



Visual C++

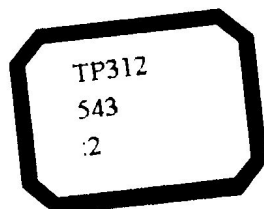
实践与提高

环境剖析 (IDE) 篇

赵明 邹传清 编著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



Visual C++ 实践与提高

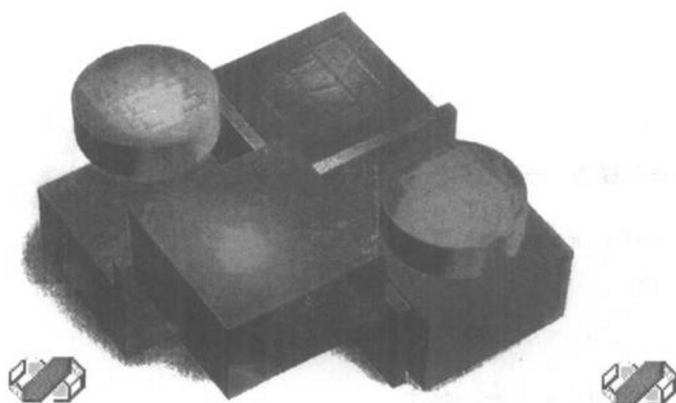
——环境剖析 (IDE) 篇

赵 明

编著

邹传清

 **Visual C++**



北方工业大学图书馆



00486598

中国铁道出版社

2001 • 北京

J5561 / 13

(京)新登字 063 号

内 容 简 介

本书共分为三大篇——基础篇、程序篇和高级篇。基础篇包括第一、二、三、四章，主要介绍 IDE 环境的概念、各种窗口的操作、编辑器的使用以及环境的定制，其中还提出了一种快速汉化 VC 界面的方法；程序篇包括第五、六、七章，主要介绍 AppWizard 的使用、在代码中快速导航以及 ClassWizard 的使用；高级篇包括第八、九、十、十一章，主要介绍调试功能的使用、编译及连接选项的设置、VBScript 宏和 Add-ins 的使用以及常用的辅助工具的使用等。

本书适合中高级 Visual C++ 的用户使用。

图书在版编目 (CIP) 数据

Visual C++ 实践与提高. 环境剖析 (IDE) 篇/赵明, 邹传清编著. —北京: 中国铁道出版社, 2000. 12
ISBN 7-113-04017-9

I. V... II. ①赵...②邹... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2000) 第 81006 号

书 名: Visual C++ 实践与提高——环境剖析 (IDE) 篇

作 者: 赵明 邹传清

出版发行: 中国铁道出版社 (100054, 北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟

特邀编辑: 鲁 凌

封面设计: 冯龙彬

印 刷: 北京兴顺印刷厂

开 本: 787×1092 1/16 印张: 19.25 字数: 459 千

版 本: 2001 年 1 月第 1 版 2001 年 1 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-113-04017-9/TP·496

定 价: 31.00 元

版权所有 盗印必究

凡购买铁道版的图书, 如有缺页、倒页、脱页者, 请与本社计算机图书批销部调换。

前 言

Visual C++ 6.0 作为现在最流行的编程工具，已经受到了越来越多开发者的青睐，虽然目前已经有很多介绍 VC 编程的书籍，但是始终没有一本全面而系统地剖析 IDE 环境的书。本书作为《Visual C++ 实践与提高》系列中的一本，目标就是填补这个空白，力求面向各个层次的 VC 使用者，系统而全面地介绍 IDE 环境的使用和技巧。

本书共分为三大篇，分别是基础篇、程序篇和高级篇。基础篇包括第 1、2、3、4 章，主要介绍 IDE 环境的概念、各种窗口的操作、编辑器的使用以及环境的定制，其中还提出了一种快速汉化 VC 界面的方法；程序篇包括第 5、6、7 章，主要介绍 AppWizard 的使用、在代码中快速导航以及 ClassWizard 的使用；高级篇包括第 8、9、10、11 章，主要介绍调试功能的使用、编译及连接选项的设置、VBScript 宏和 Add-ins 的使用以及常用的辅助工具的使用等，其中以大量实例展现了如何使用调试的高级技巧，并且讲解了代码生成及优化等一些高级选项的设置。

