

Microsoft

Microsoft

Microsoft



Visual C++[®]

自学指南

[美] Mark Andrews 著

蒋明 程昱 译

石志峰 审校



清华大学出版社



®

18717
13-2

Visual C++ 自学指南

〔美〕 Mark Andrews 著
蒋明 程昱 译
石志峰 审校

清华大学出版社

(京)新登字 158 号

Visual C++ 自学指南
Learn Visual C++ Now
Mark Andrews

Copyright © 1996 by Mark Andrews.

Original English language Edition Copyright © 1996 by Mark Andrews.

Published by arrangement with the original publisher, Microsoft Press, a division of Microsoft Corporation, Redmond, Washington, U. S. A.

本书中文版由 Microsoft Press 授权清华大学出版社出版。

中华人民共和国国家版权局著作权合同登记章 图字:01-95-317 号

版权所有,翻印必究。

本书封面贴有 Microsoft Press 激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

Visual C++ 自学指南/(美)安德鲁斯(Andrews, M.)著;蒋明,程昱译. —北京:清华大学出版社, 1997

书名原文: Learn Visual C++ Now

ISBN 7-302-02447-2

I. V... II. ①安... ②蒋... ③程... III. C 语言-程序设计-指南 IV. TP312C-62

中国版本图书馆 CIP 数据核字 (97) 第 02427 号

出版者: 清华大学出版社(北京清华大学校内, 邮编 100084)

印刷者: 北京市丰台区丰华印刷厂

发行者: 新华书店总店北京科技发行所

开 本: 787×1092 1/16 印张: 19 字数: 448 千字

版 次: 1997 年 6 月第 1 版 1997 年 6 月第 1 次印刷

书 号: ISBN 7-302-02447-2/TP · 1235

印 数: 0001~5000

定 价: 35.00 元

目 录

引言	1
0.1 使用本书时需要什么	1
0.2 获取联机帮助	1
0.3 本书内容	2
第一章 介绍 Visual C++	3
1.1 Visual C++ 编程环境	3
1.1.1 Visual C++ 编辑器	4
1.1.2 Visual C++ 向导和 App Studio	5
1.2 用 Visual C++ 编程	8
1.3 Visual C++ 的建立过程	8
1.4 MFC 库	9
1.4.1 MFC:“新的 Windows API”	9
1.4.2 MFC 和 Visual C++	10
1.5 编写 Visual C++ 程序	10
1.5.1 理解 Visual C++ 项目	10
1.5.2 分步编写 Visual C++ 程序	11
1.6 后续内容	14
第二章 Windows 程序设计简介	15
2.1 世界上第一个 C 语言程序	15
2.1.1 “Hello,world!”程序的结构	16
2.2 Windows 事件和消息:概述	16
2.2.1 消息队列和消息泵	17
2.2.2 窗口过程	17
2.2.3 消息处理器	18
2.3 例子:HELLO 程序	18
2.3.1 建立 HELLO 程序的方法	19
2.3.2 分步建立 HELLO 程序	19
2.4 HELLO 程序如何工作	22
2.4.1 WinMain 函数是如何工作的	23
2.4.2 登记一个窗口类	25
2.4.3 创建和显示窗口	27
2.4.4 在窗口中绘制文字	28
2.5 后续内容	31

