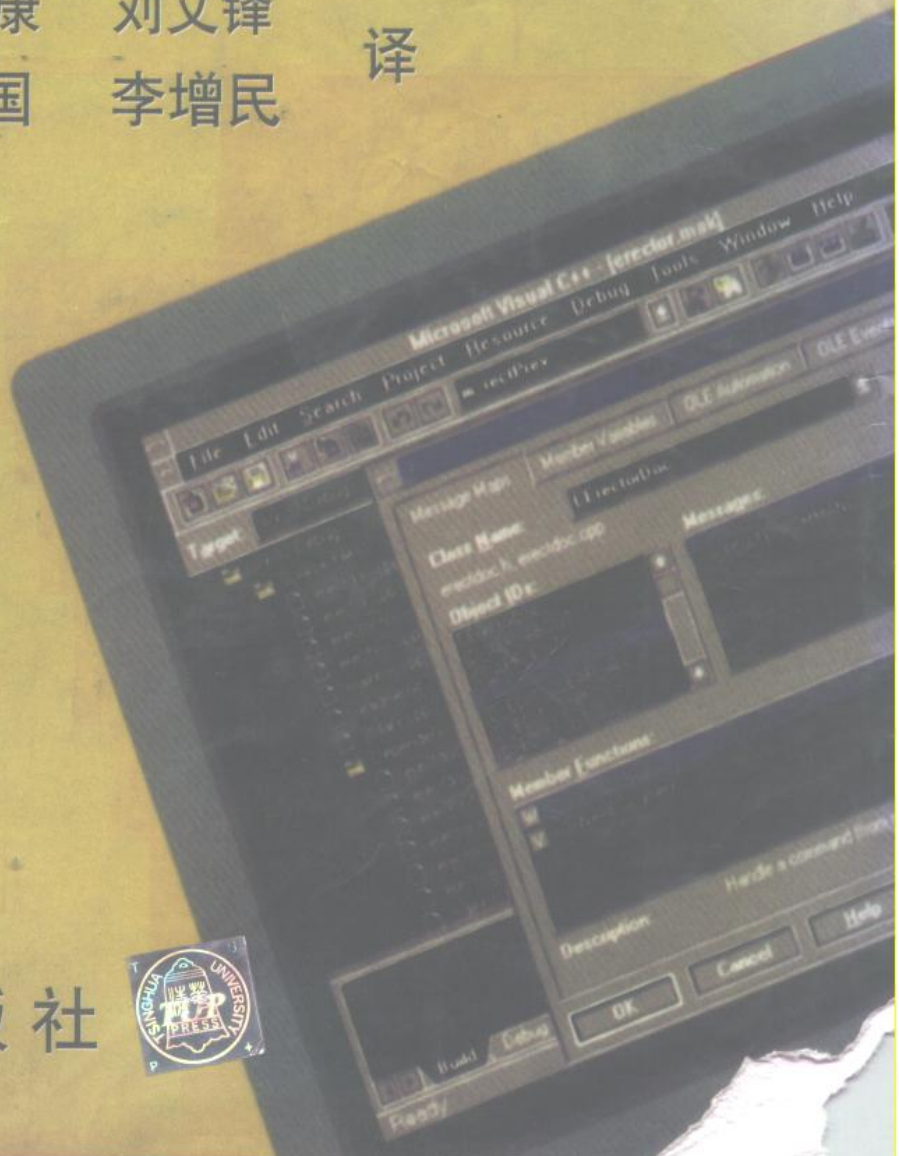


Microsoft Visual C++ 自学教程

覆盖至 Visual C++ 2.0 的所有版本

David A. Holzgang 著

曹康 刘文锋 译
吕保国 李增民



MIS:

清华大学出版社



Microsoft Visual C++ 自学教程

——覆盖至 Visual C++ 2.0 的所有版本

David A. Holzgang 著

曹康 刘文锋 译

吕保国 李增民

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书介绍了如何使用 Visual C++ 来设计、建立和调试 Windows 应用程序。全书按自学方式安排的体系结构以及书中清晰的示例,可以让读者快速且容易地学会 Visual C++ 程序设计。此外,本书还随附一张软盘,向读者提供了书中程序的源代码。

本书还讨论了如何使用 Microsoft Foundation Class Library (MFC),并讨论了如何充分利用 MFC 的强大功能函数库生成全功能的 Windows 应用程序。此外,还讨论了如何使用 Visual C++ 将应用程序从 Windows 3.1 环境下移植到 Windows NT 和 Windows 95 环境下。

中华人民共和国国家版权局著作权合同登记章 图字: 01-96-0385 号

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

Microsoft Visual C++ 自学教程: 覆盖 Visual C++ 2.0 的所有版本/曹康等译.

—北京: 清华大学出版社, 1996

ISBN 7-302-02167-8

I. M… I. 曹… III. C 语言-操作系统, Windows NT TP312C

中国版本图书馆 CIP 数据核字(96)第 07995 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: 清华大学印刷厂

发行者: 新华书店总店北京科技发行所

开 本: 787×1092 1/16 印张: 27.25 字数: 677 千字

版 次: 1996 年 6 月第 1 版 1996 年 6 月第 1 次印刷

书 号: ISBN 7-302-02167-8/TP·1031

印 数: 0001—5000

定 价: 39.00 元

目 录

第一章 引言	1
1.1 什么是 Visual C++	1
1.2 Visual C++ 2.0 环境预览	1
1.2.1 Visual Workbench	1
1.2.2 Microsoft Foundation Classes 库(MFC)	2
1.2.3 Visual C++ 版本	2
1.3 本书的目的及编排方式	3
1.3.1 目标	3
1.3.2 要求	4
1.3.3 编排方式	4
第二章 安装 Visual C++	7
2.1 Visual C++ 的 CD-ROM 版本	7
2.2 Visual C++ 1.5 和 Visual C++ 2.0	8
2.3 硬件和软件要求	8
2.3.1 Visual C++ 1.51	8
2.3.2 Visual C++ 2.0	10
2.4 安装过程.....	11
2.4.1 在 Windows 或 WFW 3.11 下安装	11
2.4.2 在 Windows NT 3.5 下安装	19
2.4.3 重新安装 Visual C++	26
2.5 测试 Visual C++ 安装.....	27
2.5.1 测试 Visual C++ 2.0 安装	27
2.5.2 测试 Visual C++ 1.51 安装	27
2.6 本章小结.....	29
第三章 使用 Visual Workbench	30
3.1 如何布置屏幕显示.....	30
3.1.1 布置和显示窗口.....	31
3.1.2 工具条和状态条.....	31
3.2 菜单条.....	34
3.2.1 File 菜单	34
3.2.2 Edit 菜单	35
3.2.3 View 菜单	37
3.2.4 Project 菜单	38
3.2.5 Browse 菜单	39

3.2.6	Debug 菜单	40
3.2.7	Tools 菜单	41
3.2.8	Options 菜单	42
3.2.9	Window 菜单	42
3.2.10	Help 菜单	43
3.3	Visual Workbench 选项	44
3.3.1	Project 选项	44
3.3.2	Debug 选项	45
3.3.3	Directories 选项	45
3.3.4	Editor 选项	45
3.3.5	Workspace 选项	46
3.3.6	Tools 选项	47
3.3.7	颜色和字体选项	47
3.4	项目结构	48
3.4.1	使用工作区	49
3.4.2	创建和维护一个项目	49
3.5	使用 QuickWin	50
3.5.1	QuickWin 概述	50
3.5.2	QuickWin 安装和执行	51
3.5.3	借助于 QuickWin 学习 C++	51
3.6	本章小结	53
第四章	使用 AppWizard	54
4.1	Windows 程序如何工作	54
4.1.1	Windows 程序	54
4.1.2	理解 Windows 通信	56
4.2	在应用程序中使用 Windows 消息	57
4.2.1	Windows 消息的类型	58
4.3	使用 AppWizard 创建一个应用程序	61
4.3.1	使用 AppWizard 创建 ErectorSet	61
4.3.2	重温 ErectorSet	68
4.4	使用 Class Browser(类浏览器)观察应用程序	73
4.4.1	Browser 窗口和查看选项	73
4.4.2	观察类	76
4.4.3	查阅函数	78
4.4.4	观察定义和引用	79
4.5	本章小结	81
第五章	在文档中绘图	82
5.1	向类中增加函数	82
5.1.1	在 ErectorSet 文档中绘图	83

