

中国文化艺术出版社

Bianchengzhinan

Visual C++6.0

编程指南

主编 / 张海棠



航空工业出版社

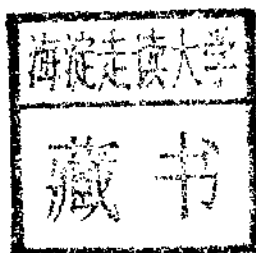
TP312
ZHT/1

Visual C++6.0 编程指南

主 编 张海棠

编 委 万 林 张 丽

邱晓光 郝敬霞



航空工业出版社

1999

055080

内 容 提 要

这本书不是供查阅各个函数和类方法的使用的参考手册，也不是一本 C++ 的教科书，它完全立足于 Visual C++ 的使用者，力图将他们引入编程的乐趣中来。“大而全”不是本书的目的，“少而精”才是我们所力图追求的。我们希望通过本书的学习，不仅能够让一位对 Visual C++ 少有所知的程序开发人员学会熟练地使用 Visual C++ 进行 Win32 应用程序的开发，而且对于初中级 Visual C++ 程序员，也能够从本书中找到一些值得一看的内容。因此，本书可能会对 MFC 中的一些内容略去不谈，但有时候可能会因为需要实现一些有趣的特性而深入到 MFC 的内部或者绕过 MFC 而直接使用 Windows API，这些都体现我们在实际编程中所遇到的实际情况——应用程序的需求有可能多种多样，其实现方式也不可能是千篇一律。本书的任务在于教会读者如何在 Visual C++ 中使用以“我”为中心的方式来开发 Windows 应用程序。

图书在版编目 (CIP) 数据

Visual C++ 6.0 编程指南 / 张海棠编. —北京: 航空工业出版社, 1999.9

ISBN 7-80134-540-1

I. V... II. 张... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (1999) 第 63832 号

航空工业出版社出版发行

(北京市安定门外小关东里 14 号 100029)

北京云浩印刷厂印刷

全国各地新华书店经售

1999 年 11 月第 1 版

1999 年 11 月第 1 次印刷

开本: 787×1092 1/16

印张: 34.5

字数: 818 千字

印数: 1-12000

定价: 46.80 元

本社图书如有缺页、倒页、脱页、残页等情况，请与本社发行部联系调换。联系电话: 010-65859701 或 64941995

前 言

Visual C++是开发运行于 Windows 95/98 和 Windows NT 环境下的 Win32 应用程序的可视化编程工具中最重要的成员之一，它为软件开发人员提供了完整的编辑、编译和调试工具和建立于 Win32 API(Application Programming Interface)基础上的 MFC 类库(Microsoft Foundation Class Library)，从而有效地缩短了 Windows 应用程序的开发周期。Windows 操作系统本身大部分是使用 C/C++ 语言写成的，而 Visual C++ 正是使用 C/C++ 语言的 Win32 应用程序集成开发环境，因此，使用 Visual C++ 来进行 Windows 应用程序的开发便有着得天独厚的优势，学习和掌握 Visual C++，也自然地成为了广大程序设计和开发人员的迫切需要。

全书共分为 14 章，涵盖了 3 个部分的内容：

第 1 部分包括第 1 章到第 3 章。这一部分介绍了使用 Visual C++ 进行 Windows 程序开发的必备知识。其中：

第 1 章：“Visual C++ 简介”介绍了 Visual C++ 6.0 的新增特性，Microsoft Developer Studio 集成开发环境的使用和定制，以及如何从 Visual C++ 6.0 的在线文档中获得所需的资料和信息。通过这一章，读者可以对 Visual C++ 及其所使用的集成开发环境有一个大致的了解，以为使用 Visual C++ 开发应用程序打下必要的基础。

第 2 章：“面向对象编程与 C++ 语言”分析和介绍一些在使用 C++ 语言进行面向对象的程序开发的过程中所常见的问题，而不是对 C++ 语言的完整的讲述。在这一章中，我们旨在通过指出一些可能引起程序出错的语言问题，来避免读者在编程过程中犯同样的错误。

第 3 章：“Win32 应用程序设计”讲述了使用 SDK 和 API 进行 32 位 Windows 应用程序设计的基本知识，这些知识是正确理解 MFC 工作方式的重要基础。

第 2 部分包括第 4 章到第 8 章。这一部分介绍了如何使用 Visual C++ 来设计 Windows 应用程序的用户界面。其中：

第 4 章：“基于对话框的应用程序”以基于对话框的应用程序为例讲述 MFC 应用程序框架、应用程序类以及应用程序的消息映射和对话框类等有关于 MFC 的基本概念。

第 5 章：“响应用户命令”讲述一般的菜单命令、工具条按钮命令和快捷键命令的响应和处理。此外，我们在本章也附带讲述了两种 Windows 控件——滑块控件和进度条的消息处理。