参加本书编写工作的有赵明、邹传清、吴晓晖、舒康、方敏、李威、刘立峰、祝晖等，感谢蔡宝忠、张严为本书策划，魏亚峰、赵志刚、郝亚鹏为本书收集资料，李建鹏、王利东、李西川、李利锋为本书校对，尚春红、程丽、张云为本书录入文稿，特别感谢铁道出版社的严晓舟老师和郭毅鹏同志，在本书的写作过程中提出了许多宝贵的意见和独到的创意。

由于编者水平有限，书中错误在所难免，请读者批评指正。

编 者

目 录

第 1 章 IDE 环境基础	1
1.1 IDE 概述	1
1.1.1 何为 IDE	1
1.1.2 Visual C++ 的 IDE 环境一瞥	2
1.2 IDE 界面元素	3
1.2.1 菜单条	3
1.2.2 工具条	8
1.2.3 Workspace 窗口	8
1.2.4 输出窗口	9
1.2.5 工作区	9
1.3 使用 IDE 与使用命令行	10
第 2 章 驾驭各种窗口	11
2.1 IDE 里面的窗口	11
2.1.1 窗口操作基础	11
2.1.2 两种类型的窗口	12
2.2 文档型(Document)窗口	12
2.2.1 平铺所有的文档型窗口	12
2.2.2 排列选定的文档型窗口	13
2.2.3 分割文档型窗口	13
2.2.4 进入全屏方式	14
2.2.5 在全屏方式下访问各种工具条	15
2.2.6 在全屏方式下访问菜单命令	16
2.3 停靠型(Docking)窗口	16
2.3.1 何为停靠型窗口	16
2.3.2 停靠型窗口的两种状态——停靠(docked)和浮动(floating)	17
2.3.3 从停靠状态到浮动状态	17
2.3.4 从浮动状态到停靠状态	17
2.3.5 显示/隐藏停靠型窗口	18
2.3.6 在停靠点保持浮动	18
2.3.7 将停靠型窗口变为文档型窗口	18



第 3 章 使用编辑器	21
3.1 代码编辑器	21
3.1.1 功能强大的代码编辑器	22
3.1.2 使用编辑命令	22
3.1.3 在代码中导航	29
3.1.4 有效地使用书签	33
3.1.5 VC6.0 的新特性——代码自动完成	35
3.1.6 在代码中查找和替换	41
3.2 资源编辑器	45
3.2.1 菜单编辑器	46
3.2.2 对话框编辑器	47
3.2.3 工具条编辑器	47
3.2.4 图标与位图编辑器	48
3.2.5 加速键编辑器	49
3.2.6 二进制编辑器	49
3.2.7 减少重复工作——资源模板的使用	50
第 4 章 定制自己的编程环境	51
4.1 定制菜单条	51
4.1.1 添加新的菜单	51
4.1.2 修改菜单外观	52
4.1.3 移动/复制菜单	54
4.1.4 删除菜单	55
4.1.5 恢复缺省菜单	56
4.2 定制工具条	56
4.2.1 创建新的工具条	57
4.2.2 添加新按钮	58
4.2.3 将按钮按功能分组	59
4.2.4 显示/隐藏工具条	60
4.2.5 删除工具条	61
4.2.6 在工具条上添加个性菜单	62
4.3 定制自己的工具集	64
4.3.1 添加自己喜欢的工具	64
4.3.2 删除不常用的工具	66
4.3.3 修改工具菜单	66
4.4 定制键盘快捷键	67
4.4.1 分配新的快捷键	67