第三章 C++ 基础	32
3.1 什么是面向对象的程序设计.....	32
3.2 老朋友,新面孔	33
3.2.1 类型说明符.....	33
3.2.2 限制符.....	39
3.3 C++ 类	42
3.3.1 声明类.....	42
3.3.2 访问说明符.....	43
3.3.3 内联成员函数.....	43
3.3.4 例子:EMPINFO 程序	44
3.3.5 构造函数和析构函数.....	47
3.3.6 缺省函数参数.....	48
3.3.7 拷贝构造函数.....	49
3.3.8 例子:CPYCONST 程序	50
3.3.9 初始化列表.....	51
3.3.10 作用域分解操作符	51
3.3.11 this 指针	52
3.4 后续内容.....	53
第四章 对象和成员函数	55
4.1 类层次结构和继承性.....	55
4.1.1 理解类层次结构.....	55
4.1.2 派生类如何工作.....	56
4.1.3 为什么使用派生类.....	56
4.2 例子:派生一个类	57
4.2.1 C++ 中继承的规则	59
4.2.2 声明一个派生类.....	59
4.3 构造派生类.....	59
4.4 重载成员函数.....	61
4.5 多态性和虚成员函数.....	64
4.5.1 例子:使用虚函数	64
4.5.2 VIRTUAL 程序是如何工作的.....	65
4.5.3 虚函数和非虚函数.....	66
4.5.4 使用虚函数的好处.....	66
4.5.5 V 表	67
4.5.6 纯虚函数与抽象类.....	71
4.5.7 在 EMPDATA 程序中是如何使用抽象类的	72
4.5.8 虚函数:优点和缺点	72
4.6 函数重载和操作符重载.....	72
4.6.1 函数重载.....	72

4.6.2 操作符重载	74
4.6.3 书写操作符重载函数	75
4.7 静态成员变量	77
4.7.1 创建静态成员变量	77
4.7.2 声明和定义静态成员变量	78
4.7.3 访问静态成员变量	78
4.7.4 私有静态成员变量	79
4.8 静态成员函数	80
4.9 友元类和友元函数	81
4.9.1 单向的友谊	81
4.10 new 和 delete 操作符	83
4.10.1 new 操作符	84
4.10.2 delete 操作符	84
4.11 后续内容	85
第五章 Visual C++ 工具	86
5.1 Visual C++ 项目	87
5.1.1 创建基本的 SCRAMBLE 项目	87
5.1.2 AppWizard 项目中的文件和类	90
5.1.3 SCRAMBLE 项目中的文件	90
5.2 加入一个位图:用 AppStudio 管理资源	91
5.3 编写代码以显示位图	94
5.3.1 对位图的操作	94
5.4 建立一个 Visual C++ 应用程序	97
5.4.1 编译一个应用程序	97
5.4.2 链接一个应用程序	99
5.5 执行 SCRAMBLE 程序	100
5.6 用 App Studio 编辑菜单	101
5.7 用 ClassWizard 创建消息处理器	103
5.8 为消息处理器编写代码	105
5.9 Visual C++ 调试器	106
5.9.1 设置断点	106
5.9.2 单步执行一个程序	106
5.9.3 打开调试器窗口	107
5.10 清单:SCRAMBLE 程序	107
5.11 后续内容	114
第六章 MFC 库	115
6.1 MFC 库概述	116
6.2 MFC 简史	117
6.2.1 MFC 版本 2.0	117

6.2.2	MFC:新的 Windows API	117
6.3	MFC 库中类的层次	119
6.4	MFC 框架类	122
6.4.1	CObject 类	122
6.4.2	CCmdTarget 类	122
6.4.3	CWinApp 类	123
6.4.4	CWnd 类	123
6.4.5	CFrameWnd 和 CMDIFrameWnd 类	124
6.4.6	CView, CScrollView 和 CDocument 类	124
6.4.7	CDocTemplate 类	125
6.5	MFC 程序是怎样工作的	125
6.5.1	WinMain 函数	125
6.5.2	InitInstance 成员函数	126
6.5.3	Run 成员函数	127
6.5.4	PumpMessage 成员函数	128
6.5.5	MFC 程序中的窗口过程	129
6.5.6	在 MFC 程序中使用文档和视图	129
6.5.7	文档模板	133
6.6	例子:改进后的 SCRAMBLE 程序	135
6.6.1	新的 SCRAMBLE 程序的实践	135
6.6.2	在新的 SCRAMBLE 程序中如何管理窗口	135
6.6.3	在 SCRAMBLE 程序中增加工具栏按钮	136
6.6.4	更新 SCRAMBLE 程序的菜单项	140
6.6.5	为 SCRAMBLE 程序创建单色的位图	142
6.6.6	为 SCRAMBLE 程序的视图增加滚动能力	144
6.6.7	定制程序窗口	146
6.7	后续内容	149
第七章	关于鼠标和消息	150
7.1	理解 Windows 消息	150
7.1.1	各种 Windows 消息	151
7.1.2	Windows API 类型的程序怎样处理命令消息	154
7.1.3	MFC 程序是怎样处理命令消息的	154
7.1.4	MFC 框架怎样分发消息	155
7.2	消息映像	156
7.2.1	使用消息映像的好处	156
7.2.2	将消息映像联接到你的程序中	157
7.2.3	说明一个消息映像	157
7.2.4	实现一个消息映像	158
7.2.5	ClassWizard 认识的消息	159

