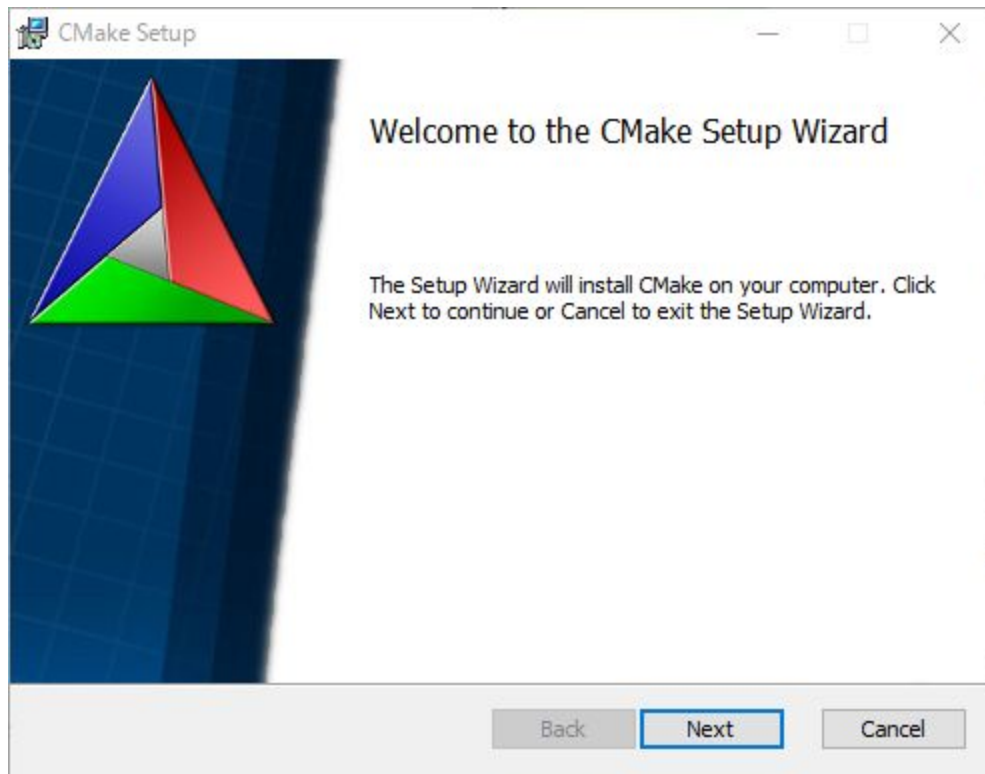


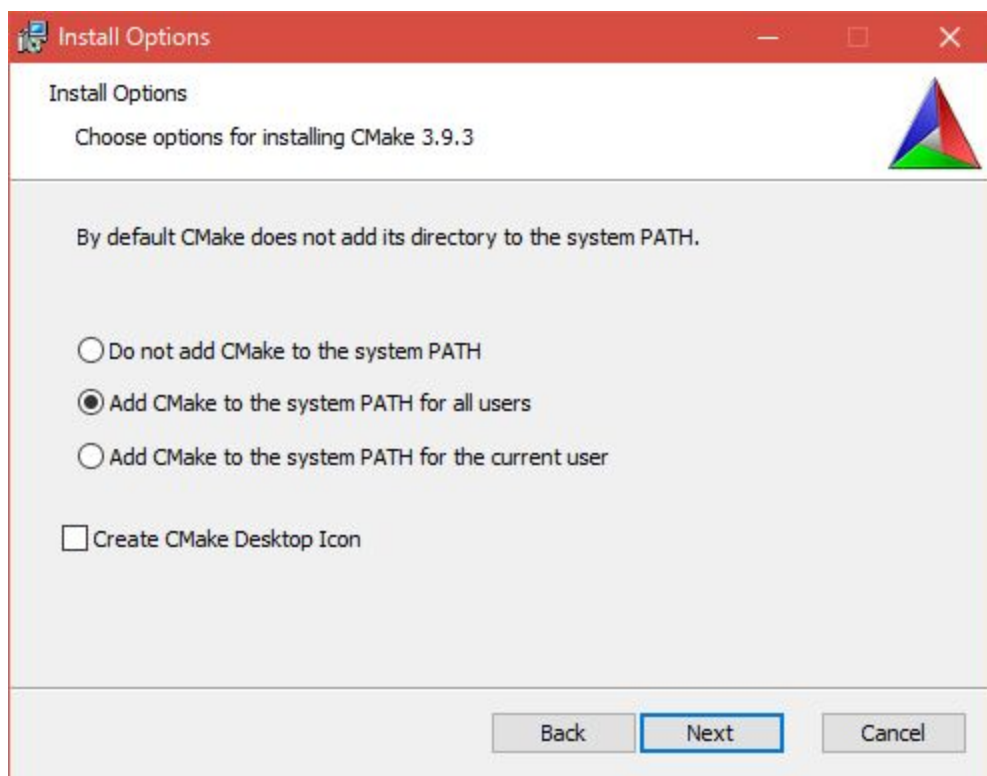
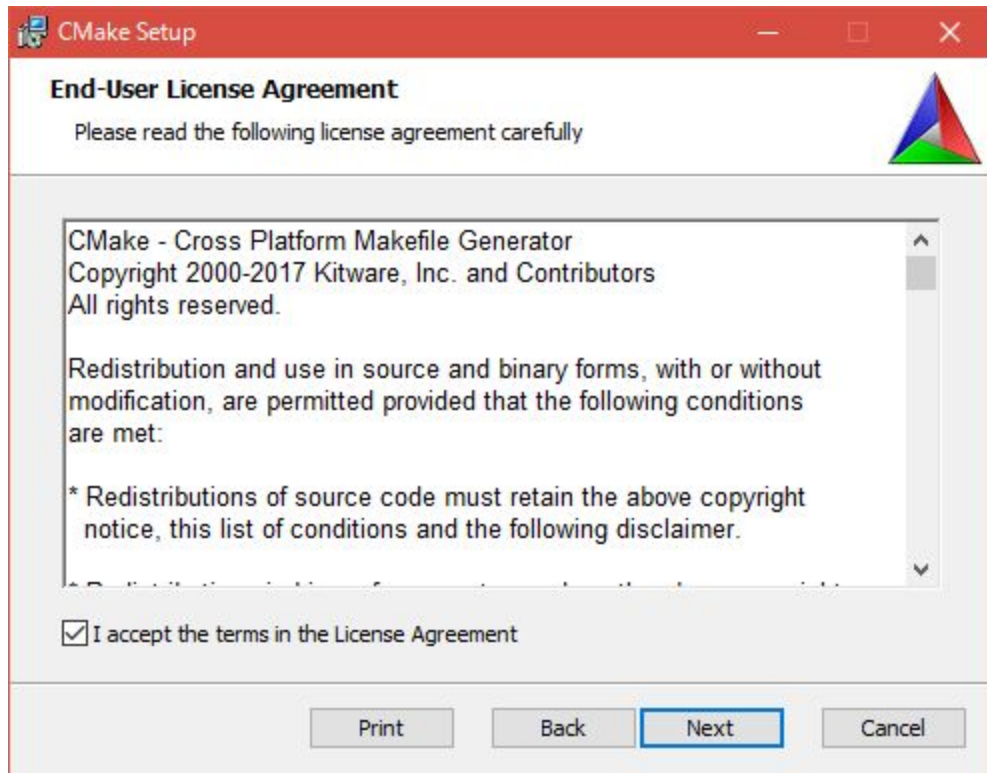
1.CMake installation

