

VC 中 cl.exe 命令参数详解

cl.exe 所在的文件夹里面有一个批处理叫做

VSVAR32.BAT

首先运行它一次，你就可以用 cl.exe 来编译你的代码了。

（在 vs2005 里面未发现该文件，怀疑因该是上级目录的 vcvarsall.bat，但是不执行该文件同样可进行命令行编译。）

CL.exe 是控制 Microsoft C 和 C++ 编译器与链接器的 32 位工具。编译器产生通用对象文件格式 (COFF) 对象 (.obj) 文件。链接器产生可执行文件 (.exe) 或动态链接库文件 (DLL)。

注意，所有编译器选项都区分大小写。

若要编译但不链接，请使用 /c。

使用 NMAKE 生成输出文件。

使用 BSCMAKE 支持类浏览。

以下是一个完整的编译器选项分类列表。

优化

选项 作用

/O1 创建小代码

/O2 创建快速代码

/Oa 假设没有别名

/Ob 控制内联展开

/Od 禁用优化

/Og 使用全局优化

/Oi 生成内部函数

/Op 改善浮点数一致性

/Os 代码大小优先

/Ot 代码速度优先

/Ow 假定在函数调用中使用别名

/Ox 使用最大优化 (/Ob1gity /Gs)

/Oy 省略框架指针

代码生成

选项 作用

/clr 启用 C++ 的托管扩展并产生在公共语言运行库上运行的输出文件

/EH 指定异常处理模型

/G3 优化代码以优选 386 处理器。在 Visual C++ 5.0 中已经停用，编译器将忽略此选项

/G4 优化代码以优选 486 处理器。在 Visual C++ 5.0 中已经停用，编译器将忽略此选项

/G5 优化代码以优选 Pentium

/GB 与 /G6 等效；将 _M_IX86 的值设置为 600

/Gd 使用 __cdecl 调用约定

/Ge 激活堆栈探测

/GF

/GF 启用字符串池

/GH 调用挂钩函数 _penter

/GH 调用挂钩函数 _pexit

/GL 启用全程序优化

/Gm 启用最小重新生成

/Gr 启用运行时类型信息 (RTTI)

/Gr 使用 __fastcall 调用约定

/GS 控制堆栈探测

/GT 支持使用静态线程本地存储区分配的数据的线程安全

/GX 启用同步异常处理

/Gy 启用函数级链接

/GZ 使用 __stdcall 调用约定

/MD 使用 MSVCRT.lib 创建多线程 DLL

/MDd 使用 MSVCRTD.lib 创建调试多线程 DLL

/ML 使用 LIBC.lib 创建单线程可执行文件

/MLd 使用 LIBCD.lib 创建调试单线程可执行文件

/MT 使用 LIBCMT.lib 创建多线程可执行文件

/MTd 使用 LIBCMTD.lib 创建调试多线程可执行文件

输出文件

选项 作用

/FA

/FA 创建汇编文件

设置列表文件名

/Fd 重命名程序数据库文件

/Fe 重命名可执行文件

/Fm 创建映射文件

/Fo 创建对象文件

/Fp 指定预编译头文件名

/FR

/FR 生成浏览器文件

/Fx 将插入的代码与源文件合并

调试

选项 作用

/GS 缓冲区安全检查

/GZ 与 /RTC1 相同

/RTC 启用运行时错误检查

/Wp64 检测 64 位可移植性问题

/Yd 将完整的调试信息放在所有对象文件中

/Yl 创建调试库时插入 PCH 引用

/Z7 生成与 C 7.0 兼容的调试信息

/Zd 生成行号

/Zi 生成完整的调试信息

预处理器

选项 作用

/AI 指定在解析传递到 #using 指令的文件引用时搜索的目录

/c 在预处理期间保留注释

/D 定义常数和宏

`/E` 将预处理器输出复制到标准输出
`/EP` 将预处理器输出复制到标准输出
`/F1` 预处理指定的包含文件
`/FU` 强制使用文件名，就像它已被传递到`#using` 指令一样
`/I` 在目录中搜索包含文件
`/P` 将预处理器输出写入文件
`/U` 移除预定义宏
`/U` 移除所有的预定义宏
`/X` 忽略标准包含目录
`/ZI` 将调试信息包含在与“编辑并继续”兼容的程序数据库中