第 6 章：“使用 Windows 标准控件”讲述了几种标准的 Windows 控件的使用。我们并没有在这一章中涵盖了所有的 Windows 标准控件，而只是选取了其中一些具有代表性的来讲述，并希望通过它们来阐述使用 Windows 标准控件的一般方法。

第 7 章：“使用 ActiveX 控件”讲述如何使用新的 ActiveX 控件来增强应用程序的功能。作为示例，我们使用了功能强大的多媒体控件 ActiveMovie 来完成了一个可以打开多种媒体文件的视频播放器。

第 8 章：“文档/视结构”讲述了如何在基于文档的应用程序中使用 MFC 的文档/视结

构, 这种结构通过将文档中数据的维护与与用户的接口相分离, 从而使得程序的结构更加合理, 更便于维护, 同时也便于实现一些有用的特性, 如同一文档的多个视图的同步更新等。

第三部分包括第 9 章到第 13 章, 在这一部分中, 我们挑选了 Windows 下的一些编程课题来进行了一些一般性的讨论。其中:

第 9 章: “图形设备接口” 讲述了如何通过 Windows 的设备上下文来使用画笔、刷子及字体等多种绘图对象来进行图形的绘制和输出。在讲述的过程中, 我们还介绍了一些很有用的编程技巧, 如使用路径和剪辑区域来绘制特殊效果等。

第 10 章: “MFC 通用类” 介绍了 MFC 通用数据类型, 它们包括集合类、字符串类和日期及时间类等。这些通用类使用 MFC 应用程序的数据管理和使用更加方便。

第 11 章: “异常处理和诊断” 介绍了如何处理程序中出现的异常事件和使用 MFC 的诊断服务来检查程序中的错误, 这些方法对于创建强大的和无错的应用程序十分必要。

第 12 章: “多线程” 介绍了 32 位编程中线程和进程的概念, 以及如何充分使用 Win32 环境下的多任务功能。

第 13 章: “动态链接库” 介绍了动态链接库的创建和使用, 以及如何使用动态链接库来扩展 MFC 等。

第 14 章: 在本章中, 我们就前面所介绍的知识作一总结, 并就一般的参考书中较少提及的实用编程方法作一介绍。

熟悉 MFC 类库的内容和 Win32 API 中的有关函数是快速高效地进行 Win32 程序设计的必要条件, 然而一般情况下我们不可能记住数千个函数的功能和用法, 由于本书篇幅有限, 在每一章中也不可能面面俱到。立足于初中级 Visual C++ 编程人员的实际需要, 作者在本书中努力介绍一些实用的编程技巧, 指出一些人多数人在编程时可能犯的错误, 而不是全面地概括性地讲述 MFC 和 Windows 程序设计 (当然, 本书出于完整性的考虑, 仍有少量的这一类的概括性文字)。这本书不是供查阅各个函数和类方法的使用的参考手册, 也不是一本 C++ 的教科书, 它完全立足于 Visual C++ 的使用者, 力图将他们引入编程的乐趣中来。“大而全” 不是本书的目的, “少而精” 才是我们所力图追求的。我们希望通过本书的学习, 不仅能够一位对 Visual C++ 少有所知的程序开发人员学会熟练地使用 Visual C++ 进行 Win32 应用程序的开发, 而且对于初中级 Visual C++ 程序员, 也能够从本书中找到一些值得一看的内容。因此, 本书可能会对 MFC 中的一些内容略去不谈, 但有时候可能会因为需要实现一些有趣的特性而深入到 MFC 的内部或者绕过 MFC 而直接使用 Windows API, 这些都体现我们在实际编程中所遇到的实际情况——应用程序的需求有可能多种多样, 其实现方式也不可能是千篇一律。本书的任务在于教会读者如何在 Visual C++ 中使用以“我”为中心的方式来开发 Windows 应用程序。

本书由北京卓越文化艺术有限公司总策划。张海棠主编, 同时参加编写的还有万林、张丽、邱晓光、郝敬霞等。尽管我们细致地对书中的每一行程序进行了反复的调试, 但是仍有可能出现疏漏, 恳请读者批评指正。

编 者

1999 年 10 月

目 录

第 1 章 Visual C++ 简介	1
1.1 Visual C++ 和 MFC 的历史	1
1.2 Visual C++ 6.0 的不同版本及特点	2
1.3 Visual C++ 6.0 的新特性	3
1.3.1 专业版	3
1.3.2 企业版	5
1.3.3 学习版	6
1.3.4 集成开发环境	6
1.4 Developer Studio 的使用	7
1.5 获得帮助	11
1.5.1 使用帮助	11
1.5.2 使用上下文相关的帮助	14
1.5.3 Developer Studio 与 Web	17
1.6 自定义 Developer Studio	18
1.6.1 自定义工具条和菜单	18
1.6.2 自定义快捷键	20
1.6.3 使用宏	20
第 2 章 面向对象编程与 C++ 语言	23
2.1 面向对象的编程技术	23
2.1.1 面向对象的程序设计	24
2.1.2 封装	25
2.1.3 继承	26
2.1.4 多态和虚函数	28
2.2 类的声明和定义	30
2.2.1 类及其成员变量和成员函数的声明和定义	30
2.2.2 成员函数和 this 指针	37
2.2.3 静态成员	41
2.2.4 联合	44
2.2.5 构造函数和析构函数	45
2.2.6 友元	52
2.2.7 运算符重载	55
2.3 类的继承	62

