**Cmake overview**

|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Version** |
| 1 | Initial Version | 1.0 |
|  |  |  |
|  |  |  |
|  |  |  |

**Contents**

|  |  |  |
| --- | --- | --- |
| **No.** | **Title** | **Page** |
| 1 | Description | 3 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Introduction:**

CMake is a cross-platform, free and open-source make system. It is used to control the software compilation process using simple platform-independent and compiler-independent configuration files. It generates native makefiles and workspaces that can be used in the compiler environment of your choice. It supports directory hierarchies and applications that depend on multiple libraries. It is used in conjunction with native build environments such as Make, Qt Creator, Ninja, Apple's XCode and Microsoft visual studio.

It has minimal dependencies, requiring only a c++ compiler on its own build system. It is quite sophisticated. It is possible to support complex environments requiring system configuration, preprocessor generation, code generation, and template instantiation.

**Compiler option:**

|  |
| --- |
| # Applies to source in the current directory and below  if (MSVC)  add\_compile\_options(/WX)  else()  add\_compile\_options(-Wall -Wextra -pedantic -Werror)  endif()  # Apply to specific target  target\_compile\_options(<target> PRIVATE -Wall) |

**Pre-processor definitions:**

|  |
| --- |
| # Applies to source in the current directory and below  add\_compile\_definitions(WITH\_OPENCV2)  # To specific target  target\_compile\_definitions(<target> PRIVATE WITH\_OPENCV2) |

**Input variables: build options:**

|  |
| --- |
| * CMAKE\_INSTALL\_PREFIX : Install directory used by install. On “make install”, this directory is prepended onto all install directories. * CMAKE\_BUILD\_TYPE: choose build type   + Debug: Debug build   + Release: Release build   + RelWithDebInfo: Release build with debug information   + MinSizeRel: Release build optimized for size |

**Setting C++ standard:**

|  |
| --- |
| set (CMAKE\_CXX\_STANDARD 11)  set (CMAKR\_C\_STANDARD 99)  set (CMAKE\_CXX\_STANDARD\_REQUIRED ON) : it will give the upfront error if standard is not supported. |

**Cmake build phases:**

|  |
| --- |
| -- The C compiler identification is GNU 7.4.0  -- The CXX compiler identification is GNU 7.4.0  -- Check for working C compiler: /usr/bin/cc  -- Check for working C compiler: /usr/bin/cc -- works  -- Detecting C compiler ABI info  -- Detecting C compiler ABI info - done  -- Detecting C compile features  -- Detecting C compile features - done  -- Check for working CXX compiler: /usr/bin/c++  -- Check for working CXX compiler: /usr/bin/c++ -- works  -- Detecting CXX compiler ABI info  -- Detecting CXX compiler ABI info - done  -- Detecting CXX compile features  -- Detecting CXX compile features - done  -- Configuring done  -- Generating done  -- Build files have been written to: /home/belotes/Documents/Learning/Workspace/CmakeLearning/1\_\_HelloWorld/Build |

* Configure time :

cmake code run here. In this step cmake parse the top level CmakeLists.txt and create the CMakeCache.txt files with cache variable.

* Generate time:

In this step CMake will generate native build tool files using information from CMakeLists.txt and variables from generated CmakeCache.txt.

* Build time:

Build tool like make runs the compiler