

适用于 IBM PC AT 286. 386. 486

3.0

Microsoft Windows

最新窗口程序设计工具

(SDK)

夏洪山 等编译

本书介绍示范的程序
设计步骤和开发过程，以
及开发工具的使用等等。

中国科学院希冀高级电脑技术公司

Microsoft Windows V3.0

最新窗口程序设计工具 (SDK)

夏洪山 等译

北京希望电脑公司

一九九一年十月

译 序

Microsoft Windows及其软件开发工具3.0版的推出,标志着PC机系统软件的发展进入了一个新的阶段,为开发从用户界面到系统特性,运行方式和执行效率诸方面都与DOS完全不同的新一代应用软件,提供了理想的环境和有效的开发手段。

Microsoft Windows 3.0软件开发工具系列技术手册包括《多窗口程序设计工具》、《多窗口程序设计指南》和《多窗口程序设计参考手册》。在本书中,详细而准确地介绍了有关Microsoft Windows软件开发工具包中提供的最新开发信息和最新开发手段。本书中示范的程序设计步骤和开发过程,以及开发工具的使用等等,在工具包中均有。对于不打算开发Windows应用程序的读者来说,读一读这套系列书,可以了解到当今微型计算机软件发展的现状和趋势。对于那些悉心钻研应用程序特性,勇于创新开拓的软件工作者来说,尤其是善长使用图象图形技术开发应用软件的程序设计人员,使用Windows 3.0软件开发工具开发图文并茂,色彩斑驳的新颖软件,这套系列手册都是必不可少、案头常驻的工具之一。

参加本书翻译和校对工作的主要人员有:夏洪山,仇钟,林志坚、丁丽亚、李纪敏和宋震等。中科院希望电脑公司对本书的印刷进行了周密的安排和组织,值此一并致谢。

译者

一九九〇年十二月

《Microsoft Windows多窗口程序设计工具》介绍怎样使用Microsoft Windows软件开发工具(SDK)提供的编程工具。本手册还将介绍生成Windows应用程序必须使用的其它一些工具,如C编译器和链接程序,它们不包括在SDK之中。

本章概要性地介绍以下内容:

- 《Windows程序设计工具》的一般组织
- 生成Windows应用程序的主要步骤
- 本书中使用的符号规则
- 有关的参考书

本书的组织

本书分成四部分,每一部分包括若干章。

第1部分“编译器和链接程序”,介绍怎样编译和链接源程序文件。第1部分包括以下几章:

■第1章“编译应用程序:C编译器”,介绍怎样使用C编译器,编译Windows应用程序的C语言源程序文件。

■第2章“链接应用程序:链接程序”介绍怎样把源文件链接或可执行的Windows应用程序。

■第3章“编译资源:资源编译器”介绍怎样使用资源编译器(RC),编译应用程序的资源,并把它们加到可执行的Windows应用程序中去。

第2部分“资源编辑器”,介绍怎样使用SDK提供的工具,创建和维护Windows程序的资源,如标象共位图等。第2部分包括以下各章:

■第4章“设计图案:SDKPaint”,介绍怎样使用SDKPaint (SDKPAINT)创建和编辑Windows应用程序的标象、光标和位图。

■第5章“设计会话框:会话编辑器”,介绍怎样使用会话编辑器(DIALOG),创建和编辑Windows应用程序的会话框。

■第6章“设计字体:字体编辑器”,介绍怎样使用字体编辑器(FONTEDIT),创建和编辑Windows应用程序的字体文件。

第3部分“调试和优化工具”,说明怎样使用SDK提供的调试和测试工具。第3部分包括以下各章:

■第7章“保护模式下调试:Windows的CodeView”,介绍怎样在保护模式下,使用Windows的CodeView (CVW)调试Windows应用程序。

■第8章“实模式下调试:符号调试器”,说明怎样在实模式下,使用符号调试器(SYMBOL)调试Windows应用程序。

■第9章“保护方式下的高级调试:80386调试器”,介绍怎样在保护模式下,使用80386调试器调试Windows应用程序。

■第10章“监督信息：SPY”，介绍怎样使用SPY监督窗口收到的系统信息。

■第11章“观察应用程序数据堆：Heap Walker”，介绍怎样使用Heap Walker来打开和检查全局堆。

■第12章“移动内存：Shaker”，介绍怎样使用Shaker来查看应用程序内存移动的效果。

■第13章“分析CPU时间：Profiler”，介绍怎样用Profiler来分析和优化可移动代码的效率。

■第14章“磁盘转贮区分析：SWAP”，介绍怎样使用Swap来分析和优化应用程序磁盘转贮操作。

第4部分“提示工具”，介绍怎样计划、编写和编译一个Windows提示系统，第4部分包括以下各章：

■第15章“提供帮助：Help系统”，以用户、Help编写者和Help程序员的观点，说明Help系统的全貌。

■第16章“规划Help系统”，说明Help编写者在计划一个Help系统时，应注意哪些方面的问题。

■第17章“创建Help主题文件”，介绍怎样编写未编码Help正文文件。

■第18章“建立提示文件”，介绍怎样建立Help资源文件。

■第19章“提示举例及编译出错信息”，列出几个字处理程序的主题文件的例子，以及它们对应的Help显示。这一章中还列出了编译器错误信息提示。

建立Windows应用程序

你可以用任意的ASCII正文编辑程序和本手册中介绍的软件工具，建立Windows应用程序。这一节简要说明建立Windows应用程序的过程，并且揭示开发工具在此过程中所起的作用。

为了建立Windows应用程序，按以下步骤做：

1. 使用正文编辑程序，建立含有WinMain函数、窗口函数和其它执行代码的C语言或汇编语言源程序文件。

2. 使用资源编辑器（SDKPAINT，DIALOG和FONTEDIT）建立应用程序需要的光标、标象、位图和字体资源。

3. 生成包含应用程序所有资源的资源描述（.RC）文件。这个利用正文编辑程序建立的描述文件，列出并命名前一步骤建立的资源。它还可以定义清单、会话框和其它资源，例如字符串表 and 应用程序定义的资源。