7.3 使用 ClassWizard 创建消息处理程序	161
7.3.1 至此的问题	162
7.4 为消息处理程序编写代码	163
7.4.1 不同的笔划	163
7.4.2 将笔划存储到数组中	164
7.4.3 逐步编写消息处理程序	164
7.4.4 把设备坐标转换为逻辑坐标	165
7.4.5 编写一个 OnMouseMove 消息处理程序	166
7.4.6 把笔划存储到文档中	169
7.4.7 在 SCRIBBLE 窗口中重绘笔划	171
7.4.8 创建和管理一个 CPen 对象	172
7.5 理解 MFC 库类	173
7.5.1 CObject 类的特性	173
7.6 文件和串行化技术	174
7.6.1 在 MFC 程序中打开文件	174
7.6.2 使用 CFile 类完成文件 I/O	175
7.7 串行化机制	176
7.7.1 流 I/O 中所使用的对象和操作符	177
7.7.2 使用<<和>>操作符	177
7.7.3 在 MFC 程序中实现串行化技术	178
7.7.4 在 AppWizard 程序中实现串行化技术	179
7.7.5 在 SCRIBBLE 程序中实现串行化技术	179
7.7.6 可串行化的数据类型	181
7.7.7 MFC 的串行化宏	182
7.7.8 在 MFC 程序中打开或关闭文档	183
7.7.9 清除 SCRIBBLE 程序的窗口	184
7.7.10 改变笔的宽度	185
7.8 后续内容	186
第八章 对话框	187
8.1 各种对话框	188
8.1.1 模式对话框	188
8.1.2 无模式对话框	188
8.1.3 消息框	188
8.2 对话框的组成元素	189
8.3 使用 App Studio 和 ClassWizard 设计一个对话框	189
8.4 把对话框集成到应用程序中	191
8.5 创建和显示一个模式对话框	191
8.5.1 调用对话框构造函数	192
8.5.2 调用 DoModal 函数	193

8.5.3 调用 OnInitDialog 函数	193
8.6 创建并显示一个无模式对话框	195
8.6.1 构造一个无模式对话框	195
8.6.2 调用 Create 函数	196
8.6.3 初始化一个无模式对话框	196
8.6.4 替换 OnOK 和 OnCancel	197
8.6.5 调用 PostNcDestroy	198
8.7 创建并显示一个消息框	199
8.8 对话框控制	199
8.8.1 按钮控制	199
8.8.2 编辑控制	202
8.8.3 静态文本控制	204
8.8.4 单选按钮	205
8.8.5 列表框	206
8.8.6 组合框	207
8.8.7 复选框	208
8.9 后读内容	209
第九章 管理数据	210
9.1 DDX 和 DDV 机制	211
9.1.1 老的方法	211
9.1.2 理解 DDX/DDV:TESTAPP 程序	211
9.1.3 创建 TESTAPP 项目并增加 DDX 支持	212
9.1.4 实现 DDX/DDV 机制	216
9.1.5 理解 UpdateData 命令	219
9.1.6 一种更方便的方法	219
9.1.7 逐步地调用 UpdateData 函数	220
9.1.8 运行 TESTAPP 程序	221
9.2 扩展 DDX/DDV:CREATION 程序	221
9.2.1 CREATION 程序的结构	224
9.2.2 创建一个 CPlayer 对象	226
9.2.3 Create A Character 对话框	229
9.2.4 Mel's Bait Shop And Fashion Boutique 对话框	235
9.2.5 Character Information 对话框	239
9.3 重温串行化技术	240
9.3.1 CREATION 程序的 Serialize 成员函数	241
9.3.2 CPlayer 类的 Serialize 成员函数	241
9.4 增加打印支持	243
9.4.1 AppWizard 的与打印有关的函数	243
9.4.2 在 CREATION 程序中进行专门打印	244