5.1.2	增加鼠标消息处理	94
5.1.3	创建桁条对象	102
5.1.4	绘制桁条对象	104
5.2	存储和恢复用户文档	108
5.2.1	串行过程	108
5.2.2	给 ErectorSet 增加串行函数	109
5.3	测试应用程序	111
5.4	本章小结	112
第六章	理解面向对象的编程和 MFC	113
6.1	面向对象的编程	113
6.1.1	为何面向对象编程	113
6.1.2	面向对象的编程特征	115
6.1.3	类和分层结构	118
6.1.4	函数和消息	123
6.1.5	向 ErectorSet 中增加 ClearAll 处理	125
6.2	MFC 应用程序结构	135
6.2.1	框架体系结构	135
6.2.2	基本类结构	136
6.3	使用 MFC 类	137
6.3.1	向 ErectorSet 中增加移动消息处理	137
6.3.2	测试新的 ErectorSet 应用程序	144
6.4	本章小结	146
第七章	视窗与文档	147
7.1	文档	147
7.1.1	文档与应用程序的关系	147
7.1.2	文档的作用	148
7.2	视窗	149
7.2.1	视窗的作用	150
7.2.2	视窗与文档的关系	151
7.2.3	文档模板	152
7.2.4	多视窗	154
7.2.5	增强视窗	158
7.2.6	更新视窗	158
7.2.7	使用图形中的无效区域	163
7.2.8	提高 ErectorSet 的绘图能力	166
7.3	滚动视窗	171
7.3.1	框架结构为滚动提供了什么	172
7.3.2	滚动窗口和视窗	174
7.3.3	在 ErectorSet 中增加滚动功能	175

7.4 本章小结	187
第八章 异常和调试	188
8.1 使用异常处理	188
8.1.1 一般执行过程	188
8.1.2 异常处理	191
8.1.3 在应用程序中加入异常处理	193
8.2 使用 Visual C++ 调试工具	197
8.2.1 在执行过程中进行调试	198
8.2.2 检查程序操作	208
8.2.3 建立调试过程	211
8.3 跟踪应用程序处理过程	214
8.3.1 在应用程序中使用跟踪	214
8.3.2 TRACE 的其它选择	220
8.4 本章小结	220
第九章 使用资源	222
9.1 资源和 Windows 程序	222
9.1.1 什么是资源	222
9.1.2 资源定义文件	224
9.2 编辑资源	227
9.2.1 调入资源进行编辑	227
9.2.2 编辑资源信息	232
9.3 增加资源	239
9.3.1 设计对话框	239
9.3.2 增加一个菜单项	241
9.3.3 定义对话框	244
9.4 本章小结	251
第十章 控制和消息	252
10.1 创建对话框处理类	252
10.1.1 创建一个新类	252
10.1.2 检查新的对话框类	254
10.2 连接按钮	257
10.3 连接对话框项	258
10.3.1 连接编辑文件框	258
10.3.2 连接组合框	260
10.4 检查对话框代码	260
10.5 增加处理代码	263
10.5.1 进行桁条对话框处理	263
10.5.2 处理桁条对话框标题	264
10.5.3 向文档类中增加代码	265

