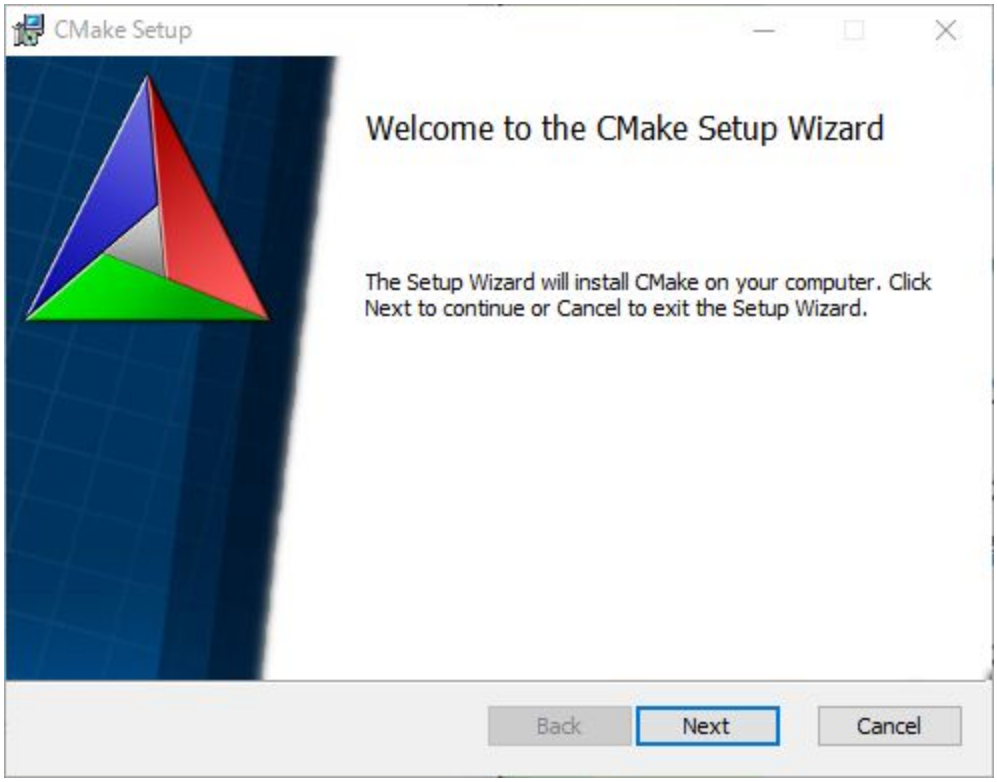
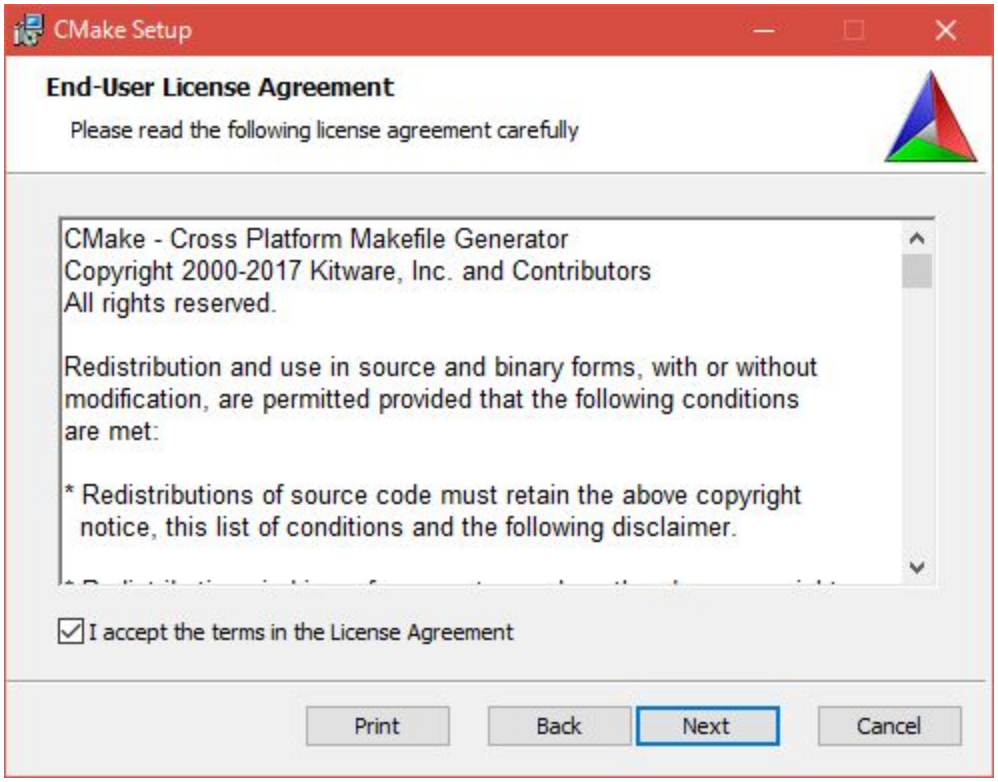
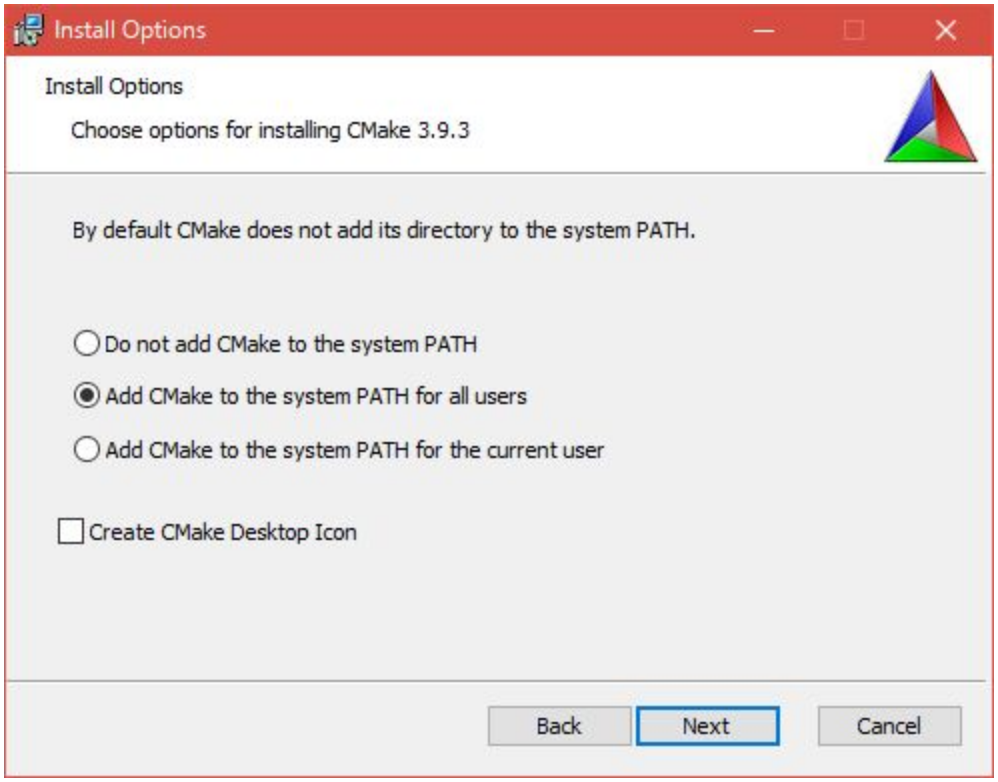
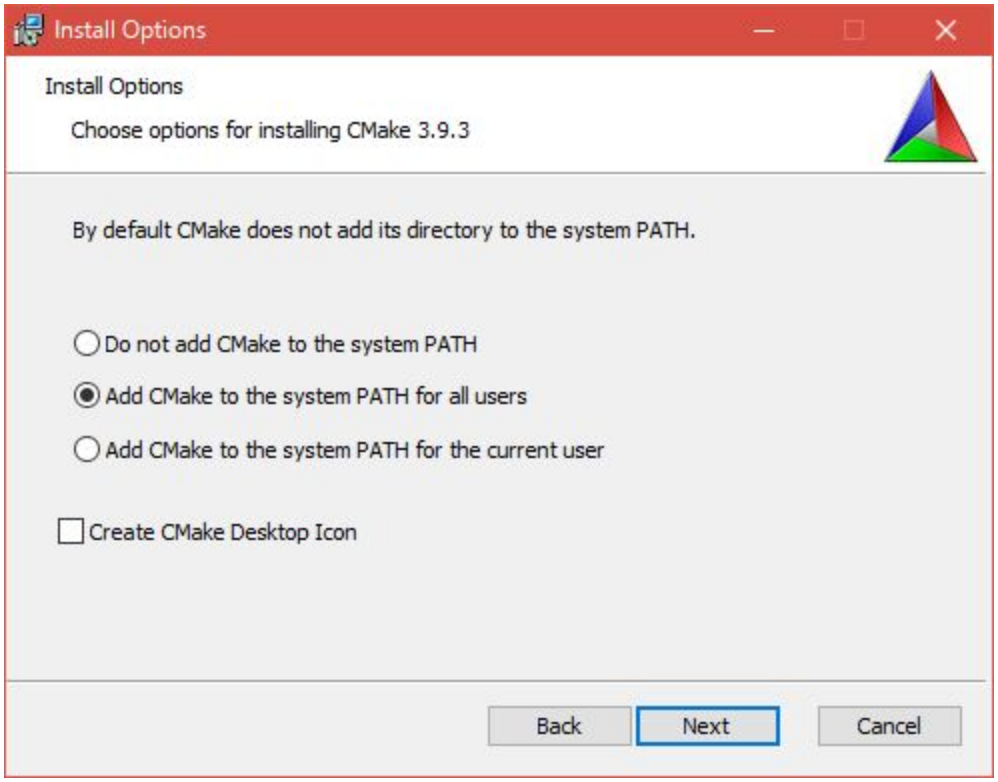
1.CMake installation