9.4.3 调用 OnPrint 成员函数	244
9.4.4 OnPrint 成员函数是怎样工作的	246
9.4.5 GetStats 成员函数	247
9.4.6 调用 GetEquipmentList 成员函数	247
9.5 后续内容	248
第十章 Visual C++ 图形	249
10.1 位图	250
10.1.1 各种 Windows 位图	251
10.2 设备相关位图(DDB)	252
10.2.1 创建 DDB	252
10.2.2 获取和保存 DDB	254
10.2.3 拷贝和显示位图	254
10.3 小型动画	258
10.3.1 逐步制作小型动画	258
10.3.2 在动画程序中使用帧缓存	259
10.3.3 避免在位图拷贝操作中出现闪烁和过快现象	260
10.3.4 在动画序列中计算边界矩形	261
10.3.5 分步动画	262
10.4 例子:GRAFDemo 程序	263
10.4.1 GRAFDemo 程序怎样工作	263
10.4.2 逐步分析 GRAFDemo 程序	263
10.4.3 在 GRAFDemo 程序中构造位图	265
10.4.4 准备好制作动画	266
10.4.5 GRAFDemo 程序的 OnDraw 函数	267
10.4.6 绘制背景	268
10.4.7 绘制玩家画像	268
10.4.8 移动小位图	270
10.4.9 计算边界矩形	271
10.5 设备无关位图(DIB)	272
10.5.1 DIB 怎样加速拷贝操作	272
10.5.2 使用 DIB 的缺点	273
10.5.3 DIB 的结构	274
10.6 调色板	279
10.6.1 系统调色板	280
10.6.2 缺省逻辑调色板	280
10.6.3 逻辑调色板	280
10.6.4 DIBDemo 程序怎样使用调色板	282
10.6.5 把系统调色板颜色映像到逻辑调色板	283
10.6.6 创建和使用逻辑调色板	283

10.6.7	使用等同调色板来简化 DIB 拷贝过程	283
10.6.8	创建一个等同调色板	284
10.7	创建和使用 DIB	284
10.7.1	DIB 拷贝操作	285
10.7.2	使用 StretchDIBits 函数	285
10.7.3	使用 SetDIBsToDevice 函数	286
10.8	例子: DIBDEMO 程序	287
10.8.1	DIBDEMO 程序怎样工作	287
10.8.2	把 DIB 取入内存	288
10.8.3	拷贝和显示 DIB	289
10.8.4	StretchDIBits 成员函数	290
10.9	后续内容	292

引 言

本书使用 Microsoft Visual C++ 来教你学习 C++ 语言,而 Microsoft Visual C++ 是编写基于 Windows 的程序的事实上的标准,也是用 C++ 创建及开发基于 Windows 的程序时使用最广泛的编译器。

本书面向的读者是那些具有 C 语言基础知识、而又想学习用 C++ 编写基于 Windows 的程序的程序员。除了教你学习 Visual C++ 的方方面面之外,本书还提供了 Windows 编程基础知识的介绍,以及一个涵盖两章的有关 C++ 语言及面向对象程序设计原则的综述。

