



Thank you for signing up for our free EBook on tool compiling plugins for Maya.

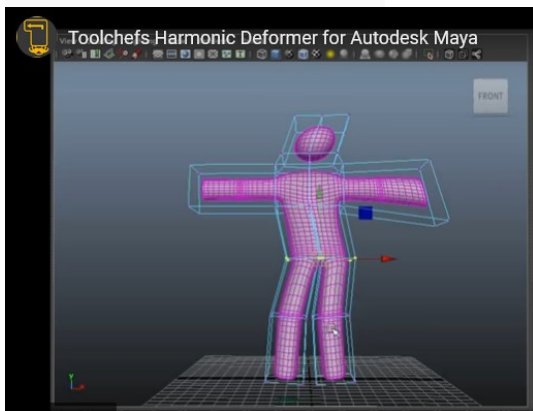
This will help you learn to build two opensource projects for Maya and give you a starting point to explore many other tools that are available while simply needing a way to make them work on your system.

Enjoy the training,  
Josh, Brad and Chad  
Rigging Dojo

## Harmonic Deformer



### Harmonic Deformer (Maya)



The [Harmonic Deformer by Toolchefs](#), is quite easy to build, luckily does not have any major dependencies.

The original repository came with a Makefile that would have to be modified to take into account of local paths.

Instead I added support for CMake and make a pull-request on GitHub, that they nicely allowed to be merged in. Now the build is much easier.

#### Software requirements:

CMake version 3.0.0+, if you can, just install the latest.

<https://cmake.org/>

Visual Studio (Community is perfectly fine, the exact version depends on which Maya you use, but usually 2017 is a safe choice).

<https://visualstudio.microsoft.com/vs/community/>

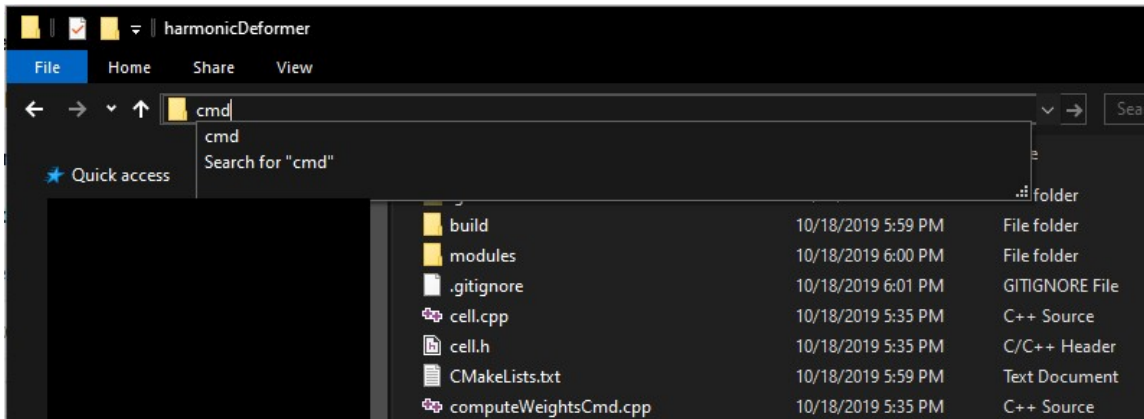
#### Build steps:

First you need to clone or download directly from the website the repository and unzip it

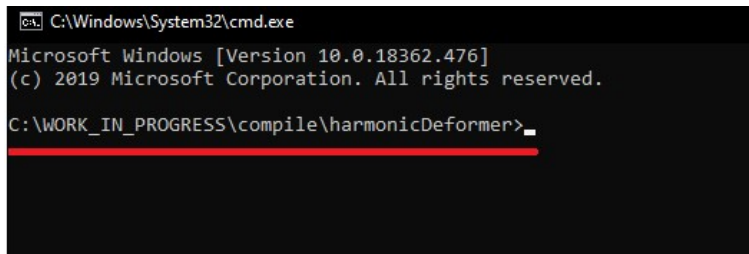
<https://github.com/Toolchefs/harmonicDeformer>

Once you did that open a command prompt pointing to the folder. To do so there are two ways:

- 1) Open a command prompt from the start menu and use cd command to navigate to the folder
- 2) Open the folder using folder explorer, then on the top of the window, where the path is type cmd



Once you do that you should have your command prompt open in the right path.



