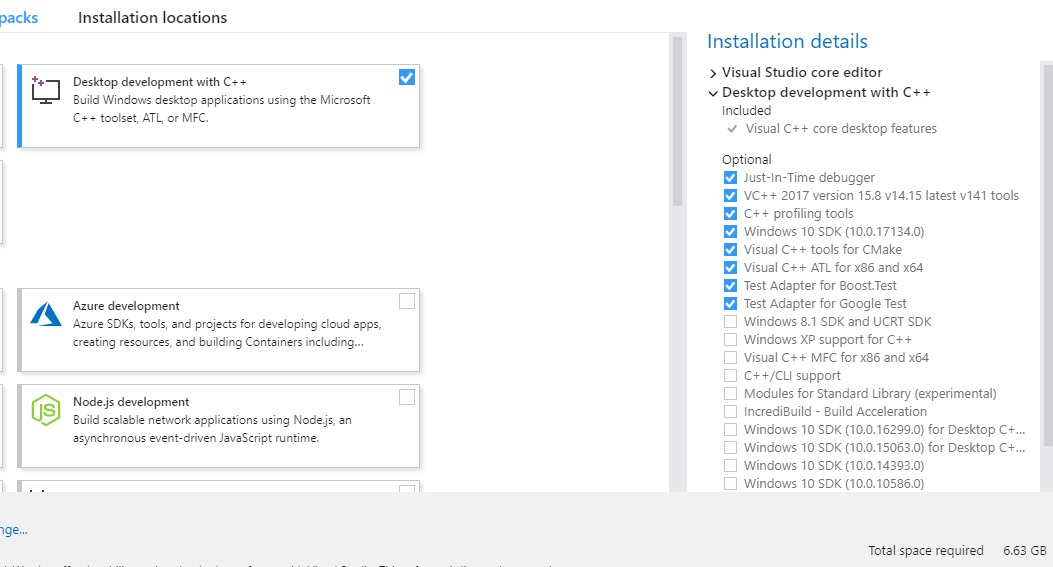
## Instructions for compiling ISOmodel using Visual Studio 2022 / VS 2022 community edition

If you plan to work from the publicly available, open source, ISOmodel git repository and keep your changes/additions open source, the terms of use allow use of VS 2022 community edition even if you are a commercial business or enterprise. If you are a commercial company and plan on keeping those changes within your company or releasing in closed source software, you cannot use VS 2022 community edition

Download VS 2022/ VS 2022 community addition and begin install:

Select Desktop development with C++ and use default settings



## Install CMake >=3.12.3

<https://cmake.org/files/v3.12/cmake-3.12.3-win64-x64.msi>

Installed using option “add Cmake path to all users”

Installed with option “Add CMake to the PATH environment variable” checked.

This will prompt for admin login

## Install Boost >=1.88 prebuilt binaries for msvc 14.3-64

(Visual Studio 2022 v > 17.10 is actually msvc 14.4-64 but things have been set to work with the newer updates to VS 2022)

<https://sourceforge.net/projects/boost/files/boost-binaries/1.88.0/boost_1_88_0-msvc-14.3-64.exe/download>

Install to a simple easy to find directory like C:\local\boost\_1\_88\_0

Graphical user interface, text, application

AI-generated content may be incorrect.

## Install Google Test >= 1.16.0

I recommend you build googletest yourself with the latest version right from github

<https://github.com/google/googletest/releases/download/v1.16.0/googletest-1.16.0.tar.gz>

<https://github.com/google/googletest/archive/release-1.8.1.zip>

Installed to C:\local\googletest\_1\_8\_1 (only the googletest directory)

## Clone the googletest repository and make googletest

git clone https://github.com/google/googletest.git

C:\git>cd googletest

C:\git\googletest>mkdir build

C:\git\googletest>cd build

C:\git\googletest\build>cmake .. -DBUILD\_GMOCK=OFF

Output should look something like this if it builds correctly

-- Building for: Visual Studio 17 2022

-- Selecting Windows SDK version 10.0.22621.0 to target Windows 10.0.26100.

-- The C compiler identification is MSVC 19.43.34810.0

-- The CXX compiler identification is MSVC 19.43.34810.0

-- Detecting C compiler ABI info

-- Detecting C compiler ABI info - done

-- Check for working C compiler: C:/Program Files/Microsoft Visual Studio/2022/Community/VC/Tools/MSVC/14.43.34808/bin/Hostx64/x64/cl.exe - skipped

-- Detecting C compile features

-- Detecting C compile features - done

-- Detecting CXX compiler ABI info

-- Detecting CXX compiler ABI info - done

-- Check for working CXX compiler: C:/Program Files/Microsoft Visual Studio/2022/Community/VC/Tools/MSVC/14.43.34808/bin/Hostx64/x64/cl.exe - skipped