4. 使用资源编辑器，以-r选择项把资源描述文件编译成二进制资源（.RES）文件。

5. 使用正文编辑程序建立模块定义（.DEF）文件。

6. 使用C语言编译器编译所有C语言源程序。使用Microsoft宏汇编（MASM）汇编所有汇编语言源程序。

7. 使用LINK，把经过汇编或编译的源程序与C运行时库进行链接，产生一个.EXE扩展名的文件。这个文件不能执行，因为它还没有包含已编译的资源。

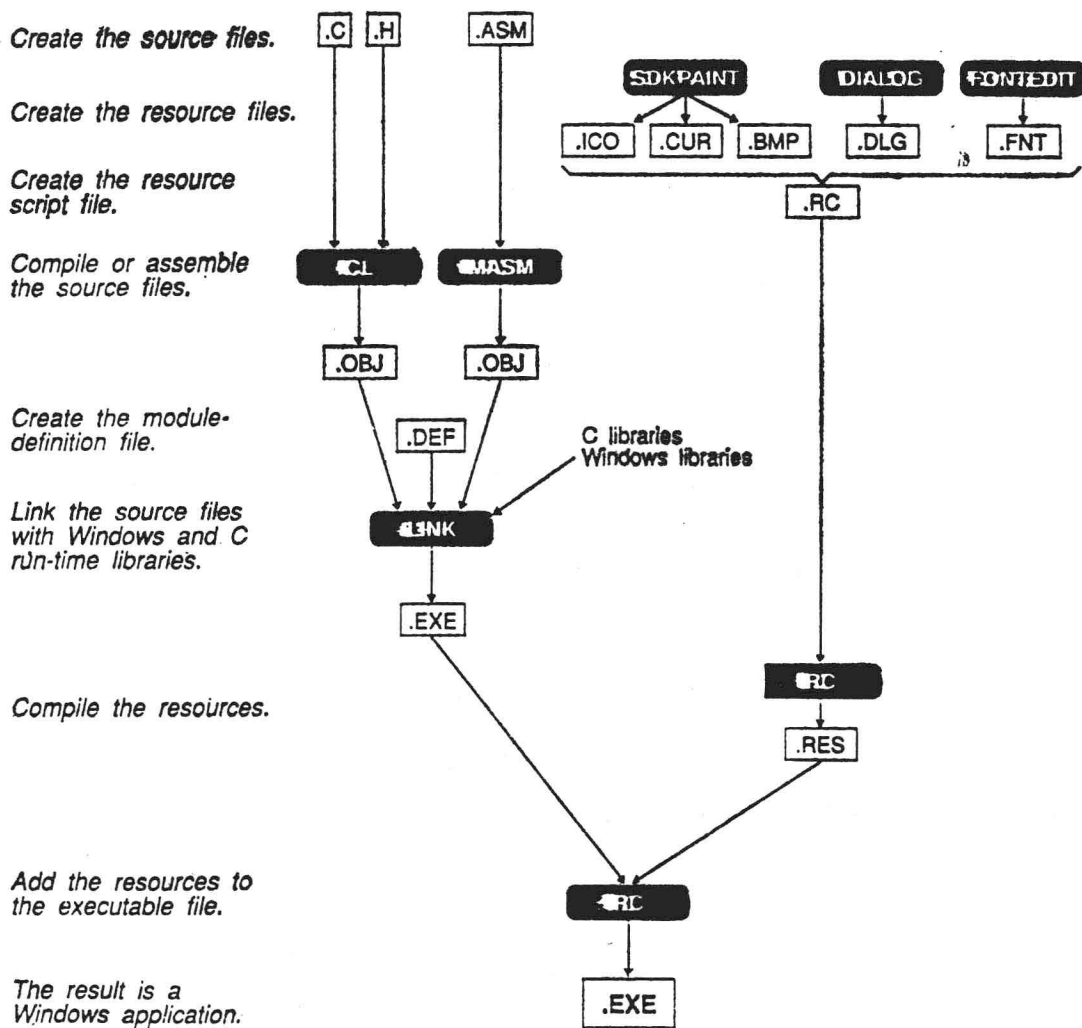
8. 使用带有一r选择项的RC程序，把二进制资源（.RES）文件加到.EXE文件，产生一个可执行的Windows应用程序。

9. 使用Windows调试器：Windows的CodeView和符号调试器，去掉程序中的错误和有疑问之处。Spy程序可以用来监测应用程序接收到的Windows信息。Shake程序模拟Windows多任务环境下可能发生的内存移动。

10. 使用Windows优化工具：Profiler和Swap，改善程序的性能，使它运行得更快而且更有效地使用内存。

11. 使用Windows帮助工具，建立应用程序的帮助系统。这一步骤可以穿插在应用程序开发过程中，而不必等到开发过程完了一后。

下图画出了建立Windows程序的各个步骤：



建立Windows应用程序

在这张图上没有画出调试、优化和帮助工具。

符号约定

在本手册中，除非特别指明，“DOS”项是指MS_DOS和PC_DOS。

以下是本手册中的符号约定。

约定	说 明
黑体正文	黑体正文表示必须逐字如实使用的特定项或标点符号：语言中的关键字或函数（如EXETYPE或CreateWindow），DOS命令，以及命令行选择项（如/zi）。你必须准确地按所示的正文送入项和标点符号。然而，并不需要区分大小写。例如，可以在DOS提示符之后送入LINK，link或Link来启动链接程序。
()	在语法说明中，括号内为传递给函数的一个或几个参数。
斜体正文	斜体正文表示一个替代符，需要你提供实际值。例如，SetCursorpos 函数的语法：SetCursorPos (X,Y)，指示你必须替换X和Y坐标的值。
等线体	代码例子用笔划粗细一样的字体印出。
BEGIN	程序例子中的垂直省略号表示省略了这部分程序。
:	
END	
...	一个项后面的水平省略号表示多个同种形式的项可能存在。在下例中，水平省略号表示你可以在g命令中指定多个breakaddress值： g [(=startaddress)][(breakaddress)]...
[()]	双方括号内为命令行或语法说明的可选择域或参数。在下例中，option和executable-file是RC命令的可选择参数： PC [(aoption)]filename[(executable-file)]
	竖线表示你可以输入它两边的项中的一个。以下的命令行语法说明了竖线的用法： DB [(address range)] 竖线表明你可以在DB(Dump Bytes) 命令之后，指定address或range。
" "	双引号表示正文形式的项。
{ }	大括号表示你可以指定其中一项。
小的大写字母	小的大写字母表示键或键序列的名称，例如：ALT+SPACEBAR。