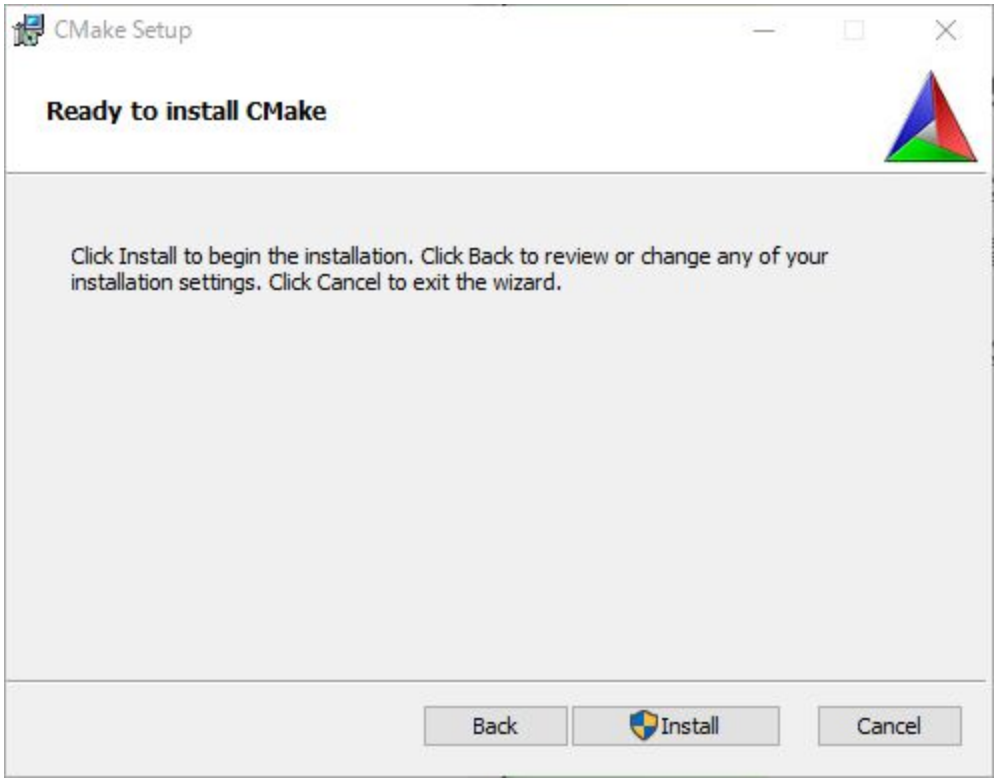
https://cmake.org/downCQCload/
Download from the above URL.
This time, install the latest version cmake-3.9.3-win64-x64.msi. Click Next Click Agree Add CMake to the system Select PATH for all users Click Next Click Next Click Install Click Finish Click Finish

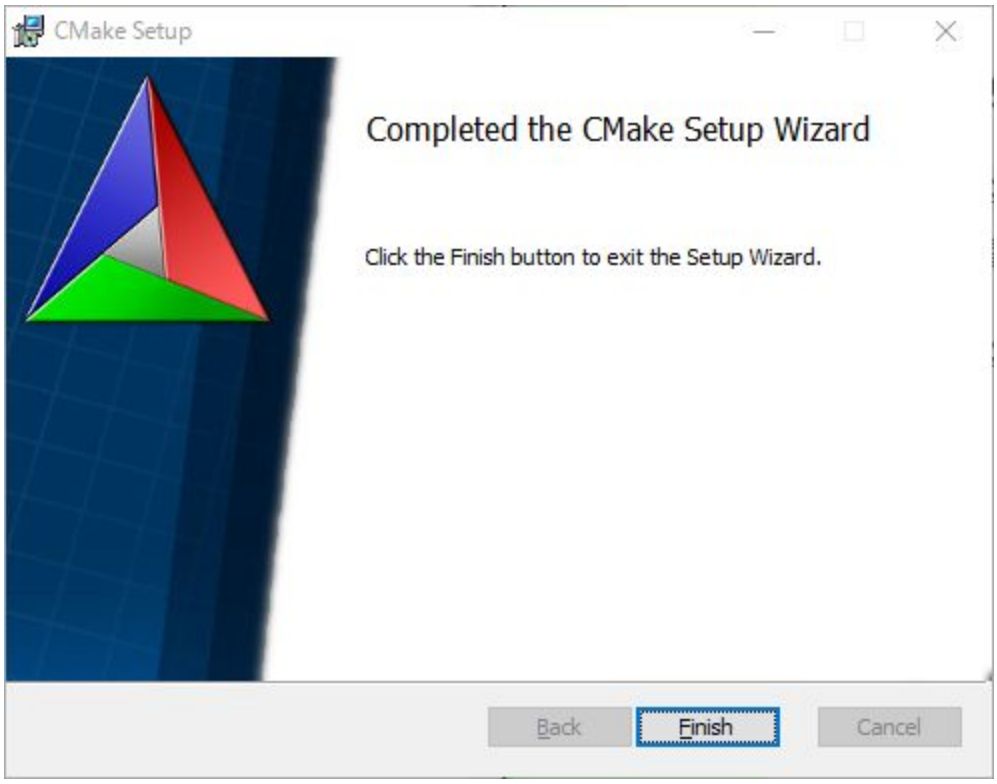



CMake Setup
End-User License Agreement
Please read the following license agreement carefully
CMake - Cross Platform Makefile Generator
Copyright 2000-2017 Kitware, Inc. and Contributors
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
I accept the terms in the License Agreement
Print
Back
Next
Cancel


Install Options
Install Options
Choose options for installing CMake 3.9.3
By default CMake does not add its directory to the system PATH.
Do not add CMake to the system PATH
Add CMake to the system PATH for all users
Add CMake to the system PATH for the current user
Create CMake Desktop Icon
Back
Next
Cancel


Install Options
Install Options
Choose options for installing CMake 3.9.3
By default CMake does not add its directory to the system PATH.
Do not add CMake to the system PATH
Add CMake to the system PATH for all users
Add CMake to the system PATH for the current user
Create CMake Desktop Icon
Back
Next
Cancel


CMake Setup
Ready to install CMake
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.
Back
Install
Cancel



2.Qt5 installation

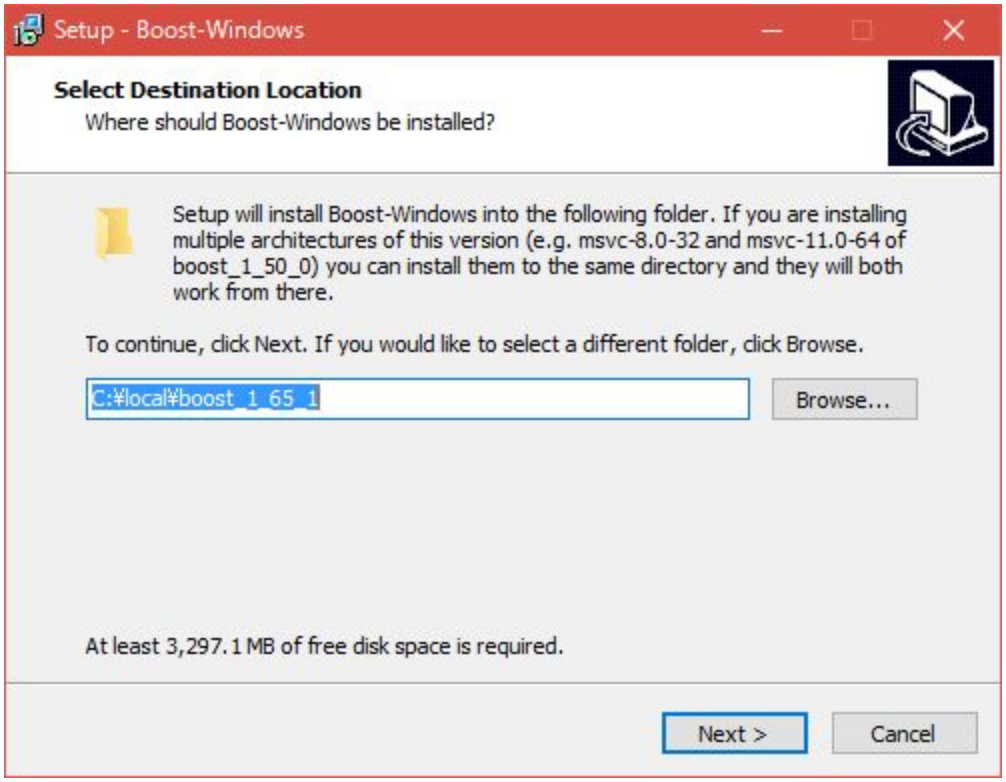
Download and execute qt-opensource-windows-x86-5.9.1.exe from https://download.qt.io/archive/qt/5.9/5.9.1/ .

