

计算机软件开发系列丛书

Visual C++ 入门与应用

何福林 编著
刘 莹



希望

学苑出版社

计算机软件开发系列丛书

Visual C++入门与应用

何福林 刘 莹 编著

史小君 燕卫华 审校

学 苑 出 版 社

(京)新登字 151 号

内 容 提 要

Visual C++ 的基本内容大致可分成四部分:基础知识、类库、函数库和 Windows 编程。本书集中介绍了用 Visual C++ 开发应用程序所涉及到的各项内容,主要包括 Visual C++ 入门与安装、OOP 技术、C++ 的特性和如何在 Visual C++ 中建立与使用函数库及类库。全书可分成两部分:第一部分(第一章到第十七章)完整地讨论了程序开发过程中的各种问题;第二部分(最后四章)在总结第一部分的基础上给出了若干个完整的实例程序,并对这些程序作了较透彻的剖析。

有关函数库、类库和 Windows 编程的内容,读者可参阅相关书籍。对 C++(尤其是 Visual C++) 语言程序员而言,本书是一本很实用的技术指导书籍,并可供 Visual C++、OOP 和 Windows 程序员参考。

需要本书者,请与北京海淀 8721 信箱书刊部联系,邮政编码 100080,电话 2562329。

计算机软件开发系列丛书

Visual C++ 入门与应用

编 著:何福林 刘 莹

审 校:史小君 燕卫华

责任编辑:汪亚文

出版发行:学苑出版社 邮政编码:100036

社 址:北京市海淀区万寿路西街 11 号

印 刷:保定列电印刷厂

开 本:787×1092 1/16

印 张:39 字 数:911 千字

定 数:1~5000 册

版 次:1994 年 7 月北京第 1 版第 1 次

ISBN 7-5077-0779-2/TP·11

本册定价:48.00 元

学苑版图书印、装错误可随时退换

目 录

第一章 Microsoft Visual C++ 环境	1
1.1 Microsoft Visual C++ 快速入门	1
1.2 安装 Microsoft Visual C++	5
1.3 构造 DOS MFC 库文件	8
1.4 使用 Visual Workbench	9
1.5 在 Visual Workbench 中获取帮助	11
1.6 使用 Visual Workbench 菜单	13
1.7 编辑一个程序	19
1.8 编译、链接及运行一个程序	20
1.9 浏览	22
1.10 设置工程文件选项	23
1.11 调试	26
1.12 使用 CL	26
1.13 用 CL 进行编译和链接	26
1.14 使用 CL 环境变量	27
1.15 CL 选项	28
1.16 一个 Visual Workbench 练习	35
1.17 小结	36
第二章 Visual C++ 中的程序开发工具	37
2.1 LINK 链接器	37
2.2 LINK 选项	41
2.3 库文件管理程序 LIB	44
2.4 程序维护实用程序 NMAKE	46
2.5 CodeView 调试程序	51
2.6 小结	61
第三章 ANSI 标准 C 概述	62
3.1 一个 C 程序的结构	62
3.2 ANSI C 转义序列和 trigraph(三字母一音)	64
3.3 预处理器伪指令	65
3.4 变量的声明和定义	69
3.5 结构、联合及位字段	71
3.6 表达式	74
3.7 语句	78
3.8 函数	82

3.9	ANSI C 库	84
3.10	小结	85
第四章	Microsoft Visual C++ 对标准 C 的扩展	86
4.1	Microsoft Visual C++ 特有的关键字	86
4.2	定制内存模式	89
4.3	预定义全局变量及预处理器宏	93
4.4	预编译指令	96
4.5	基本数据类型的大小和容量	97
4.6	小结	98
第五章	面向对象的程序设计(OOP)基础	99
5.1	什么是面向对象的程序设计	99
5.2	面向对象程序设计技术术语	107
5.3	C 语言中的面向对象的程序设计	110
5.4	小结	126
第六章	C++ 和面向对象的程序设计	128
6.1	C++ 面向对象的程序设计	129
6.2	C++ 的几何图形	135
6.3	小结	142
第七章	C++ 和 ANSI 标准 C	143
7.1	C++ 的特性	143
7.2	小结	154
第八章	用于标准 I/O 的 C++ 类	155
8.1	C++ I/O 库	155
8.2	C++ 中的流 I/O	155
8.3	文件 I/O	164
8.4	串 I/O	170
8.5	小结	171
第九章	用类创建对象	173
9.1	作为对象的类	173
9.2	实现类	178
9.3	使用类	182
9.4	小结	193
第十章	定义对象的操作	194
10.1	参数和返回值	194
10.2	创建和释放对象	198
10.3	定义函数和操作符	209
10.4	小结	224
第十一章	在 C++ 中使用继承	225
11.1	派生类	225