Microsoft Windows软件开发工具资料集

在这个资料集中，“SDK”特指Microsoft Windows软件开发工具及其内容。SDK包括下列手册：

标题	内容
《安装和更新指南》	提供SDK的设计方向，说明怎样安装SDK软件，着重说明3.0版的改进。
《编程指南》	说明怎样编写Windows应用程序，并提供一些应用程序的例子，作为用户编写应用程序的参考。《编程指南》还涉及一些编写高级Windows程序的技术。
《编程工具》	说明怎样使用诸如调试器和特定的SDK 编辑器这样一些软件开发工具，建立用户的Windows应用程序。

《参考手册》

对Microsoft Windows应用程序编程界面 (API) 作了广泛详尽的描述,《参考手册》按照字母表顺列出当前 API 所有的函数、信息和数据结构,并且进一步说明怎样使用API,

《普通用户可涉及的系统应用程序结构: 高级界面设计指南》

提供编程指导和规范格式,以保持所编写的程序与其它 Microsoft Windows应用程序的一致性。

小结

这一章概要地介绍了《工具》的组织,并简要说明建立Windows应用程序的步骤和所用的工具。

有关建立Windows应用程序的其它介绍,参见《程序设计指南》。

目 录

引言.....	(1)
本书的组织.....	(1)
建立 Windows 应用程序.....	(2)
符号约定.....	(3)
小结.....	(5)

第一部分 编译器和链接程序

第一章 编译应用程序：C 编译器.....	(1)
1.1 编译C语言写的 Windows 应用程序.....	(1)
1.2 编译器选择项.....	(3)
1.2.1 内存模式选择项.....	(3)
1.2.2 开发应用程序选择项.....	(3)
1.2.3 动态链接库选择项.....	(4)
1.3 小结.....	(4)
第二章 链接应用程序：链接程序.....	(5)
2.1 创建模块定义文件.....	(5)
2.1.1 创建应用程序的模块定义.....	(6)
2.1.2 创建库的模块定义.....	(7)
2.2 引进动态链接库.....	(8)
2.3 链接一个应用程序.....	(8)
2.3.1 使用LINK命令.....	(8)
2.3.2 指定LINK命令选择项.....	(9)
2.3.3 在LINK命令行中指定库.....	(11)
2.4 检查可执行文件头.....	(12)
2.5 小结.....	(12)
第三章 编译资源：资源编译器.....	(13)
3.1 在应用程序中引入资源.....	(13)
3.2 创建资源描述文件.....	(13)
3.3 使用资源编译器.....	(15)
3.3.1 分别编译资源.....	(17)
3.3.2 定义预处理使用的名.....	(18)
3.3.3 更换已编译资源文件名.....	(18)
3.3.4 控制 RC 查找的路径.....	(18)

3.3.5 显示进展信息.....	(19)
3.4 小结.....	(19)

第二部分 资源编辑器

第四章 设计图案：SDKPaint.....	(20)
4.1 SDKPaint怎样处理文件.....	(20)
4.1.1 文件类型.....	(20)
4.1.2 标象和光标数据：SDKPAINT.DAT 文件.....	(21)
4.2 SDKPaint窗口.....	(22)
4.3 打开文件和图案.....	(23)
4.3.1 把文件转换成 3.0 格式.....	(23)
4.3.2 打开位图.....	(23)
4.3.3 打开标象和光标.....	(24)
4.4 用SDKPaint工具.....	(25)
4.5 使用SDKPaint调色板.....	(25)
4.5.1 处理不透明色、屏幕色相反.....	(26)
4.6 自定义调色板.....	(27)
4.6.1 编辑颜色.....	(27)
4.6.2 保存调色板.....	(28)
4.6.3 装入自定义调色板.....	(28)
4.7 定义光标热点.....	(28)
4.8 使用数据传递区.....	(28)
4.9 使用ZoomIn检查图案.....	(29)
4.10 小结.....	(29)
第五章 设计会话框：会话框编辑器.....	(29)
5.1 会话框编辑器怎样处理文件.....	(30)
5.1.1 会话框描述.....	(31)
5.1.2 资源文件.....	(32)
5.1.3 包含文件.....	(32)
5.2 安装和撤除用户控制框.....	(33)
5.2.1 安装用户控制框.....	(33)
5.2.2 撤消用户控制框.....	(33)
5.3 观察会话框：会话框编辑器窗口.....	(33)
5.3.1 模式显示.....	(35)
5.3.2 工具箱.....	(35)
5.3.3 选择项状态窗口.....	(35)
5.4 打开文件和会话框.....	(36)
5.4.1 打开资源文件.....	(36)

