Table of Contents

[Build Zap3D Library 2](#_Toc207456481)

[Option 1: Build Library with command line 2](#_Toc207456482)

[Option 2: Build Library with Visual Studio 2022 IDE 3](#_Toc207456483)

[Build demo application Zap3DDemo using Zap3D Library 4](#_Toc207456484)

[Option 1: Build application with command line 4](#_Toc207456485)

[Option 2: Build application with Visual Studio 2022 IDE 5](#_Toc207456486)

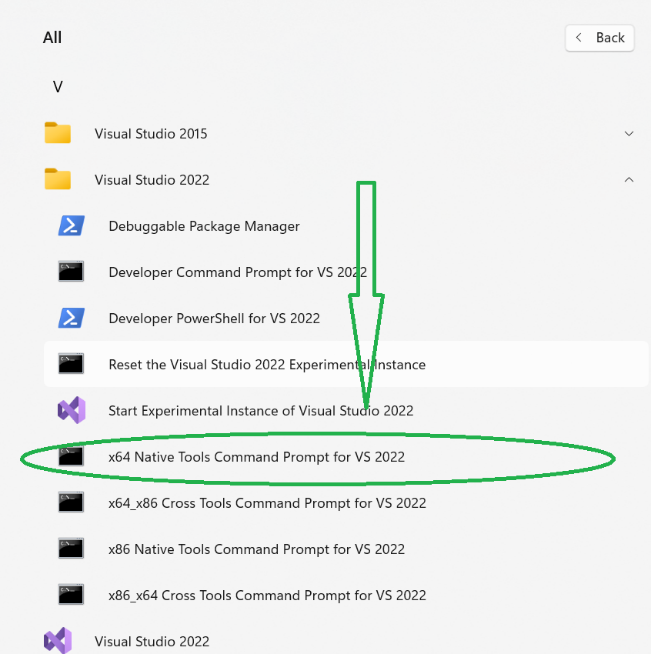
[Create own projects using Zap3D Library 6](#_Toc207456487)

[Build own project with Visual Studio 2022 IDE 6](#_Toc207456488)

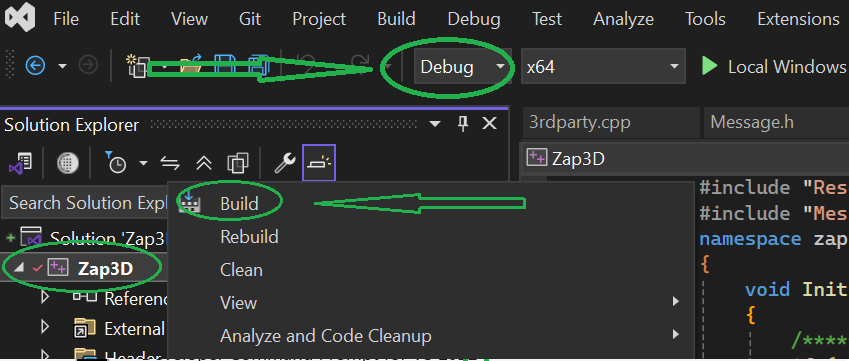
# Build Zap3D Library

## Option 1: Build Library with command line

1. In the folder Zap-Library\3rdparty run the script **extract\_libs.cmd**, so you can see the generated folders  
   A screenshot of a computer

   AI-generated content may be incorrect.
2. Start x64 Native Tools for VS 2022  
   
3. Go to Zap3D folder:  
   cd /D c:\....\Zap-Library\Zap3D
4. Build the project in Debug and Release modes  
   msbuild Zap3D.vcxproj -p:Configuration=Debug  
   msbuild Zap3D.vcxproj -p:Configuration=Release
5. As a result, a folder named **Zap-Library\export** will be generated, notice it is outside Zap3D folder. It will contain **include** folder with exported header files, and **lib\_win** folder containing lib files built in Debug and Release modes. Also, lib\_win folder will contain pdb files, not needed for building, but helping with debug.

## Option 2: Build Library with Visual Studio 2022 IDE

1. In the folder Zap-Library\3rdparty run the script **extract\_libs.cmd**, so you can see the generated folders
2. Start Visual Studio 2022 IDE
3. Choose Open An Existing Project and open Zap-Library\Zap3D\ Zap3D.vcxproj
4. A sln file will be generated, just for convenience.
5. Choose Debug configuration, then right click the project in solution explorer and click Build:  
   W
6. Then choose Release configuration and click build
7. As a result, a folder named **Zap-Library\export** will be generated, notice it is outside Zap3D folder. It will contain **include** folder with exported header files, and **lib\_win** folder containing lib files built in Debug and Release modes. Also, lib\_win folder will contain pdb files, not needed for building, but helping with debug.

# Build demo application Zap3DDemo using Zap3D Library

## Option 1: Build application with command line

1. Before all you should build the Zap3D Library as described in **Build Zap3D Library** section
2. Start x64 Native Tools for VS 2022  
   A screenshot of a computer

   AI-generated content may be incorrect.
3. Go to Zap3D folder:  
   cd /D c:\....\Zap-Library\Zap3DDemo
4. Build the project in Debug and Release modes  
   msbuild Zap3DDemo.sln -p:Configuration=Debug  
   msbuild Zap3DDemo.sln -p:Configuration=Release
5. Run application from where it is generated

## Option 2: Build application with Visual Studio 2022 IDE

1. Before all you should build the Zap3D Library as described in **Build Zap3D Library** section
2. Open Visual Studio 2022 IDE
3. Choose Open Existing Project and open following solution file   
   Zap-Library\Zap3DDemo\ Zap3DDemo.sln
4. Build it in Debug or Release mode. Then run the program.

# Create own projects using Zap3D Library

## Build own project with Visual Studio 2022 IDE

1. For advanced Visual Studio users, notice the **Zap3DTriangleBasic.prop** file. When creating your own projects, instead of manually configuring each include/library folders and files inputs, the **.props** file can be imported to your project and with minimal changes it will help you to spare lots of time:  
   Zap-Library\Zap3DDemo\ Zap3DTrianglesBasic\Zap3DTriangleBasic.props  
   The only changes required are the user macros zaplibdir and thirdpartydir. This is subject to be described more detaily in further releases of this document.