11.2	使用继承.....	243
11.3	小结.....	261
第十二章	虚拟函数和多态性.....	263
12.1	动态链接.....	263
12.2	虚拟函数.....	265
12.3	小结.....	273
第十三章	C++的高级技术	274
13.1	模板.....	274
13.2	异常处理.....	277
13.3	小结.....	282
第十四章	在C++中使用C库	283
14.1	C与C++的链接	283
14.2	使用ANSI标准C库	287
14.3	编译器指定的库.....	305
14.4	小结.....	306
第十五章	在C++中建立类库	307
15.1	在C++中建立类库	307
15.2	C++类的公用接口	315
15.3	小结.....	317
第十六章	使用Microsoft Foundation Class 库	318
16.1	用Microsoft Foundation Class 设计Microsoft Windows 程序	318
16.2	Microsoft Foundation Class 的层次	331
16.3	小结.....	336
第十七章	用C++建立MS-DOS应用程序	337
17.1	Forms 软件包	337
17.2	表格的存储和检索.....	338
17.3	表格的组成.....	342
17.4	显示一个表格.....	353
17.5	Form 类	367
17.6	创建一个表格.....	373
17.7	填写表格.....	380
17.8	小结.....	390
第十八章	类与对象设计实例.....	391
18.1	抽象类.....	391
18.2	多态行为.....	410
18.3	结构类型.....	422
18.4	状态驱动对象.....	435
18.5	禁止对象.....	440
18.6	迟后联编和重联编.....	444

18.7	小结.....	456
第十九章	类体系设计实例.....	458
19.1	特权实例.....	458
19.2	对成员函数的访问.....	492
19.3	小结.....	507
第二十章	通用 MFC 类应用实例	508
20.1	CString 类	508
20.2	扩充 CString 类	524
20.3	数组类.....	525
20.4	表类.....	537
20.5	映射类.....	552
20.6	小结.....	565
第二十一章	类模板应用实例.....	566
21.1	TEMPLDEF 实用程序	566
21.2	通用队列类.....	567
21.3	建立 int 型队列类	574
21.4	建立字符串队列类.....	590
21.5	CObList 类的使用	605
21.6	小结.....	615

第一章 Microsoft Visual C++ 环境

本书的主要目的是帮助 C 程序员学习用 Microsoft Visual C++ 编译器进行 Windows 程序设计、面向对象的程序设计(OOP)和 C++ 程序设计的基本概念。同时,本书还可以作为 C++ 类(Microsoft Foundation Classes V2.0)和伴随 Microsoft Visual C++ 的 C 函数库的参考指南。当然,在学习 OOP 和 Microsoft Visual C++ 库之前,读者应了解 Microsoft Visual C++ 提供了哪些工具来帮助用户执行必需的编辑、编译、链接和调试等程序设计任务。为了帮助用户熟悉 Microsoft Visual C++ 环境,本章对 Microsoft Visual C++ 产品进行概述,包括其组成部分、如何安装以及如何运行 Windows 的 PC 机上使用。Visual C++ 从本质上讲是一个 Windows 应用程序。尽管可以在一个 DOS 窗口中进行编译,但 Visual C++ 最好的程序设计界面是 Visual Workbench。

Visual C++ 有两个版本:

(1) 专业版本,它包括一个更高级的优化编译器,更广泛的文档并能设计基于 DOS 的应用程序。

(2) 个人版,其费用较低,没有那么高级的编辑器,文档较少,且不能设计基于 DOS 的应用程序。

本书都使用专业版编译器,价格的差别对编译器性能的降低使专业人员更倾向于专业版编译器。

本章还将介绍称为 Visual Workbench 的交互式开发环境,以及编译和链接的 DOS 命令行接口 CL。第二章进一步介绍了程序开发工具的其他方面,如帮助用户管理建立应用程序的“编辑—编译—链接—调试”周期的 LIB, LINK, NMAKE 和 CV。当然,有了 Visual Workbench,用户不必再关心实际产生可执行程序机理。Visual Workbench 为用户做了绝大部分工作。