From there it will be quite simple to build with CMake.

1) run: **mkdir build && cd build**

This will create a folder called build and change the directory to build.

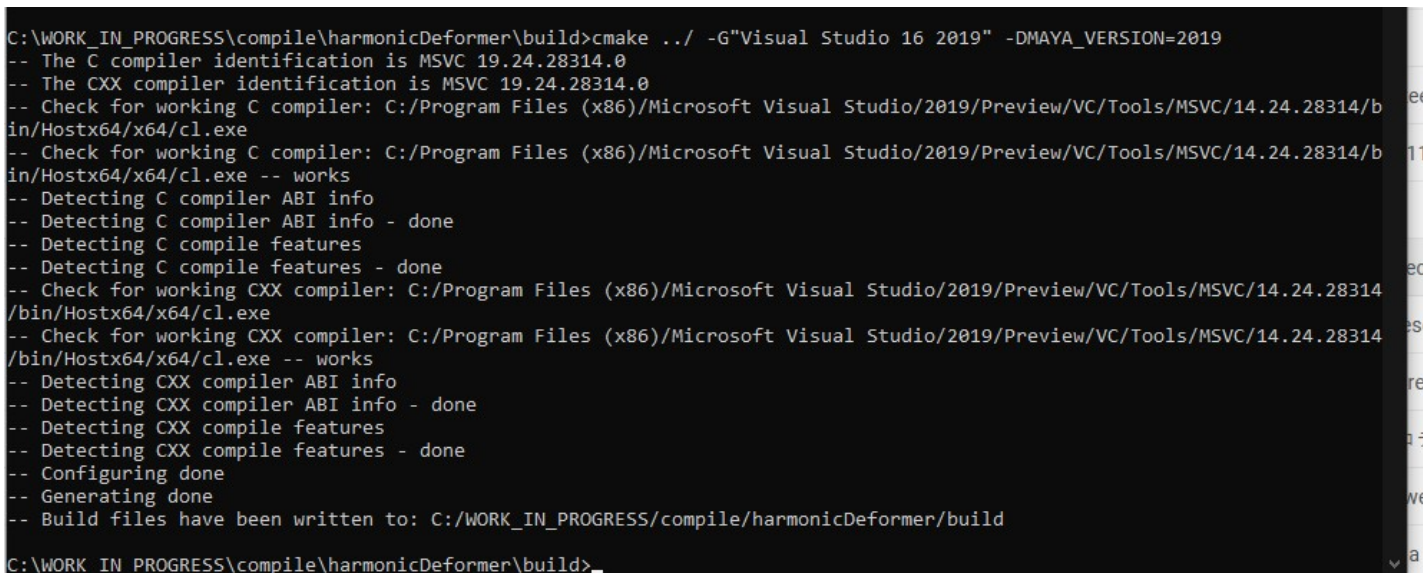
2) Next you need to run: **cmake ../ -G"Visual Studio 15 2017 Win64" -DMAYA\_VERSION=2018**

Ok there is a bit to unpack in this command, first of all we tell cmake to execute, and to find the cmake file, one level above ("../"), next we tell cmake to generate a visual studio solution with the **-G** flag, if you use VS 2017, you need to specify Win64, it is not needed anymore for VS 2019.

Finally to compile we have a dependency on Maya, so we use the Chad Vernon cgcmake (<https://github.com/chadmv/cgcmake>) to do so, by passing which version of Maya we want to use, here I used **Maya 2018**.

If Maya is installed in the standard directory it will find it, otherwise if you have a custom installation path, please refer to CMake documentation.