2.3.1	单一派生.....	62
2.3.2	多重继承.....	64
2.4	多态与虚函数.....	74
2.4.1	虚函数.....	75
2.4.2	纯虚函数与抽象类.....	76
2.5	ClassView 和 WizardBar.....	78
2.5.1	使用 ClassView.....	78
2.5.2	使用 WizardBar.....	82
第 3 章	Win32 应用程序设计.....	85
3.1	事件驱动的应用程序.....	86
3.2	Win32 API 和 SDK.....	87
3.3	使用 SDK 编写 Windows 应用程序.....	88
3.4	32 位编程的特点.....	105
第 4 章	基于对话框的应用程序.....	115
4.1	使用 AppWizard 生成应用程序框架.....	115
4.2	应用程序类.....	121
4.3	MFC 应用程序的消息循环.....	130
4.4	对话框类.....	135
第 5 章	响应用户命令.....	148
5.1	菜单消息响应.....	148
5.2	工具条.....	174
5.3	快捷键消息响应.....	185
5.4	滑块控件消息响应.....	200
5.5	进度条消息响应.....	207
5.6	上下控件消息响应.....	212
第 6 章	使用 Windows 标准控件.....	216
6.1	使用对话框编辑器和 ClassWizard.....	217
6.2	所有窗口类的基类: CWnd.....	223
6.3	按钮.....	233
6.3.1	下压按钮.....	233
6.3.2	位图按钮.....	242
6.3.3	组框.....	244
6.3.4	单选钮.....	245
6.3.5	复选框.....	249

6.4	静态控件	254
6.5	文本编辑控件	257
6.6	列表框控件	267
6.7	组合框	277
6.8	滚动条控件	286
第 7 章	使用 ActiveX 控件	292
7.1	什么是 ActiveX 控件	292
7.2	使用 ActiveMovie 控件的视频播放器	293
第 8 章	文档/视结构	316
8.1	文档/视结构概述	316
8.2	使用 AppWizard 创建框架应用程序	319
8.3	生成文档	321
8.3.1	概述	322
8.3.2	把文档数据保存到成员变量中	322
8.3.3	串行化数据	326
8.3.4	使用集合类管理数据	331
8.3.5	串行化对象	342
8.4	生成视	343
8.5	视类	373
8.6	同一文档的多个视	376
8.7	添加对多文档类型的支持	387
第 9 章	图形设备接口	388
9.1	设备上下文	389
9.1.1	几个与图形绘制有关的简单数据类型	390
9.1.2	显示设备上下文	392
9.2	画笔对象	399
9.2.1	创建画笔	399
9.2.2	使用画笔在设备上下文中输出	400
9.2.3	图形输出函数	404
9.3	刷子对象	410
9.4	字体对象	415
9.4.1	创建字体对象	415
9.4.2	LOGFONT 结构	415
9.4.3	使用字体对象并枚举系统中的所有字体	417
9.4.4	创建特殊的字体效果	426

9.5 映射模式	433
9.5.1 预定义的映射模式	434
9.5.2 自定义的坐标系统	442
第 10 章 MFC 通用类	446
10.1 数组类	446
10.2 列表类	453
10.3 映射类	458
10.4 字符串类	462
10.5 日期和时间类	465
第 11 章 异常处理和诊断	470
11.1 处理 C++ 异常	470
11.2 MFC 异常	470
11.3 诊断服务	474
第 12 章 多线程	478
12.1 创建线程	478
12.2 线程间通信	480
12.3 线程同步	485
第 13 章 动态链接库	494
13.1 概述	494
13.2 创建和使用动态链接库	496
13.2.1 DLL 的结构和导出方式	496
13.2.2 链接应用程序到 DLL	500
13.3 使用动态链接库扩展 MFC	503
第 14 章 一些扩展	509
附 录	531
附表 1 MFC 类库层次表	531
附表 2 ASCII 码表 (0~127)	537
附录 3 虚拟键码	537

