|  |
| --- |
|  |
|  |
| conan 삭제 |
|  |
| conan install |
|  |
| pip install conan==2.0.0 |
| pip install conan==1.58 |
| D:\git\conan\conan\_src>pip install conan==1.58  Defaulting to user installation because normal site-packages is not writeable |
| ... |
| Running setup.py install for conan ... done  Successfully installed conan-1.58.0 |
|  |
| C:\Users\User>conan  'conan'은(는) 내부 또는 외부 명령, 실행할 수 있는 프로그램, 또는  배치 파일이 아닙니다.  C:\Users\User>cd C:\Users\User\AppData\Roaming\Python\Python311\Scripts  C:\Users\User\AppData\Roaming\Python\Python311\Scripts>conan  Consumer commands  install Installs the requirements specified in a recipe (conanfile.py or conanfile.txt).  config Manages Conan configuration.  get Gets a file or list a directory of a given reference or package.  info Gets information about the dependency graph of a recipe.  search Searches package recipes and binaries in the local cache or a remote. Unless a  remote is specified only the local cache is searched.  Creator commands  new Creates a new package recipe template with a 'conanfile.py' and optionally,  'test\_package' testing files.  create Builds a binary package for a recipe (conanfile.py).  upload Uploads a recipe and binary packages to a remote.  export Copies the recipe (conanfile.py & associated files) to your local cache.  export-pkg Exports a recipe, then creates a package from local source and build folders.  test Tests a package consuming it from a conanfile.py with a test() method.  Package development commands  source Calls your local conanfile.py 'source()' method.  build Calls your local conanfile.py 'build()' method.  package Calls your local conanfile.py 'package()' method.  editable Manages editable packages (packages that reside in the user workspace, but are  consumed as if they were in the cache).  workspace Manages a workspace (a set of packages consumed from the user workspace that  belongs to the same project).  Misc commands  profile Lists profiles in the '.conan/profiles' folder, or shows profile details.  remote Manages the remote list and the package recipes associated with a remote.  user Authenticates against a remote with user/pass, caching the auth token.  imports Calls your local conanfile.py or conanfile.txt 'imports' method.  copy Copies conan recipes and packages to another user/channel.  remove Removes packages or binaries matching pattern from local cache or remote.  alias Creates and exports an 'alias package recipe'.  download Downloads recipe and binaries to the local cache, without using settings.  inspect Displays conanfile attributes, like name, version, and options. Works locally,  in local cache and remote.  help Shows help for a specific command.  lock Generates and manipulates lock files.  frogarian Conan The Frogarian  Conan commands. Type "conan <command> -h" for help |
| Conan이 없는 명령이라고 나왔던 이유는, Python설치할때, PATH를 등록안했기 때문이다.  Python을 다시 설치하여 문제 해결한다 |
| C:\Users\User\AppData\Roaming\Python\Python311\Scripts |
| C:\Users\User\AppData\Roaming\Python\Python311\Scripts\conan.exe |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| CMake+Conan 설명 |
| https://www.youtube.com/watch?v=cwg886syRcU |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| 이 conanfile.py를 편집한다. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| ... |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| https://github.com/CppIndia-UserGroup/CppIndia-SessionDocuments/tree/master/19-12-2020 |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| 샘플 |
| https://github.com/conan-io/training |
| * conanfile.py is a Conan "recipe". It declares dependencies, how to build a package from sources. The same recipe can be used to manage different configurations, like different platforms, compilers, build types, etc * conanfile.txt is a text simplification of conanfile.py, that can be used exclusively to consume dependencies, but not to create packages. conanfile.py can be used both to consume dependencies and to create packages * a profile file is a text file containing configuration values like os=Windows and compiler=gcc. You can pass these values in the command line, but it is better to have them in files, easier to manage and more convenient. * settings.yml is a declaration of what values can the settings take. To validate the inputs and make sure there are no typos, and use a common set of configurations so people can collaborate together.   I suggest following the tutorials in the docs, like <https://docs.conan.io/en/latest/getting_started.html>, or if you are into video-format, this free training is good: <https://academy.jfrog.com/path/conan>  Regarding the versions, you need to use those defined in the settings, for Visual Studio you need to use 14, 15, etc. The new msvc compiler setting, experimental, will be using the compiler version, like 19.xx. In general, it is not needed to specify the compiler version down to the patch, because this is mostly for the binary model, and it is typically not needed to go to that level. If you want to learn how to customize the settings value, [read this section](https://docs.conan.io/en/latest/extending/custom_settings.html) |
|  |
|  |
| install packages from the new conan center remote |
| conan install spdlog/1.11.0 |
| conan install boost/1.72.0 |
|  |
|  |
|  |
| CMake 강좌 |
| https://www.youtube.com/watch?v=QhKkZ\_7Wgfg |
|  |
|  |
| Visual Studio 2022에서는 bullet CMake가 된다. |
|  |
|  |
| 2022에서 이것이 된다. |
| D:\git\projects\modern-cmake\examples |
|  |
| Conan-project 테스트 |
| D:\git\projects\examples2\tutorial\consuming\_packages\simple\_cmake\_project |
|  |
| https://docs.conan.io/en/2.0/tutorial/consuming\_packages/build\_simple\_cmake\_project.html |
| “C:\Program\_Files\Microsoft\_Visual\_Studio\2022\Professional\Common7\IDE\CommonExtensions\Microsoft\CMake\CMake\share\cmake-3.24\Modules |
| FindPackageHandleStandardArgs.cmake:230(message) |
| Could Not find ZLIB(missing ZLIB\_LIBRARY ZLIB\_INCLUDE\_DIR) |
|  |
|  |
|  |
|  |
|  |
|  |