本书还提供了本地球上最好的 Visual C++ 老师的私人帮助,这位老师就是 Visual C++ 向导。Visual C++ 向导包括 AppWizard 和 ClassWizard。只要通过菜单命令的选择,AppWizard 就可以生成一个能够工作的 Visual C++ 应用程序;ClassWizard 可以创建和管理 Visual C++ 程序中的类。如果想从这些向导中得到私人辅导的话,你所要做的就是提问。你可以通过研究 Visual C++ 向导生成的代码来学习 C++,Visual C++ 和好的程序设计风格。向导从来不会写错误的代码,因此你完全可以放心,你得到的将是符合 Microsoft 标准的、表现优秀的代码。

随着对 Visual C++ 编程的进一步熟悉,你就可以注意到这些向导是如何完成使应用程序运行起来所需的那些繁琐工作的,从而加深对基本原理的理解。这可能是目前所发现的学习 Visual C++ 和 Windows 程序设计的最不痛苦的办法。

0.1 使用本书时需要什么

本书带的 Visual C++ 版本在 Microsoft Windows 95 和 Microsoft Windows NT 以及更早版本的 Windows 下运行。它生成 16 位的应用程序,可以立即执行,无须编辑或改造。

要运行本书中的例子,需要有一个 Intel 386,486 或奔腾处理器及 Microsoft Windows 95,Microsoft Windows NT,或者 Microsoft Windows 3.0 或 3.1 版(最好是 Windows 3.1 版)。还需要一个有足够磁盘空间的硬盘来安装所需的选件,最少需要 4MB(最好 16MB)内存(依操作系统而定)。

注 如果在看完本书之后你决定升级到高版本的 Visual C++,那么你可以使用新的编译器重新编译本书给出的任何一个例子程序及任何一个用 Visual C++ 1.0 版编制的程序。甚至可以用 Borland C++ 5.0 及其以后版本(支持 MFC 库及用 Visual C++ 写的源代码)重新编译本书中给出的程序和你自己的 Visual C++ 1.0 版程序。

0.2 获取联机帮助

任何时候,只要查阅 Visual C++ 1.0 版所带的联机帮助文件,就可以得到帮助。进入

联机帮助,所要做的就是,在 Visual C++ 编辑器的 Help 菜单中根据需要帮助的类型选择相应的主题。帮助主题包括 C/C++ 语言,Windows 软件开发工具箱(SDK)及 Microsoft 基本类(MFC)库 2.0 版。Visual C++ 联机帮助文件还可以给你提供有关 Visual C++ 软件开发环境所带工具的信息。

0.3 本书内容

下面介绍一下本书所覆盖的主题。

第一章“介绍 Visual C++”让你熟悉一下 Visual 工作台,AppWizard 以及其它一些在用 C++ 开发应用程序时所用到的重要工具和编程原则。

第二章“Windows 程序设计简介”概述了一个基于 Windows 的程序的结构。在后面章节中研究 AppWizard 如何为 Visual C++ 程序构造框架时会用到本章所学的知识。

第三章“C++ 基础”介绍了普通 C++ 和面向对象程序设计的一些最重要的特征。本章解释了 C 和 C++ 之间的区别,并教给你如何创建及实现 C++ 类、成员变量和成员函数。

第四章“对象和成员函数”与第三章一样,旨在研究普通 C++ 及面向对象程序设计的基本原理。在本章中,我们要介绍在 C++ 程序中是如何使用对象及成员函数的,还要介绍 C++ 程序设计的其它一些重要原理,包括继承性、多态性、虚函数、函数重载和友元函数。

第五章“Visual C++ 工具”详细描述了如何使用 Visual C++ 中的编程工具,并通过一个简单易懂的例子程序的帮助,说明用 Visual C++ 编程与传统的 Windows API 风格的编程有何不同。

第六章“MFC 库”介绍了 MFC 库是如何通过加入一些新的为 Windows 程序员专门设计的类及成员函数来增强普通 C++ 和 Windows API 的。本章包括的主题有 CObject 类、CWnd 类及其它一些 MFC 库中的重要类。