10.6	测试应用程序	270
10.7	本章小结	271
第十一章	增强应用程序	272
11.1	创建不变信息	272
11.1.1	在文件中存储信息	272
11.1.2	使用应用程序简要表	274
11.2	分割窗口	280
11.2.1	分割口和视窗	281
11.2.2	向 ErectorSet 中增加分割窗口	283
11.3	使用可选的 MFC 类	287
11.4	状态条处理	288
11.4.1	定义状态条的位置	289
11.4.2	更新状态条	290
11.4.3	使用 CTime 辅助类	290
11.4.4	向状态条中增加显示	292
11.5	本章小结	297
第十二章	测量与演示	298
12.1	打印增强	298
12.1.1	测量系统	299
12.1.2	改变为与设备无关的测量系统	302
12.1.3	打印体系结构	309
12.1.4	打印标题信息	313
12.2	绘制位图	319
12.2.1	位图类型	319
12.2.2	增加图案填充位图	320
12.3	跟踪内存遗失	338
12.3.1	MFC 内存诊断	339
12.3.2	使用内存跟踪	340
12.4	本章小结	343
第十三章	更深入的议题	344
13.1	增加新的 MFC 函数	344
13.1.1	如何更新应用程序	344
13.1.2	增加上下文相关帮助	345
13.2	增加对象链接和嵌入(OLE)	357
13.2.1	OLE 文档	358
13.2.2	OLE 应用程序支持	360
13.3	OLE 2.0 MFC 支持	363
13.3.1	增加 OLE 服务器支持	364
13.3.2	OLE 容器支持要求	400

13.4	本章小结.....	401
第十四章	升级到 Visual C++ 2.0	402
14.1	使用 Visual C++ 2.0 环境	402
14.1.1	开发环境的新特征.....	402
14.1.2	创建一个项目.....	409
14.1.3	浏览用户的应用程序.....	413
14.1.4	新的调试特征和支持.....	417
14.2	把项目从 16 位转换为 32 位.....	423
14.2.1	32 位应用程序中的差别	423
14.2.2	升级 ErectorSet	424
14.3	本章小结.....	426

第一章 引言

本章对本书进行了概括介绍,描述了本书的内容和使用方法。本章从解释什么是 Visual C++ 及其工作环境开始,然后介绍了本书的编排方式,并对每章内容进行了简要概述。

1.1 什么是 Visual C++

Visual C++ 是为开发 Windows 应用程序而创建的一种新的循序渐进的编程工具。它的高性能及高度集成的开发环境,可使用户比以前更快更方便地创建、建立和调试 Windows 应用程序。Visual C++ 基于以下两个重要概念:首先,Visual C++ 在 Windows 环境下运行,所以用户可以使用一组丰富的基于 Windows 的全部工具,这可以帮助用户创建和管理 Windows 下的项目和应用程序;其次,它使用了目前著名的包括新工具的“可视”用户界面(在某些方面类似于 Visual Basic)。

1.2 Visual C++ 2.0 环境预览

Microsoft Visual C++ 基于以下三个元素:C/C++ 编译器和连接程序,Visual Workbench 以及 Microsoft Foundation Classes(MFC, Microsoft 基础类)库。其中,作为综合开发环境,每一组件均对 Visual C++ 的成功作出了重大贡献。

1.2.1 Visual Workbench

在 Visual C++ 2.0 以前的版本中,Visual Workbench 是 Visual C++ 开发环境的核心(在 2.0 版本中,整个环境只称为“开发环境”)。Visual Workbench 包括了一个优秀的代码编辑器,且该代码编辑器具有许多卓越的功能,如自动标记功能——一旦用户在放置代码的地方做了标记,则下一行将自动地在同一点处开始放置,该功能对缩进排列代码非常有用。Visual Workbench 提供了完整的项目管理,可使用户将各种代码模块连接到一个项目中,然后该项目可作为一个单元来处理,以使用户建立自己的应用程序。资源编辑器允许用户创建和编辑各种类型的 Windows 资源,而且集成的程序调试器允许用户在监控方式下运行自己的程序、检查和改变数据以及在程序中设置断点。最新的 2.0 版甚至允许用户实时调试,以便当错误刚出现时,用户可迅速转入程序调试器。