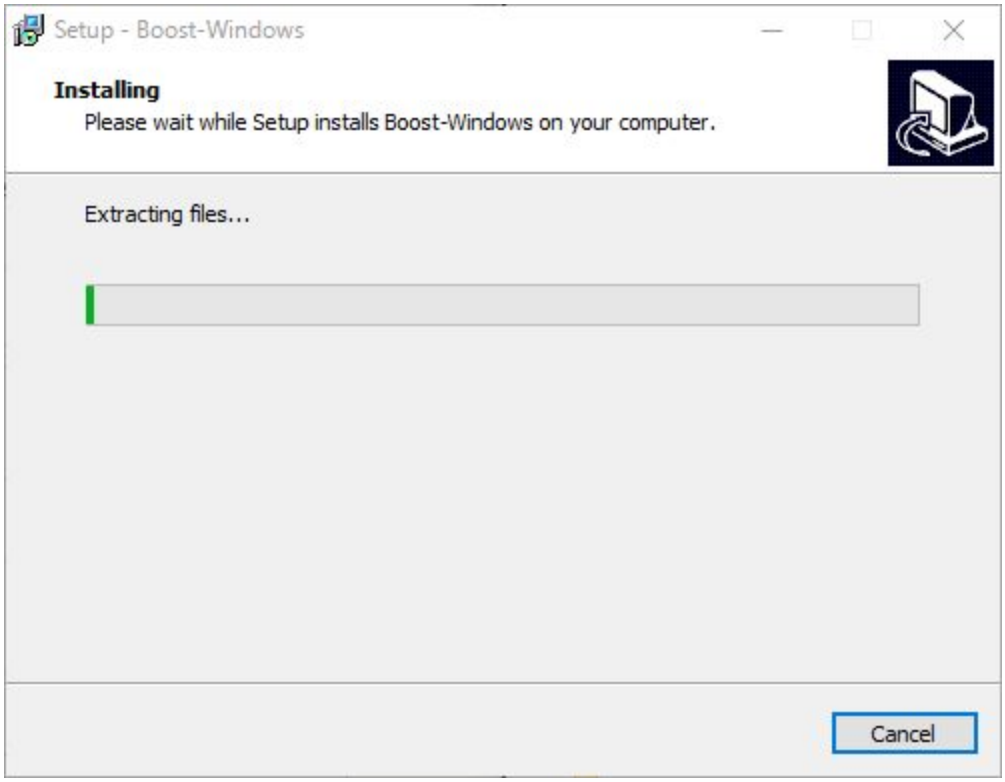
(If required to run a sample that requires Qt, installation not required if not used)

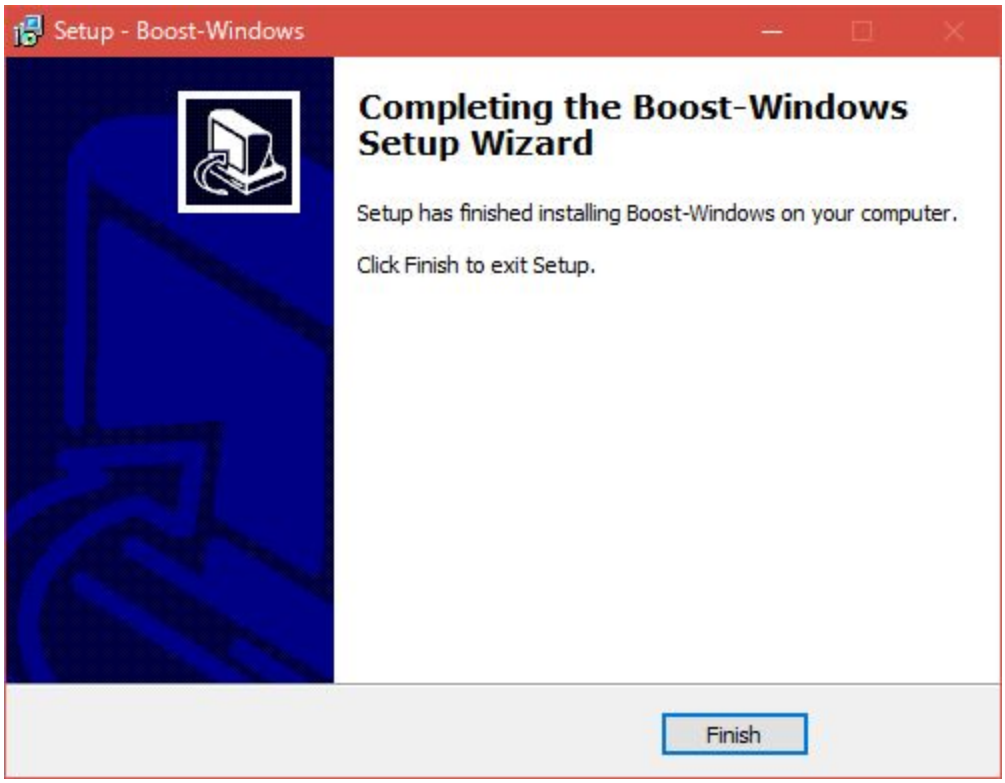
Install and configure Boost

Install

https://sourceforge.net/projects/boost/files/boost-binaries/
Download from the above URL.
This time I selected version 1.65.1.
I don't know variously, so if you think about 7z of all-msvc-32-64 for now, it is 7 hours to unzip, so
download boost_1_65_1-msvc-14.1-64.exe and execute Because it becomes, it proceeds as it is. (Next) Installation progress. Finish with Finish.

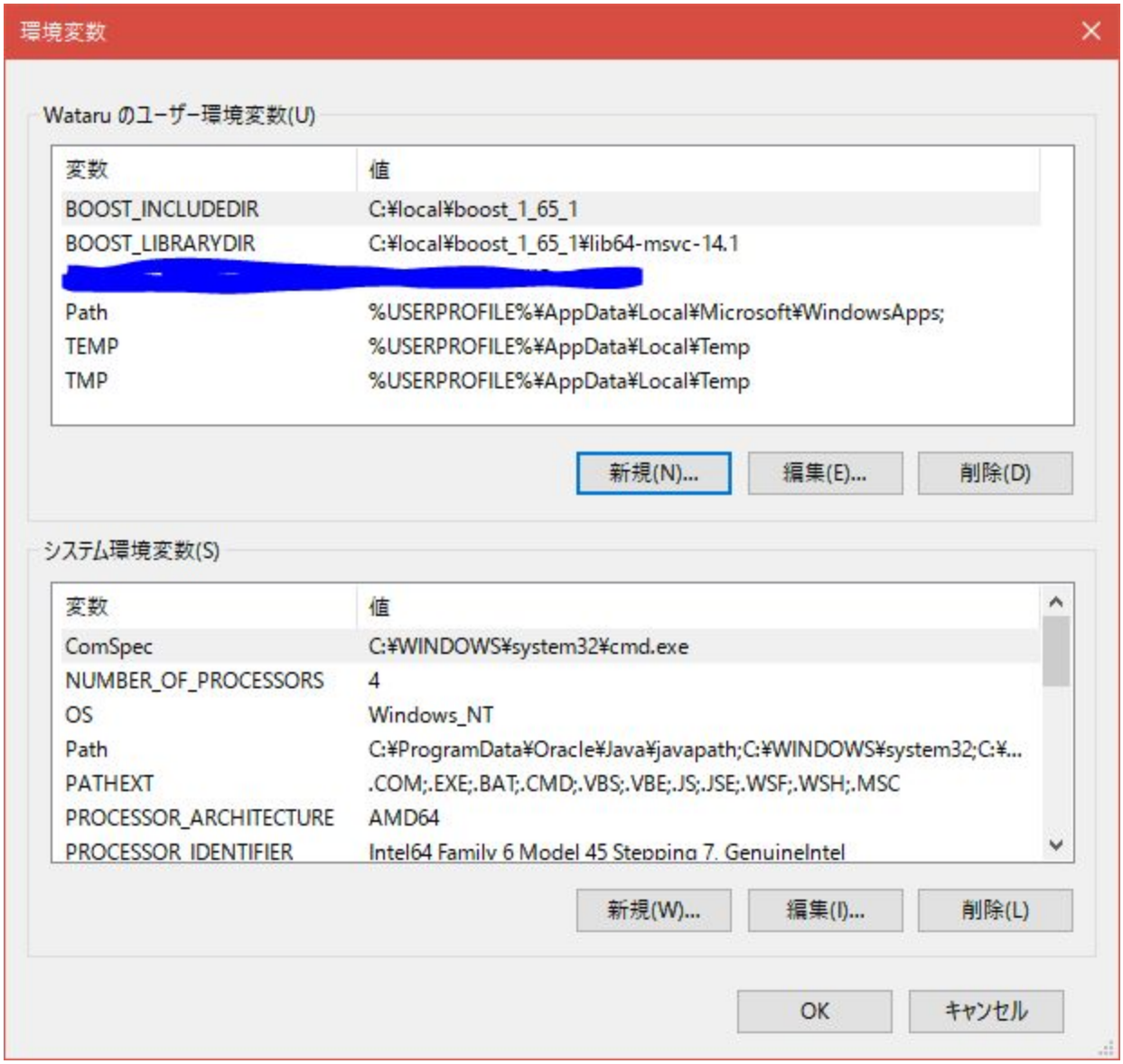



Setup - Boost-Windows
Installing
Please wait while Setup installs Boost-Windows on your computer.
Extracting files...
Cancel