4.4.2 删除已有快捷键.....	68
4.4.3 恢复缺省快捷键.....	69
4.5 定制小技巧集锦.....	69
4.5.1 在状态条上显示时间.....	69
4.5.2 启动 VC 时自动打开上次的项目.....	70
4.5.3 打开项目时不加载文档.....	70
4.5.4 控制 Window 菜单显示的窗口个数.....	71
4.5.5 控制系统记录最近编辑项目的个数.....	71
4.5.6 控制系统记录最近编辑文件的个数.....	71
4.6 定制的最高境界——汉化 VC 界面.....	72
第 5 章 建立自己的应用程序框架.....	75
5.1 概述.....	75
5.2 定制基于 MFC 的可执行程序框架.....	77
5.2.1 单/多文档可执行程序.....	77
5.2.2 基于对话框的可执行程序.....	89
5.3 生成动态链接库的框架.....	92
5.3.1 使用 MFC AppWizard (dll).....	92
5.3.2 生成 Win32 动态链接库框架.....	96
5.4 其他 Wizard.....	97
第 6 章 在程序中导航.....	99
6.1 概述.....	99
6.1.1 常用的查找功能.....	99
6.1.2 使用书签标记我们的程序.....	105
6.1.3 打开和关闭导航功能.....	105
6.1.4 利用 MFC 库的信息导航.....	106
6.2 浏览文件中各成员的信息.....	107
6.3 全面了解成员在工程中的方位.....	108
6.4 快速定位一个成员.....	109
6.5 全面掌握函数的调用关系.....	109
6.5.1 所调用的函数.....	109
6.5.2 被哪些函数调用.....	111
6.6 了解类的派生与继承关系.....	112
6.6.1 派生类.....	112
6.6.2 基类.....	114
6.7 其他工具的导航作用.....	114
6.7.1 ClassWizard.....	115
6.7.2 WizardBar.....	116



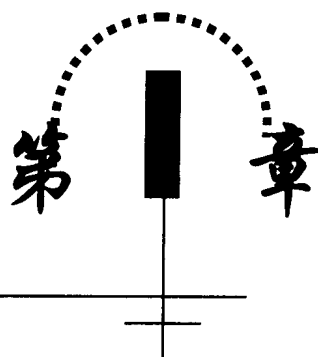
6.7.3	Workspace.....	118
第 7 章 在程序中添加代码..... 123		
7.1	概述.....	123
7.2	您的好帮手——ClassWizard.....	123
7.2.1	认识的 CLW 文件.....	124
7.2.2	特殊的注释形式.....	128
7.2.3	消息映射.....	130
7.2.4	产生新类.....	134
7.2.5	增加成员变量.....	143
7.3	灵活运用 WizardBar 和 ClassView.....	146
7.3.1	WizardBar.....	146
7.3.2	ClassView.....	155
7.4	管理工程中的类.....	156
7.4.1	引入外部类.....	156
7.4.2	改变类的名称.....	157
7.4.3	改变类所在的文件.....	158
7.4.4	删除一个类.....	158
第 8 章 有效地使用调试工具..... 159		
8.1	概述.....	159
8.2	调试版本的建立.....	160
8.3	VC 调试环境的一些特性.....	161
8.3.1	环境简介.....	161
8.3.2	拖放操作.....	163
8.3.3	方便的数据表现形式.....	164
8.3.4	在调试中修改代码.....	164
8.4	控制程序的运行.....	168
8.4.1	基本命令.....	169
8.4.2	设置断点.....	169
8.5	监视程序的运行.....	177
8.5.1	Output 窗口.....	179
8.5.2	Watch 窗口.....	182
8.5.3	QuickWatch 对话框.....	185
8.5.4	Call Stack 窗口.....	186
8.5.5	Memory 窗口.....	187
8.5.6	Variables 窗口.....	188
8.5.7	Registers 窗口.....	189



8.5.8 Disassembly 窗口	190
8.6 输出调试信息	192
8.6.1 TRACE 宏	192
8.6.2 输出对象信息	195
8.7 在程序中断言	199
8.7.1 条件断言	199
8.7.2 类型断言	201
8.7.3 对象有效性断言	202
8.8 内存泄漏检查	204
8.8.1 调试环境下的内存管理	204
8.8.2 内存检查	206
第 9 章 掌握 VC 编程环境的高级特性	211
9.1 设置项目编译选项	212
9.1.1 代码生成	215
9.1.2 Processor 选择	216
9.1.3 代码优化	228
9.1.4 生成列表文件	236
9.1.5 生成浏览信息文件	243
9.1.6 设置 C++ 语言特性	245
9.2 设置项目连接选项	248
9.2.1 连接器的一般选项	248
9.2.2 生成调试信息	250
9.2.3 设置输入模块	251
9.2.4 设置输出参数	252
9.2.5 定制连接	253
9.3 设置环境变量	254
9.4 创建项目的 Release 版本	255
9.4.1 Release 版本和 Debug 版本的差异	256
9.4.2 设置 Release 版本	256
9.4.3 解决可能出现的问题	256
第 10 章 任务自动完成	259
10.1 概述	259
10.2 VBScript 宏的使用	260
10.2.1 VBScript 简介	260
10.2.2 IDE 环境中的各种对象	264
10.2.3 快速录制宏	265
10.2.4 播放宏	267