第 1 章 Visual C++ 简介

只要提到在 Windows 95/98 和 Windows NT 下进行 32 位的应用程序开发,就不能不提到 Visual C++。相比其他的编程工具而言,Visual C++在提供可视化编程方法的同时,也适用于编写直接对系统进行底层操作的程序,其生成代码的质量,也要优于其他的许多开发工具。Visual C++所提供的 Microsoft 基础类库(Microsoft Foundation Class Library, 简称为 MFC),对 Windows 95/98/NT 所用的 Win32 应用程序接口(Win32 Application Programming Interface)进行了十分彻底的封装,这使得可以使用完全的面向对象的方法来进行 Windows 95/98/NT 应用程序的开发,从而大大地缩短了应用程序的开发周期,降低了开发成本,也使得 Windows 程序员从大量的复杂劳动中出来,随着对 Visual C++了解的逐步深入,你会亲自感受到这一点。Visual C++使 Windows 编程在确保应用程序性能的同时,成为一件有趣的事情。

在本章中,我们将讲述:

- Visual C++ 和 MFC 历史。
- Visual C++ 6.0 的不同版本和它们的区别。
- Visual C++ 6.0 的新特性。
- Visual C++ 6.0 集成开发环境的使用。

1.1 Visual C++ 和 MFC 的历史

Visual C++的核心是 Microsoft 基础类库,即通常所说的 MFC。尽管使用 Visual C++编程并不一定要使用 MFC,使用 MFC 也不一定就要使用 Visual C++, Borland C++的新版本也提供了对 MFC 的支持,然而事实上,在很多情况下,我们提到 Visual C++时指的就是 MFC,而提到 MFC 时指的也就是 Visual C++。因此,在绝大多数情况下,Visual C++和 MFC 是指同一样东西。

MFC 相当彻底地封装了 Win32 软件开发工具包(Software Development Kit, 即通常所说的 SDK)中的结构、功能,为编程者提供了一个应用程序框架。这个应用程序框架为编程者完成了很多 Windows 编程中的例行性工作,如管理窗口、菜单和对话框、执行基本的输入和输出、使用集合类来保存数据对象等等。并且, MFC 使得在程序中使用很多过去很专业、很复杂的编程课题变得更加容易,如 ActiveX、OLE、本地数据库和开放式数据库互联(Open Database Connectivity, 简称为 ODBC)、Windows 套接字和 Internet 应用程序设计等,以及其他的应用程序界面特性,如属性页(也叫标签对话框)、打印和打印预览、浮动的和可定制的工具条。

早在 1989 年, Microsoft 的程序员开始试图将 C++ 和面向对象的编程概念应用于

Windows 编程中,以编写出一个可以使 Windows 编程工作更加简便的应用程序框架。他们把这个应用程序框架叫做 AFX (AFX 这个词来源于 Application Framework,但奇怪的是这个词组中并没有包含“X”这个字母)。直到今天,AFX 小组早已不存在了,AFX 这个名称也于 1994 年初不再使用,但在 Visual C++ 和 MFC 中,AFX 的影子却随处可见,很多全局函数、结构和宏的标识符都被加上了 AFX 的前缀。

最初的 AFX 版本在经过一年的艰苦工作之后诞生,却未能被大多数 Windows 程序员所接受。AFX 的确是经过了精心的规划和编码,并且,它也提供了对 Windows API 的高度抽象,建立了全新的面向对象的 AFX API,但最要命的是 AFX API 库根本不兼容现有的 Windows API。由此导致的最严重后果是大量的 SDK 代码无法移植,而程序员将学习两种完全不同的编程方法。

AFX 不得不重新做所有的一切,他们所创建的新的应用程序框架是一套扩展的 C++ 类,它封装和映射了 Windows API,这就是 MFC 的前身。过去的 AFX 小组也变成了 MFC 小组。最终,MFC 的第一个公开版本于 1992 年 3 月随 Microsoft C/C++ 7.0 (而不是 Visual C++ 1.0)一起推出。那时距 Windows 3.1 发布尚有好几个月。在 MFC 1.0 中还没有文档/视图结构,但有 CObject 和 CArchive 类。在 12 个月之后,MFC 2.0 随 Microsoft 新的编程工具 Visual C++ 1.0 一道出炉。与 MFC 1.0 一样,MFC 2.0 仍是 16 位的,因为 32 位的 Windows NT 3.1 直到 1993 年 7 月才问世。在 MFC 2.0 中,增加了对文档/视图结构、OLE 1.0、Windows 3.1 公用对话框的支持和消息映射等功能。在 Windows NT 3.1 面世一个月以后,Microsoft 推出了 32 位版本的 Visual C++ 和 MFC 2.1,它实际上是 MFC 2.0 的 Win32 接口。