Setup - Boost-Windows
Completing the Boost-Windows Setup Wizard
Setup has finished installing Boost-Windows on your computer.
Click Finish to exit Setup.
Finish

Configuration

Set environment variables.
Right-click on Windows mark → System → Advanced system settings → Click 新建 (N) in Advanced tab New click above 2 Add two environment variables. (Example: BOOST_INCLUDEDIR) Also, add to the already existing PATH . To do this, select the variable Path and click Edit. Insert and add to the back of an already existing variable . When everything is finished, press OK to finish. Even if you make a mistake, do not delete the contents of the existing Path.



```
BOOST_LIBRARYDIR=C:\local\boost_1_65_1\lib64-msvc-14.1
BOOST_INCLUDEDIR=C:\local\boost_1_65_1
```

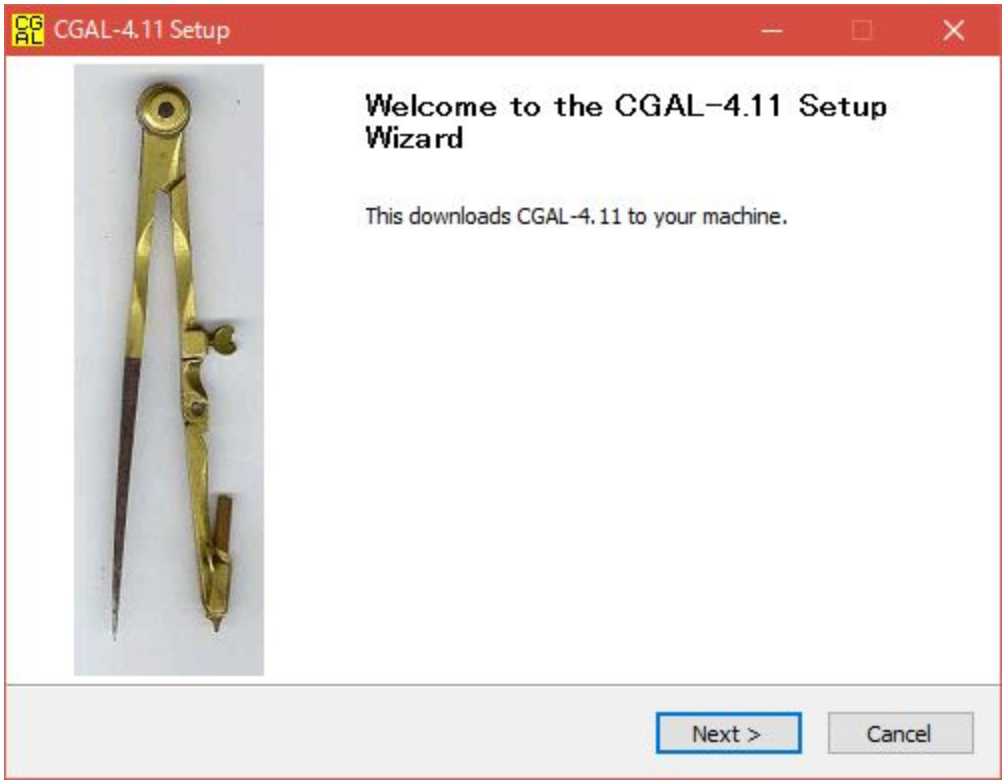

```
C:\local\boost_1_65_1\lib64-msvc-14.1
```