10.2.5	手工编写宏.....	267
10.2.6	调试宏.....	272
10.2.7	将宏放置在工具条上.....	272
10.3	Add-ins 的使用.....	273
10.3.1	Add-ins 与 COM.....	273
10.3.2	用 AppWizard 创建 Add-ins.....	274
10.3.3	向 Add-ins 里添加命令	274
10.3.4	将 Add-ins 连接至 VC 开发环境.....	275
10.4	VBScript 宏与 Add-ins 的比较	276
第 11 章	使用辅助工具.....	279
11.1	捕获 Windows 消息——Spy++	279
11.2	Tracer	283
11.3	版本控制——Visual Source Safe	284
11.4	Visual Modeler	285
附录 1	VC 所支持的文件类型	287
附录 2	常用的键盘快捷键	289



IDE 环境基础

本章主要内容:

1. IDE 概念剖析
2. Visual C++的 IDE 环境介绍
3. IDE 与命令行的比较

我们可以看到, 现在流行的开发工具清一色全是可视化的, 你可以在一个统一的环境里面完成程序界面的设计以及代码的撰写、编译、调试。这就意味着作为一个现代的程序员, 你不仅要懂得编程语言, 更要懂得使用各种各样的工具, 你要用菜单编辑器去设计菜单, 用对话框设计器去设计对话框, 甚至要用位图编辑器去设计一幅位图, 而所有这些工具都集成在 IDE 环境下。

开发工具不同, 也就意味着它所提供的 IDE 环境不同。作为 Visual Studio 的一员, Visual C++的 IDE 开发环境与其它产品具有显著的不同之处, 因为从本质上来说, Visual C++并不是一种真正的可视化开发工具, 它同那些所谓的 RAD (Rapid Application Development) 开发工具是不同的, 它更侧重于比较低层的开发, 能完成其它一些工具所不能完成的任务, 因此它的 IDE 要相对复杂一点。为了使用 Visual C++强大的功能, 就必须熟悉它所提供的 IDE 开发环境, 下面将简要地介绍一下 Visual C++的 IDE 环境, 为以后熟练地使用它打下基础。

1.1 IDE 概述

自从 IDE 出现以后, 它就为程序员开发程序带来了很大的方便, 可以使得在一个统一的界面下完成一个程序的全部开发过程。Visual C++的 IDE 环境既吸收了一些其它 IDE 环境的优点, 又具有自己本身扩展的一些功能强大的特性, 特别是逐渐符号软件工程对编译器的要求, 具有一定的智能, 可以在编辑代码时就去掉一些语法错误。

本小节首先介绍 IDE 的概念, 使你能对 IDE 的发展和来龙去脉了解得更清楚, 然后详细地介绍 Visual C++的 IDE 环境的各个部分, 主要是个总体的介绍, 为以后的内容打好基础。

1.1.1 何为 IDE

IDE 的全称是 Integrated Development Environment, 即为集成开发环境, 这个概念早在 DOS 时代就已经有了。在非常古老的 DOS 时代, 程序员开发一个程序的流程是这样的: 首



先，在一个文本编辑器（如 DOS 下的 Edit）里面输入程序代码；然后，用编译程序对编辑好的程序进行编译，如果出现编译错误，再用文本编辑器打开程序进行修改，直到编译通过；接着，用连接程序将编译后生成的目标文件连接成可执行文件，运行此可执行文件，看是否按自己的想法工作，如果不是，那么还要用调试程序对可执行文件进行调试，直到程序完全按照自己的想法工作。后来，各个编译器厂家都将程序编辑器、编译程序、连接程序以及调试程序统统集成在一个开发环境里，使得在这一个开发环境里就能完成所有的开发工作，这就是 IDE 的由来。