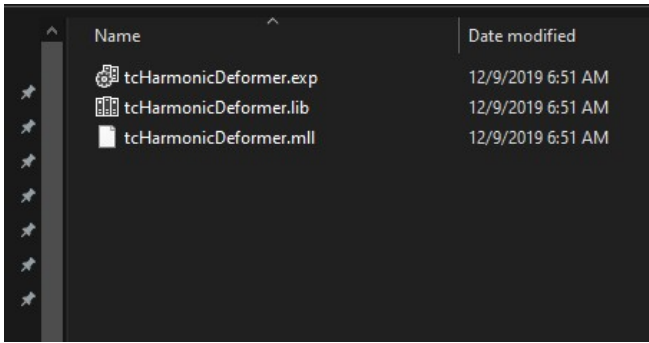
Here the actual commands I ran on my machine:



If you see "Build files have been written to ..." It means everything went well.

To build the actual plugin you have two options, open visual studio and build from there or build directly from command line using cmake, just run: **cmake --build . --config Release**

Once it is done you should see your built file inside the build/Release folder



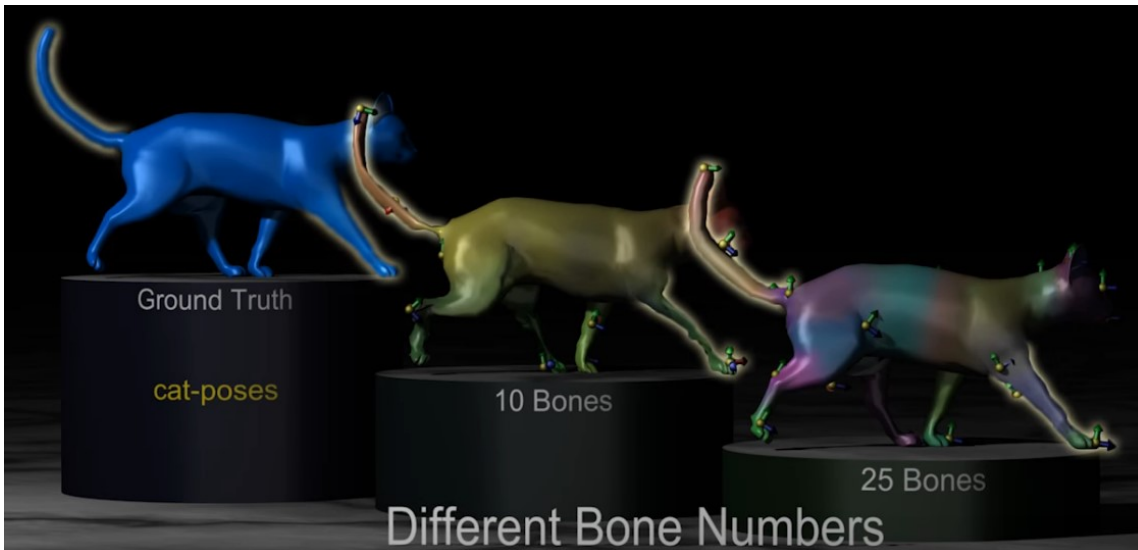
This is it!

You can now install it as a Maya Plugin and test it out.

## Introducing *Dem Bones*: an Open Source Library for Skinning Decomposition



Dem Bones



Skinning decomposition refers to the family of automated methods to convert any animated mesh sequences to skinning models and bone transformations. The power and usefulness of these methods in game production pipeline is any arbitrary, highly deformable models controlled by complex deformation rigs can be generated offline, and then converted to skinning model for real time replaying in game engines. This solution allows artists using all possible rigging tools without worrying about the performance penalty in the game play. Skinning decomposition can be used to compress animations of blendshapes, muscle simulation or cloth simulation. Independent components of the method can also be used to track bones animation or to solve skinning weights from range of motion (ROM) poses.

The state-of-the-art skinning decomposition solution was introduced in the SIGGRAPH Asia 2012 technical paper "Smooth Skinning Decomposition with Rigid Bones" by Binh Le and Zhigang Deng. Since then, this method has been implemented in some animation tools for game production, including Hans Godard's [skinning converter](#), Autodesk Maya's [bake deformer](#), or SideFX Houdini's [skinning converter](#), to name a few. At EA, we also use this technique in our content pipeline quite successfully since 2015.