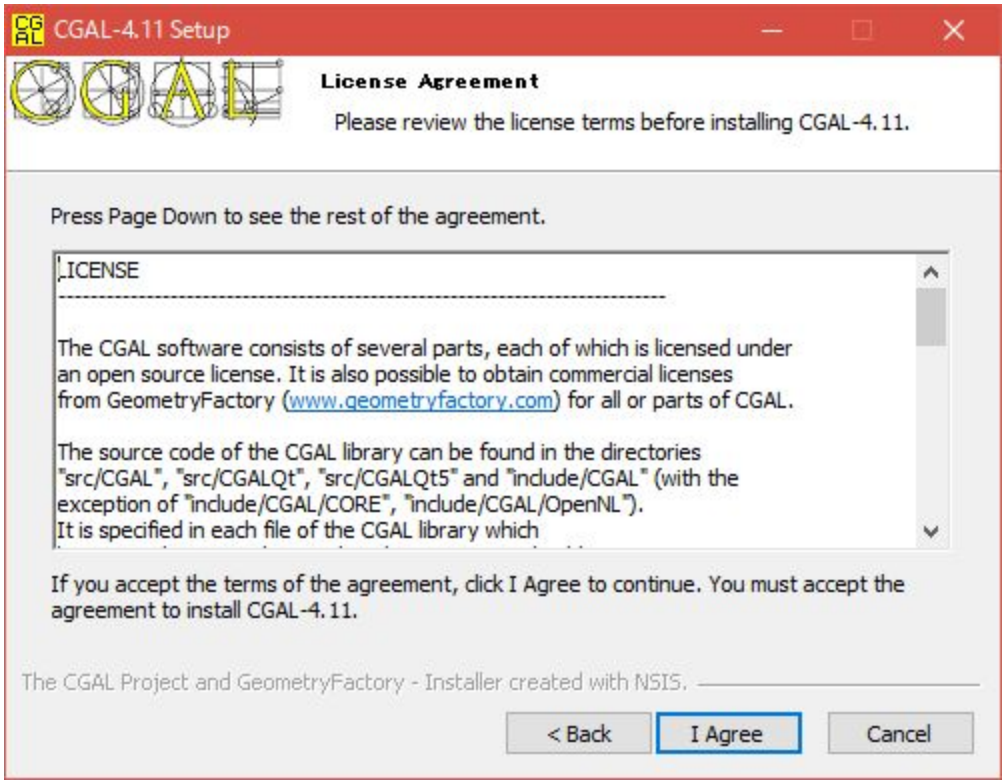
```
;
```

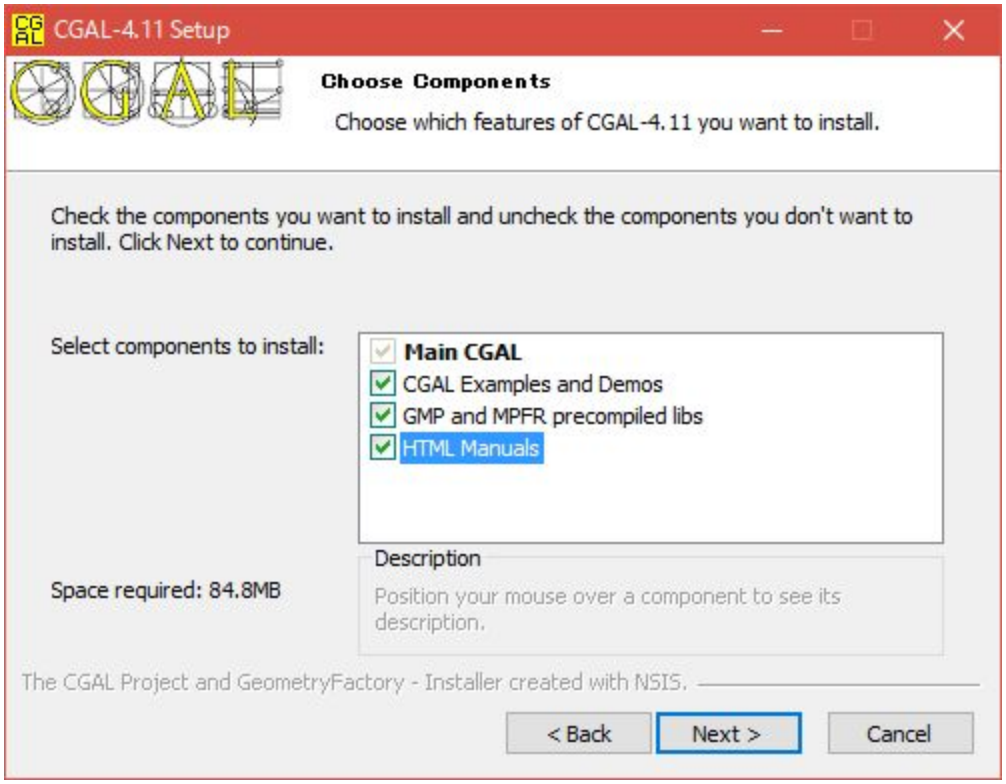
CGAL installation

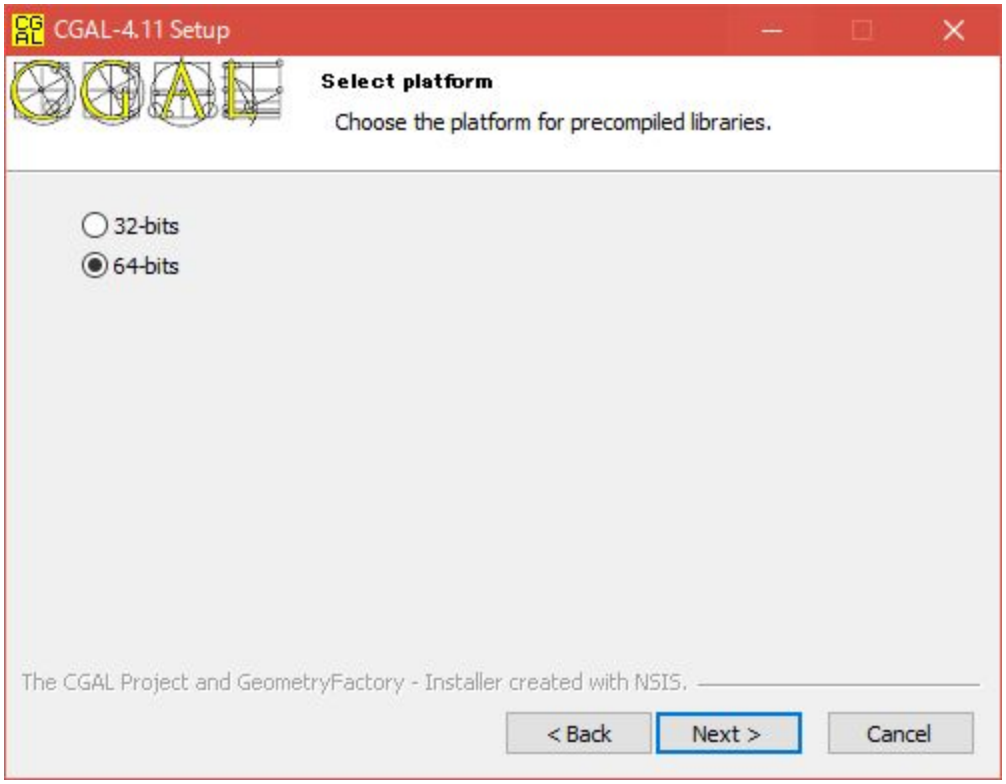
CGAL download and installation

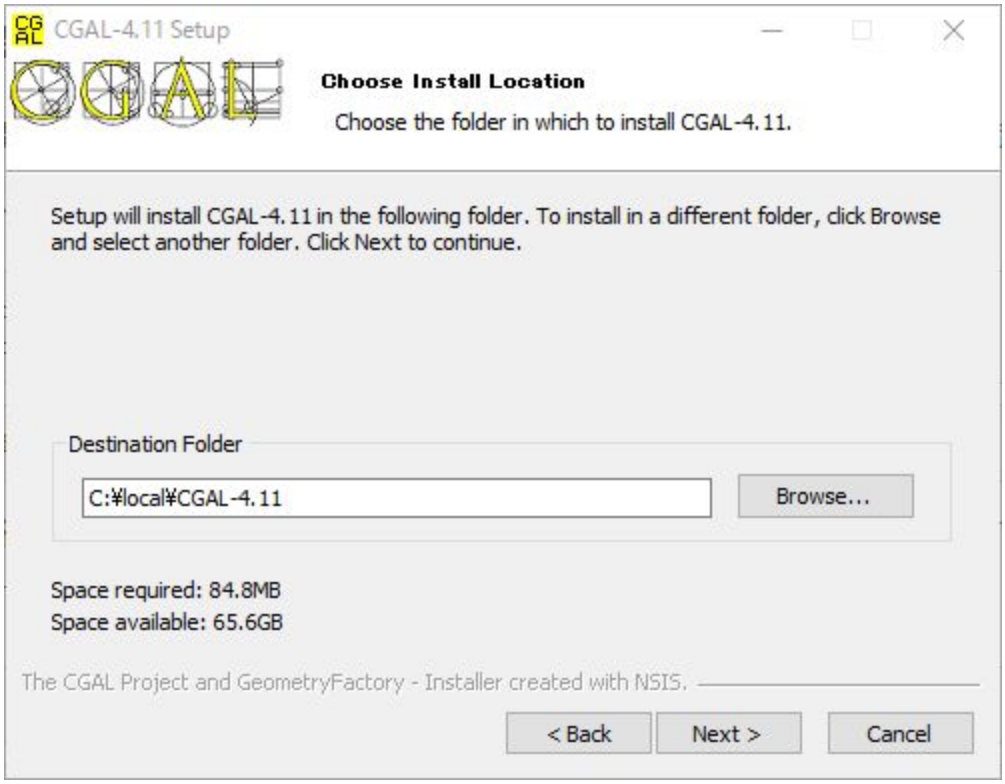
https://www.cgal.org/download.html
https://github.com/CGAL/cgal/releases
Download and execute CGAL-4.11-Setup.exe. Next Agree Personally, I also want HTML Manual, so check all and click Next. Select 64bit. Make the directory C: \ local \ the same as Boost. (Somehow) I made it all users. The others are not changed. Once complete, Next Downloading is complete, so compile with CMake. This came out. In my environment, gmp was not included in the environment variable, so I will set it later.


CGAL-4.11 Setup
Welcome to the CGAL-4.11 Setup Wizard
This downloads CGAL-4.11 to your machine.
Next >
Cancel

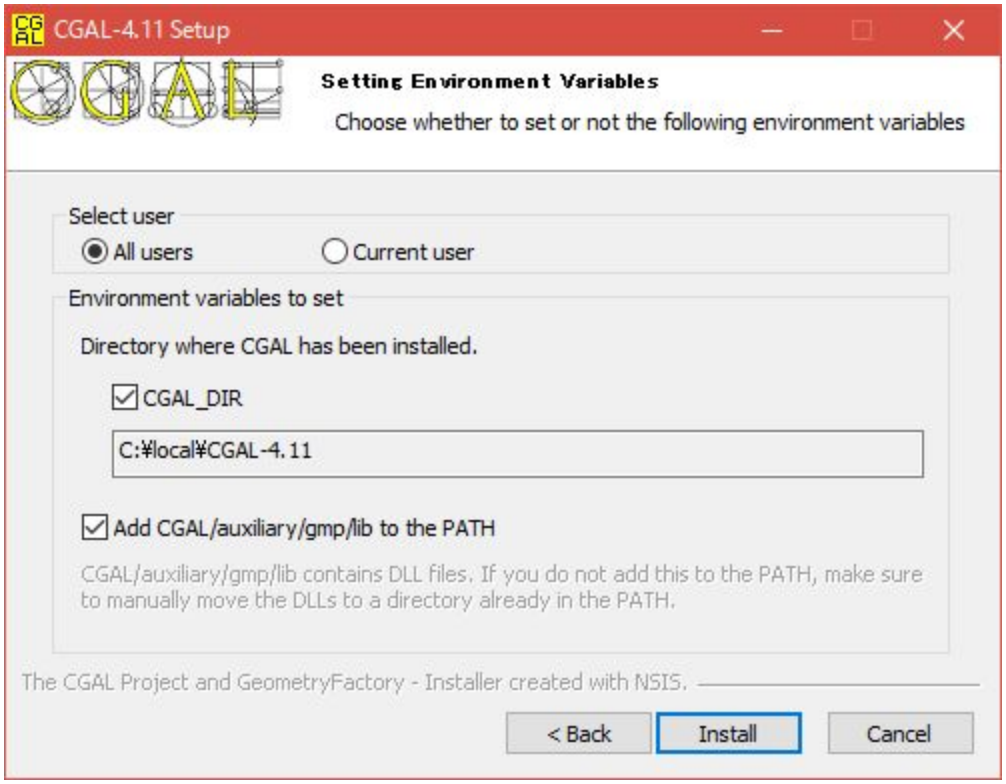

CGAL-4.11 Setup
License Agreement
Please review the license terms before installing CGAL-4.11.
Press Page Down to see the rest of the agreement.
LICENSE
The CGAL software consists of several parts, each of which is licensed under an open source license. It is also possible to obtain commercial licenses from GeometryFactory (www.geometryfactory.com) for all or parts of CGAL.
The source code of the CGAL library can be found in the directories "src/CGAL", "src/CGALQt", "src/CGALQt5" and "include/CGAL" (with the exception of "include/CGAL/CORE", "include/CGAL/OpenNL").
It is specified in each file of the CGAL library which
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install CGAL-4.11.
The CGAL Project and GeometryFactory - Installer created with NSIS.
< Back
I Agree
Cancel