Visual Workbench 也采用了 Microsoft 的专用辅助工具 AppWizard 和 ClassWizard,这两种工具可使用户很容易地创建和定制基于 MFC 的应用程序。用户可使用它们从 MFC 中选择模块,以建立初始的框架程序,然后把新的函数和类加到该框架中。正如本书练习中介绍的那样,这种工具很有益处,它可以帮助用户得到所需要的各种完整的支持,而不会被 MFC 的复杂性所吓倒。

1.2.2 Microsoft Foundation Classes 库(MFC)

就本书而言,由 Visual C++ 2.0 提供的编程环境就是学习和使用 Visual C++ 很有力的理由。尽管如此,不论用户是否相信,还有更强有力的理由,即该程序可以访问 MFC 库。

与其它任何开发方法相比,MFC 使得用 C++ 编写 Windows 应用程序更快且更方便。此类库由一系列代表应用程序框架的 C++ 类构成。这些类被设计成一起使用,以建立应用程序结构框架,它可提供大部分用户界面和功能,而这些正是用户在 Windows 应用程序中期望得到的。因为应用程序框架使用 C++ 类,所以对用户而言,仅需要通过采用提供新函数的新类去替换适当的类,这样可方便地扩展框架应用程序功能。该新函数正是用户希望加入到内置函数中的函数。

Visual C++ 环境通过内置在 Visual Workbench 中的两个 Wizard (AppWizard 和 ClassWizard) 连接到 MFC 中。这些工具可使用户既方便又安全地选择和定制自己的框架应用程序,并增加新的处理功能,该功能可直接连接到当前的代码库中。另外,这两个工具均可生成代码,调用 MFC 后可完成基本任务。在本书的后面,读者将会发现,当用户使用这些工具时,Wizard 是采用大部分 MFC 功能的最重要组件。

1.2.3 Visual C++ 版本

目前,已有多种版本的 Visual C++: 版本 1.0、版本 1.5 及当前版本,同时还包括为 16 位开发环境研制的 Visual C++ 1.51 和为 32 位开发环境研制的 Visual C++ 2.0。

本书尽可能覆盖了各种版本的 Visual C++。此书的编写目的是帮助用户学习 Visual C++ 的基本内容。从第一版开始,其环境就被优化设计了,因此用户在本书中学到的内容对所有的版本都适用。尽管如此,本书还是主要面向拥有 1.5 版和 1.51 版的用户。所以,每一帧画面和例子都是基于这种版本的。如果用户使用的是 1.0 版或 2.0 版,则在这些方面需要做些小幅度调整,而且用户屏幕上所显示的内容与本书所看到的会略有不同。本书给出的所有代码均可以由上述版本的 Visual C++ 生成,且不应该会出现什么大的困难。因此,读者所需的调整量很少。

Visual C++ 的第一版(1.0 版)有特定的限制。特别地,该版不支持 OLE 2.0,OLE 2.0 仅包含在 1.5 版及其后续版本的 MFC 中。如果使用 1.0 版,用户将不能做第十三章中的练习,因为这些练习需要 OLE 支持。要做这些,就得需要支持 OLE 2.0 的 MFC 版本。

运行 Visual C++ 2.0 的用户需要阅读第十四章。开始练习前,必须把应用程序代码由 16 位转换成 32 位。尽管在 Visual C++ 2.0 中的主要变化是支持 32 位编程的,但在其它方面仍然有所不同,特别是在使用 AppWizard 和使用资源编辑器方面。在开始其它章节的练习前,读者必须了解它们才行。再者,变化是很小的,主要是在方便性和明确性方面的改进,用户应该能够很容易地把文字上的描述轻松地转换到应用环境中去。