最后一个 16 位的 Visual C++ 编译器是 1993 年 12 月推出的 Visual C++ 1.5,直到今天,一些为 Windows 3.1 编写 16 位应用程序的程序员还在使用这个版本。1994 年 9 月,32 位的 MFC 3.0 伴随着 Visual C++ 2.0 一道面市。在今天的计算机图书市场上,还有大量的关于 Visual C++ 2.0 和 MFC 3.0 的图书,因此,可以想象得出 Visual C++ 2.0 所取得的成功和它所产生的影响。在 Visual C++ 5.0 中包括的 MFC 版本不是 MFC 5.0,而是 MFC 4.21。在 1998 年,Microsoft 公司终于推出了其最新的 Visual C++ 版本——Visual C++ 6.0。发展到今天,MFC 已发展成一个稳定和涵盖极广的 C++ 类库,为成千上万的 Win32 程序员所使用。MFC 库是可扩展的,和 Windows 技术的发展始终是同步的。并且,MFC 类库使用了标准的 Windows 命名约定和编码格式,所以有经验的 Windows SDK 程序员很容易过渡到 MFC。MFC 结合了 Windows SDK 编程概念和面向对象的程序设计技术,从而具有极大灵活性和易用性。

1.2 Visual C++ 6.0 的不同版本及特点

Visual C++ 6.0 是 Microsoft 于 1998 年 6 月推出的最新的 Visual C++ 编译器,它包括三个版本。各个版本之间的区别如表 1-1 所示:

表 1-1 Visual C++ 6.0 的不同版本

版 本	特 点
学习版 (Learning Edition)	除了代码优化、剖析程序(一种分析程序运行时行为的开发工具)和到 MFC 库的静态链接外, Visual C++ 6.0 学习版提供了专业版的其他所有功能。学习版的价格要比专业版本低很多, 这是为了使希望使用 Visual C++ 6.0 来学习 C++ 语言的个人也可以负担得起。但不能使用 Visual C++ 6.0 学习版来开发供发布的软件, 其授权协议明确禁止这种做法
专业版 (Professional Edition)	此版本的 Visual C++ 6.0 可用来开发 Win32 应用程序、服务和控件。在这些应用程序、服务和控件中可使用由操作系统提供的图形用户界面或控制台 API
企业版 (Enterprise Edition)	可用来开发和调试为 Internet 或企业内部网(Intranet)设计的客户-服务器应用程序。Visual C++ 6.0 企业版还包括了开发和调试 SQL 数据库应用程序和简化小组开发的开发工具

1.3 Visual C++ 6.0 的新特性

如果你没有使用过 Visual C++ 6.0, 或者虽然使用过 Visual C++ 6.0, 但对它还不是非常熟悉, 那么可以跳过这一节的内容, 继续阅读本书的其他内容。在这一节里列举了 Visual C++ 6.0 中新增的所有特性, 这些内容对熟练掌握了 Visual C++ 6.0 的程序员来说是很 useful 的, 他们可以通过阅读本节了解到 Visual C++ 6.0 所做的改进, 从而知道自己应该补充的是哪一方面的内容。而对于 Visual C++ 的初学者和刚入门者来说, 了解这些内容就不是那么有必要了, 尤其是当你急切地想进入 Visual C++ 6.0 的编程实践中去的时候。

下面我们分版本来讲述 Visual C++ 6.0 相对于上一个版本新增加的内容。

1.3.1 专业版

在 Visual C++ 6.0 专业版中包括如下的新增特性(其中一些特性是在 Visual C++ 5.0 中就已经提出的, 我们在这里也顺便列出, 以使读者对其构成有较全面的认识):

1. C++ 语言

新增下列 C++ 关键字: bool、explicit、false、mutable、true 和 typename。

允许使用 __declspec 来声明指定的存储类属性是应用于某一类型还是某一类型的一个变量。

2. 编译器、链接器和 NMAKE

编译器添加了对 COM 的支持, 从而简化了使用 COM 对象的 C++ 客户的开发。为了演示如何使用该特性来支持 COM, 新增了如表 1-2 所示的示例程序。

表 1-2 为演示编译器对 COM 的支持而新增的示例程序

示例程序	演示内容
ACDUAL	MFC 应用程序中双界面的支持

INPROC	进程内自动化服务器
MFCALC	使用自动化服务器实现的一个简单计算器
COMEXCEL	单独运行的自动化客户程序。该程序创建一个新的 Microsoft Excel 电子表格, 并生成饼图
COMIDE	单独运行的自动化客户程序。该程序自动操纵 Microsoft Developer Studio
COMMAIL	单独运行的自动化客户程序。该程序为 Microsoft Exchange 4.0 自动操纵了 MAPI
COMMAP	不同 COM 接口入口映射宏的使用
LABRADOR	ATL 的使用。该程序实现了一个没有用户界面的 EXE 服务器
FRETHREAD	使用编译器的 COM 支持编写多线程客户程序和自由线程服务器
ALLINONE	MAC、STL、ATL 和 COM 的使用

使用/O 选项编译生成的代码大小将比 Visual C++ 4.2 版小 5%~10%。

链接器使用了/FIXED 选项来创建更小的供发行的应用程序。因此, 在使用剖析程序时, 由于需要重定位信息, 链接器必须使用/PROFILE 和/FIXED:NO 选项。这同样适用于其他如 BoundsChecker 或 Purify 之类的链接后(post-link)工具。