5.4.2 打开包含文件.....	(36)
5.4.3 打开会话框.....	(36)
5.5 编辑会话框中的控制框.....	(36)
5.5.1 增加控制框.....	(38)
5.5.2 处理独立的控制框.....	(38)
5.6 处理控制框组.....	(40)
5.6.1 移动控制框组.....	(41)
5.6.2 定义输入光标序列.....	(41)
5.7 处理会话框.....	(42)
5.7.1 重设会话框的大小.....	(42)
5.7.2 会话框改名.....	(43)
5.7.3 定义格式.....	(43)
5.7.4 设置内存标志.....	(43)
5.7.5 放弃编辑.....	(43)
5.8 在资源之间移动会话框.....	(43)
5.9 处理包含文件.....	(44)
5.9.1 创建新的包含文件.....	(44)
5.9.2 装入存在的包含文件.....	(44)
5.9.3 编辑包含文件.....	(45)
5.9.4 保存包含文件.....	(45)
5.10 小结	(45)
第六章 设计字体：字体编辑器.....	(46)
6.1 打开一种字体.....	(46)
6.2 编辑字符.....	(47)
6.2.1 把象素置亮或置暗.....	(47)
6.2.2 改变象素行或列.....	(47)
6.2.3 移动象素块.....	(49)
6.2.4 改变字符宽度.....	(49)
6.2.5 保存对字符的修改.....	(50)
6.2.6 取消对字符的修改.....	(50)
6.3 编辑字体.....	(50)
6.4 改变字体文件头信息.....	(51)
6.5 小结.....	(52)

第三部分 调试工具和优化选择

第七章 保护方式下的调试：Windows CodeView.....	(53)
7.1 使用要求.....	(54)
7.2 CVM与其他Microsoft调试器的比较.....	(54)

7.3 准备进行 CVW.....	(55)
7.4 启动调试器.....	(57)
7.5 存贮画面信息.....	(60)
7.6 CVW 屏幕操作.....	(61)
7.7 CVW 联机提示.....	(64)
7.8 显示程序数据.....	(64)
7.9 修改程序数据.....	(72)
7.10 控制程序执行.....	(72)
7.11 应用程序异常终止的处理.....	(76)
7.12 结束 CVW 画面.....	(77)
7.13 重新启动 CVM 调试画面.....	(77)
7.14 CVW 高级技术.....	(78)
7.15 TOOLS.INI 文件关于 CVW 的说明.....	(79)
7.16 CVW 中的操作举例.....	(79)
7.17 小结.....	(82)
第八章 实地址方式下的调试: SYMDEB.....	(84)
8.1 准备符号文件.....	(84)
8.2 安装调试终端.....	(86)
8.3 启动 Windows 的 SYMDEB.....	(86)
8.4 使用符号映象.....	(89)
8.5 启动应用程序.....	(90)
8.6 显示分配信息.....	(91)
8.7 退出 SYMDEB.....	(92)
8.8 SYMDEB 命令一览表.....	(92)
8.9 SYMDEB 命令说明.....	(97)
第九章 保护方式下的高级调试: 80386 调试器.....	(110)
9.1 准备 80386 调试器的符号文件.....	(110)
9.2 启动调试器.....	(110)
9.3 应用程序的失败.....	(111)
9.4 调试器命令格式.....	(112)
9.5 常用命令目录.....	(114)
第十章 监督信息: SPY.....	(142)
10.1 显示信息.....	(142)
10.2 选择项.....	(143)
10.3 选择窗口: 窗口清单.....	(143)
10.4 打开或关闭 SPY: SPY 清单.....	(144)
10.5 小结.....	(144)
第十一章 观察应用程序数据堆: HeapWalker.....	(145)
11.1 Heap Walker 如何查看内存.....	(145)