如果用户对 Visual C++ 2.0 还不太熟悉,则可能会奇怪为什么不能直接从最新的版本开始,并让它顺其自然呢? 问题是 Visual C++ 2.0,它是以 32 位应用程序为目标研制的开发环境,本身就是一个 32 位的应用程序。这就意味着,它只能运行在 Windows NT 3.5 和 Windows 95 上。而且,在 Visual C++ 上开发的应用程序,作为 32 位应用程序,只能在这些

系统上运行。因此,大多数打算学习 Visual C++ 的用户,很可能正在使用 Windows 3.0 或 3.1,并准备开发在这些环境下的应用程序。本书主要面向这部分用户,正如以前指出的,这似乎并不是一种严格的限制,因为 Visual C++ 2.0 的用户可以很容易地利用这些进行练习。

1.3 本书的目的及编排方式

本书的目的就是向用户介绍(更确切地说,是帮助用户自学)基本的 Windows 应用程序编程。为完成此任务,本书采用了 Visual C++ 开发环境和 MFC 库,从而使用户创建和增强一个简单的样本应用程序。

本书的目的不在于向用户详细介绍如何进行 Windows 编程。当用户致力于开发 Windows 应用程序时,还会存在很多问题。在本书中,用户使用 MFC 就可基本避开大部分 Windows 应用编程中固有的复杂问题,这就是本书的初衷。尽管如此,如果准备开发具有专业水准的 Windows 应用程序,用户就应意识和了解到很多这方面的内容。尽管未介绍很多 MFC 细节,用户仍会发现,这种理解在开发时会有意外的收获。

问题在于,本书只能覆盖那么多内容。本书主要集中研究使用 Visual C++ 环境和 MFC。Windows 编程问题只有与它们有关时才予讨论,而不使用通用术语。因此,用户只能在练习中学习 Windows 编程基础。为了得到详细的理论,可在其它课程中深入钻研。

本书包含许多实例代码,详细解释了这些代码的产生、编辑和作用。在学习这些实例时,用户就能明白一个典型的 Windows 应用程序是如何组织的,明白如何使用 C++ 的类和继承,以及 MFC 如何构造以使用户增加自己的代码来完成案头的任务。

1.3.1 目标

帮助用户自学 Windows 编程这一目标太广了,以致于在组织和选择本书的材料时涉及面十分广泛。在帮助设计和确定本书内容时,有四个特定的目标一直占主导地位。

第一个目标是使用户方便地使用 Visual Workbench 和其它 Visual C++ 工具。本书结束时,用户可以很容易地在 Visual Workbench 上处理任何大小或类型的项目。这就意味着,查阅任何东西都不必使用参考资料。也就是说,在设置某一编译器或连接器选项,或调用在简单应用程序中没有涉及到的高级 API 时,都可不用参考资料。

第二个目标是让用户明白如何使用 MFC。因为在 Visual Workbench 内,MFC 构成了大多数工作的基础。用户必须熟悉如何访问和使用 MFC,利用这些工具(例如 AppWizard 和 ClassWizard),可将它集成到自己的应用程序中。

第三个目标与第二个目标有关但又有些不同。当学完本书时,读者应该熟悉构成 MFC 核心的基础类。完整的 MFC 包含了很大范围的类和函数,要精通这些有效的大量组合需要一定的时间,而且必须处理多种 Windows 应用程序。在本书范围内,相信读者能够熟悉构成 Windows 应用程序所必需的组成部分的那些类,这些类很可能出现在用户创建的任何应用程序中。

最后,本书打算帮助用户学习典型的 Windows 应用程序的组织 and 结构。单击或多次单击鼠标,用户就可利用 MFC 建立一个应用程序。但是,用户必须明白一个应用程序是如何

构成的以及为何必须用特定的方式来组织。否则,就像一名仅仅明白如何操作自动驾驶仪的驾驶员一样,冒险进入自己并不太熟悉的环境。在这样的开发环境中,自己的应用程序未能正常地发挥作用,但并不知道它为何不能工作,而且不知道如何解决这些问题。