新增的/EH 编译选项可以更有效地控制 C++ 异常处理。C++ 同步异常处理允许编译器生成更小的代码, 因此它是 Visual C++ 6.0 新的默认 C++ 异常处理模式。

对用来控制代码优化所面向的编译器的编译器选项/G3、/G4、/G5、/G6 和/GB 作了修改。

将/GX 编译器选项映射为/EHsc。

允许使用链接器选项/PDBTYPE 指定包括调试信息的程序数据库(PDB)。该选项可以节省磁盘空间并加快链接。

在 NMAKE 中支持批处理规则。

3. AppWizard

新的 AppWizard 可以自动管理基于对话框的应用程序中的对话框类。只需要简单的创建一个基于对话框的应用程序, 并选择对自动化的支持, 就可以像早期版本的 AppWizard 一样, 得到一个支持基本自动化的基于对话框的应用程序。通过单独的代理类, 对话框类也可以通过自动化导出。可以添加方法和属性来导出对话框中的元素。

定制的 AppWizard 可以改变工程创建时的设定。例如, 可以在目标创建之后调整编译器、链接器和查看设定或者添加定制的创建步骤。

4. MFC

asynchronous (URL) moniker 允许应用程序异步地下载文件和控件属性, 以便在任务完成后为其他进程释放系统资源。

可以在 Web 浏览器(如 Internet Explorer 3.0)或支持 ActiveX 文档的 OLE 容器(如 Microsoft Office Binder)的整个客户区显示活动的文档。

Win32 Internet API (WinInet)使 Internet 成为任意应用程序的一个完整部分并简化了 Internet 服务, 如 FTP、HTTP 和 gopher 的访问。

增加了对 DAO 3.5 的支持。

增加了对 ODBC 3.0 的支持, 并对 MFC ODBC 类作了几个重要的修改。

COleDateTime 成员函数 SetDate、SetDateTime、SetTime 的返回值从 BOOL 改变为 int。每一个成员函数当 COleDateTime 对象被正确设置时返回 0, 否则返回 1。该返回值基于 DateTimeStatus 枚举类型。

新增示例程序 IMAGE。该程序生成一个可以异步下载数据的 ActiveX 控件。

5. Active Template Library (ATL) 2.1

ATL 2.1 版支持创建既小又快的 ActiveX 控件。

6. C Runtime Library

新增的函数 _itoa、_i64toa 和 _ui64toa 将数据转换为一个以 null 结尾的字符串。所对应的宽位字符版本为 _itow、_i64tow 和 _ui64tow 为 _itoa、_i64toa 和 _ui64toa。

改善了下列的通用浮点超越函数的性能: pow、sqrt、log、log10、sin、cos、tan、asin、acos、atan。

改善了内存移动和内存拷贝函数的性能。

7. ANSI 标准 C++ 库

Visual C++ 的标准库遵从 1996 年 9 月 24 日公布的 ANSI C++ (X3J16) 工作单——ANSI Doc No. X3J16/96-0178 WG21/N0996。该标准于 1996 年 7 月在 Stockholm 会议上制定。

8. OLE DB

OLE DB 是一组 OLE 接口, 可以使应用程序以统一的方式访问保存在不同信息源中的数据。这些接口支持适合于数据源的大量数据库功能性, 并允许数据源共享其数据。配套的 OLE DB 软件开发工具包提供的一组软件部件、工具和文档可以开发 OLE DB 客户并提供帮助。

9. ERRLOOK 工具

ERRLOOK 工具可以使用系统错误的值来检索相应的错误消息, 其中包括 OLE HRESULT。错误值可以通过拖放、编辑命令等多种方法给出。由 ERRLOOK 所返回的错误消息文本可以复制并粘贴到其他应用程序中。

除了以上新增特性之外, 在 Visual C++ 6.0 光盘上的 \DEVSTUDIO\VC\SAMPLES 目录下还包括了一些新增的示例程序。

1.3.2 企业版

除包括专业版中的所有特性外, Visual C++ 6.0 企业版还包括下列特性以支持企业级应用程序的开发:

Microsoft Transaction Server (Microsoft 事务服务器)用于创建基于事务的应用程序。

Visual Database Tools (可视化数据库工具)提供了数据库和 SQL 查询的图形化设计。

调试数据库连接时具有更好的性能, 该进程将比过去快许多。

扩展的 SQL 数据类型支持可以方便地将局部变量变为除 text 和 image 外的所有 SQL 数据类型。这种变换包括 money 类型和 datetime 类型。可以在局部变量和 NULL 值之间相互变换。还可以方便地查看包括 text 和 image 在内的所有 SQL 数据类型。

在光盘上的\DEVSTUDIO\VC\SAMPLES 目录下包括了特定于企业版的新增示例程序。

1.3.3 学习版

Visual C++ 6.0 学习版包括了学习 C/C++ 和使用 MFC、OLE、ODBC、DAO、ActiveX 和 COM 的各种工具，但不包括下面的特性：

