

- [Visual Studio libtorch使用教程](#)
 - [预备软件](#)
 - [libtorch准备](#)
 - [检查cmake版本及功能](#)
 - [项目准备](#)
 - [项目说明](#)
- [在HM中调用代码](#)

Visual Studio libtorch使用教程

预备软件

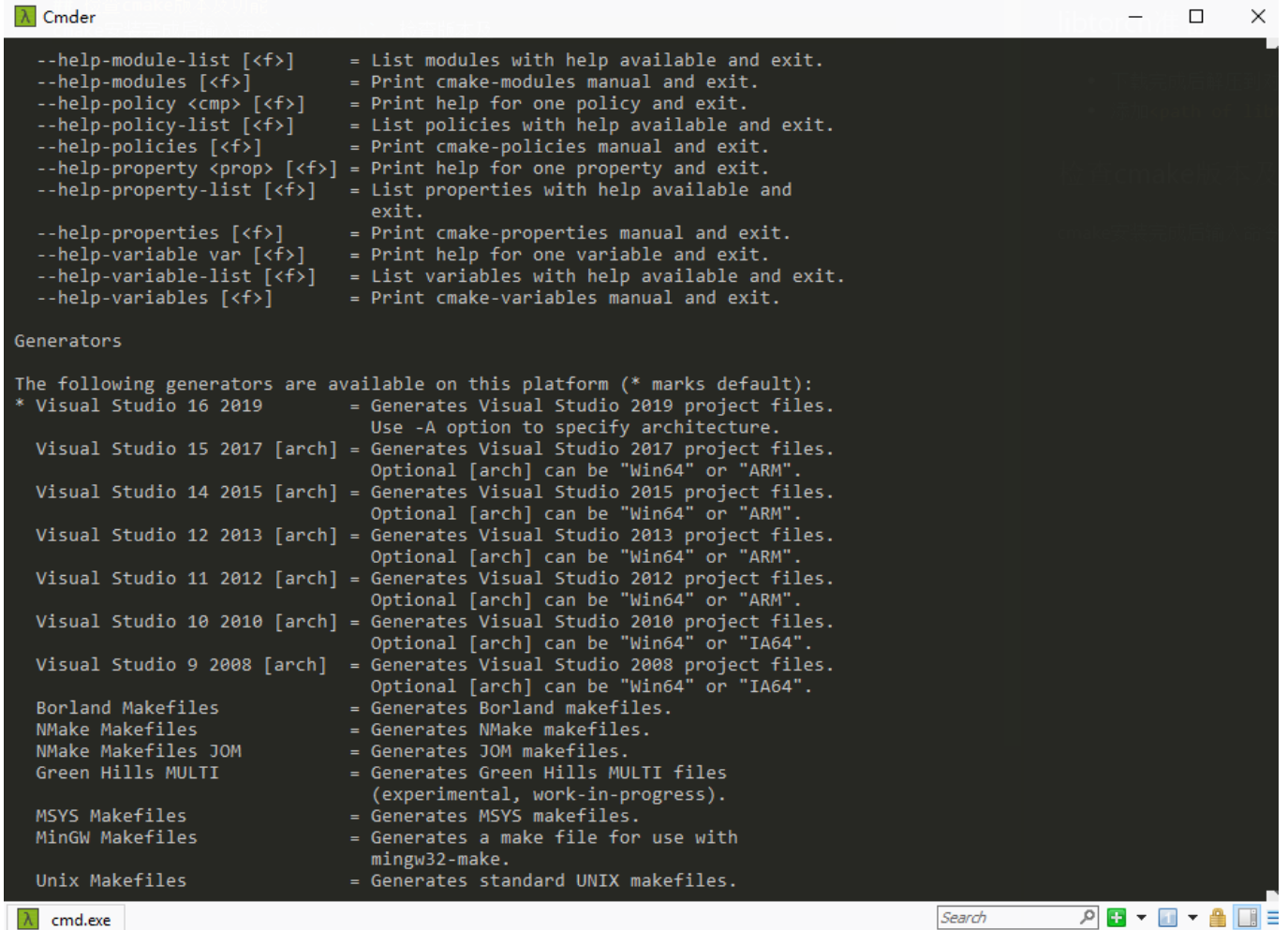
- **libtorch**: 下载windows下libtorch的nightly build, stable版本头文件引用有问题, 推荐下载cpu版本, 因为gpu版本环境配置比较复杂
- **visual studio**: 本教程使用的是vs2019, 但其他版本使用方法应没有太大差别
- **cmake**: vs2015不自带cmake工具, 需要自行去官网下载, 下载好加入系统PATH