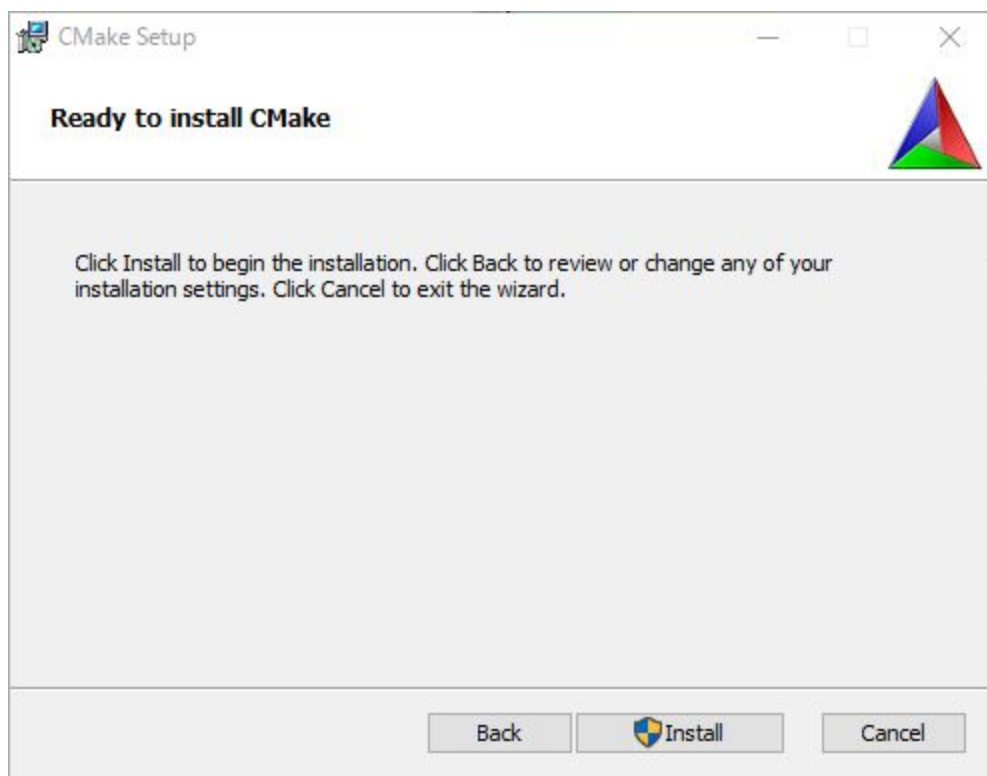
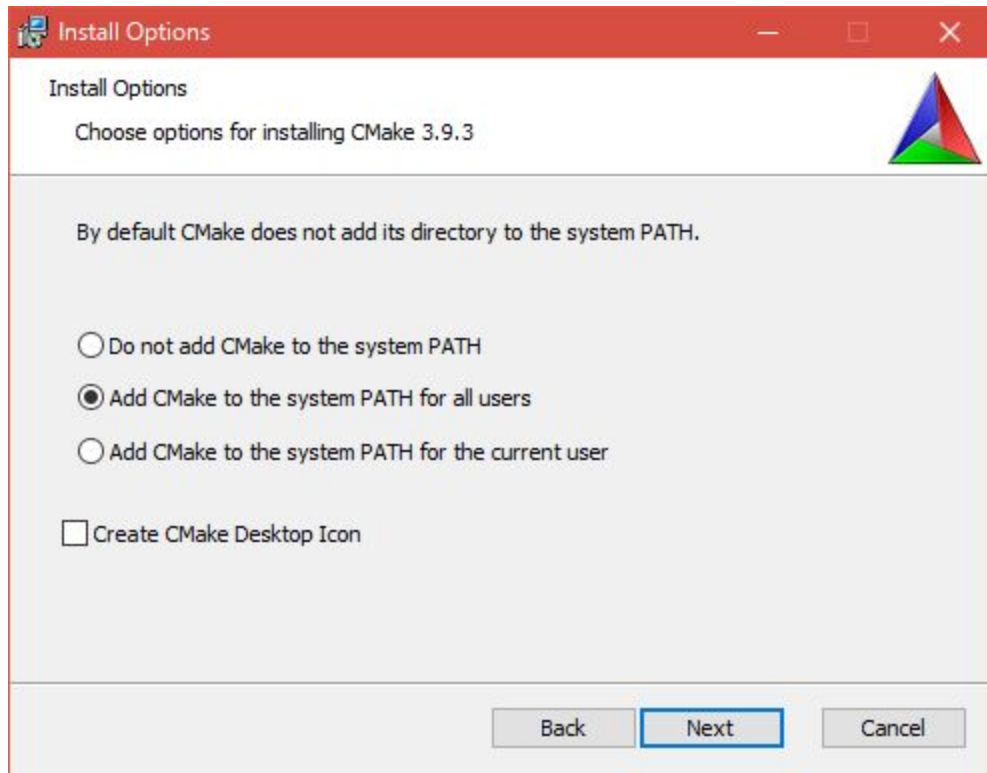
<https://cmake.org/download>

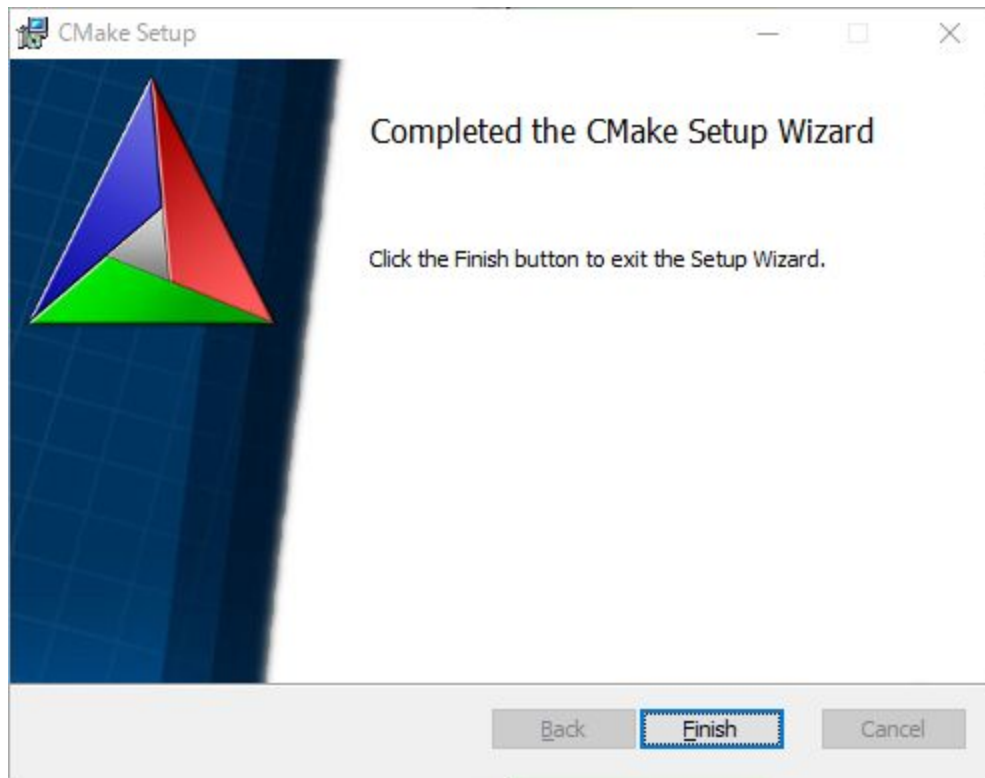
Download from the above URL.