- 与 MFC 的静态链接。

使用 Visual C++ 6.0 学习版编写的应用程序只能在运行时链接到 MFC 动态链接库。

- 代码优化。

不能使用/O 选项来生成更小和更快的代码。

- 程序剖析。

不能使用剖析程序来分析程序代码中的某一部分是否可以从性能改进中获益。

- RemoteData 控件及其他的数据绑定控件。

除了上面的内容外，Visual C++ 6.0 学习版包括专业版中的其他新特性。

1.3.4 集成开发环境

Microsoft Developer Studio 可用于 Visual J++ 1.1、Visual InterDev、Visual C++ 6.0 和 MSDN 等程序的开发。新的 Developer Studio 包括以下的新特性：

- 自动化和宏。

可以使用 Visual Basic 脚本来自动操纵例行的和重复的任务。可以将 Visual Studio 及其组件当作对象来操纵，还可以使用 Developer Studio 对象模型创建集成的附加程序。

- ClassView。

使用文件夹来组织 C++ 和 Java 中的类，包括使用 MFC、ATL 创建或自定义的新类。

- 可定制的工具条和菜单。

- 连接到正在运行的程序并对其进行调试，还可以使用宏语言来自动操作调试器。

- 可以在 Developer Studio 中查看 Internet 上的 World Wide Web 页。

- 可以在一个工作空间中包括多个不同类型的工程。

工作空间文件使用扩展名.dsw 来代替过去的扩展名.mdp，工程文件使用扩展名.dsp 来代替过去的扩展名.mak。

- 改进的资源编辑器。

在 Visual C++ 中，可以使用 WizardBar 来将代码与程序中的可视元素挂钩。

快捷键、二进制、对话框和字符串编辑器支持定位至快捷键、ASCII 字符串、十六进制字符串、控件 ID 和标签及指定字符串的 Find 命令更方便地一次修改多个项(可以快捷键、对话框、菜单和字符串)的属性。

- 改进的文本编辑器。

可以使用正确的句法颜色设置来显示无扩展名的头文件。

可以定制选定页边距的颜色来更好地区分同一源代码窗口中的控件和文本区域。Find in Files 命令支持两个单独的窗格。

- 上下文相关的 What's This 帮助 (需要说明的是, 在 Visual C++ 6.0 中, 当我们通过开发环境中的 Help 菜单寻找“帮助”时, 系统将调用单独的帮助程序, 这应该是相对于 Visual C++ 5.0 的一个相当大的改进)。

- 便捷的类库提示。

相信使用过 VB 的读者一定会注意到在 VB 中的便捷的函数提示。在 Visual C++ 6.0 中, 在输入时在线的丰富提示也实现了。的确, 这使得程序员从记忆众多的函数参数中解脱出来, 从而极大地提高编程效率。

- 改进的 WizardBar。

可用于 Visual J++ 程序的编写。在 Visual C++ 6.0 中, 集成环境有了更大的用处: 在 Visual Studio 中, 不仅可以进行 Visual C++ 程序的编写, 而且也可以同时在其中编写 Visual J++ 程序。

- 新增的向导。

添加了集成到 Visual J++ 和 Visual InterDev 中的向导。

1.4 Developer Studio 的使用

在一整套的 Visual Studio 98 中, Visual C++ 6.0、Visual J++ 和 Visual InterDev 都使用同一个开发环境, 称作 Developer Studio。你可在 Developer Studio 中创建所开发的应用程序的源文件、各种资源文件及其他文档。这些文件以工作空间和工程的形式进行组织。Developer Studio 中一次只能打开一个工作空间, 在同-个工作空间中可以包括多个工程。一般来说, 每一个工程就是开发的一个应用程序。这些工程相互之间可以具有某种联系和从属关系, 也可以彼此完全独立。此外, 这些工程还可以是不同类型的。比如说, 可以在已经包括有 Visual C++ 工程的工作空间中添加一个 Visual J++ 或是 Visual InterDev 工程。工程中除了包括应用程序所用到的源文件、资源文件外, 还可以包括其他类型的文件, 如应用程序的规格说明书、流程图、开发日程等等。对于那些由 ActiveX 部件(如 Microsoft Word 等)所创建的 ActiveX 文档, 可以在 Developer Studio 中直接打开。而对于那些与其他类型的应用程序相关联的文档, 也可以通过 Developer Studio 在独立的窗口中打开。

Developer Studio 所包括的内容是很丰富的。本节只讲述一些基本的概念和用法。这些用法是进行下一步所不可缺少的。对一些特殊的用法, 我们将在本书后面的章节中需要用时再作说明。