libtorch准备

- 下载完成后解压到对应目录, 如C:\libtorch
- 添加<libtorch下载目录>\lib至系统PATH

检查cmake版本及功能

cmake安装完成后输入命令`cmake -h`, 检查generator选项, 样例输出如下图



```

--help-module-list [<f>]      = List modules with help available and exit.
--help-modules [<f>]         = Print cmake-modules manual and exit.
--help-policy <cmp> [<f>]    = Print help for one policy and exit.
--help-policy-list [<f>]     = List policies with help available and exit.
--help-policies [<f>]       = Print cmake-policies manual and exit.
--help-property <prop> [<f>] = Print help for one property and exit.
--help-property-list [<f>]   = List properties with help available and
                             exit.
--help-properties [<f>]     = Print cmake-properties manual and exit.
--help-variable var [<f>]    = Print help for one variable and exit.
--help-variable-list [<f>]  = List variables with help available and exit.
--help-variables [<f>]      = Print cmake-variables manual and exit.

Generators

The following generators are available on this platform (* marks default):
* Visual Studio 16 2019      = Generates Visual Studio 2019 project files.
                             Use -A option to specify architecture.
  Visual Studio 15 2017 [arch] = Generates Visual Studio 2017 project files.
                             Optional [arch] can be "Win64" or "ARM".
  Visual Studio 14 2015 [arch] = Generates Visual Studio 2015 project files.
                             Optional [arch] can be "Win64" or "ARM".
  Visual Studio 12 2013 [arch] = Generates Visual Studio 2013 project files.
                             Optional [arch] can be "Win64" or "ARM".
  Visual Studio 11 2012 [arch] = Generates Visual Studio 2012 project files.
                             Optional [arch] can be "Win64" or "ARM".
  Visual Studio 10 2010 [arch] = Generates Visual Studio 2010 project files.
                             Optional [arch] can be "Win64" or "IA64".
  Visual Studio 9 2008 [arch]  = Generates Visual Studio 2008 project files.
                             Optional [arch] can be "Win64" or "IA64".
  Borland Makefiles           = Generates Borland makefiles.
  NMake Makefiles             = Generates NMake makefiles.
  NMake Makefiles JOM         = Generates JOM makefiles.
  Green Hills MULTI           = Generates Green Hills MULTI files
                             (experimental, work-in-progress).
  MSYS Makefiles              = Generates MSYS makefiles.
  MinGW Makefiles             = Generates a make file for use with
                             mingw32-make.
  Unix Makefiles              = Generates standard UNIX makefiles.

```

项目准备

若已安装vs2017及以上版本可以直接创建cmake项目, 若没有则手动编写CMakeLists.txt文件, 示例文件如下

```

cmake_minimum_required (VERSION 3.8)

set(CMAKE_CXX_STANDARD 11)

find_package(Torch REQUIRED)

add_executable (CMakeProject "CMakeProject.cpp" "CMakeProject.h")

target_link_libraries(CMakeProject ${TORCH_LIBRARIES})

target_include_directories(CMakeProject PUBLIC TORCH_INCLUDE_DIRS)

```

进入项目目录, 执行命令 `cmake -DCMAKE_PREFIX_PATH=<libtorch下载目录> -DCMAKE_BUILD_TYPE=Debug|Release -G"<项目类型>" .`

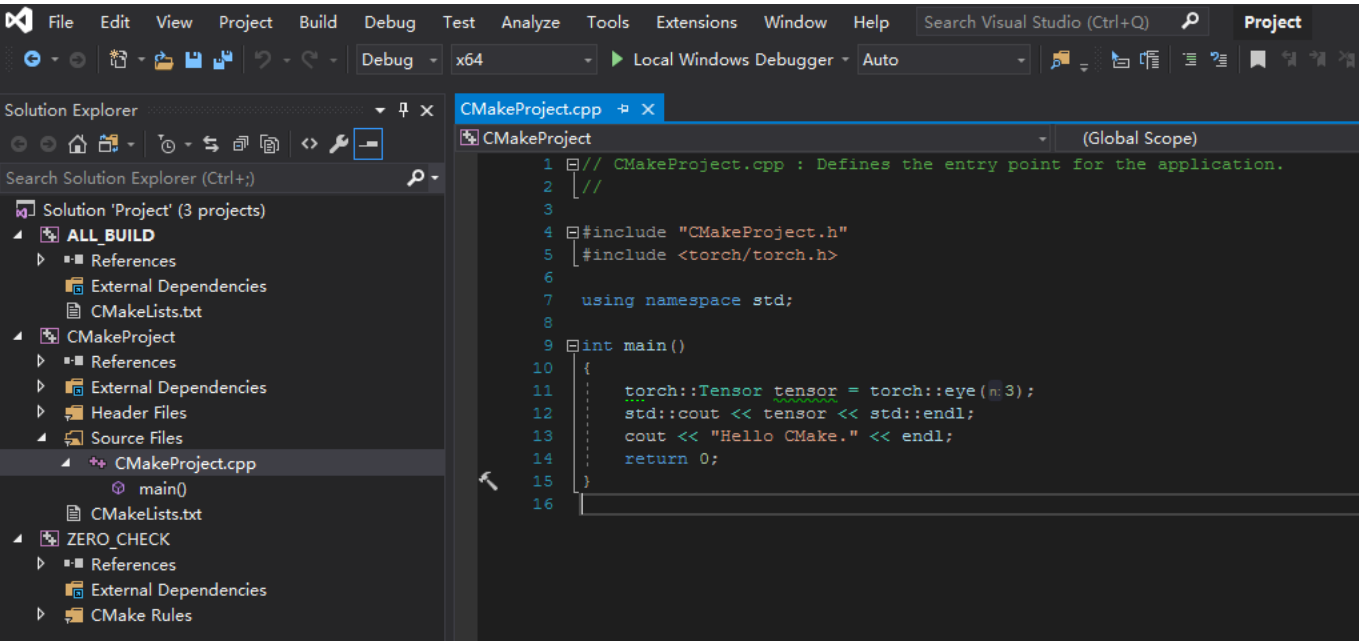
如对于libtorch安装在C:\libtorch目录下, 要生成vs2019项目, 项目类型为Debug的情况执行 `cmake -DCMAKE_PREFIX_PATH=E:\libtorch -DCMAKE_BUILD_TYPE=Debug -G"Visual Studio 16 2019" .`