SEED is pleased to open source *Dem Bones*, a library for skinning decomposition, implemented by Binh Le, the creator of the method. *Dem Bones* offers a lot of refinements for better quality and faster performance compared to the original research paper. *Dem Bones* core library is C++ header-only solvers using [Eigen](#) and [OpenMP](#). We also include a minimalist command line tool for Windows and MacOS that can import and export data from standard asset formats: [FBX](#) and [Alembic](#). The library and command line tool provide core functions of skinning decomposition such as auto generation of joints, solving sparse, non-negative, affine skinning weights and solving rigid joint transformations. The package is ready to use in research and prototyping, or it can be integrated into the production pipeline with minimum effort.

Check out Dem Bones at: <https://github.com/electronicarts/dem-bones>

Dem Bones is released under the [BSD 3-Clause license](#).

### Build setup:

This plugin is a bit more involved to build, but it comes with a CMake setup, the main issue is the number of dependencies it has which needs to be downloaded and copied correctly.  
Let us get started!

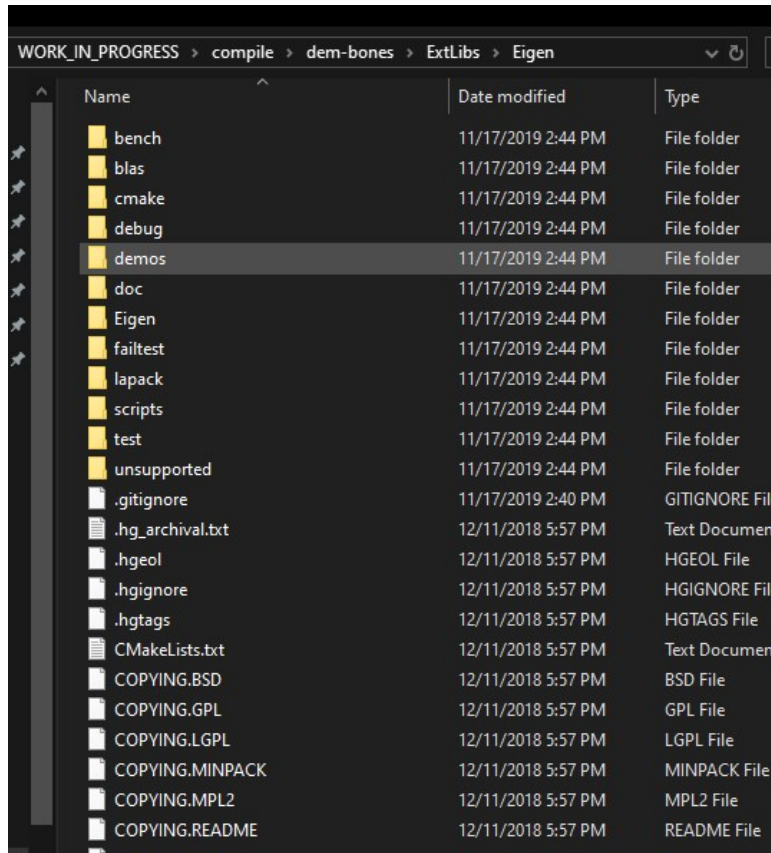
### Software requirements:

Other than the normal dependencies we will download, you will need CMake 3.15, which is one of the latest at the time of speaking. CMake checks for the required version so build will fail unless minimum required version is present on your system.