到了 Windows 时代，由于 GUI（Graphic User Interface）的大量使用，IDE 环境集成的工具更多了。由于要处理菜单，所以就有了菜单编辑器；要处理对话框，就有了对话框编辑器；要处理各种各样的资源，就有了资源编辑器。总之是不停地有新的工具被集成进 IDE，使得 IDE 这个名称也越来越贴切了，也使得编程工具越来越向工具这个方向发展，可以说在现在这个时代，不会使用工具，就不可能编出好的程序。

1.1.2 Visual C++的 IDE 环境一瞥

当我们启动 Visual C++（以后简称 VC）后，就可以看到它所提供的 IDE 环境，如图 1-1 所示。

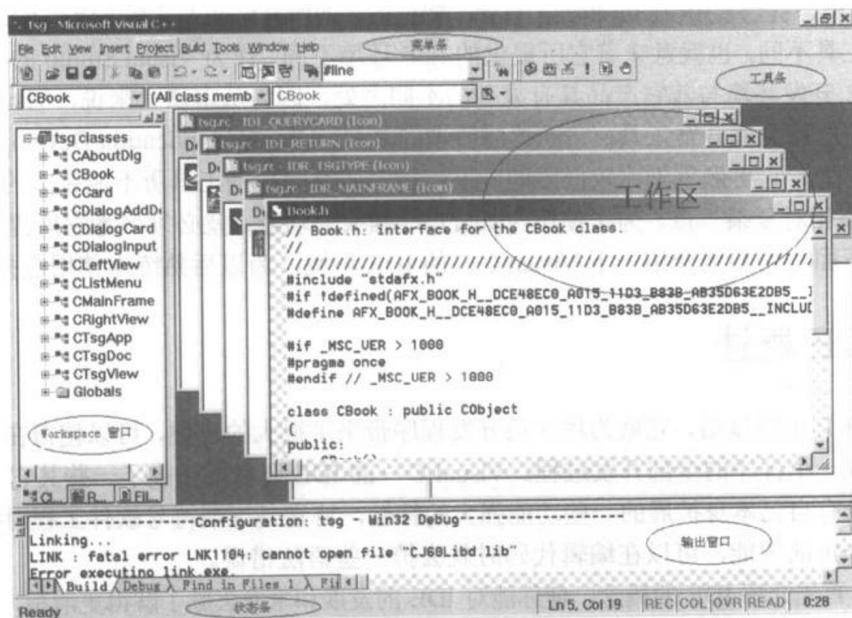


图 1-1 VC 的 IDE 界面

从图 1-1 中我们可以看到，VC 的 IDE 界面和传统的 Windows 程序一样，包含有菜单条、工具条以及状态条。除了这些以外，主界面共分为三大部分，分别是 Workspace 窗口、工作区和输出窗口，这些基本界面元素互相配合，形成了具有 VC 自己特色的 IDE 环境，下面将对这些基本界面元素作详细的介绍。



1.2 IDE 界面元素

我们知道，整个 IDE 环境是由各个 IDE 界面元素组成的，学会了使用各个 IDE 界面元素，就学会了整个 IDE 环境的使用。本节分别介绍菜单条、工具条、Workspace 窗口、输出窗口以及工作区等基本的界面元素，使得你对整个 IDE 环境有个比较感性的认识，其中在菜单条的介绍中，列出了所有的菜单命令，你可以在阅读以后的章节时，再回头来翻一下这些命令的介绍。

1.2.1 菜单条

菜单条提供了 VC 几乎所有的命令，通过这些命令，你可以访问到 VC 的大部分功能。除了提供标准的“File”、“Edit”、“View”、“Window”和“Help”菜单之外，VC 还提供了编程专用的功能菜单，例如“Project”、“Build”、“Debug”等。需要注意的是，VC 的菜单条是随着上下文而动态变化的，在一个状态下是看不到所有的菜单的，比如在调试程序的时候，“Debug”菜单就会替换“Build”菜单。下面我们按照功能，分类介绍 VC 提供的各个主要菜单。