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









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

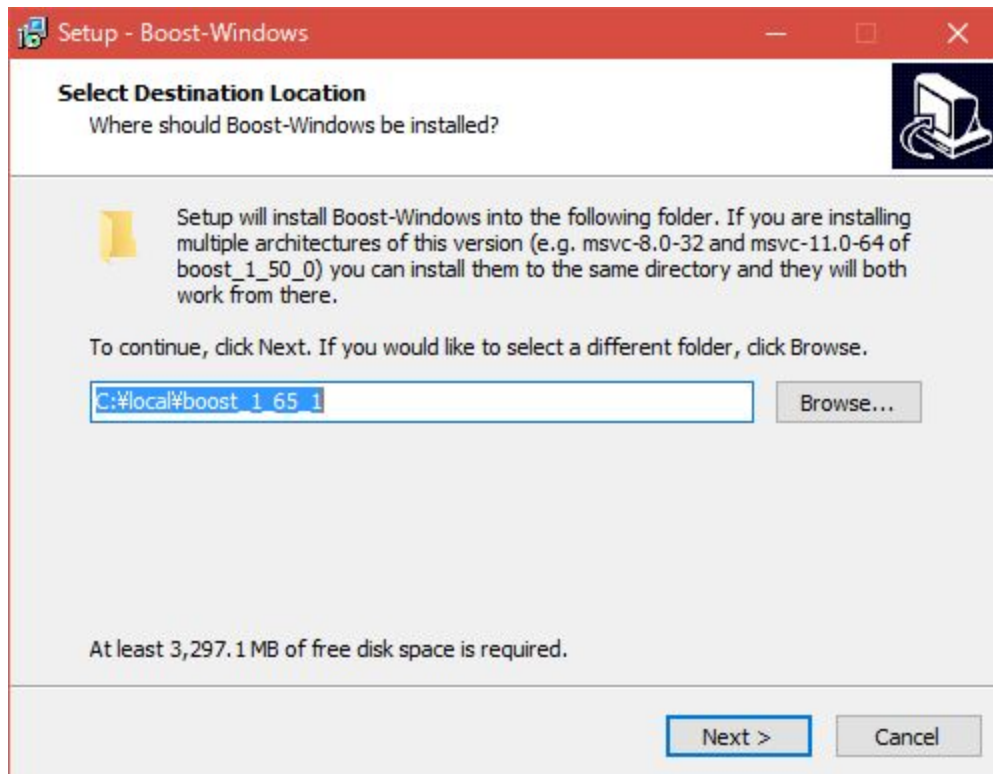
<https://sourceforge.net/projects/boost/files/boost-binaries/>

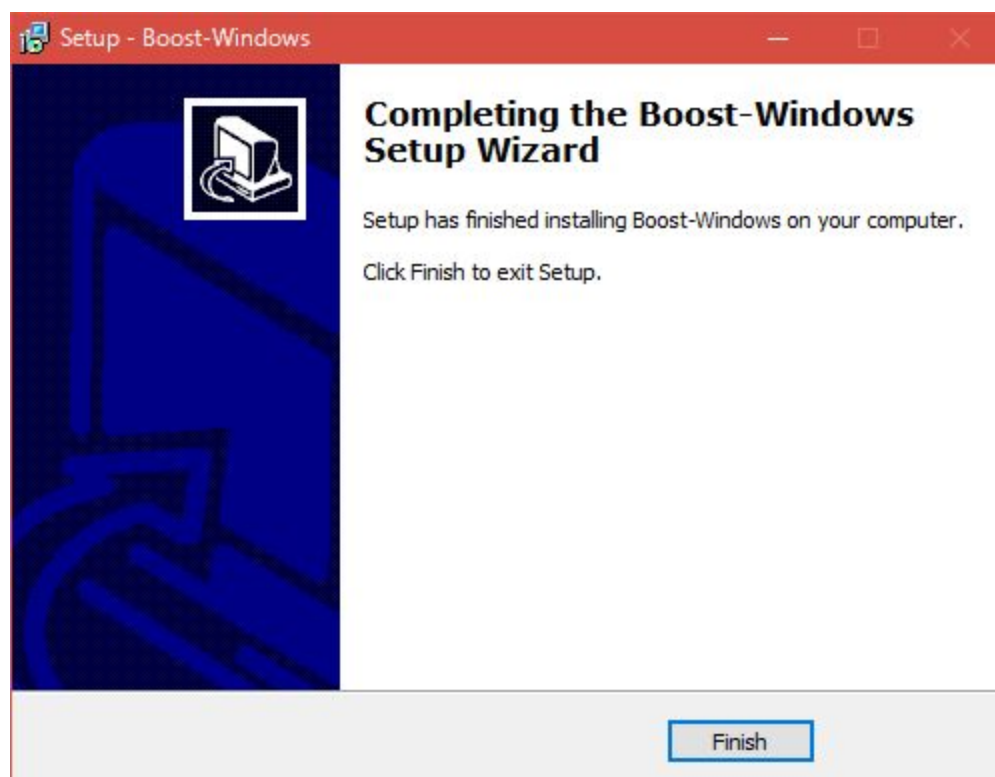
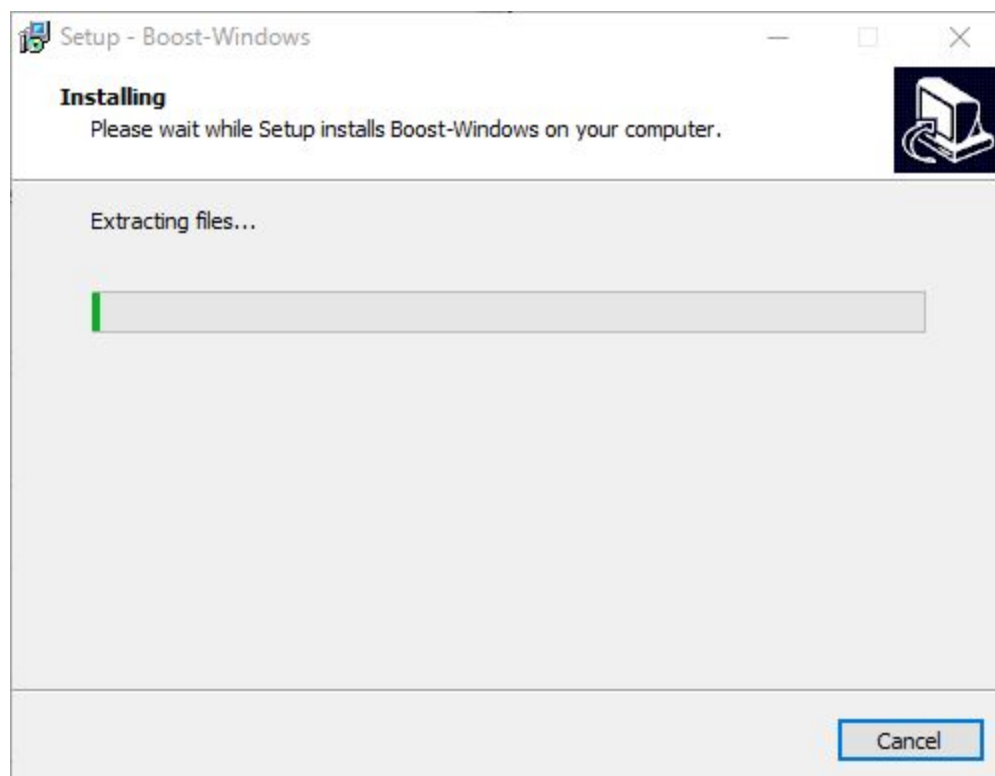
Download from the above URL.

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.