若上述操作无误项目文件夹应该如下图, Project.sln为生成项目

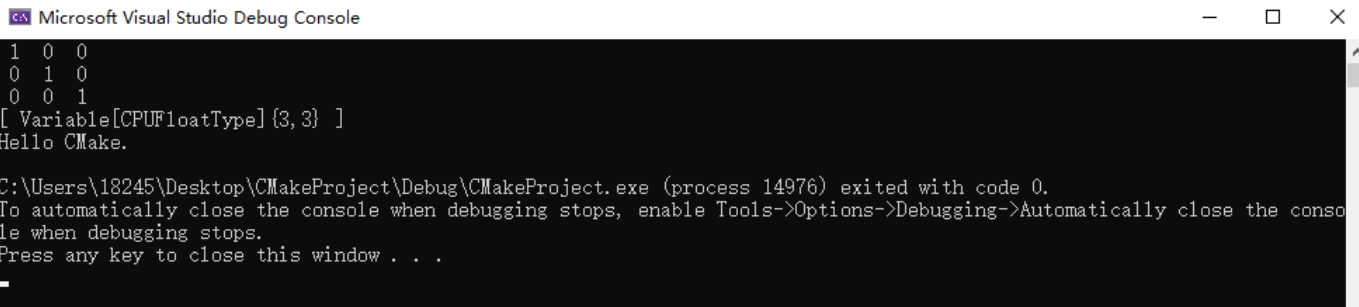
名称	修改日期	类型	大小
CMakeFiles	2019/4/5 20:46	文件夹	
ALL_BUILD.vcxproj	2019/4/5 20:46	VC++ Project	45 KB
ALL_BUILD.vcxproj.filters	2019/4/5 20:46	VC++ Project Fil...	1 KB
cmake_install.cmake	2019/4/5 20:46	CMAKE 文件	2 KB
CMakeCache.txt	2019/4/5 20:46	文本文档	15 KB
CMakeLists.txt	2019/4/4 20:59	文本文档	1 KB
CMakeProject.cpp	2019/4/5 9:28	C++ 源文件	1 KB
CMakeProject.h	2019/4/4 19:42	C Header 源文件	1 KB
CMakeProject.vcxproj	2019/4/5 20:46	VC++ Project	57 KB
CMakeProject.vcxproj.filters	2019/4/5 20:46	VC++ Project Fil...	1 KB
Project.sln	2019/4/5 20:46	Visual Studio Sol...	4 KB
ZERO_CHECK.vcxproj	2019/4/5 20:46	VC++ Project	44 KB
ZERO_CHECK.vcxproj.filters	2019/4/5 20:46	VC++ Project Fil...	1 KB

项目说明

Project.sln共包含三个项目: ALL_BUILD, CMakeProject, ZERO_CHECK, 其中CMakeProject是我们想要的



CMakeProject下的External Dependencies里包含了所有与libtorch有关的头文件, 若无错误就可以直接编译运行了



运行成功!

在HM中调用代码

由于现阶段libtorch调用必须通过cmake执行, 我在github上提供了一个HM16.20的cmake项目, 大家可以在对应位置修改CMakeLists.txt来在指定项目调用libtorch

- 在C++中加载pytorch模型教程https://pytorch.org/tutorials/advanced/cpp_export.html
- HM cmake工程https://github.com/chjz1024/HM_vc2015

注意: vs2017及以上版本虽然支持直接编译cmake项目, 但由于未知问题总会出现unresolved external symbol "__declspec(dllimport)"错误, 推荐使用上述生成sln项目方法