第七章“关于鼠标及消息”介绍鼠标事件的概念,以及在 Visual C++ 程序中如何使用鼠标事件与用户交互。还介绍菜单、消息、消息处理函数和消息映射表。

第八章“对话框”向你介绍如何使用 AppStudio 设计对话框并用控制来装备它们。本章解释并将演示普通的对话框控制以及用户自己画的控制。一个例子程序给出如何在自己的 Visual C++ 程序中使用具有多种不同类型控制的消息框、无模式对话框和模式对话框。

第九章“管理数据”继续对话框的研究。本章介绍如何用 ClassWizard 为对话框类创建成员变量,以及如何在 Visual C++ 应用程序中将些成员变量作为对话框控制与成员函数之间的联接点。这一技术的关键就是 Visual C++ DDX(对话数据交换)和 DDV(对话数据确认)机制,用它们就可以在应用程序和对话框控制之间来回传递信息。

第十章“Visual C++ 图形”介绍 Windows 图形和动画,并给出如何将已有的图形例程融入你的 Visual C++ 应用程序。在本章中,你将学习使用设备相关位图(DDB)、设备无关位图(DIB)、sprite 图形、step 图形和透明位图拷贝。本章提供两个例子程序:一个用 DDB 来演示 sprite 图形的使用,一个用 DIB 来完成类似的任务。本章还包括一个有关 Visual C++ 及面向对象程序设计的其它参考资料的清单。

第一章 介绍 Visual C++

学习用 Microsoft Visual C++ 编程与学习用其它计算机语言开发软件有所不同。用旧的、传统的计算机语言(如 C, Basic 或 Pascal)设计程序时,一般需要自己去做每一件事情;想要计算机做些什么的时候,都要写一行或一块代码。

用 Visual C++ 编程时,很多工作是已经做好了。在大部分情况下,Visual C++ 都提供了计算机所需的用于完成你所交给它的任务的普通代码。然后,你可以对 Visual C++ 提供的代码进行添加或裁剪,从而创建更加具体的应用程序。

这一简便的方法可以使你不必编制为完成重复性任务(如处理键盘和鼠标操作及把窗口拖动到屏幕)所需的每行代码。用 Visual C++ 编程时,你就可以集中精力予那些更富创造力的工作——如编制那些使你所开发的应用程序真正有所创新、有所不同的代码。

本章介绍 Visual C++, 并带你熟悉组成 Visual C++ 程序设计环境的那些工具。还介绍 Microsoft 基本类(MFC)库 2.0 版,它是一个 C++ 类和成员函数的大库,这些类和成员函数是为用 Visual C++ 开发 Windows 程序而专门设计的。在本章最后,将会看到一个 Visual C++ 工具,一个名叫 AppWizard 的实用程序,是如何通过单击菜单项来生成一个完全可以运作起来的 Windows 应用程序的。在后面的章节中,将会学到如何编写代码,将由 AppWizard 生成的应用程序框架扩充为更加复杂的 Visual C++ 应用程序。

本章所包含的主题内容如下:

- Visual 工作台编程环境,它介绍 Visual C++ 编辑器和 Visual C++ 向导。
- Visual C++ 向导如何帮助你学习 Visual C++。
- 如何编译及链接 Visual C++ 程序。
- MFC 库——“新的 Windows API”——如何帮助你编制强有力的 Visual C++ 程序。
- 逐步指导你使用 Visual C++ 编写框架 MFC 程序。

1.1 Visual C++ 编程环境

用 Visual C++ 开发应用程序时,使用的是 Visual 工作台,有时简称为 VWB。Visual 工作台是 Visual C++ 带的一个主要的编辑与调试工具。它的角色就是 Visual C++ 程序设计的命令中心,并提供了许多程序设计用的实用程序,包括:

- AppWizard 应用程序生成器。
- AppStudio 资源管理器。
- ClassWizard 类管理器。
- Visual C++ 源浏览器。
- Visual C++ 调试器。

