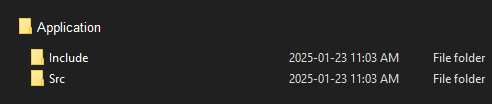
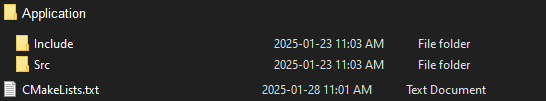
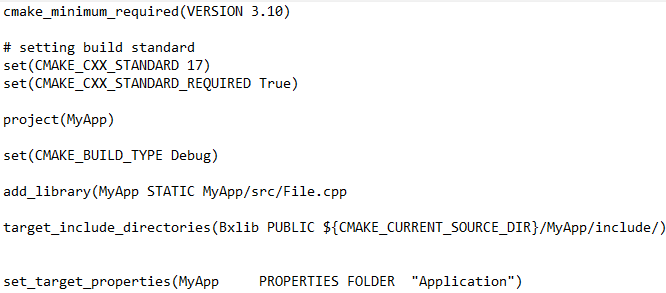
Start with **CMake** and **Conan**

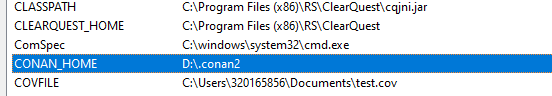
1. Download and install the CMake from <https://cmake.org/download/>
2. Create a sample project with only Source and Header files in it.



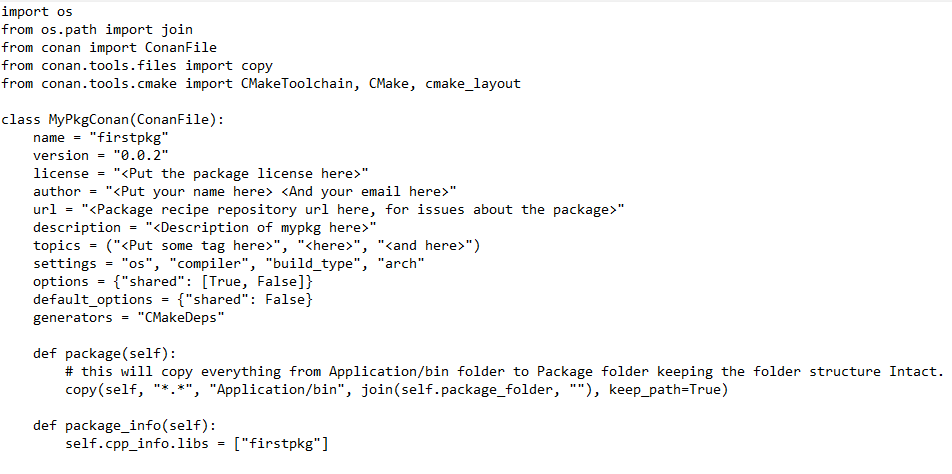
1. Create a *CMakeLists.txt* at the root folder which lists all the source files and the include path as shown in the below CMake File.   
   



1. Navigate to the Application folder.
2. Create a new folder and name it as “build”.
3. Open the VS Dev command prompt and Navigate to the build folder root folder i.e. CD ..\Application\build
4. Run the command “**cmake ../.**”, this will pick the *CMakeLists.txt* and start building the files. All the generated cmake file and the solution will be placed in the build folder.
5. Once the cmake generation is successful, run “**cmake –build .**”, this will build the generated solution and generate the library.
6. The library will be published to *Application/Build/bin/Debug* folder.
7. Now to start with generating and publishing the Conan Package.
8. Install the Conan using “**pip install conan**”.
9. Set the “**.conan2**” folder file path as **CONAN\_HOME** in the environment variable.



1. Create new text file in the root folder*(../Application/)* and name it as *conanfile.py* and place the below content.



1. Open the VS Dev command prompt navigate to the root folder i.e. *../Application* and run “**conan create .**”
2. This will generate the conan package in “**.conan2**” folder (exact path of the file will be mentioned in the command line post successful generation of the package).
3. Check the generated conan package by running the command “**conan list**”
4. Once the conan package is generated we need to publish it to the conan server.
5. In our case we need to create our own local conan server where we can publish our local packages.
6. Run “**pip install conan-server**” in command prompt, this will create the conan server to host our local repository.
7. Open different command prompt terminal and run “**conan\_server**”, this will start the conan server. Keep this terminal open execute any new command on the previously open terminal.
8. Now we can upload our generated conan package from the local cache to the remote server.
9. First, we need to create our own remote server, for this run “**conan remote add my\_local\_server http://localhost:9300**”
10. Run “**conan list**” and copy the package name to be published to the remote server.
11. Run “ **conan upload firstpkg/0.0.2 --remote my\_local\_server**”, this will upload the generated conan package to the local antifactory server.
12. Next step is to download the conan package from the local remote server.
13. To test this, create a conanfile.txt in the root folder. Modify something like below:



1. Now run “**conan install -of build**”, here build is the folder name where the conan will install the configuration and other artifacts.