第三章和第四章总结了 C 程序设计语言的特性。

1.1 Microsoft Visual C++ 快速入门

Microsoft Visual C++ 是多个产品的集成。因此,随软件带了许多手册。Visual C++ 有三种风格:标准版本。专业版本和 C7 升级版。对于这三种风格,都提供了针对 Visual C++ 变体的文档。

作为主产品的专业版 Visual C++ 具有比标准版更多、功能更强的特性。专业版提供如下手册:

Visual C++ 介绍

Visual Workbench 用户指南

App Studio 用户指南

C++ 教程

类库用户手册
程序设计技术
面向 Windows 的程序设计工具
CodeView Debugger 用户指南
命令行实用工具用户指南
Source Profiler 用户指南
类库参考
C 语言参考
运行时间库参考
iostream 类库参考
综合索引
MFC 快速参考

Windows 3.1 SDK 入门

Windows 3.1 SDK 程序员参考第一卷:概述

Windows 3.1 SDK 程序员参考第二卷:函数

Windows 3.1 SDK 程序员参考第三卷:消息、结构和宏

Windows 3.1 SDK 程序员参考第四卷:资源

标准版 Visual C++ 的功能要弱一些(针对非职业软件设计人员),它只带较少的手册:

Visual C++ 介绍

Visual Workbench

App Studio 用户指南

C++ 教程

类库用户指南

程序设计技术

类库参考

MFC 快速参考卡片

围绕 Visual C++ 专业版提供的 C7 升级版,适用于 Microsoft C7.0 老用户,升级版带以下手册:

Visual C++ 介绍

Visual Workbench 用户指南

App Studio 用户指南

类库用户指南

类库参考

MFC 快速参考卡片

C/C++ 7.0 升级

购买 Visual C++ 能额外买到 SDK 资料,包括:

入门

程序设计工具

程序设计指南

多媒体程序设计参考

多媒体程序设计指南

面向 Windows 的 Pen Computing 程序员参考

Windows 界面:应用程序设计指南

Win32 应用程序设计界面概述

Visual C++ 中包含了一个订单表,可用来订购这些书。

Microsoft 还包含一 Phar Lap 的 286/DOS-Extender Lite 的拷贝,该产品允许 DOS 应用程序访问 2M 以上的扩展内存。

Visual C++ 提供完全的面向 C 和 C++ 的 Windows 程序设计环境,并且包含所有的支持工具及编写 Microsoft Windows 和 DOS 应用程序所需的库。作为一名程序设计师,用户一般先用如下工具:

(1) Visual Workbench,用于编辑源文件,并与包括 C 和 C++ 编译器在内的许多实用支持工具接口,Visual Workbench 允许用户建立应用程序,并对其进行调试,而不必退出 Visual Workbench。通常,Visual Workbench 将帮助用户使用其他开发工具。例如,运行编译器、链接器和资源编译器以建立可执行文件。

(2) Application Wizard,通常称为 AppWizard,它是用于建立初始 Windows 应用程序的程序开发工具。

(3) Class Wizard,用于管理 C++ 类的实用工具(用 App Wizard 进行程序开发时使用)。

(4) Application Studio,通常称为 App Studio,用于编辑如菜单、对话框、位图、图标和光标等的 Windows 应用程序资源。

(5) LINK,用于将多个目标模块结合成一个可执行文件的基于 DOS 的链接器。

(6) LIB,用户在一个文件中管理目标模块集合的基于 DOS 的库管理程序。

(7) NMAKE,用于自动程序开发的基于 DOS 的制作实用程序。

(8) CV 和 CVW,分别用于 DOS 和 Windows 的 CodeView 调试器。

也有一些实用工具(都是基于 DOS 的)用户通常不使用,但一旦用户想使用都能很容易地得到,这些实用工具包括:

(1) EXEHDR,一种用来显示和修改包含在 EXE 文件头部中的信息的实用工具。

(2) CVPACK,用来压缩包含 CodeView 调试器所需信息的可执行文件的大小。

(3) HELPMAKE,一种用于将文本文件(具有特殊格式)转换为数据库的实用工具,这些数据库可被许多像 Visual Workbench 的 Microsoft 应用程序使用,用来显示联机帮助信息。

(4) IMPLIB,一个用来从一个或多个动态链接库创建一个特殊类型库的输入库。输入库用于 Microsoft Windows 程序。

(5) MSD,让用户查看有关自己 PC 机硬件和软件配置等重要信息的 Microsoft Diagnostics 工具。

另外,Microsoft Visual C++ 包含一系列基于 Windows 的实用工具。这些 Windows 实用工具有:

(1) SPY,一个允许用户检查将发往一个窗口的消息和消息参数的程序。

(2) DDESPY,一个允许用户查看一个 DDE 服务程序和一个客户应用程序之间 DDE 交换的程序。

(3) HEAPWALKER, 一个允许用户检查 Windows 内存, 以查看存在哪些内存对象、每个运行应用程序的内存对象的地址及 Windows 内存是怎么被利用的实用工具。

(4) STRESSAPP, 一个允许用户强制 Windows 分配资源, 直到没有资源可分配给应用程序的程序, 这就允许用户在某些资源不再可以利用时, 查看应用程序出了什么问题。

(5) ZOOMIN, 一个小应用程序, 它允许用户放大屏幕的一小部分。

(6) DEBUGWIN, 一个实用工具, 它允许用户查看 Windows 产生的调试版本信息(通常用户需要一台连到 COM1 的终端以查看调试信息)。

(7) MFC TRACE OPTIONS, 一个允许用户查看 MFC 使用的调试选项的实用工具。

(8) HOTSPOT EDITOR, 一个用户可用来建立或编辑超图形的程序, 超图形是一幅有一个或多个用户可在其上引起动作的热点的位图。

用户如何使用 Microsoft Visual C++ 要根据用户的需要决定。用户可以用以下方式之一编译和链接自己的应用程序:

(1) 从 Visual Workbench 调用编译器和链接器, 可使用 Project, Build 和 Project, Rebuild All 菜单选项。

(2) 从 DOS 命令行运行编译器和链接器。

(3) 使用 NMAKE 使编译和链接过程自动化。

使用 Visual Workbench 是 Visual C++ 开发应用程序的最效的方法。也可以在 DOS 下运行编译器。但是, 这样效率不高, 因为用户必须先运行一编辑器编辑源代码, 然后再编译(和链接)源代码。

总之, 用户会发现 Visual Workbench 的编辑器是一高效的程序设计工具, 其编辑器在有限范围内可以配置用户的链接操作。使用 Visual Workbench 的工具条, 用户可以直接访问编译器和链接器。Visual Workbench 是一具有相当标准的 Windows MDI 用户界面的 Windows 应用程序。当 Windows 以增强模式(80386 保护模式)运行时, 用户可以用调试器运行和调试应用程序, 该调试器是 Visual Workbench 集成环境的一部分。Visual Workbench 允许用户完成整个程序开发循环, 即开发和编辑源文件; 建立 Windows 资源; 编译、链接及运行程序; 调试程序。所有这些操作都不必退出 Visual Workbench。

因为大量软件产品都包含许多源程序模块, 所以使用 Visual Workbench 非常必要, Visual Workbench 将建立一个工程文件(与 NMAKE 的 MAK 文件类似), 它包含 Visual Workbench 编译和链接必要模块所需要的所有伪指令。有了工程文件, 当用户编辑了一个或多个源文件后, 选择 Visual Workbench 的 Project Build 图标将编译所有受影响的文件。例如, 修改一个头文件就必须编译所有包含该头文件的源文件。

正如用户将从本章有关 Visual Workbench 的描述将了解到的, 工程文件描述了源文件、目标文件和可执行文件的相互依赖关系。Visual Workbench 使用依赖关系知识来编译适当文件并建立新的可执行文件。

用户使用 Options 菜单为编译和链接声明各种选项, 不论某个工程是第一次被建立还是工程文件已经建立后。在本章后一部分中, 概要说明将显示如何从 Visual Workbench 设置编译和链接选项。在本章末尾, 有一个如何建立 Visual Workbench 工程的教程。但是, 下节首先描述如何安装 Microsoft Visual C++。再下一节概述 Visual Workbench 和 CL, CL 是供编译和链接用户的命令接口。Microsoft Visual C++ 所包括的许多其他工具将在第二章中介绍。

注意,Microsoft Visual C++需要 Windows 运行在 Enhanced Mode,80386 或 80486 系统,至少 4MB 内存才能运行。编译器是一 32 位 Windows/DOS 应用程序,必须运行在保护模式的 Intel 80386 或 80486 微处理器上。