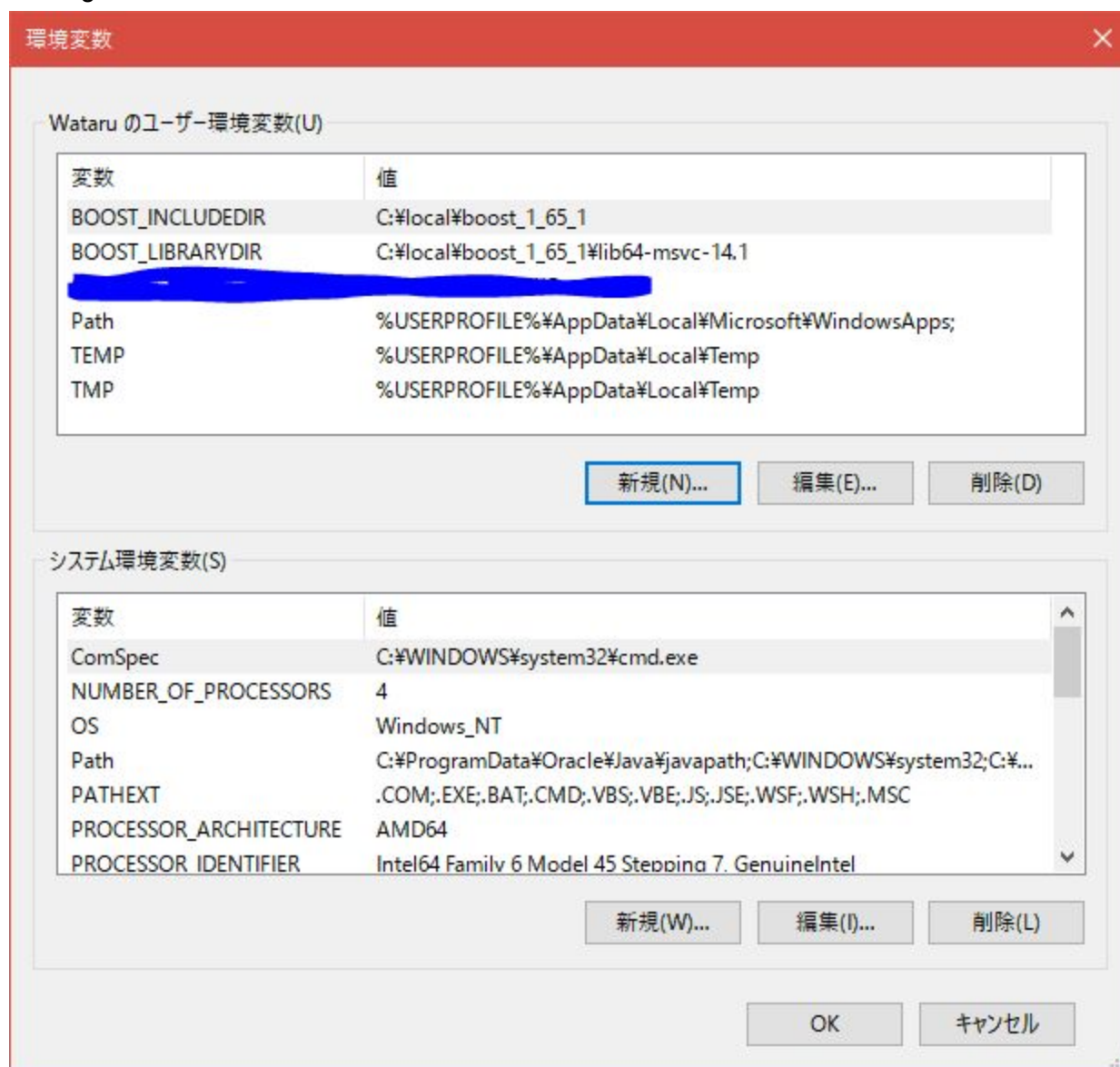




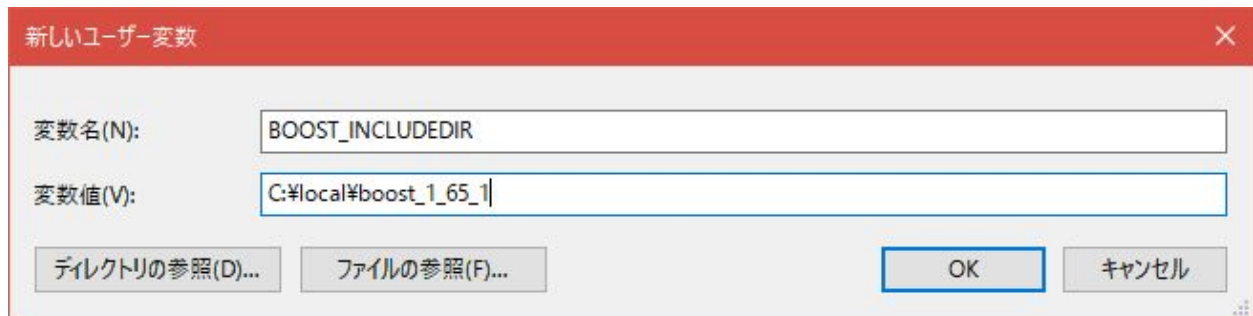
Configuration

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
```

;

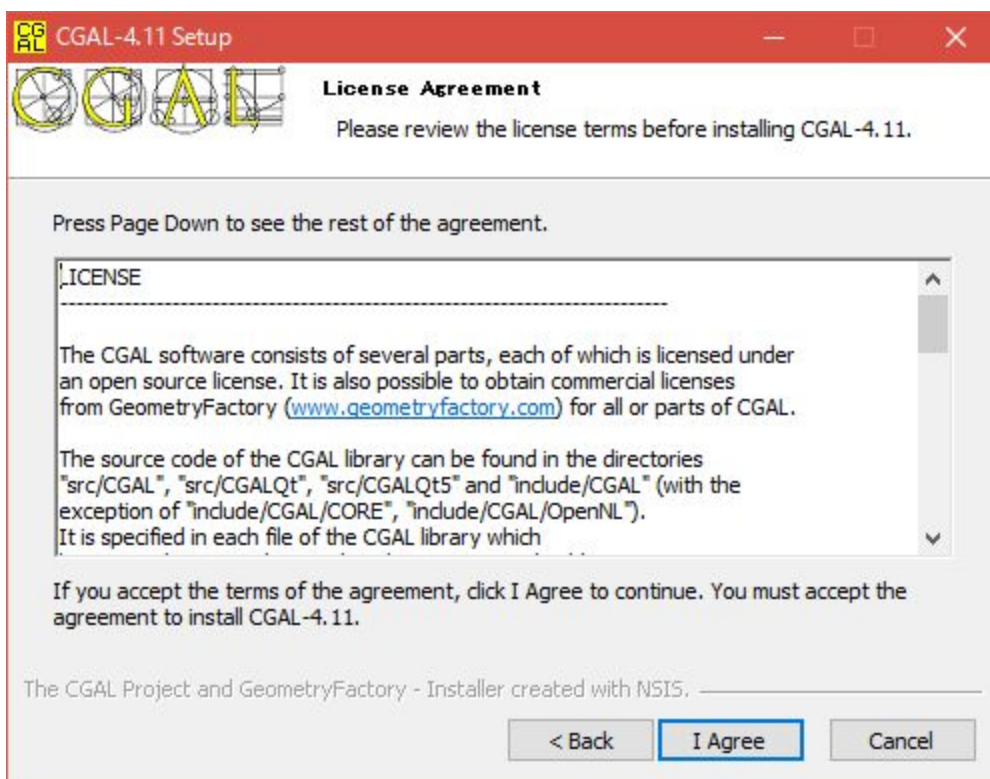
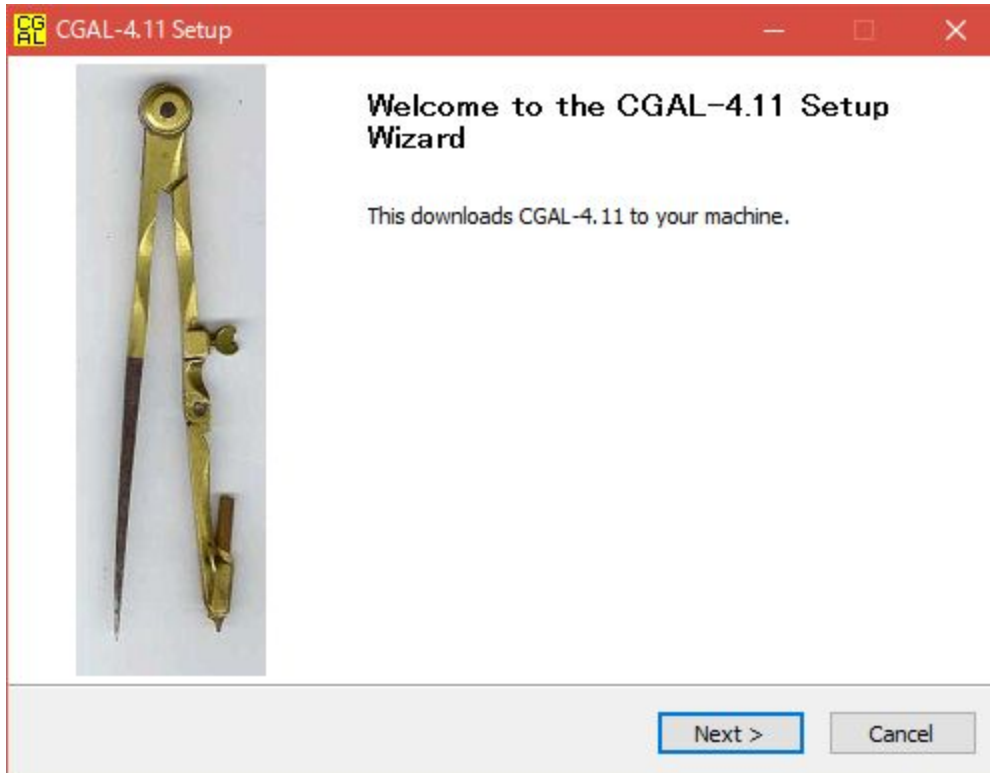
CGAL installation

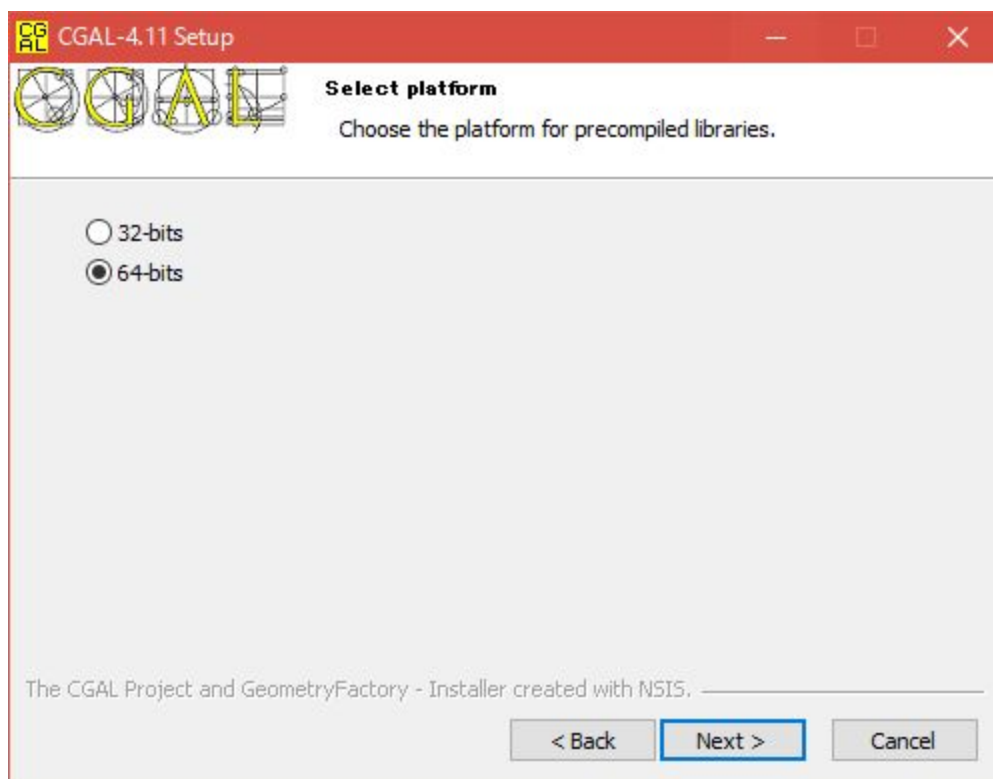
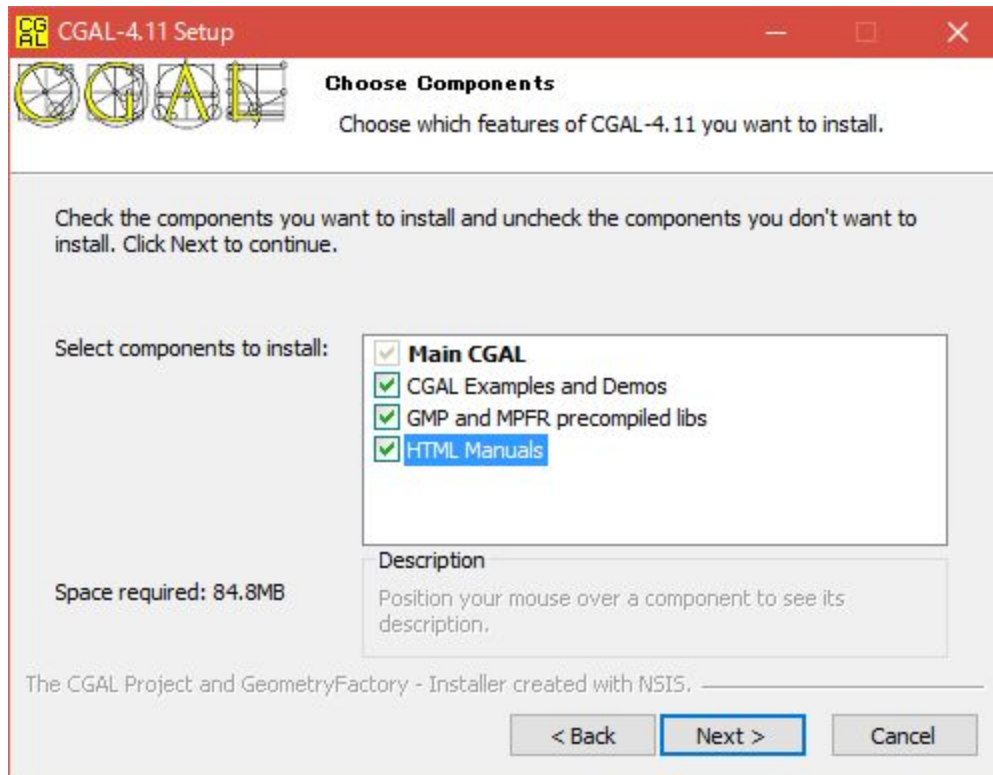
CGAL download and installation

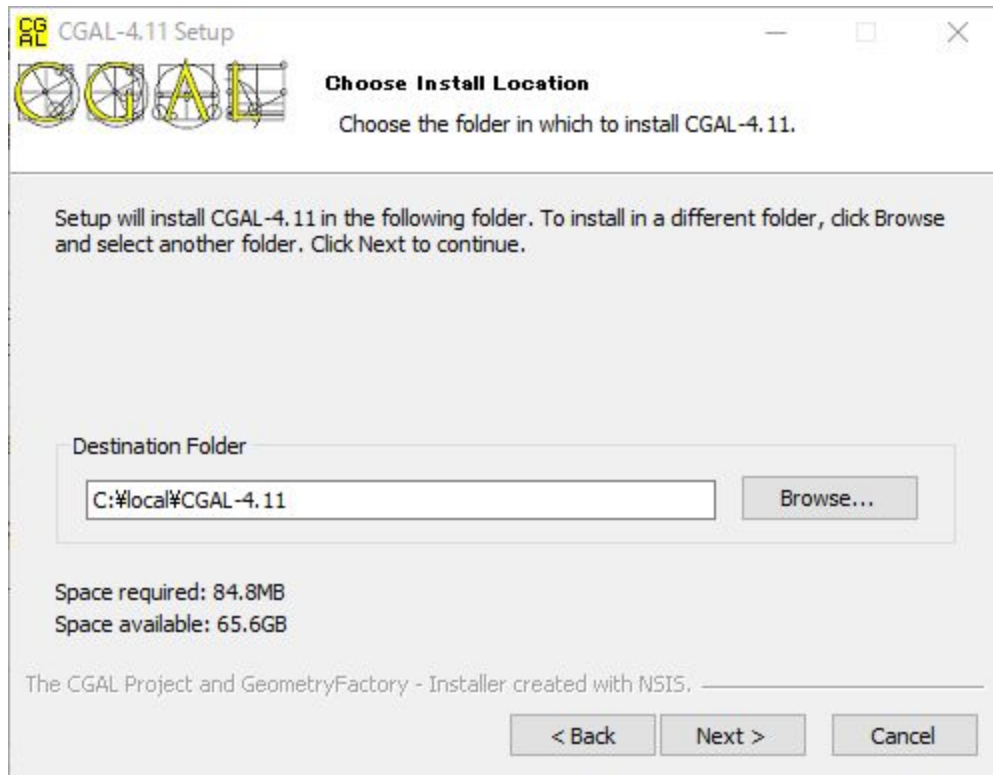
<https://www.cgal.org/download.html>

<https://github.com/CGAL/cgal/releases>

