<https://www.cnblogs.com/coderfenghc/archive/2012/06/16/CMake_ch_01.html>

 公司的一个项目使用CMake作为跨平台构建工具；业务有需求，当然要好好研读一下官方的技术手册。目前的计划是先把官方手册翻译一下，了解清楚CMake中的各种命令、属性和变量的用法。同时在工作中也会阅读CMake的真实源码，后续会基于此陆续写一些工程中使用CMake的心得。CMake的版本也在不停更新，有些新的命令和变量会随着版本更新添加进来，这是后事了，暂且不管；现在锁定CMake 2.8.3作为手册翻译的版本。

      作为园子里的新丁，文章在术语和表达等等方面会有欠缺的地方，还请大侠们慷慨指点。另外，罗马不是一天建成的，长长的手册翻译完也不知道要经历多少日升月落；不过还是希望自己能够坚持下去:-)。

**CMake2.8.3 主索引**

* **命令名称**
* **用法**
* **描述**
* **命令选项**
* **生成器**
* **命令**
* **属性**
* **全局域属性**
* **目录属性**
* **目标属性**
* **测试属性**
* **源代码属性**
* **Cache Entries属性**
* **兼容性命令**
* **CMake 标准模块**
* **CMake策略**
* **变量**
* **改变行为的变量**
* **描述系统的变量**
* **语言变量**
* **控制构建的变量**
* **提供信息的变量**
* **版权**
* **其他参考资料**

**命令名称**

     cmake - 跨平台Makefile生成工具。

**用法**

　　cmake [选项] <源码路径>

　　cmake [选项] <现有构建路径>

**描述**

　　cmake可执行程序是CMake的命令行界面。它可以用脚本对工程进行配置。工程配置设置可以在命令行中使用-D选项指定。使用-i选项，cmake将通过提示交互式地完成该设置。

　　CMake是一个跨平台的构建系统生成工具。它使用平台无关的CMake清单文件CMakeLists.txt，指定工程的构建过程；源码树的每个路径下都有这个文件。CMake产生一个适用于具体平台的构建系统，用户使用这个系统构建自己的工程。

**选项**

-C <initial-cache>: 预加载一个脚本填充缓存文件。  
　　当cmake在一个空的构建树上第一次运行时，它会创建一个CMakeCache.txt文件，然后向其中写入可定制的项目设置数据。-C选项可以用来指定一个文件，在第一次解析这个工程的cmake清单文件时，从这个文件加载缓存的条目(cache entries)信息。被加载的缓存条目比项目默认的值有更高的优先权。参数中给定的那个文件应该是一个CMake脚本，其中包含有使用CACHE选项的SET命令；而不是一个缓存格式的文件。

-D <var>:<type>=<value>: 创建一个CMake的缓存条目。

　　当cmake第一次运行于一个空的构建数时，它会创建一个CMakeCache.txt文件，并且使用可定制的工程设置来填充这个文件。这个选项可以用来指定优先级高于工程的默认值的工程设置值。这个参数可以被重复多次，用来填充所需要数量的缓存条目(cache entries)。

-U <globbing\_expr>: 从CMake的缓存文件中删除一条匹配的条目。  
　　该选项可以用来删除CMakeCache.txt文件中的一或多个变量。文件名匹配表达式(globbing expression)支持通配符\*和?的使用。该选项可以重复多次以删除期望数量的缓存条目。使用它时要小心，你可能因此让自己的CMakeCache.txt罢工。

-G <generator-name>: 指定一个makefile生成工具。

　　在具体的平台上，CMake可以支持多个原生的构建系统。makefile生成工具的职责是生成特定的构建系统。可能的生成工具的名称将在生成工具一节给出。

-Wno-dev: 抑制开发者警告。

　　抑制那些为CMakeLists.txt文件的作者准备的警告信息。

-Wdev: 使能开发者警告信息输出功能。

　　允许那些为CMakeLists.txt文件的作者准备的警告信息。

-E: CMake命令行模式。  
　　为了真正做到与平台无关，CMake提供了一系列可以用于所有系统上的的命令。以-E参数运行CMake会帮助你获得这些命令的用法。可以使用的命令有：chdir, copy, copy\_if\_different copy\_directory, compare\_files, echo, echo\_append, environment, make\_directory, md5sum, remove\_directory, remove, tar, time, touch, touch\_nocreate, write\_regv, delete\_regv, comspec, create\_symlink。

-i: 以向导模式运行CMake。

　　向导模式是在没有GUI时，交互式地运行cmake的模式。cmake会弹出一系列的提示，要求用户回答关于工程配置的一行问题。这些答复会被用来设置cmake的缓存值。

-L[A][H]: 列出缓存的变量中的非高级的变量。  
　　-L选项会列出缓存变量会运行CMake，并列出所有CMake的内有被标记为INTERNAL或者ADVANCED的缓存变量。这会显示当前的CMake配置信息，然后你可以用-D选项改变这些选项。修改一些变量可能会引起更多的变量被创建出来。如果指定了A选项，那么命令也会显示高级变量。如果指定了H选项，那么命令会显示每个变量的帮助信息。

=========================================================================

好了，今天先翻到这里。开饭喽~~

<<<------------- 欢迎转载；转载请标明出处。 ------------->>>