尽管 Microsoft Visual C++编译器、链接器及其他程序通用的实用程序可运行在 DOS 窗口中,但一旦用户开始用 Visual Workbench 后,一般不会经常把编译器作为 DOS 应用程序来使用。

1.2 安装 Microsoft Visual C++

安装 Microsoft Visual C++只须直接运行发行盘中的 SETUP 程序。SETUP 将发行盘中的文件恢复并拷贝到用户所指定的硬盘上。用户有权选择希望 Microsoft Visual C++安装时使用的驱动器和目录。在启动 Windows 后,安装 Visual C++基本步骤如下:

(1) 将第一张盘(#1)插入到适当的驱动器中(A:或 B:),根据软件大小及系统配置而定。

(2) 从 Program Manager 中选 File 菜单的 Run 选项,在命令行方框中,输入如下程序名(使用正确的驱动器字母):

A:SETUP

(3) 一旦 SETUP 开始运行(如图 1.1),阅读并响应屏幕显示。任何时刻按下屏幕上的 HELP 按钮(如图 1.2)都可调出联机帮助信息。通过响应 SETUP 的屏幕所提的问题,用户能完成如下操作。

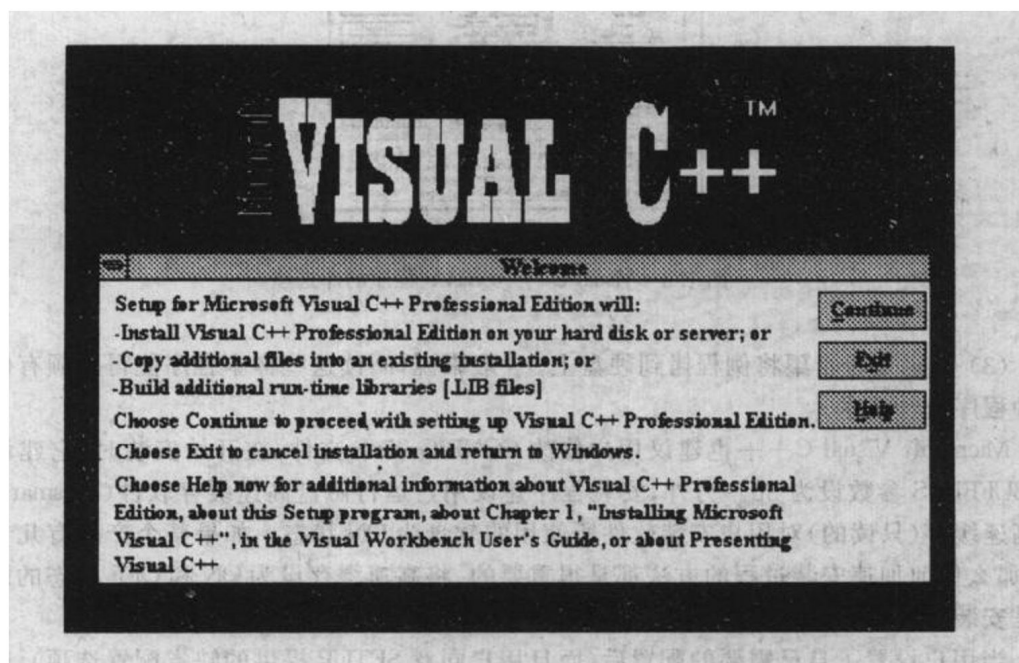


图 1.1 Visual C++ SETUP 程序打开的屏幕

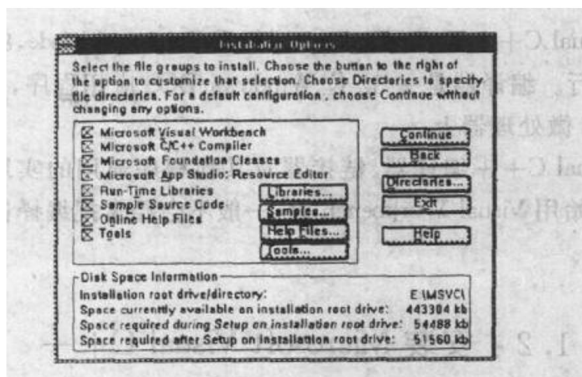


图 1.2 Visual C++ SETUP 程序的初始化选项

(1) 选择用户希望 Microsoft Visual C++ 装在其中的驱动器号和目录。注意 Visual C++ 需要多少硬盘空间。在缺省安装情况下,需要 52MB 硬盘空间。如果用户安装了任何 Visual C++ 的可选部分,实际可能需要更多的硬盘空间。