“File”菜单

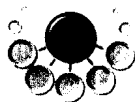
“File”菜单属于 Windows 标准菜单，提供对磁盘文件的存取，下面列出了“File”菜单下面的子菜单及其功能描述。如表 1-1。

表 1-1

菜单名称	功能描述
New	新建一个文件或项目
Open	打开一个文件
Close	关闭当前文件
Open Workspace	打开一个工作空间，工作空间可以包含多个项目
Save Workspace	保存当前的工作空间
Close Workspace	关闭当前的工作空间
Save	保存当前的文件
Save As	将当前的文件另存一份
Save All	保存所有的文件
Page Setup	打印时的页面设置
Print	打印
Recent Files	记录最近编辑过的文件，以供以后快速打开
Recent Workspaces	记录最近编辑过的项目，以供以后快速打开
Exit	退出 VC

“Edit”菜单

“Edit”菜单属于 Windows 标准菜单，提供常用的编辑功能，表 1-2 列出了“Edit”菜单



下面的子菜单及其功能描述。

表 1-2

菜单名称	功能描述
Undo	恢复上次操作
Redo	重复上次操作
Cut	剪切选中文本
Copy	复制选中文本
Paste	粘贴选中文本
Delete	删除选中文本
Select All	选中文件的全部内容
Find	在当前文件中查找字符串
Find in Files	在某个目录中的所有文件中查找字符串
Replace	在当前文件中查找并替换字符串
Go to	定位到文件中某一地方
Bookmarks	设置、管理书签
Advanced→ Incremental Search	渐增型搜索
Advanced→ Format Selection	将选中的文本转换成标准格式
Advanced→ Tabify Selection	将选中的文本中的四个空格转换成 Tab
Advanced→ Untabify Selection	将选中的文本中的 Tab 转换成四个空格
Advanced→ Make Selection Uppercase	将选中的文本全部转换成大写字母
Advanced→ Make Selection Lowercase	将选中的文本全部转换成小写字母
Advanced→ View Whitespace	查看文件中的空格和制表符 (Tab)，空格用 ` 来显示，Tab 用 ^ 来显示
Breakpoints	设置、管理断点
List Members	列出类的成员
Type Info	列出类的类型信息
Parameter Info	列出类的参数信息
Complete Word	自动补全代码

“View” 菜单

“View” 菜单属于 Windows 标准菜单，提供不同视图之间的切换功能，表 1-3 列出了“View” 菜单下面的子菜单及其功能描述。



表 1-3

菜单名称	功能描述
ClassWizard	启动类设计向导
Resource Symbols	查看资源符号
Resource Includes	查看资源头文件
Full Screen	在全屏幕方式下查看
Workspace	切换 Workspace 窗口的显示
Output	切换输出 (Output) 窗口的显示
Debug Windows→ Watch	显示 Watch 窗口, 在调试时, Watch 窗口缺省为打开
Debug Windows→ Call Stack	显示调用堆栈窗口
Debug Windows→ Memory	显示存储器窗口, 可以查看指定地址的内存内容
Debug Windows→ Variables	显示变量窗口, 可以查看局部变量, This 指针等, 在调试时, 变量窗口缺省被打开
Debug Windows→ Registers	显示寄存器窗口, 可以查看 CPU 的寄存器内容
Debug Windows→ Disassembly	显示反汇编窗口
Refresh	刷新显示
Properties	查看当前对象的属性

“Insert”菜单

“Insert”菜单是 VC 提供的辅助编程的菜单, 主要功能是向项目中添加内容, 下面列出了“Insert”菜单下面的子菜单及其功能描述。如表 1-4。

表 1-4

菜单名称	功能描述
New Class	添加一个新类, 类型不受限制
New Form	添加一个新的对话框类, 基类只能在 CDialog、CFormView 和 CRecordView 之间选择
Resource	添加一个新的资源, 可以是任何 VC 所支持的资源类型
Resource Copy	复制当前选中的资源, 并添进项目中
File As Text	将一个文件以文本形式添进当前项目
New ATL Object	插入一个新的 ATL 对象