- Visual C++ 编辑器。
- Visual C++ 编译器。
- Visual C++ 链接器。

图 1.1 给出了 Visual 工作台窗口刚一打开时的样子。从这个主 Visual 工作台窗口可以访问一些编辑和编程的部件。Visual 工作台环境的大部分主要部件将在后面几节中介绍。随着本书的深入,你将学到有关其余部件的更多知识。

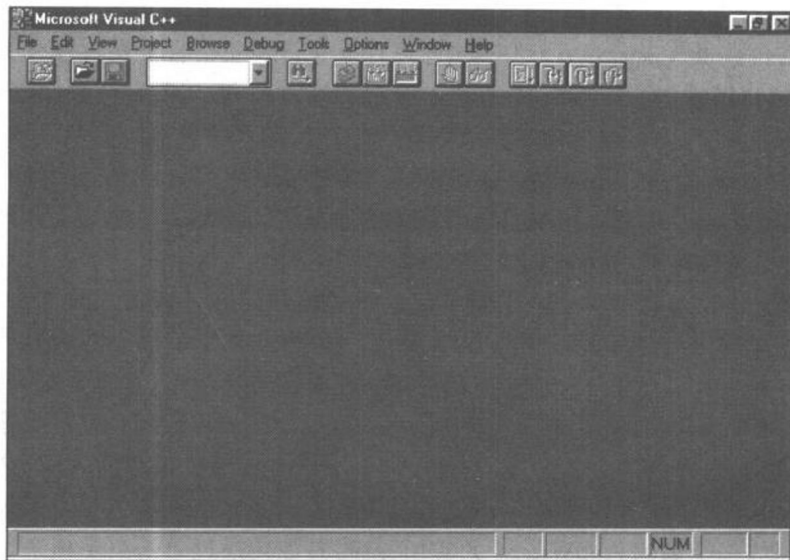


图 1.1 主 Visual 工作台窗口

1.1.1 Visual C++ 编辑器

启动 Visual C++ 时会看到 Visual C++ 编辑器窗口。Visual C++ 编辑器是一个标准的 Windows 文字编辑程序,它具有一些为编制 Visual C++ 程序而设计的特别特征。

其中一个特征就是一个内置的源浏览器,它用于记录对变量和 C++ 类的定义及引用。另一个特征是用不同的颜色来突出 Visual C++ 中用到的不同关键字。

这种彩色编码的语法特点可以使你很容易地发现 Visual C++ 程序中出现的某些特殊类型的单词或短语。例如,C 语言关键字是蓝色的,C++ 关键字用红色显示,注释是绿色的。调试器断点是红带中的反转显示。(这些是缺省设置;如果喜欢其它约定,则可以选择 Options 菜单的 Color 项,改变 Visual C++ 编辑器的缺省颜色设置。)

Visual C++ 编辑器其它特别的特点包括函数内的程序行的自动缩进,一个查找及替换的实用程序,可以使你立刻访问一行选中文本的书签,以及一个内置的源代码调试器。你可以从菜单栏或工具栏访问这些工具中的任何一个以及许多其它的工具,从工具栏访问时只要将鼠标指向它并单击,一共可访问 14 个普通的菜单命令。

快捷键

用 Visual C++ 编辑器写源代码时,可以使用一些特殊的键在源文件中编辑文本和来回移动。表 1.1 列出了 Visual C++ 编辑器承认的一些普通击键组合。

表 1.1 编辑操作的快捷键

快捷键	编辑操作
Ctrl-左箭头	向左移一个单词
Ctrl-右箭头	向右移一个单词
Home	移到本行的第一个行首缩进处
Home,Home	移到本行行首
Ctrl-Enter	移到本行的第一个行首缩进处
End	移到本行行尾
Ctrl-Home	移到文件首
Ctrl-End	移到文件尾
Ctrl-Z 或 Alt-Backspace	废除上一次编辑所做的改变
Ctrl-A	重复上次编辑
Ctrl-T	一直删到单词结束
Ctrl-C 或 Ctrl-Ins	将选定文本拷贝到剪贴板
Ctrl-X 或 Shift-Del	将选定文本剪切到剪贴板
Ctrl-V 或 Shift-Ins	从剪贴板粘贴文本
Ctrl-]	移到与之相对应的括号
Tab	插入一个制表符
Ctrl-Alt-T	切换制表符标记的显示