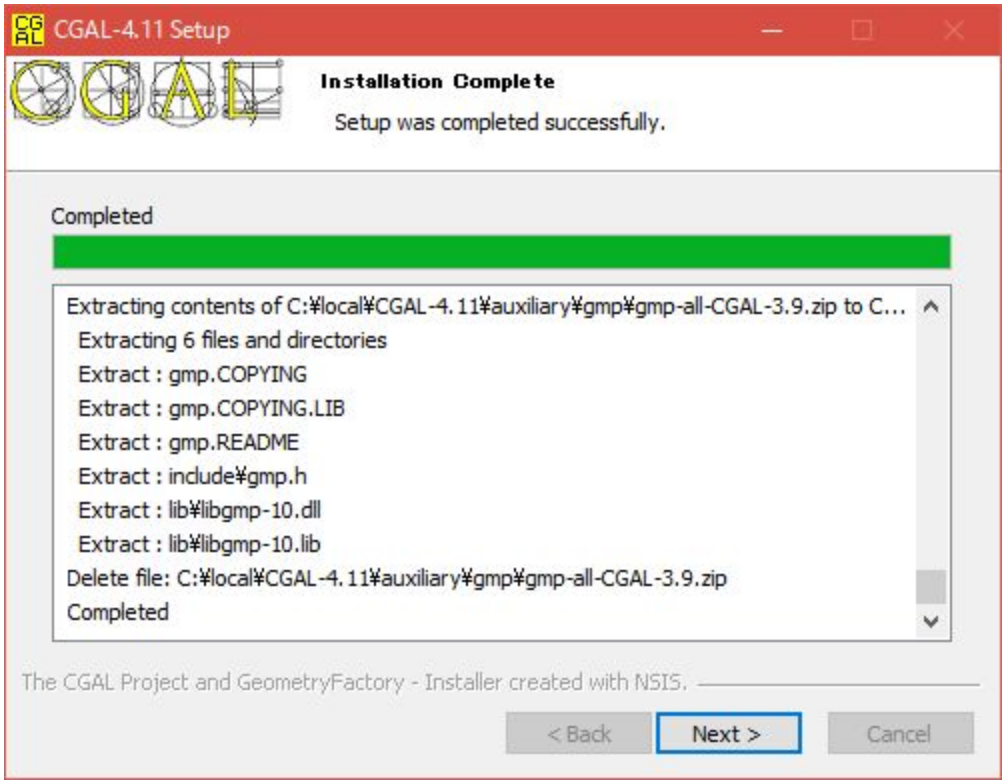

CGAL-4.11 Setup
Choose Components
Choose which features of CGAL-4.11 you want to install.
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.
Select components to install:
Main CGAL
CGAL Examples and Demos
GMP and MPFR precompiled libs
HTML Manuals
Description
Position your mouse over a component to see its description.
Space required: 84.8MB
The CGAL Project and GeometryFactory - Installer created with NSIS.
< Back
Next >
Cancel

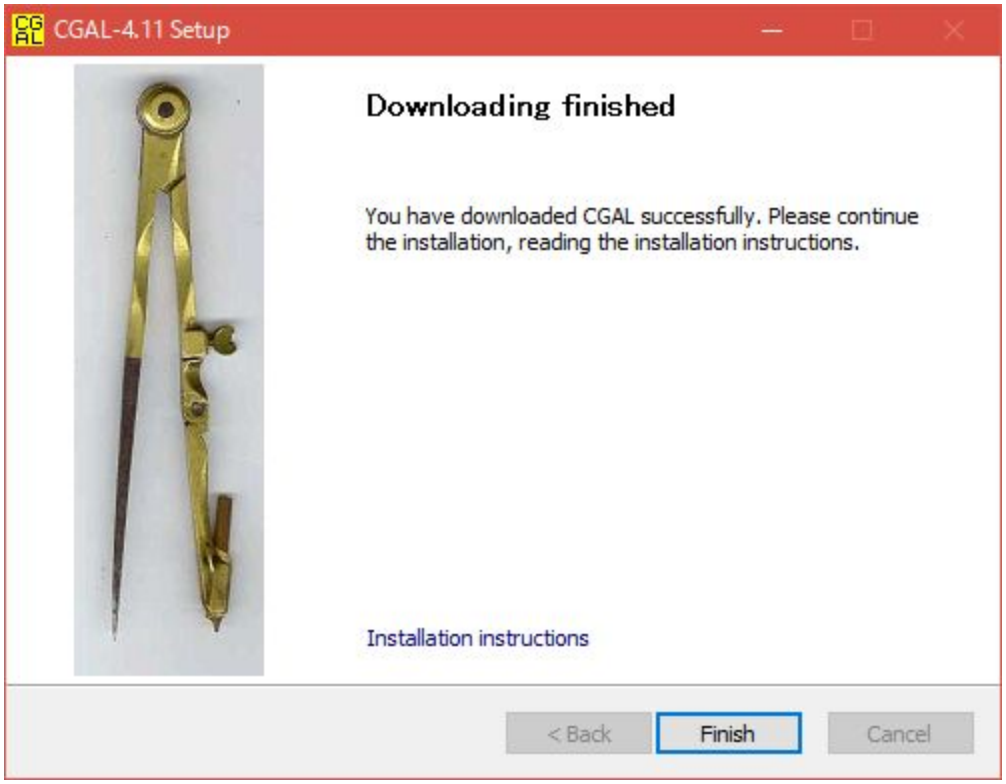

CGAL-4.11 Setup
Select platform
Choose the platform for precompiled libraries.
32-bits
64-bits
The CGAL Project and GeometryFactory - Installer created with NSIS.
< Back
Next >
Cancel



`C:\local\CGAL-4.11`


CGAL-4.11 Setup
Setting Environment Variables
Choose whether to set or not the following environment variables
Select user
All users
Current user
Environment variables to set
Directory where CGAL has been installed.
CGAL_DIR
C:¥local¥CGAL-4.11
Add CGAL/auxiliary/gmp/lib to the PATH
CGAL/auxiliary/gmp/lib contains DLL files. If you do not add this to the PATH, make sure to manually move the DLLs to a directory already in the PATH.
The CGAL Project and GeometryFactory - Installer created with NSIS.
< Back
Install
Cancel


CGAL-4.11 Setup
Installation Complete
Setup was completed successfully.
Completed
Extracting contents of C:¥local¥CGAL-4.11¥auxiliary¥gmp¥gmp-all-CGAL-3.9.zip to C...
Extracting 6 files and directories
Extract : gmp.COPYING
Extract : gmp.COPYING.LIB
Extract : gmp.README
Extract : include¥gmp.h
Extract : lib¥libgmp-10.dll
Extract : lib¥libgmp-10.lib
Delete file: C:¥local¥CGAL-4.11¥auxiliary¥gmp¥gmp-all-CGAL-3.9.zip
Completed
The CGAL Project and GeometryFactory - Installer created with NSIS.
< Back
Next >
Cancel