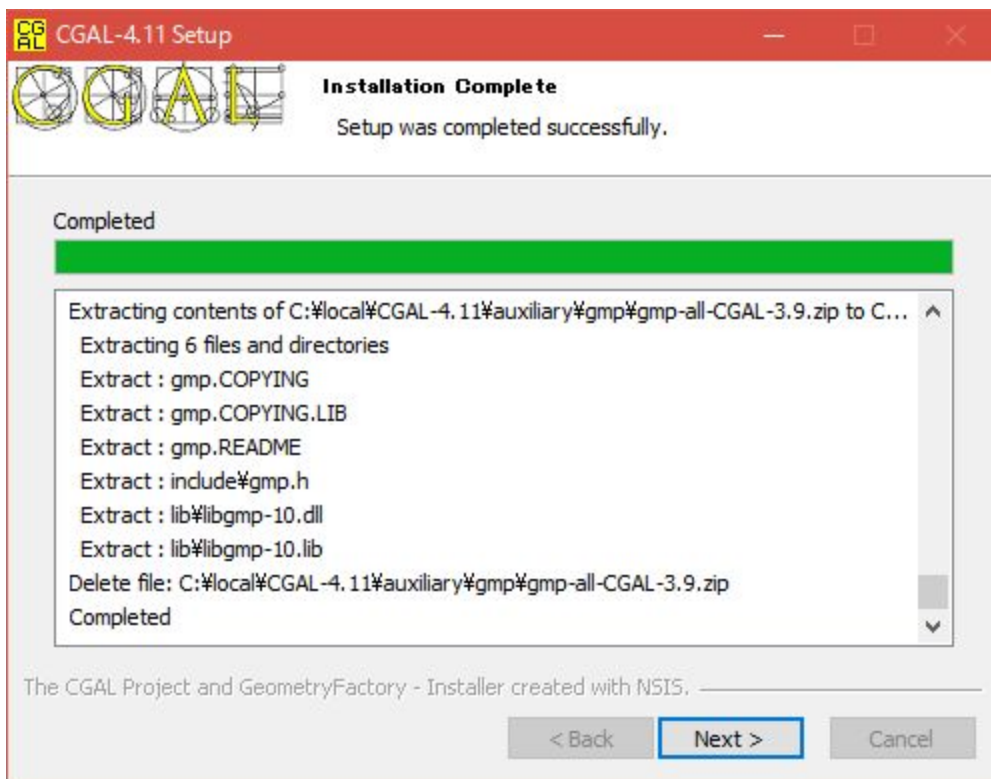
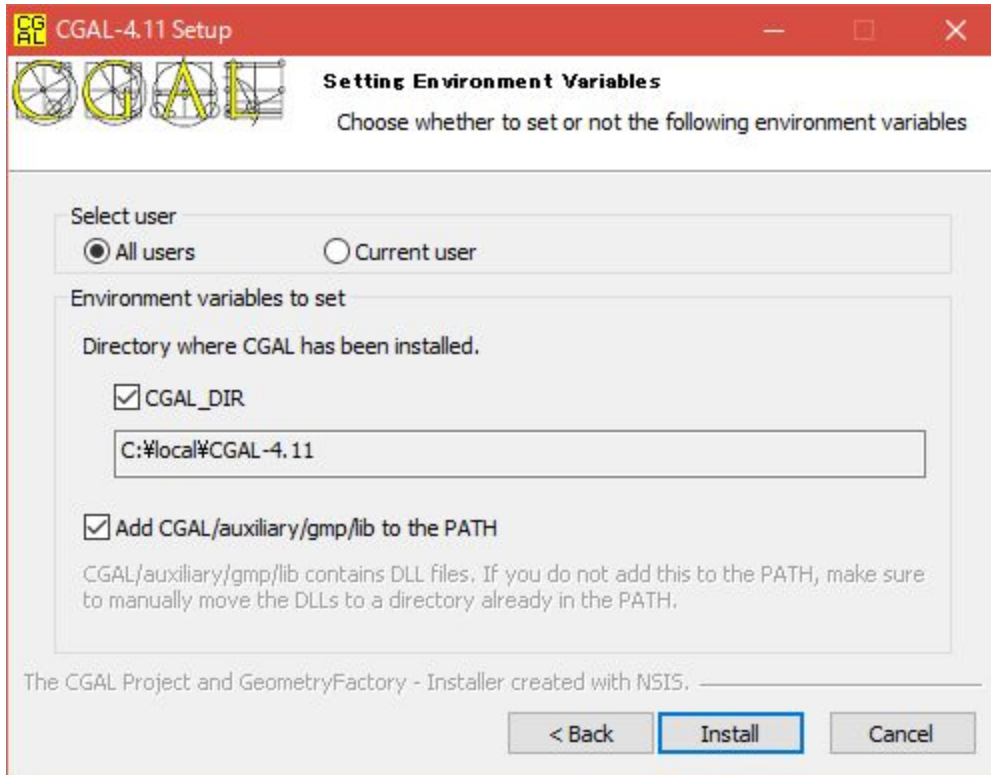
Download and execute CGAL-4.11-Setup.exe. Next Agree Personally, I also want HTML Manual, so check all and click Next. Select 64bit. Make the directory C: \ local \ the same as Boost. (Somehow) I made it all users. The others are not changed. Once complete, Next Downloading is complete, so compile with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

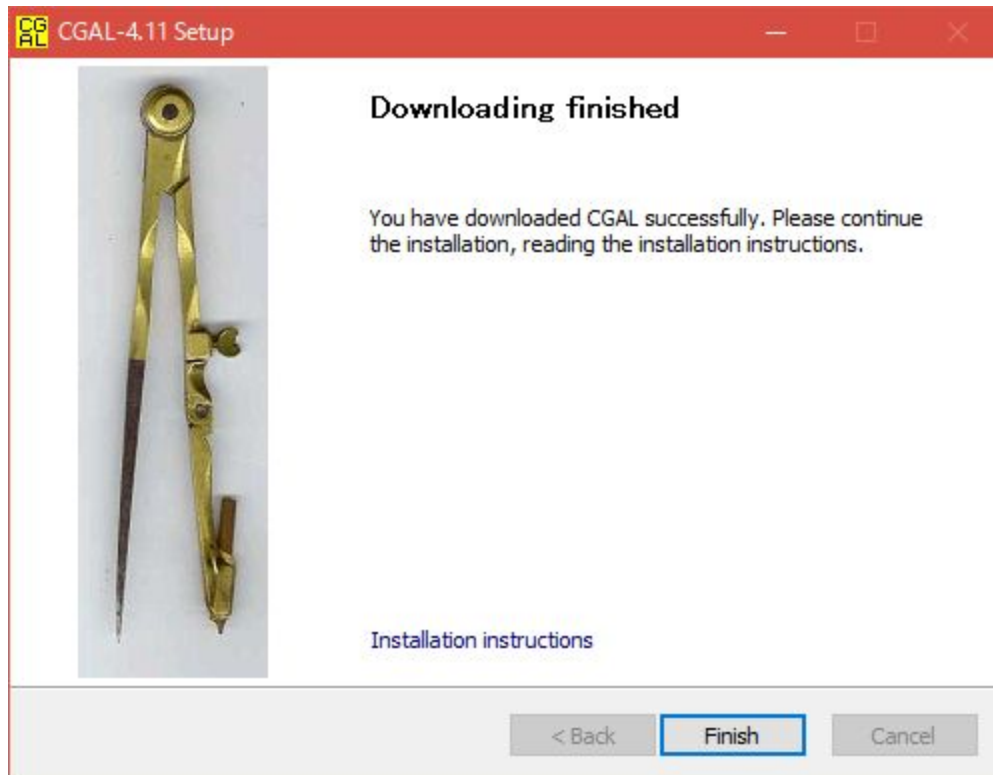






C:\local\CGAL-4.11