(2) 为用户需要安装的库选择 memory models 复选框,如图 1.3 所示。如果用户不熟悉内存模式,可以选择一种安全的选择,即安装除 Compact 模式外所有的内存模式,专业程序员一般不使用 Compact 模式。

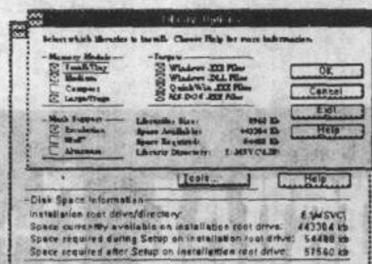


图 1.3 Visual C++ SETUP 程序的库选项

(3) 决定是否希望将例程拷到硬盘上。一般来说,阅读这些样本程序能得到颇有价值的各种程序设计技术。

Microsoft Visual C++ 也建议用户修改 CONFIG.SYS 文件,在开始安装时,它建议用户将 BUFFERS 参数设为 30。另外,安装程序建议用户运行磁盘高速缓存软件(如 smartDrv),使高速缓存(只读的)对用户安装软件所使用的软盘为 ON 状态。如果某个产品有几十张软盘,那么任何加速安装过程的方法都是很重要的。将高速缓存设为 ON 和 OFF 状态的区别可能是安装时间相差数倍。

当用户设置了自己需要的配置后(而且用户同意 SETUP 提供的缺省配置选项),就应选择 Continue 按钮。这样就通知了 SETUP 为用户所选的内存模式安装编译器和建立库文件。SETUP 也创建一个程序组(见图 1.4)从而使用户能由此选择 Visual C++ 实用程序。还有许多图标(在 Visual C++ 的 Program Manager 的程序组中)可供用户使用 Visual C++ 所带的

帮助文件。



图 1.4 Visual C++ Program Manager 程序组

为了能从 DOS 命令行运行编译器和其他实用程序,用户必须使用批处理文件 MSVCVARS. BAT。该文件由 SETUP 创建,SETUP 向 MSVCVARS. BAT 加入针对所需环境变量的定义。用户可从一 DOS 窗口运行该批处理文件。MCVCVARS. BAT 将为用户设置必需的环境变量。如果用户愿意,可以通过直接包含 MCVCVARS. BAT 中行或调用 MCVCVARS. BAT 来将该文件包含到 AUTOEXEC. BAT 中。如果用户将 MSVCVARS. BAT 包含到 AUTOEXEC. BAT 中,就必须确信将环境变量的设置放在前面。

MSVCVARS. BAT 文件一般包含如下命令:

```
@echo off
set TOOLROOTDIR=D:\MSVC
set PATH=D:\MSVC\BIN;%PATH%
set INCLUDE=D:\MSVC\INCLUDE;D:\MSVC\MFC\INCLUDE;%INCLUDE%
set LIB=D:\MSVC\LIB;D:\MSVC\LIB;%LIB%
```

从用户对 MS-DOS 的经验来看,用户应该知道像 PATH 这样的环境变量一般是一任意字符串的名字,而且是用 SET 命令定义的。

在 DOS 提示符下输入 SET 将显示当前环境变量的设置,如

```
SET PATH=C:\;C:\DOS6
```

定义 PATH 为字符串 C:\;C:\DOS6,从而代替了它的先前定义。下面是 SETUP 定义的环境变量:

TOOLROOTDIR	存放 Visual C++ 工具的目录, 例如: SET TOOLROOTDIR=D:\MSVC
PATH	当输入一个不是 DOS 内部命令的命令后, DOS 搜索的目录路径。如: SET PATH=D:\MSVC\BIN;%PATH% 请注意, 原路径是如何用 %PATH% 环境变量加到新路径变量后的。
INCLUDE	由分号分隔的目录名, C 和 C++ 包含文件(一般带 H 扩展名)存于其中。例如: SET INCLUDE=D:\MSVC\INCLUDE;D:\MSVC\MFC\INCLUDE;%INCLUDE% 请注意: 原包含路径是如何通过用 %INCLUDE% 环境变量加到包含路径变量后的。
LIB	由分号分隔的目录名, 这些目录是用来保存库文件的。例如: LIB=D:\MSVC\LIB;D:\MSVC\MFC\LIB;%LIB% 请注意: 原库路径是如何通过用 %LIB% 环境变量附加到新库路径变量后的。