### Setup of dependencies:

Once you cloned or downloaded the repository you will need to work on your dependencies, let us start by getting Eigen 3.3.7: <http://bitbucket.org/eigen/eigen/get/3.3.7.tar.bz2>

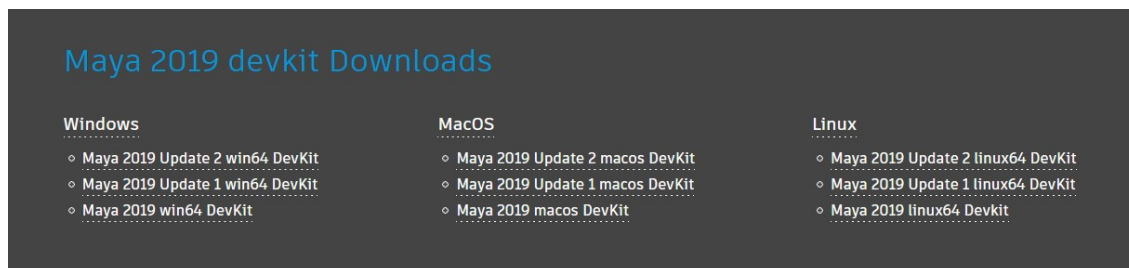
Once you unzip it, just **copy the top level folder inside ExtLibs** and **rename it to Eigen**:



What I did was renaming my top folder eigen-eigen-323c052e1731 to Eigen and copy it.

Next in the list we have is Alembic, which needs to come from the Maya 2019 devkit. You can find it at this link <https://www.autodesk.com/developer-network/platform-technologies/maya>

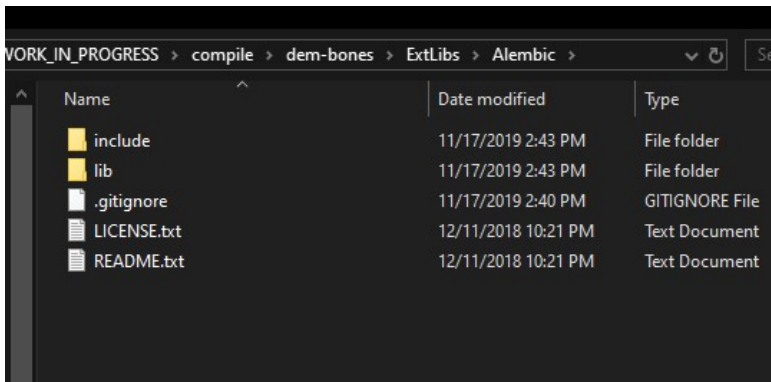
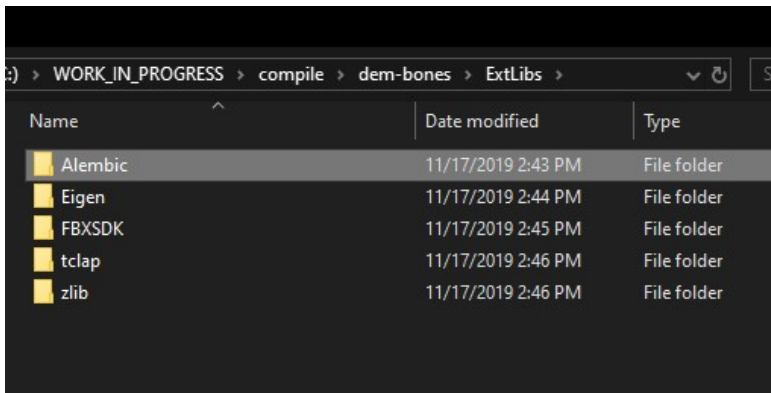
Scroll all the way to the bottom and you will find all the SDK.



Pick the right one for you, in my case I will be using the Win64 update2 which you can get directly from this link

[https://autodesk-adn-transfer.s3-us-west-2.amazonaws.com/ADN+Extranet/M%26E/Maya/devkit+2019/Autodesk\\_Maya\\_2019\\_2\\_Update\\_DEVKIT\\_Windows.zip](https://autodesk-adn-transfer.s3-us-west-2.amazonaws.com/ADN+Extranet/M%26E/Maya/devkit+2019/Autodesk_Maya_2019_2_Update_DEVKIT_Windows.zip)

Navigate inside the extracted zip at path devkitBase\devkit, there you will find the Alembic folder, copy it inside ExtLibs.