-- Detecting CXX compile features

-- Detecting CXX compile features - done

-- Performing Test CMAKE\_HAVE\_LIBC\_PTHREAD

-- Performing Test CMAKE\_HAVE\_LIBC\_PTHREAD - Failed

-- Looking for pthread\_create in pthreads

-- Looking for pthread\_create in pthreads - not found

-- Looking for pthread\_create in pthread

-- Looking for pthread\_create in pthread - not found

-- Found Threads: TRUE

-- Configuring done (21.7s)

-- Generating done (0.2s)

-- Build files have been written to: C:/git/googletest/build

In the build directory you should now have a “googletest-distribution.sln” file. This is a visual studio project file that you can use to make/build googletest with visual studio 2022

Next we should make it easier to link to the googletest by copying things to a parallel directory to boost. Insite the googletest directory is another googletest directory (parallel to the build directory you just made). Copy that directory to c:\local\googletest

On my computer that means copy

C:\git\googletest\googletest

To

C:\local\googletest

Create a new directory for the libs and copy over the gtest.lib and gtest-main.lib that were created by the build you just did

mkdir C:\local\googltest\lib

Copy the following files into the new lib directory you just made

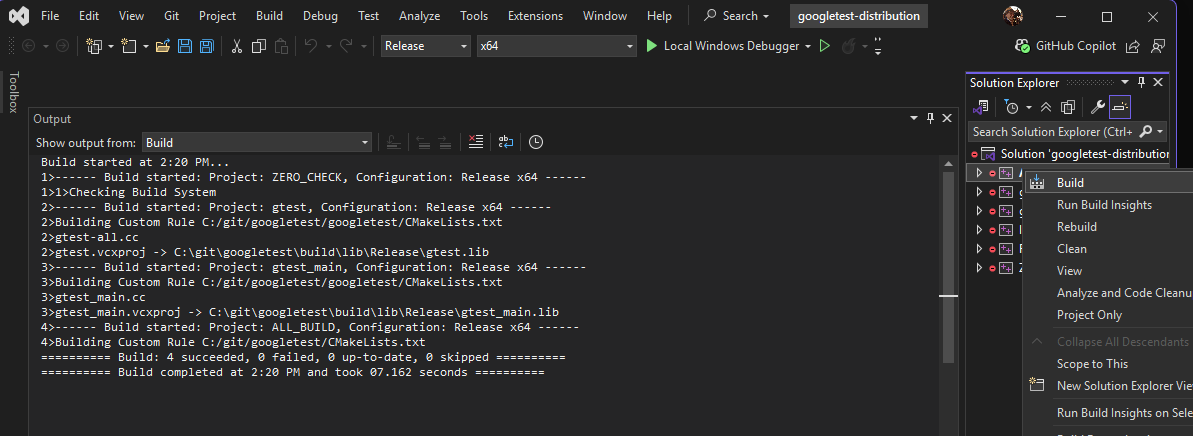
googletest/build/lib/Release/gtest.lib

googletest/build/lib/Release/gtest-main.lib

:

## Make googletest

On windows with Visual Studio you should find the file “googletest-distribution.sln” in the build directory which is a visual studio project file. Open in Visual Studio, set things to release and build with “ALL\_BUILD” option. If it completes correctly the output should look something like this:



This should create “gtest.lib” and “gtest\_main.lib” files in the build\lib\Release directory

## Installing Googletest

At the main root directory for the googletest repository is a directory “googletest” that will have subdirectories “cmake”, “docs” “include”, “samples”, “src”, “test” and files CMakeLists.txt and README.md.

Copy this to a place parallel to the boost libraries like c:\local\

A screenshot of a computer

AI-generated content may be incorrect.

In this googletest directory, make a new directory “lib”

A screenshot of a computer

AI-generated content may be incorrect.

Copy the files “gtest.lib” and “gtest-main.lib” from the git googletest/build/lib/Release directory to this new lib directory

This googletest directory now has include and lib directories with the require include files and full libraries for use with Visual Studio 2022.

## Compiling isomodel using Visual Studio IDE

Update cmake min ver to 3.6

First we need to create the .sln files

Go to isomodel directory root (e.g. c:\git\isomodel\isomodel\) and make a binary directory

cd C:\git\idomodel\isomodel\bin

mkdir bin

cd bin

cmake ../src -G "Visual Studio 17 2022" -DBoost\_LIBRARY\_DIR="C:/local/boost\_1\_88\_0/lib64-msvc-14.3" -DBoost\_INCLUDE\_DIR="C:/local/boost\_1\_88\_0" -DGTEST\_ROOT="c:/local/googletest"

If this works without errors the output should look like below and will have created an isomodel\_project.sln file

Text

AI-generated content may be incorrect.

Text

AI-generated content may be incorrect.

Load isomodel\_project.sln into Visual Studio

Change build to Release, x64 and build the entire project

Graphical user interface, application

AI-generated content may be incorrect.

Graphical user interface, text, application

AI-generated content may be incorrect.