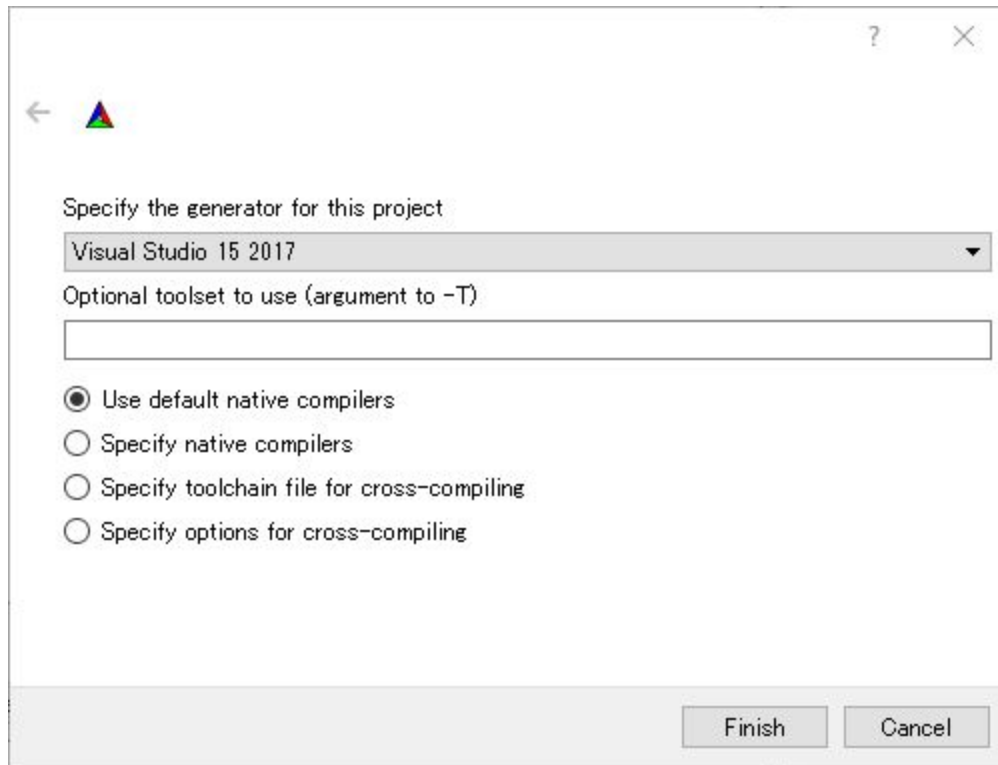


Configuration

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.

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	<input checked="" type="checkbox"/>
BUILD_TESTING	<input type="checkbox"/>
CGAL_CXX_FLAGS	-D_CRT_SECURE_NO_DEPRECATED -D_SCL_SECURE_NO_DEPRECATED -D_C...
CGAL_DONT_OVERRIDE_CMAKE...	<input checked="" type="checkbox"/>
CGAL_ENABLE_PRECONFIG	<input checked="" type="checkbox"/>
CGAL_HEADER_ONLY	<input type="checkbox"/>