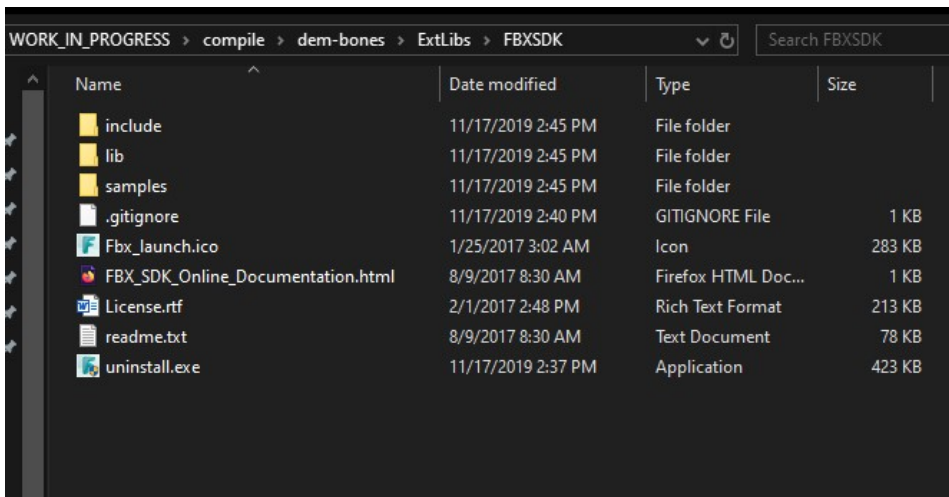


Another one down! Let us move forward, next dependencies is FBX, you can download it from <https://www.autodesk.com/developer-network/platform-technologies/fbx-sdk-2019-0>

Where the windows SDK link is:  
[http://download.autodesk.com/us/fbx/2019/2019.0/fbx20190\\_fbxsdk\\_vs2015\\_win.exe](http://download.autodesk.com/us/fbx/2019/2019.0/fbx20190_fbxsdk_vs2015_win.exe)

Now unluckily this is an executable, so you will have to install it and then go hunt for it in the installation folder, in my case the folder is :C:\Program Files\Autodesk\FBX\FBX SDK\2019.0

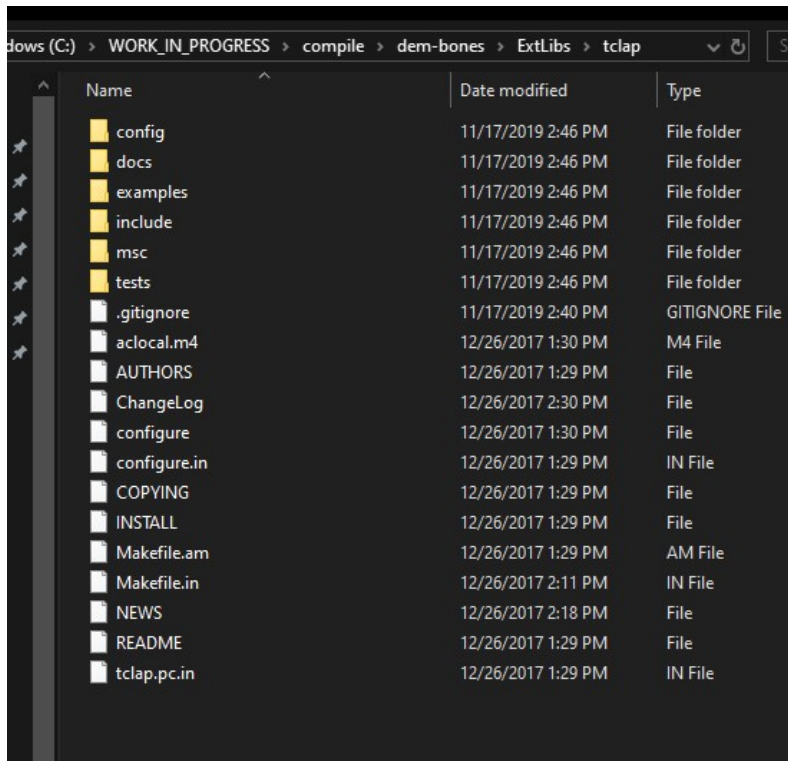
What I did was to go inside that folder and copy the whole content inside ExtLibs/FBX\_SDK like in the below image.