要想了解更多的有关 Visual C++ 编辑器及其诸多特征和功能的知识,就从 Visual 工作台的 Help 菜单选择 Visual Workbench 项,然后单击标有 Using The Editor 的那个按钮。

1.1.2 Visual C++ 向导和 App Studio

Visual 工作台带了一套称为“向导(Wizard)”的程序设计工具。在 Visual C++ 开发环境中,向导替你完成一系列复杂的任务,因此不必记住所有的细节。例如,叫做 AppWizard 的工具就可以通过单击一个菜单项来建立一个应用程序。另一个 Visual C++ 向导 ClassWizard 让你使用对话框控制将诸如菜单和对话框之类的资源与 Visual C++ 程序中的代码联系起来。让我们再详细一些地看看这些向导及一个称为 App Studio 的相关工具。

一、AppWizard

由 AppWizard 生成的应用程序有时被称为“应用程序框架”。应用程序框架是个最小的 Visual C++ 程序,你可以加入任何一些你自己的应用程序所需的具有特殊目的的代码,从而定制这个最小的程序。

图 1.2 给出了 AppWizard 打开时的样子。

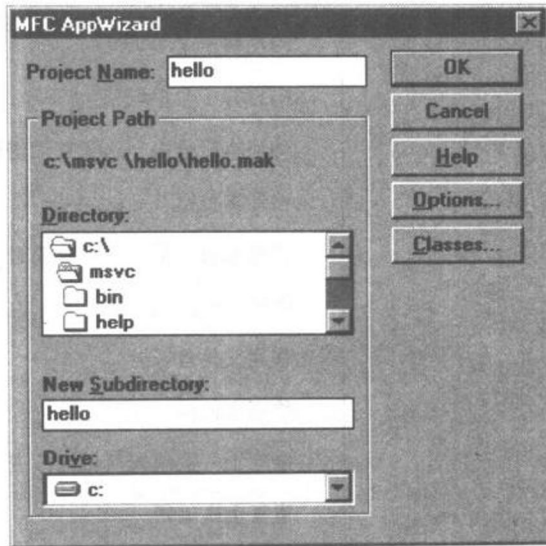


图 1.2 MFC AppWizard 对话框

只要从 Visual C++ Project 菜单选择 AppWizard 项就可以启动 AppWizard。然后 AppWizard 会显示各种对话框,你可以用它们来指定你要创建的应用程序的属性。结束的时候,AppWizard 生成一个能正常运行的 Visual C++ 应用程序,它符合你的指定并包含一个 Windows 应用程序所有的基本成分,这些成分包括以下内容:

- 一个主框架窗口及由被创建应用程序类型所需的其它窗口。
- 带有标准 Windows 菜单的一个菜单栏,如 File, Edit 及 Help。
- 打开文件、保存文件、打印文件及实现打印预览功能所需的所有菜单项和对话框。
- 一个工具栏和一个状态栏。
- 对容器和服务对象的 OLE 支持。
- 用 MFC 库中的 Visual C++ 类创建应用程序时所需的所有源文件和资源文件。

当 AppWizard 已经生成了应用程序框架时,就该由你来把这个框架变为你要设计的那种 Windows 应用程序。

在本章末尾,你将会有一个使用 AppWizard 创建应用程序框架的机会。在以后各章中,你将学会如何给一个应用程序框架添加代码,使得程序可以具有自己的一套有趣的功能。结束本书的时候,你就会知道用很多种不同的方法将 AppWizard 框架扩充为许多种不同的应用程序。

在第五章“Visual C++ 工具”中你还将学到更多的有关 AppWizard 和 MFC AppWizard 对话框的知识。