本书假定读者已经是一个熟练的 Windows 95/98/NT 用户, 因此, 安装 Visual C++ 的过程应该是一个相对很容易的事, 所以我们不作介绍。下面的过程中, 我们假定已在计算机中安装上了 Visual C++ 6.0, 在一般情况下, 这同时也安装了 Developer Studio, 并以它作为 Visual C++ 6.0 的集成开发环境。

这时, 可从“开始”菜单中运行 Visual C++ 6.0, 屏幕显示如图 1-1 所示。图 1-1 还标注了 Developer Studio 中的几个常提到的部件的名称, 以便以后提到这些名词时, 你可以知道它们所指的是 Developer Studio 中的哪一部分。

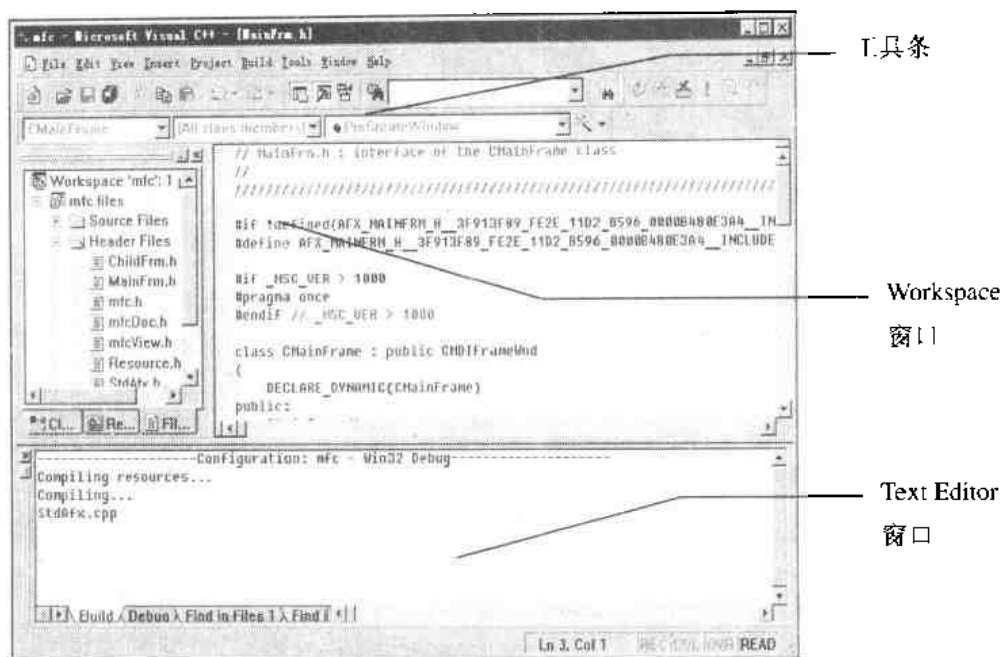


图 1-1 Microsoft Developer Studio - Visual C++ 6.0 的集成开发环境

当然，你所看到的内容和工具条等也许会有不同，这取决于具体的设置。下面我们来讲述其中的主要部分。在 Developer Studio 中，整个窗口被分成了若干个部分，需要注意的是，随着设置的不同，或者所安装的软件包的不同，或者是处于开发的不同阶段(典型地，在输入源代码和调试程序的两个不同阶段)，你所见到的 Developer Studio 组件和相互之间的位置也会不一样。

前面说到过，在 Developer Studio 中，我们是以工作空间(workspace)和工程(project)来组织文件和进行工作的。工作空间位于这个结构的最顶层，因此，我们首先需要创建一个工作空间。创建工作空间通常有两种方法：

第一种方法是显式地创建一个空白的工作空间，然后向工作空间中添加工程。我们从 Developer Studio 的 File 菜单下选择 New...命令，出现如图 1-2 所示的对话框。

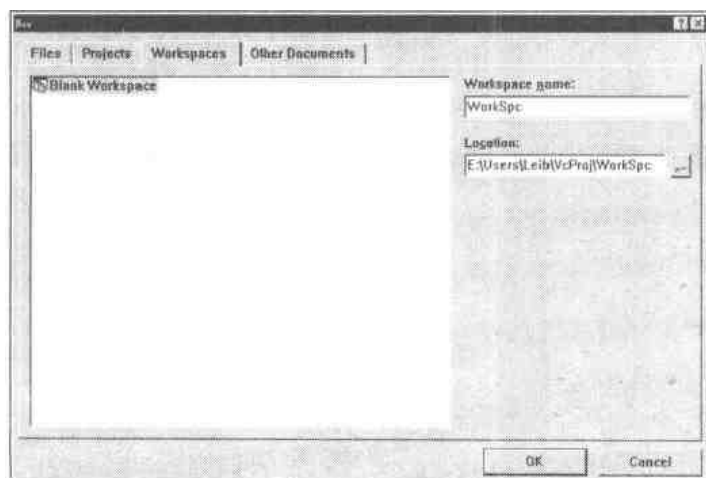


图 1-2 创建空白工作空间