11.2 Heap Walker窗口.....	(145)
11.3 使用Heap Walker命令.....	(146)
11.4 对使用Heap Walker的几点建议.....	(149)
11.5 小结	(150)
第十二章 移动内存: Shaker	(150)
12.1 Shaker用法.....	(150)
12.2 小结	(151)
第十三章 分析CPU时间: Profiler.....	(151)
13.1 Profiler一览	(151)
13.2 准备运行 Profiler.....	(152)
13.3 使用 Profiler 函数.....	(152)
13.4 样本代码.....	(156)
13.5 显示样本: SHOWHITS.EXE	(157)
13.6 小结	(158)
第十四章 磁盘转贮区分析: Swap.....	(159)
14.1 准备运行Swap.....	(159)
14.2 运行Swap.....	(160)
14.3 显示输出.....	(160)
14.4 小结	(161)
第十五章 提供帮助: Help系统.....	(162)
15.1 创建一个Help系统: 开发阶段.....	(162)
15.2 如何对用户显示Help.....	(163)
15.3 如何对Help的作者显示.....	(164)
15.4 如何对Help 编程者显示.....	(165)
15.5 小结	(165)
第十六章 筹划Help系统	
16.1 准备方案.....	(165)
16.2 决定主题文件结构.....	(168)
16.3 Help主题词的显示设计.....	(170)
第十七章 创建 Help 主题文件.....	(173)
17.1 选择编辑工具.....	(173)
17.2 结构化Help主题.....	(173)
17.3 编码Help主题词文件.....	(174)
17.3.1 分配构造标记.....	(175)
17.3.2 分配内容相关的字符串.....	(175)
17.4 插入图像.....	(181)
17.5 管理主题文件.....	(182)
17.6 小结	(183)
第十八章 建立提示文件.....	(184)

18.1 创建提示规划文件.....	(184)
18.2 确定主题词文件: Files项.....	(184)
18.3 确定建立标记: BuildTags 项.....	(185)
18.4 确定可选项: options项.....	(185)
18.5 指定多个主题词内容: Alias项.....	(189)
18.6 映射与内容有关的主题词: MAP 项.....	(190)
18.7 通过引用嵌入位图: Bitmaps 项.....	(191)
18.8 编译提示文件.....	(192)
18.9 为应用编程而访问提示文件.....	(193)
18.10 小结.....	(198)
第十九章 提示举例及编译出错信息.....	(198)
19.1 提示主题词的 例子.....	(198)
19.2 提示编译 出错信息.....	(203)

第一部份 编译器和链接程序

第一部份中介绍怎样使用C编译器编译C语言源代码模块，怎样用链接程序把已经编译过的文件或已经汇编过的文件与Microsoft Windows库以及C库链接在一起，怎样利用资源编译器生成可执行的Windows应用程序。

第一章 编译应用程序：C编译器

许多Microsoft Windows应用程序是用C语言编写的，可以用Microsoft C编译器(CL)编译它们。本章介绍以下内容：

■ 简要说明怎样编写C语言Windows应用程序。

■ 怎样使用Microsoft C编译器编译C语言Windows应用程序。

本章只研究具有特别复杂要求的C语言Windows应用程序。有关C编译器的完整信息，参见有关的Microsoft C手册。有关使用C语言编写Windows应用程序的说明，参见《编程指南》。

1.1 编译C语言编写的Windows应用程序

编译C语言写的Windows应用程序的C编译器，由Microsoft C提供，不包含在Microsoft Windows软件开发工具(SDK)中。Microsoft Windows V3.0要求配备5.1以上版本的Microsoft C编译器，或是2.0以上版本的Microsoft Quick C。用C命令启动Microsoft C编译器。表1.1列出了编译Windows应用程序时通常使用的选择项及其说明。