“Project”菜单

“Project”菜单是 VC 提供的辅助编程的菜单, 主要用来对项目进行管理、设置, 下面列出了“Project”菜单下面的子菜单及其功能描述。如表 1-5。



表 1-5

菜单名称	功能描述
Set Active Project	设置当前激活的项目，只在同时打开多个项目的时候有用
Add To Project→ New	在“New”对话框里创建新的文件或项目，并添加到当前的工作空间中
Add To Project→ New Folder	添加新的文件夹到当前的项目中
Add To Project→ Files	添加文件到当前的项目中
Add To Project→ Data Connection	添加数据库连接到当前的项目中，只对数据库程序有用
Add To Project→ Components and Controls	添加 VC 提供的组件或已经登记的 ActiveX 控件到当前的项目中
Dependencies	设置项目间的依赖关系
Settings	设置项目的各个属性选项
Export Makefile	将项目文件导出成 Make 文件
Insert Project into Workspace	向当前的工作空间中插入项目，可以多项目同时打开

“Build”菜单

“Build”菜单是 VC 提供的辅助编程的主要菜单，用来对项目进行编译、连接并生成可执行文件，下面列出了“Build”菜单下面的子菜单及其功能描述。如表 1-6。

表 1-6

菜单名称	功能描述
Compile	编译当前激活的源文件或头文件
Build	编译并连接当前激活的项目配置
Rebuild All	对当前激活的项目配置先 Clean，再 Build
Batch Build	可以同时选中多个项目来 Build
Clean	将当前激活的项目配置产生的所有中间文件和输出文件删除
Start Debug→ Go	开始调试当前项目
Start Debug→ Step Into	单步跟踪，进入函数内部
Start Debug→ Run to Cursor	执行到光标处
Start Debug→ Attach to Process	将系统进程挂到 VC 调试器中进行调试
Debugger Remote Connection	调试远程机器上的程序，可以通过 TCP/IP 协议连接
Execute	执行当前的项目
Set Active Configuration	设置哪个项目配置被激活
Configurations	设置项目的各个配置，包括“Debug”、“Release”版本，还可以添加自己的配置
Profile	生成 Profile 文件，只在 Build 时允许了 Profiling 才有效



“Tools” 菜单

“Tools” 菜单是 VC 提供的辅助编程的菜单，主要用来对 IDE 环境进行定制，同时提供了一些外部工具，表 1-7 列出了“Tools”菜单下面的子菜单及其功能描述。

表 1-7

菜单名称	功能描述
Source Browser	使用浏览信息在代码中导航，需要在 Build 时生成浏览信息文件（BSC 文件）
Close Class Browser File	关闭浏览信息窗口
Visual Component Manager	启动 Component Manager 工具
Register Control	启动 Register Control 工具
Error Lookup	启动 Error Lookup 工具
ActiveX Control Test Container	启动 ActiveX Control Test Container 工具
OLE/COM Object Viewer	启动 OLE/COM Object Viewer 工具
Spy++	启动 Spy++ 工具
MFC Tracer	启动 MFC Tracer 工具
Customize	定制菜单条、工具条、Tools 菜单、快捷键等
Options	设置 IDE 开发环境的一些选项
Macro	管理、编辑、运行宏
Record Quick Macro	将鼠标、键盘操作录制成宏
Play Quick Macro	播放录制的宏

“Window” 菜单

“Window” 菜单属于 Windows 标准菜单，提供对各个窗口的管理，以及在窗口之间的切换，表 1-8 列出了“Window”菜单下面的子菜单及其功能描述。

表 1-8

菜单名称	功能描述
New Window	复制当前激活的窗口
Split	分割当前窗口
Docking View	切换 Docking 窗口的状态，要么为 Docked 状态，要么为 Floating 状态
Close	关闭当前的窗口
Close All	关闭所有打开的窗口
Next	切换到下一个窗口
Previous	切换到前一个窗口
Cascade	层叠放置所有打开的窗口
Tile Horizontally	水平平铺所有打开的窗口
Tile Vertically	垂直平铺所有打开的窗口
Windows	显示所有打开的窗口的列表