Microsoft Visual C++ 的工具使用这些环境变量来定位所使用的文件。

1.3 构造 DOS MFC 库文件

尽管 Visual C++ 软件有许多张软盘, 但不是所有编译器的库文件都由安装盘提供的。缺少的一部分库就是 Microsoft Foundation Classes(MFC) 实模式(DOS) 库。

这个缺点很容易克服。Microsoft 提供优秀的 MAKE 文件以使用户能构造(或重构造)任何 MFC 库。另外, 还可以修改原始 MFC 代码(不提倡), 也可重新构造库以包含用户的修改。但是, 在作任何修改前, 用户必须仔细考虑两件事: 首先, 这些修改可能产生什么副作用; 其次, 在以后安装新版 Microsoft MFC 库时, 怎样完成修改。

要建立新的 MFC DOS 库, 请先打开一 DOS 窗口, 然后执行如下步骤:

(1) 使用 MSVCVARS.BAT 文件(如果用户没有在 AUTOEXEC 中加入对库 MSVCVARS 的调用)为 DOS 命令行版的编译器建立环境。

(2) 转到包含 MFC 源文件的目录(MSVC\MFC\SRC)。

(3) 阅读文件 README.TXT, 其中包含有关构造 MAKE 文件过程的详细指令。

(4) 构造每个所需的库。建议用户或者为已安装了运行时间库的模式构造所有内存模式, 或者简单地构造所有 MFC 库。用户应同时构造每个内存模式的发行版和调试版。

例如, 构造 MFC DOS medium 内存模式, 调试版库的命令是:

```
NMAKE DEBUG=1 TARGET=R MODEL=M
```

一旦用户构造好了 MFC DOS 库, 就可以建立使用 MFC 库的 DOS 工程。

1.4 使用 Visual Workbench

当用户成功地安装了 Microsoft Visual C++ 后,就可以从 Visual Workbench 试用 C 和 C++ 编译器, Visual Workbench 是一个集所有功能于一身的集成程序设计环境。Visual Workbench 是一个相当标准的 Windows MDI 应用程序,而且如果用户熟悉 MDI 应用程序的使用,使用该产品就不会有什么困难。使用 Visual Workbench,用户能进行如下操作:

- (1) 编辑一个或多个文件
- (2) 全屏幕浏览源程序文件(以只读方式编辑)
- (3) 通过编译源程序文件和链接一个或多个目标文件构造一可执行文件
- (4) 运行新创建的程序
- (5) 使用内部调试器发现程序错误

另外, Visual Workbench 提供大量联机帮助。事实上,学习 Visual Workbench 的最佳方法是开始使用它,并阅读联机帮助信息。尽管如此,为了用户方便,下节将简要介绍 Visual Workbench 的菜单所提供的一些重要选项,特别是对 C 和 C++ 程序设计比较重要的选项。

用户按照 C\C++ 语言和关键字可定制 Visual Workbench 的屏幕颜色及其彩色能力。Visual Workbench 不像老的 Programmer 的 Workbench 那样容易扩展或定制。遗憾的是, Visual Workbench 的编辑器没有像许多程序设计员所偏爱的灵活性。以后的版本的产品相信会具有强大的灵活的编辑器可供用户重新配置,并为其写宏定义。除屏幕颜色外,目前用户所见的是 Visual Workbench 的缺省配置。

1.4.1 Visual Workbench 入门

用户可以从 Microsoft Windows 启动 Visual Workbench。要启动 Visual Workbench,只需从 Visual C++ 程序组中选择适当的图标(如图 1.4 所示)。

选择 Visual C++ 图标将调出 Visual Workbench(如图 1.5)。屏幕上出现什么内容依赖于用户上次使用 Visual Workbench 时最后的操作。如果用户在某个工程处于打开状态时退出 Visual Workbench, Visual Workbench 下次启动后将自动打开该工程。

注意:用户如需要退出 Visual Workbench,可接 ALT+F4。另一种退出方法是用鼠标或按 ALT+F 调出 FILE 菜单,然后选择 EXIT 选项(用鼠标或按 X 键)。

在启动时 Visual Workbench 从名为 MSVC. INI 的文件中读某些初始化设置并自动装入 Visual Workbench 上次最后装入的工程。通过编辑该文件,用户可以修改许多 Visual Workbench 使用的缺省设置。与许多. INI 文件不同, MSVC. INI 中的许多语句都通过编辑该文件而修改。用户应养成习惯不在应用程序运行时修改. INI 文件。但是,许用应用程序在退出时都重写. INI 文件,从而可能改变用户所作的修改。在修改 MSVC. INI 文件时,务必要先备份,以防在编辑时出错。