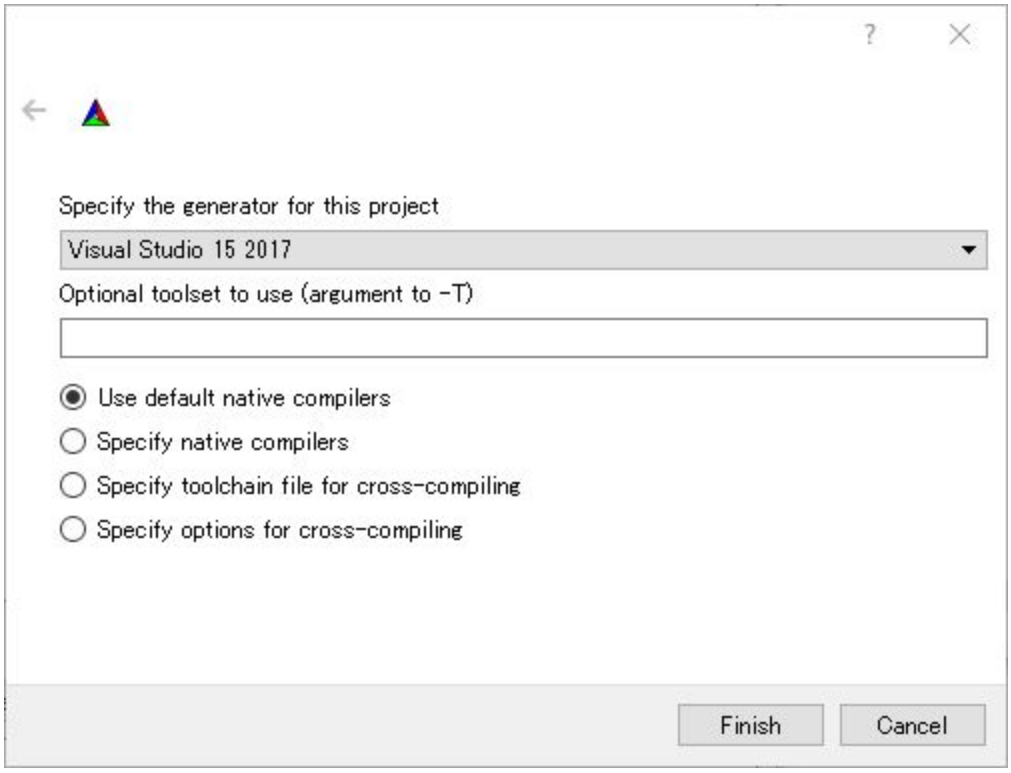


Configuration

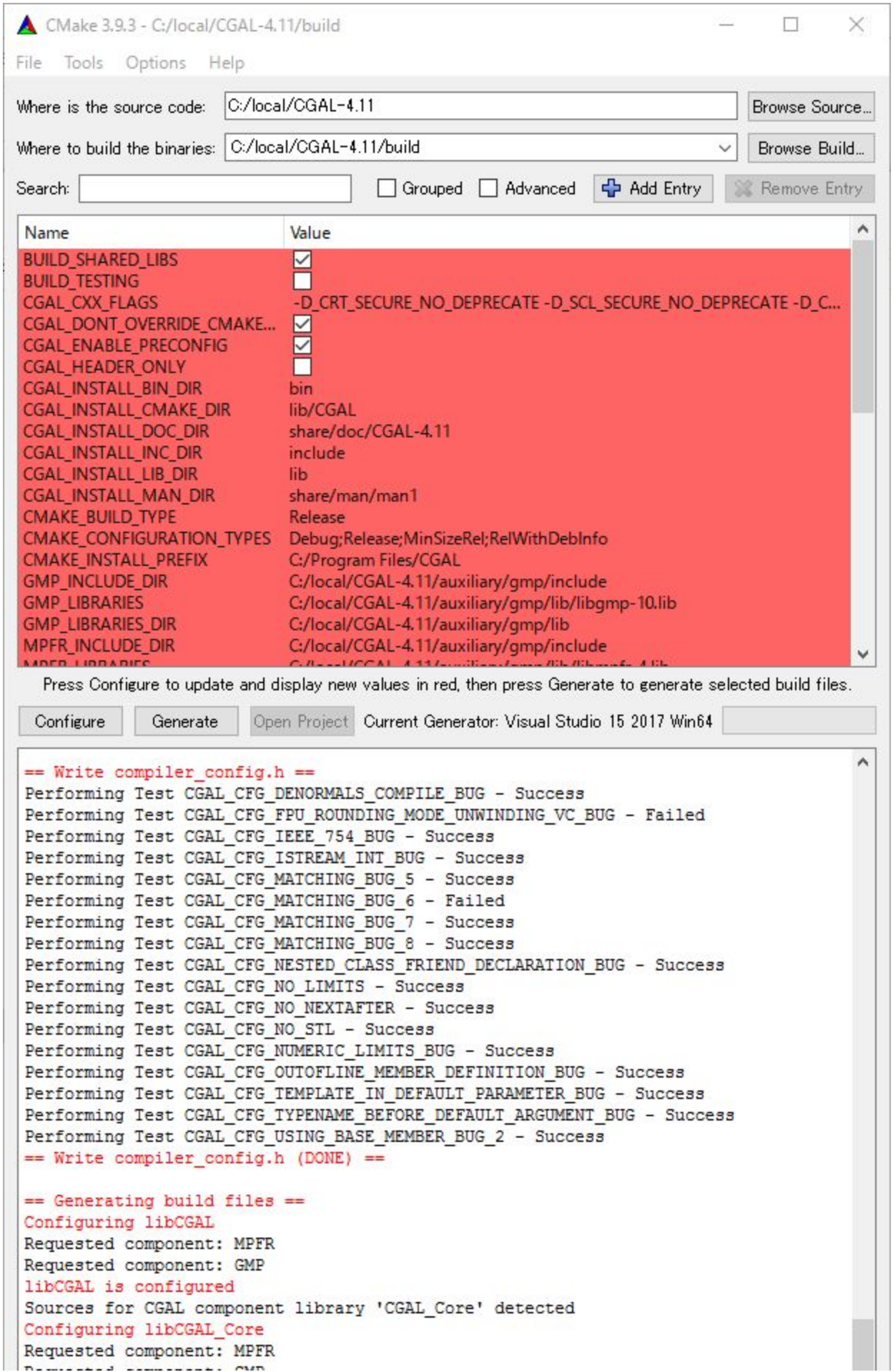
Let Source code be and Build the binaries be. Press Configure.