语言

选项 作用

`/noBool` 取消 C++ `bool`、`true` 和 `false` 关键字
`/vd` 取消或启用隐藏的 `vtordisp` 类成员
`/vmb` 对指向成员的指针使用最佳的基
`/vmg` 对指向成员的指针使用完全一般性
`/vmm` 声明多重继承
`/vms` 声明单一继承
`/vmv` 声明虚拟继承
`/Za` 禁用语言扩展
`/Zc` 在 `/Ze` 下指定标准行为
`/Ze` 启用语言扩展
`/Zg` 生成函数原型
`/Zl` 从 `.obj` 文件中移除默认库名
`/Zp n` 封装结构成员
`/Zs` 只检查语法

链接

选项 作用

`/F` 设置堆栈大小
`/LD` 创建动态链接库
`/LDd` 创建调试动态链接库

/link 将指定的选项传递给 LINK
/MD 使用 MSVCRT.lib 编译以创建多线程 DLL
/MDd 使用 MSVCRTD.lib 编译以创建调试多线程 DLL
/ML 使用 LIBC.lib 编译以创建单线程可执行文件
/MLd 使用 LIBCD.lib 编译以创建调试单线程可执行文件
/MT 使用 LIBCMT.lib 编译以创建多线程可执行文件
/MTd 使用 LIBCMTD.lib 编译以创建调试多线程可执行文件

预编译头

选项 作用

/Y- 忽略当前生成中的所有其他预编译头编译器选项
/Yc 创建预编译头文件
/Yd 将完整的调试信息放在所有对象文件中
/Yu 在生成期间使用预编译头文件
/YX 自动处理预编译头

杂项

选项 作用

@ 指定响应文件
/? 列出编译器选项
/c 编译但不链接
/H 限制外部（公共）名称的长度
/HELP 列出编译器选项
/J 更改默认的 char 类型
/NOLOGO 取消显示登录版权标志
/QI0f 确保 Pentium 0F 指令没有问题
/QIfdiv FDIV、FPREM、FPTAN 和 FPATAN 指令有缺陷的 Intel Pentium 微处理器的变通方法
QIfist 当需要从浮点类型转换为整型时取消 Helper 函数 _ftol 的调用
/showIncludes 在编译期间显示所有包含文件的列表
/Tc
/Tc 指定 C 源文件
/Tp
/Tp 指定 C++ 源文件

`/V` 设置版本字符串
`/w` 设置警告等级
`/w` 禁用所有警告
`/Wall` 启用所有警告，包括默认情况下禁用的警告
`/WL` 在从命令行编译 C++ 源代码时启用错误信息和警告消息的单行诊断
`/Zm` 设置编译器的内存分配限制

CL 命令行使用下列语法：

```
CL [option...] file... [option | file]... [lib...] [@command-file]  
[/link link-opt...]
```

下表说明 CL 命令的输入项意义

option 一个或多个 CL 选项。请注意，所有选项都应用于所有指定的源文件。选项是由一个正斜杠 (/) 或一个短划线 (-) 指定的。如果某个选项带有参数，则该选项的说明指定在选项和参数之间是否允许有空格。选项名 (/HELP 选项除外) 区分大小写。有关更多信息，请参阅 CL 选项的顺序。

file 一个或多个源文件、.obj 文件或库的名称。CL 编译源文件并将 .obj 文件和库的名称传递给链接器。有关更多信息，请参阅 CL 文件名语法。

lib 一个或多个库名。CL 将这些名称传递给链接器。

command-file 包含多个选项和文件名的文件。有关更多信息，请参阅 CL 命令文件。

link-opt 一个或多个链接器选项。CL 将这些选项传递给链接器。

您可以指定任意数目的选项、文件名和库名，条件是命令行上的字符数不超过 1024，该限制是操作系统指定的。

CL 命令文件请参见

设置编译器选项 | 编译器选项

命令文件是一个文本文件，它包含您另外在命令行上键入或使用 CL 环境变量指定的选项和文件名。CL 接受在 CL 环境变量中或命令行上用作参数的编译器命令文件。与命令行或 CL 环境变量不同，命令文件允许使用多行选项和文件名。

命令文件中的选项和文件名将根据 CL 环境变量中或命令行上的命令文件名的位置被进行处理。但是，如果 /link 选项出现在命令文件中，则该行其余部分的所有选项将被传递给链接器。命令文件的后面几行中的选项和命令行上命令文件调用之后的选项仍被作为编译器选项接受。

命令文件一定不能包含 CL 命令。每个选项必须在同一行上开始和结束；不能使用反斜杠（\）跨行组合一个选项。

命令文件用一个 @ 符后接一个文件名指定；该文件名可指定绝对路径或相对路径。