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
MPFR_LIBRARIES	C:/local/CGAL-4.11/auxiliary/gmp/lib/libmpfr-4.lib

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

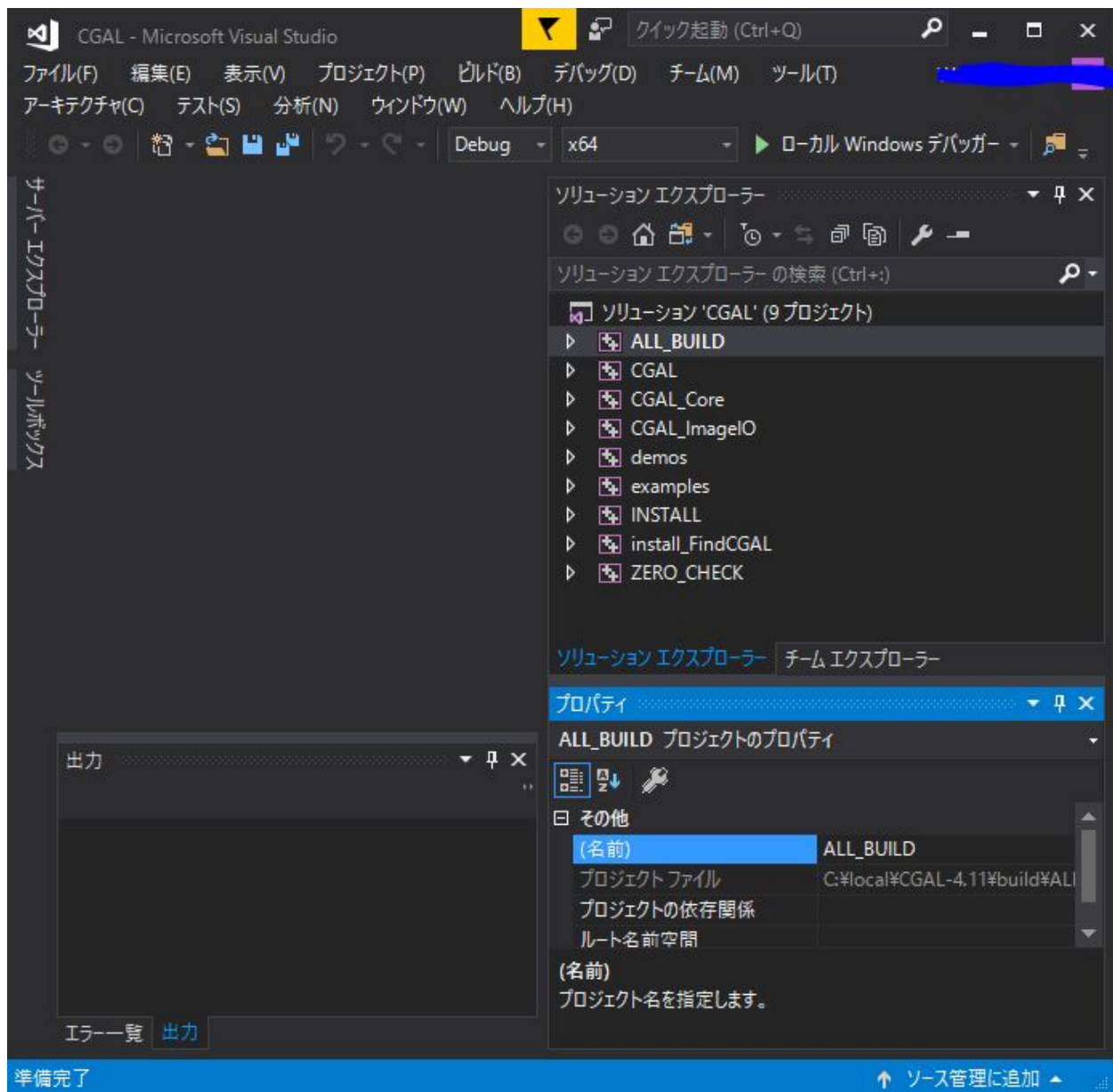
```

== 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) ==

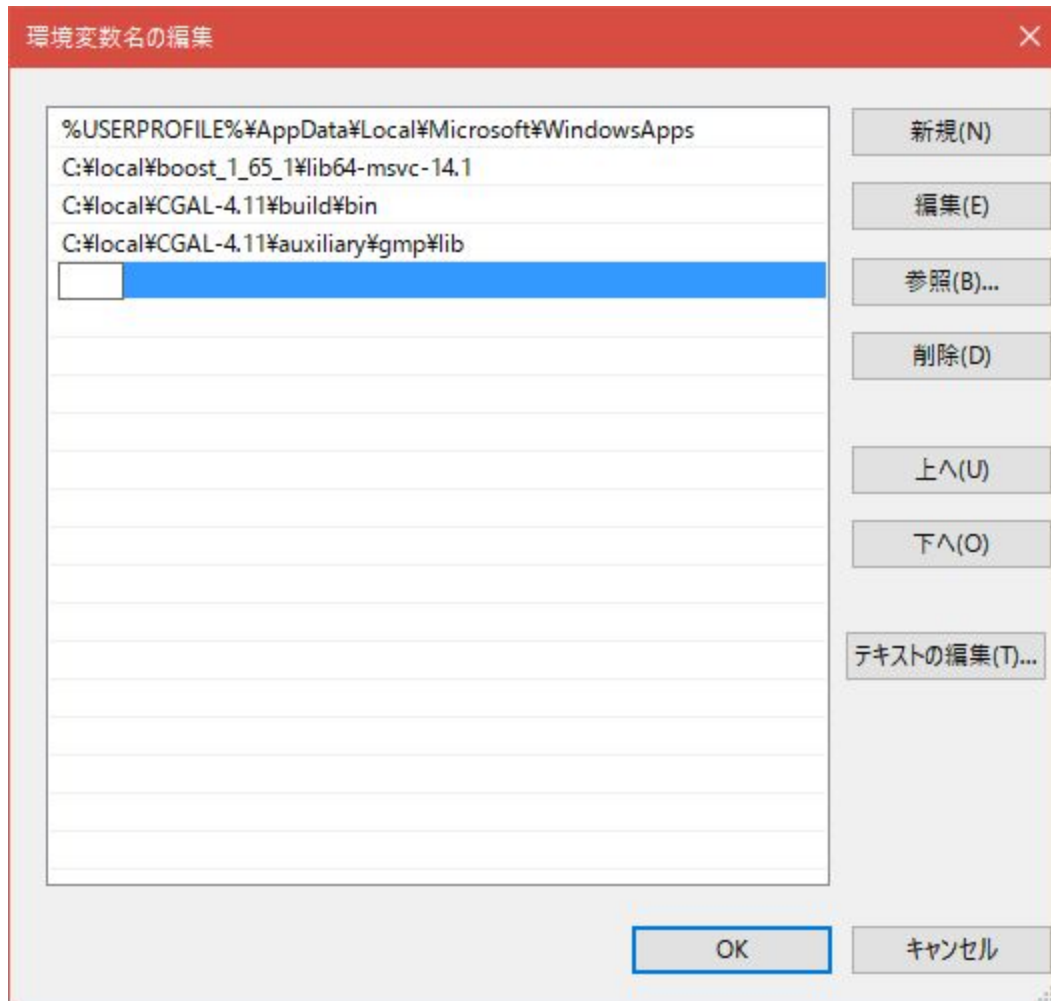
== Generating build files ==
Configuring libCGAL
Requested component: MPFR
Requested component: GMP
libCGAL is configured
Sources for CGAL component library 'CGAL_Core' detected
Configuring libCGAL_Core