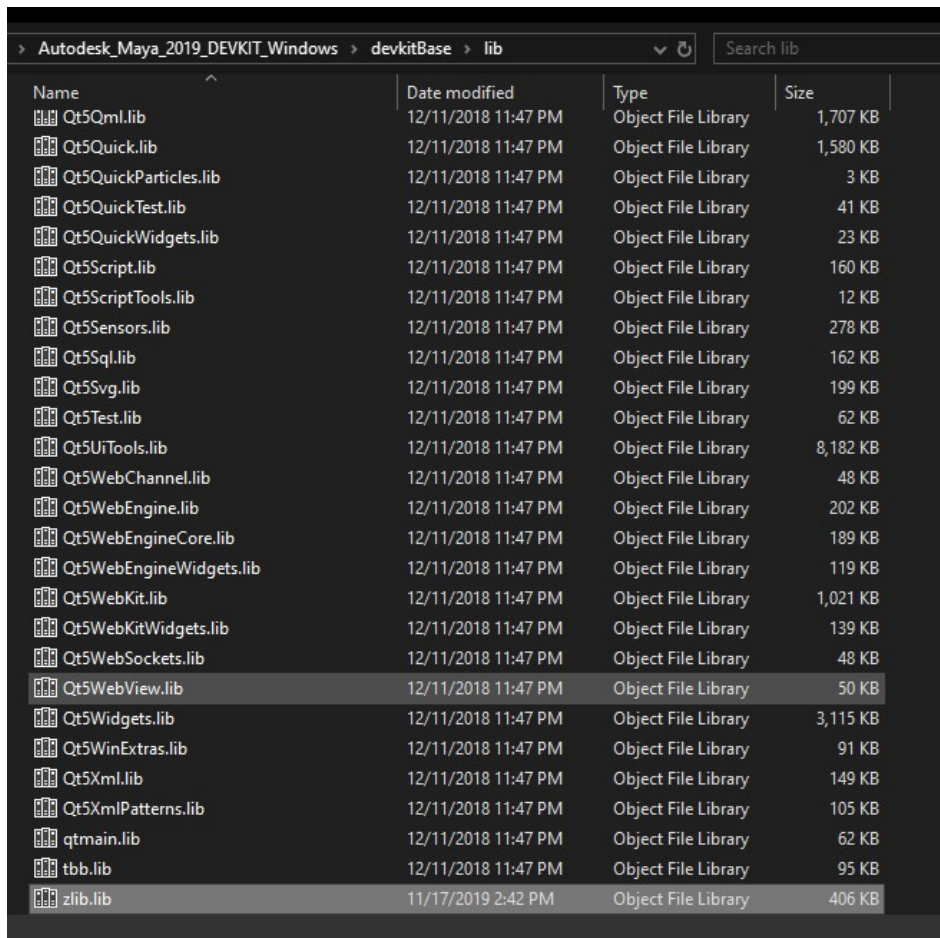
Basically, you just need to get rid of the 2019 folder.

Getting there, next we have tclap, this is another fairly straight forward dependency, download, unzip and copy. Download from: <https://sourceforge.net/projects/tclap/files/tclap-1.2.2.tar.gz/download> Then unzip and copy the content inside ExtLibs/tclap like shown below:

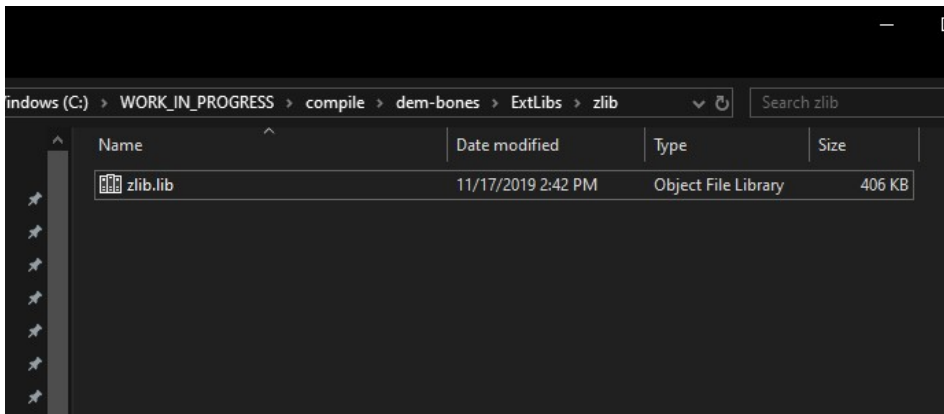




Finally, we have last dependency (windows only), zlib, this is coming again from the Maya devkit so go back to the folder we downloaded, inside **devkitBase/lib** you will find **zlib.lib**



Copy it inside ExtLib/zlib and we are good to go



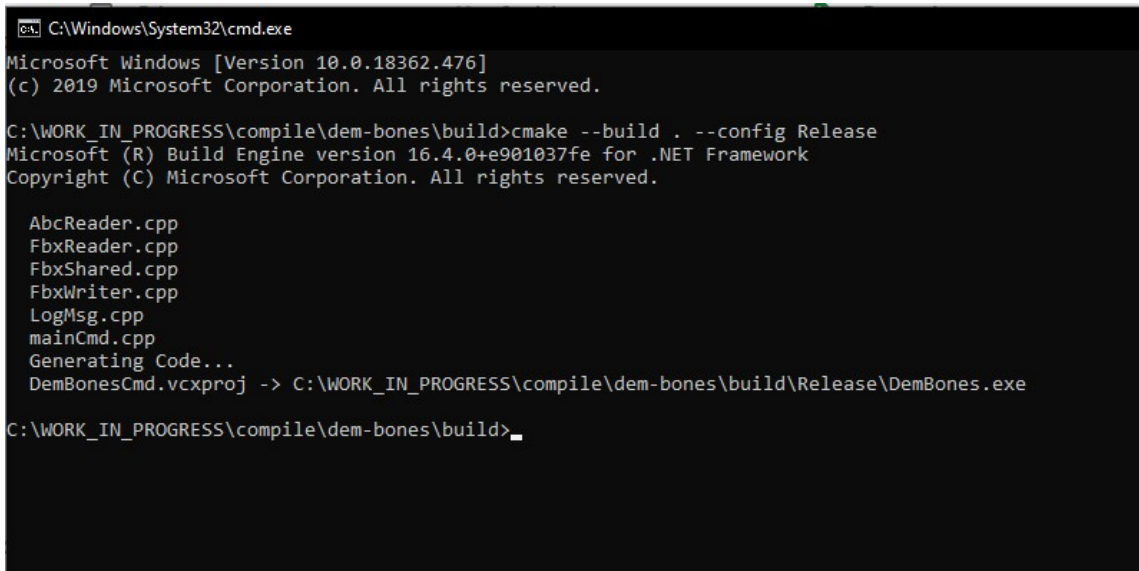
This is it for the dependencies finally we can go about building the plugin,

### Build steps:

Being a cmake build steps are very similar, create a build folder and run the cmake with the generator you wish.

```
mkdir build
cd build
cmake ../ -G"Visual Studio 15 2017 Win64"
```

Finally to build:  
 cmake --build . --config Release



To note, they do provide a pre-compiled windows version in case you need it and do not want to go through all the troubles, you can find it inside dem-bones/bin, there is also a Readme with usage information there.

Hope it helps!

Rigging Dojo API C++ Mentor,  
 Marco Giordano

Interested in learning more on how to write your own tools using the API in Maya? You can come learn how in our API training class with mentor help as you learn, find out more at <https://www.riggingdojo.com/maya-api-c-course/>