表1.1 与Windows应用程序有关的C编译器选择项

选择项	用法	功能
-AC	以紧凑内存模式编译	当程序有一个代码段和多个数据段时使用。
-AL	以大内存模式编译	当程序有多重代码段和多重数据段时使用。
-AM	以中模式编译	当程序有多重代码段和单个数据段时使用。也可以用于生成使用混合内存模式的应用程序。混合内存模式参见《编程指南》。
-AS	以小模式编译	当应用程序有一个代码段和一个数据段时使用。也可以用于生成使用混合内存模式的应用程序。
-AW	确保指针在被强置成32位地址时，能得到	当编译动态链接库时使用。

	正确的段地址	
-C	只编译（不链接）	当有多个C源模块，并且需要把编译和链接分别进行时使用。
-Gs	取消栈检查，以提高性能	所有Windows应用程序在开发阶段完成之后，最好使用该项。
-Gw	为所有函数添加Windows前序代码和收尾代码	所有Windows源代码都需要使用该项。（也可以用于不包含输出的（回调）函数的源代码模块，但那时最好使用-GW）。
-GW	用经过缩减的Windows源代码模块前序和收尾代码替换应用程序中远程调用的函数（只有C6.0以上版本才有）	不包含输出的或回调函数的Windows源代码模块尽可能使用该项。
-Os	对代码大小进行优化，不考虑速度	所有Windows源代码模块都可以使用。
-Ow	在Windows环境设置的强置区中放松别名检查（仅对C6.0以及以后的版本有效）	用它代替非Windows程序的-Oa放松别名检查选项。
-Zd	生成可用于符号调试（程序SYMDEB）或80386调试程序（WDB386）的目标文件	在需要使用SYMDEB或WDB386调试源代码模块的场合指定该项。
-Zi	生成可用于Windows CodeView（CVW项。）的目标文件	在需要使用CVW调试源代码模块的场合指定该项。
-Zp	压缩结构成单字节边界	适用于所有使用结构的Windows源代码模块。

除了表1.1列出的选择项之外，也可以根据需要加入其它选择项。1.2节“编译器选择项”中有全面的介绍。

下例使用了开发小模块 Windows 应用程序源文件时推荐使用的一些选择项，编译TEST.C源文件：

```
CL -c -Os -Gw -AS -Zdp TEST.C
```

根据以上选择项，编译器不进行链接（-c），对程序大小进行优化（-Os），为输出的函数添加Windows前序代码和收尾代码，使用小内存模式（-AS），提供行号信号（-Zd），

压缩结构 (-Zp)。

1.2 编译器选择项

本节介绍编译Windows应用程序时可能用到的一些选择项。完整的说明参见《Microsoft C优化编译器用户手册》。

本节介绍以下几类编译器选择项：

■ 内存模式选择项，用来编译中模式、紧凑模式和大模式的应用程序。（编译器默认地使用小模式。）

■ 在应用程序开发阶段设置的选择项。

■ 编译动态链接库时使用的选择项。

1.2.1 内存模式选择项

Windows应用程序可以使用小、中、紧凑和大内存模式。（Windows不支持巨内存模式）。

在编译源文件时，指定CL命令行上适当的选择项，就可以指定编程模式。这样必须根据应用程序对代码和数据的需要作出选择。有关内存模式的编译器选择项有以下这些：

内存模式	编译器选择项
小模式	-AS
中模式	-AM
紧凑模式	-AC
大模式	-AL

由于Windows要求紧凑模式和大模式应用程序中的数据段都固定不动，这样一来Windows就只能运行具有远程数据段的应用程序的仅仅一个实例。正因为这样，Windows应用程序应避免使用紧凑模式和大模式，除非有必要把一个存在于DOS环境下的紧凑模式或大模式应用程序改造成一个Windows应用程序。任何紧凑模式或大模式应用程序的模式定义文件（.DEF）中，都必须包含以下语句：

DATA FIXED

在编译中、紧凑和大模式源文件时，可以指定源程序归属的代码段和数据段名称：-NT用于指定代码段；-ND用于指定数据段。如果不指定这些选择项，C编译器假定源程序归属于标准代码段_TEXT和标准数据段_DATA。

有关内存模式的详细说明，参见《编程指南》和适当的Microsoft C资料。

1.2.2 开发应用程序选择项

为了方便地开发和调试应用程序，C编译器提供以下选择项：

■ 提供行号信息以便显示程序的行。

■ 省略优化步骤，以便调试更方便。

■ 允许屏蔽栈检查。

为调试作准备