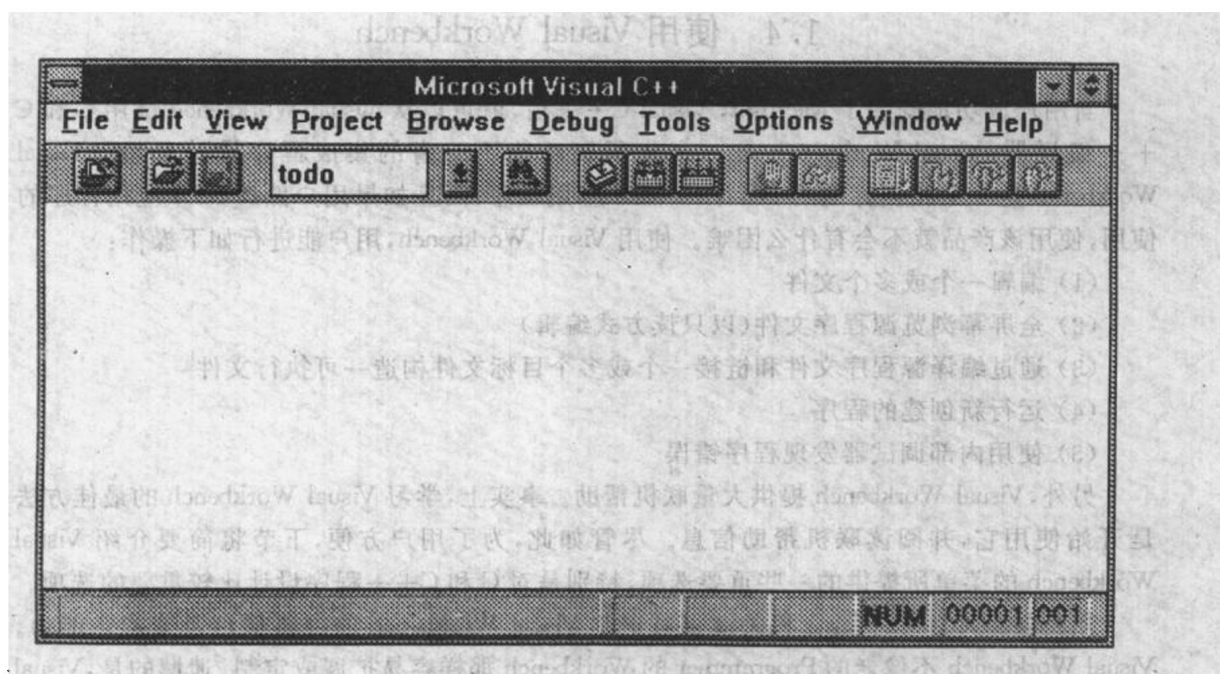


图 1.5 Visual C++ 的 Visual Workbench

请注意本书的命令语法。用方括号([...])括起的参数是可选的,而用尖括号(<...>)括起的参数则是必需的,英文形式的参数是替换项,用户在实际使用该命令时必须输入具体的值来替代之。例如,在下列命令中,<filename>是一替换项:

CL <filename>

另一方面,CL 是必须直接输入的项,因为它是黑体字。

1.4.2 Visual Workbench 的命令行参数

Visual Workbench 允许用户在启动时选用一些选项。这些选项只在用户使用 Visual Workbench 遇到问题时才需要(通常是编译时)。要了解 Visual Workbench 如何运行编译器,以及 Visual Workbench 怎样才能与编译器接口,用户必须清楚实际的编译器(和连接器及资源编译器)不是 Windows 应用程序。相反,它们是用一个名为 WinTee 的程序链接到 Visual Workbench 的 DOS(32 位)应用程序。

WinTee 是一个小的实用程序,它在 Visual Workbench 启动编译器时运行。该程序从编译器接收输出并将其发送到 Visual Workbench。这些输出信息(比如出错信息)然后被显示在 Visual Workbench 的 Output 窗口中。

由于这些运行编译器的方法比较复杂,所以经常出错。为了帮助用户消除这些错误,Visual Workbench 有一系列的选项,如表 1.1 所示。