

# CMakeList -- Compile Format

## ForeWord

此文档的编写目的是记录在学习 CMake 的安装使用，以及 CMakeList.txt 编译文档的编写中的心得体会和知识总结

以备将来在有需要的时候能够查询 同时也可在编写的过程中作为参考手册使用，以辅助编写，提高效率 文中 "[ ]" 表示指定位置替换为对应格式字符串 本文档会不定时更新，同时有不足或者错漏请各位大佬指正

## Introduce

任何一个软件项目，除了写代码之外，还要考虑如何组织和管理这些代码，使项目代码层次结构清晰易读，这对以后的维护工作大有裨益

同时决定代码的组织方式及其编译方式，也是程序设计的一部分

因此，我们需要 CMake 和 AutoTools 这样的工具来帮助我们构建并维护项目代码

以下列出 CMake 的一些主要特点（选择 CMake 的理由）：

1. 跨平台

（这我个人认为最重要，基本支持市面上所有主流操作系统 — Windows, Linux, MacOS）

2. 开放源代码,使用类 BSD 许可发布

3. 能够管理大型项目

4. 简化编译构建过程和编译过程。 Cmake 的工具链非常简单：cmake + make

5. 可扩展,可以为 cmake 编写特定功能的模块,扩充 cmake 功能

再附上 CMake 官方网站，[点此跳转](#)

## Composition

## Content

### Variable

CMake 中所有的变量都是 string 类型

### Single-Valued Variable

声明（设置）变量

```
set( var-name value )
```

移除变量

```
unset( var-name )
```

引用变量

```
${ var-name }
```

打印变量

```
message = (" var-name = ${ var-name } ")
```

## Multi-Valued Variable (List)

声明（设置）列表

```
set( list-name value-1 ... value-N ) set( list-name "value-1; ...  
;value-N" )
```

移除列表

```
unset( list-name )
```

引用列表

```
${ list-name }
```

打印列表

```
message = (" list-name = ${ list-name } ")
```

## Basic Format

```
command( argument-1 ... ) 多个参数用空格分隔
```

## Command（按照字典序排序）

```
# [message-string] --- 注释内容
```

```
add_library( <name> [mode] [EXCLUDE-FROM-ALL] source-1 ... source-N )
```

参数说明：

<name>: 库文件名字

[mode]: **STATIC**（静态库） / **SHARED**（共享动态库）

[EXCLUDE-FROM-ALL]:

如果指定属性，对应的属性会在目标被创建时被设置（是否从默认构建中排除，包括子目录） **source-N**: 被添加入库的源文件

-----

```
aux_source_directory( . [var-name] ) --- 查找源文件并保存到相应的变量中
```

```
cmake_minimum_required( VERSION [number] ) --- 指定最低 CMake 版本
```

```
set() --- 声明设置变量值（详情见 Variable 内容）
```

```
unset() --- 移除变量（详情见 Variable 内容）
```