`C:/local/CGAL-4.11`

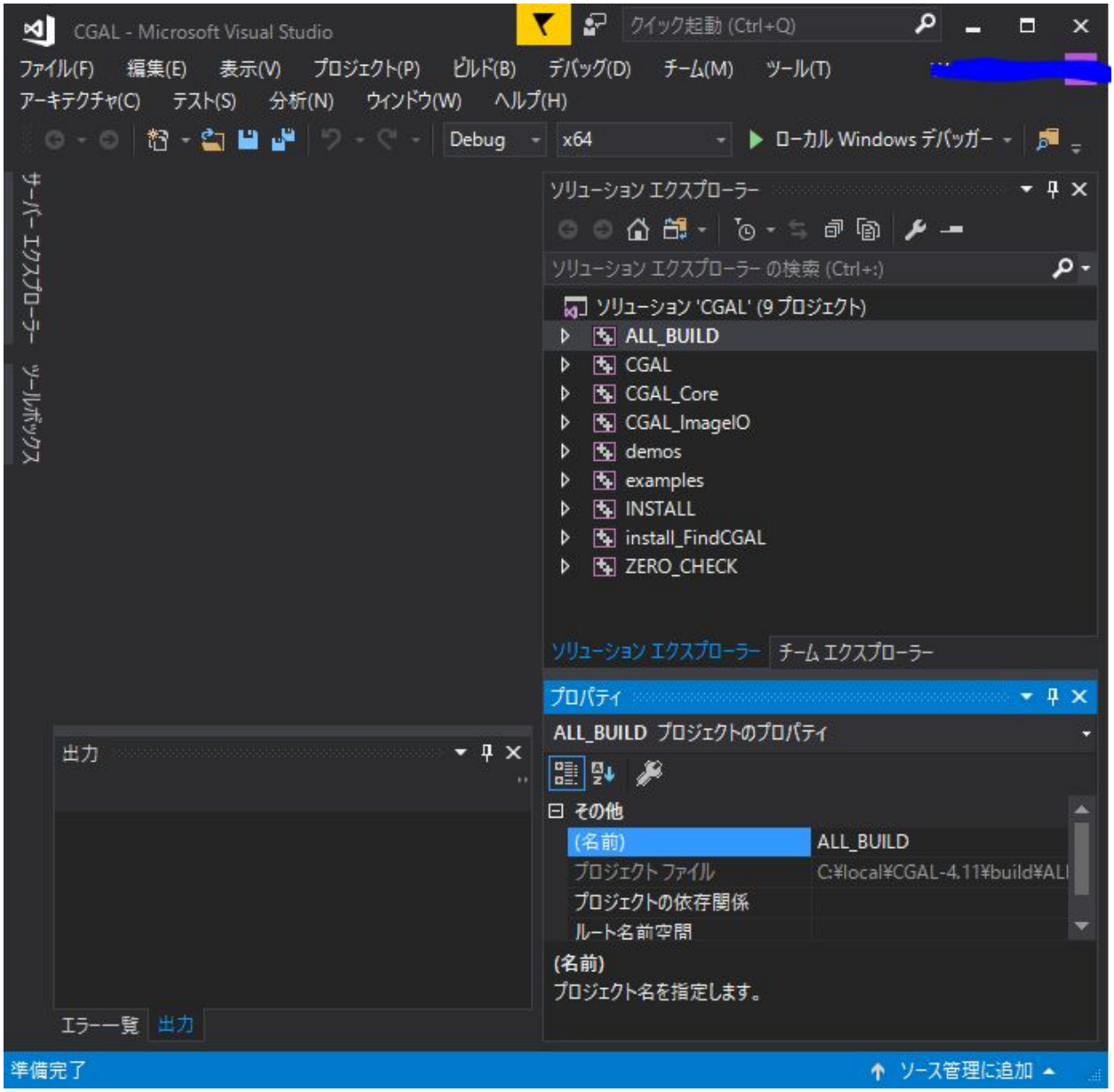
`C:/local/CGAL-4.11/build`



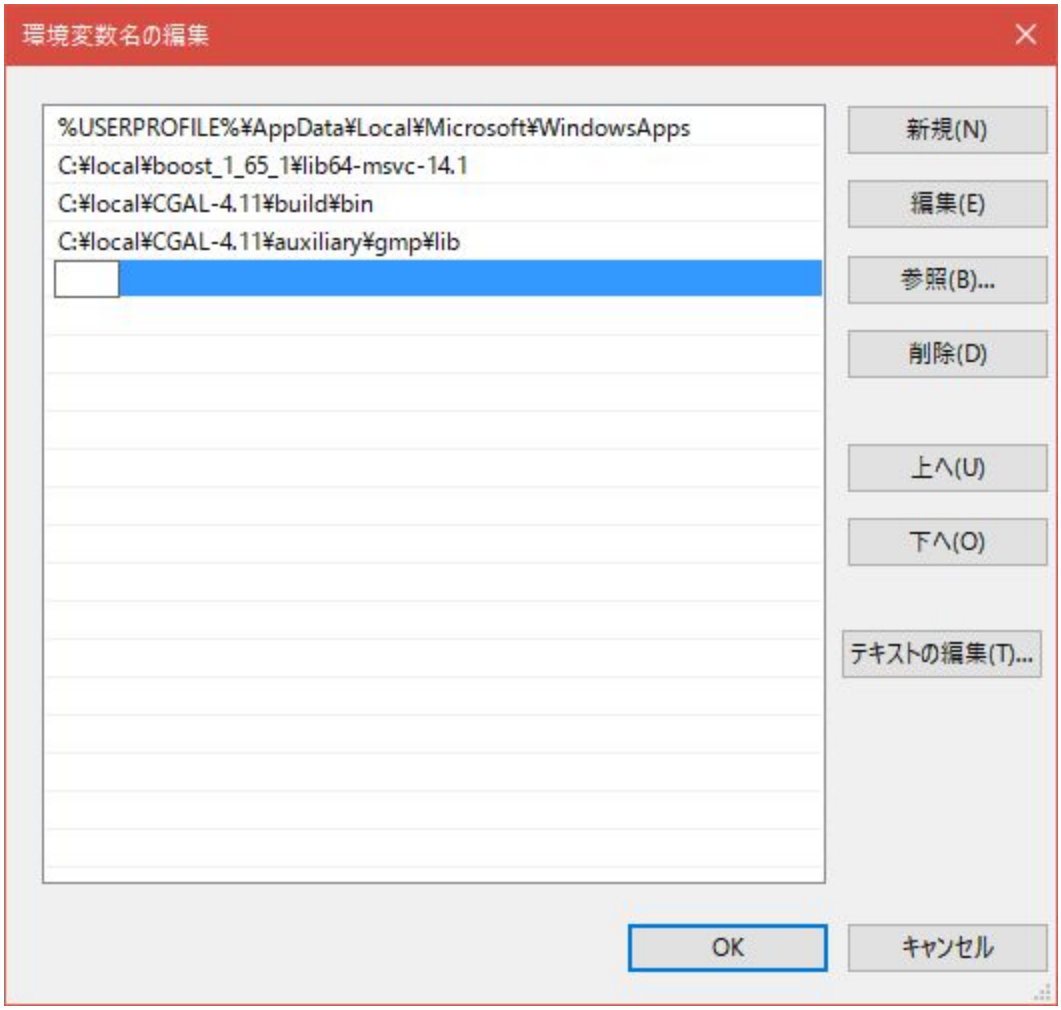
Here, select Visual studio 15 2017 Win64 and click Finish.


CMake 3.9.3 - C:/local/CGAL-4.11/build
File Tools Options Help
Where is the source code: C:/local/CGAL-4.11
Browse Source...
Where to build the binaries: C:/local/CGAL-4.11/build
Browse Build...
Search:
Grouped
Advanced
Add Entry
Remove Entry
Name
Value
BUILD_SHARED_LIBS
BUILD_TESTING
CGAL_CXX_FLAGS
-D_CRT_SECURE_NO_DEPRECATE -D_SCL_SECURE_NO_DEPRECATE -D_C...
CGAL_DONT_OVERRIDE_CMAKE...
CGAL_ENABLE_PRECONFIG
CGAL_HEADER_ONLY
CGAL_INSTALL_BIN_DIR
bin
CGAL_INSTALL_CMAKE_DIR
lib/CGAL
CGAL_INSTALL_DOC_DIR
share/doc/CGAL-4.11
CGAL_INSTALL_INC_DIR
include
CGAL_INSTALL_LIB_DIR
lib
CGAL_INSTALL_MAN_DIR
share/man/man1
CMAKE_BUILD_TYPE
Release
CMAKE_CONFIGURATION_TYPES
Debug;Release;MinSizeRel;RelWithDebInfo
CMAKE_INSTALL_PREFIX
C:/Program Files/CGAL
GMP_INCLUDE_DIR
C:/local/CGAL-4.11/auxiliary/gmp/include
GMP_LIBRARIES
C:/local/CGAL-4.11/auxiliary/gmp/lib/libgmp-10.lib
GMP_LIBRARIES_DIR
C:/local/CGAL-4.11/auxiliary/gmp/lib
MPFR_INCLUDE_DIR
C:/local/CGAL-4.11/auxiliary/gmp/include
Press Configure to update and display new values in red, then press Generate to generate selected build files.
Configure
Generate
Open Project
Current Generator: Visual Studio 15 2017 Win64
== Write compiler_config.h ==
Performing Test CGAL_CFG_DENORMALS_COMPILE_BUG - Success
Performing Test CGAL_CFG_FPU_ROUNDING_MODE_UNWINDING_VC_BUG - Failed
Performing Test CGAL_CFG_IEEE_754_BUG - Success
Performing Test CGAL_CFG_ISTREAM_INT_BUG - Success
Performing Test CGAL_CFG_MATCHING_BUG_5 - Success
Performing Test CGAL_CFG_MATCHING_BUG_6 - Failed
Performing Test CGAL_CFG_MATCHING_BUG_7 - Success
Performing Test CGAL_CFG_MATCHING_BUG_8 - Success
Performing Test CGAL_CFG_NESTED_CLASS_FRIEND_DECLARATION_BUG - Success
Performing Test CGAL_CFG_NO_LIMITS - Success
Performing Test CGAL_CFG_NO_NEXTAFTER - Success
Performing Test CGAL_CFG_NO_STL - Success
Performing Test CGAL_CFG_NUMERIC_LIMITS_BUG - Success
Performing Test CGAL_CFG_OUTOFLINE_MEMBER_DEFINITION_BUG - Success
Performing Test CGAL_CFG_TEMPLATE_IN_DEFAULT_PARAMETER_BUG - Success
Performing Test CGAL_CFG_TYPENAME_BEFORE_DEFAULT_ARGUMENT_BUG - Success
Performing Test CGAL_CFG_USING_BASE_MEMBER_BUG_2 - Success
== Write compiler_config.h (DONE) ==
== Generating build files ==
Configuring libCGAL
Requested component: MPFR
Requested component: GMP
libCGAL is configured
Sources for CGAL component library 'CGAL_Core' detected
Configuring libCGAL_Core
Requested component: MPFR

t turns red, but it is not an error because it is DONE. Click Generate.



Since CGAL.sln appears in C: \ local \ CGAL-4.11 \ build, it opens with VS2017. Among these, build ALL_BUILD only. Output such as (Rebuild only the project) is output.



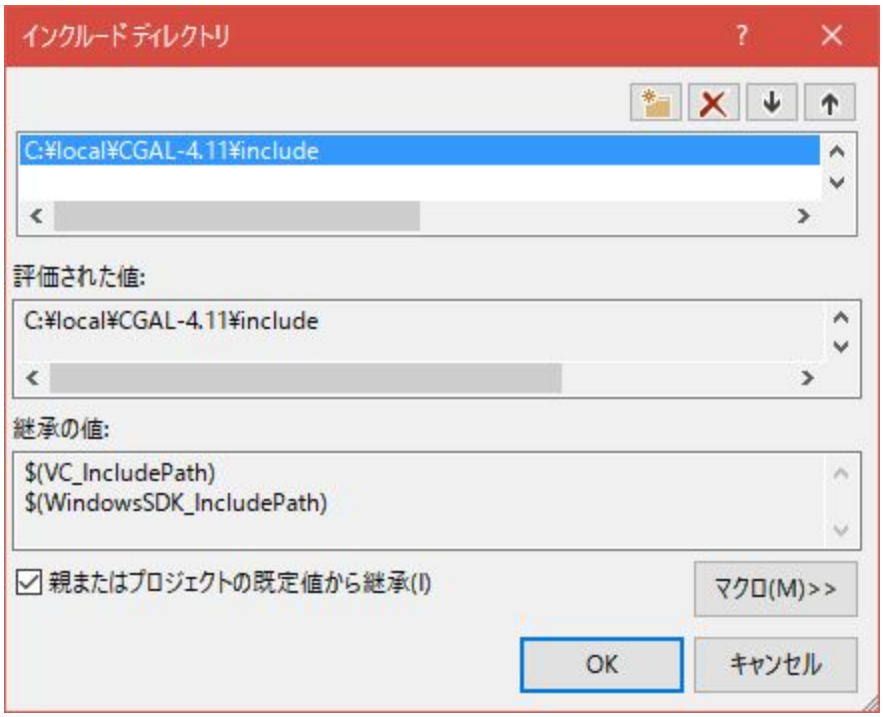
When this is complete, add "C: \ local \ CGAL-4.11 \ build \ bin" and "C: \ local \ CGAL-4.11 \ auxiliary \ gmp \ lib" to environment variable "Path". Move to the environment variable setting screen in the same way as before, and click Edit for Path. When you click New, the input field blinks, so add the above two environment variables to it. ※ Do not delete variables that already exist! !

Test building

Configuration

Create new → Project → win32 Console application

※ Make an empty project. Not precompiled



From the project settings , add to the include directory . Add to library directory . Add to the dependent library directory . These make it possible to refer to and execute all functions without intellisense errors.

C:\local\CGAL\auxiliary\gmp\include
C:\local\boost_1_64_0
C:\local\CGAL\build\include
C:\local\CGAL\include

C:\local\CGAL\auxiliary\gmp\lib
C:\local\boost_1_64_0\libs
C:\local\CGAL\build\lib

```
libgmp-10.lib
libmpfr-4.lib
kernel32.lib
user32.lib
```

```
CGAL_Core-vc140-mt-gd-4.13.1.lib
CGAL_ImageIO-vc140-mt-gd-4.13.1.lib
CGAL-vc140-mt-gd-4.13.1.lib
```