Requested component: MPFR
Requested component: GMP

```

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.



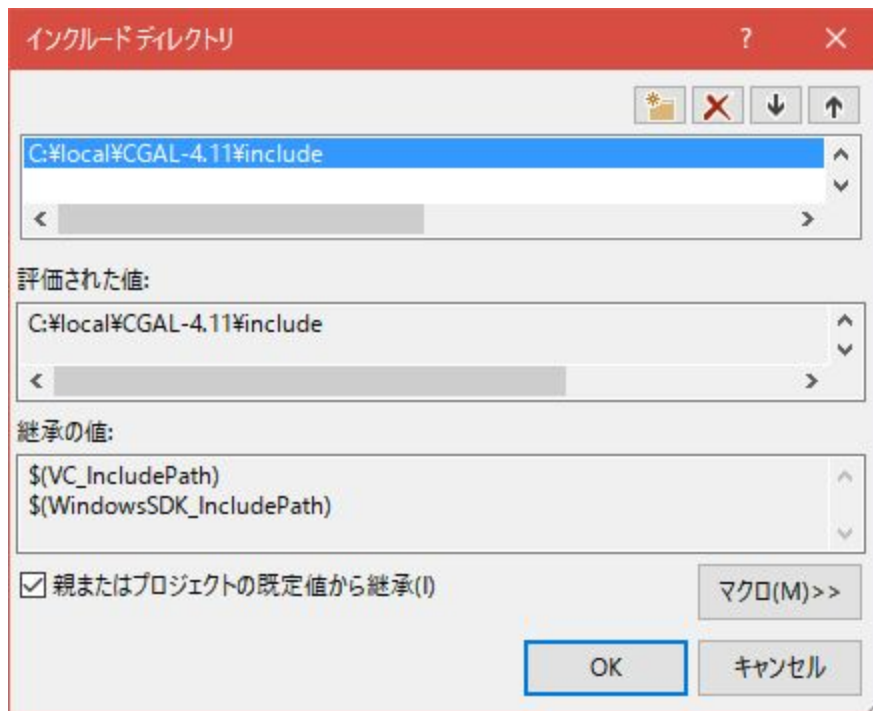
When this is complete, add "C: \ local \ CGAL-4.11 \ build \ bin" and "C: \ local \ CGAL-4.11 \ auxiliary \ gmp \ lib" to environment variable "Path". Move to the environment variable setting screen in the same way as before, and click Edit for Path. When you click New, the input field blinks, so add 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