这些目标是密切相关和互相交叉的,每一个目标均是互相支持和互相利用的。像其它好的书籍一样,本书没有区分每一个目标,而是设法把它们联系在一起。因此,用户可用同样的方式学习 Visual C++、MFC 和 Windows 编程。

1.3.2 要求

本书对读者有一些要求,要求分为两大类:学习工具和编程知识。如果喜欢的话,也可称为“硬件”和“软件”。首先从实际要求开始,读者需要有 Visual C++ 的拷贝版。如果还没有在自己的系统里安装它,那样正好,可以在此阅读一些安装指南。如果已经安装了它,那也不错,只需简单地跳过前面的内容,去学习开发环境和第一个练习就可以了。除此之外,所需的其它内容则都在本书或本书随附的磁盘中给出了。或者,本身就是 Windows 环境中的正常内容。

在编程知识方面,读者至少需要对 C 语言有所了解。读者需要基本的编程知识,且能够阅读和理解 C 语言代码。

除了 C 语言之外,本书还包含创建和修改标准 Windows 应用程序所需的其它基本信息。本书没有探讨创建 DLL 或可视控制的有关更高层领域,而是坚持探讨创建一个典型 Windows 应用程序所需的基本工作及其用户界面。

本书不是传授 C++,而是解释 C++ 的最基本的概念。如果读者不懂 C++,仍可从实例中学习和了解代码。C++ 是一种丰富而又多变的语言。笔者强烈建议读者在开始用 Visual C++ 和 MFC 大量编程前,应认真学习和掌握该语言。许多 C++ 概念是 MFC 如何工作和处理项目的核心,对 C++ 的正确理解将有助于读者使用 MFC。

1.3.3 编排方式

本书是按一系列讲座和练习形式编排的,目的在于使读者能够用所学习的每个新概念和功能迅速工作。为了帮助读者了解 Windows 开发过程的本质,应用程序作为实例贯穿全书,这可帮助读者明白实际的 Windows 应用程序是如何建立的,一个完整而又复杂的 Windows 应用程序是如何从开始逐步演变而成的。

这里有一种方法,即通过对 Visual C++ 工具和 MFC 的使用而不断加深理解和掌握,这样,应用程序的开发便会水到渠成。很多开发人员都会提供帮助,而且是完全正确的,即在开始编程前,读者必须完全明白自己的应用程序如何工作。事实上,大多数人,即使像编者这样的专业开发人员,开始时对所做的工作也只是一般的想法。甚至对于一个详细的设计,读者仍然会发现需要钻研某一领域或方向,而这是开始时所预料不到的。这里要看到的过程,是编者尽力模拟的创建一个简单的 Windows 应用程序的实际过程。

通过使用 MFC 和 Visual C++,读者可比以前有更大的灵活性去修改和调整自己的应用程序,这的确是一个好消息。这里的练习将展示给读者如何使用这些工具,既安全又方便地改变和维护自己的应用程序。

■ 本书约定的表示法

本书使用标准的印刷惯例来帮助读者明确特定的项目。

`int i=x+y;` 采用单一间隔字体的文本是代码或代码段,例如函数名、变量名或已定义的常数。

File 文本中某个字母下面划线时,代表命名项目的访问键。这种方法主要是帮助用户使文本与显示器相匹配。在屏幕上的文本尚未检验时,用户不能使用这些键去访问。

本书还采用“本书习惯用法”来标出具有特殊含义的文本。所有“本书习惯用法”如下:

注意:提醒读者这些信息特别有趣或特别重要。

警告:提醒读者工作时可能遇到的特定问题或危险。

概念与术语:表示 C++ 术语和概念的定义和讨论。如果读者熟悉 C++, 可以跳过该项下的正文段,该处很可能是读者已知道的知识。因为有些读者不像其它读者一样熟悉 C++, 所以编者尽量补充那些对理解或使用实例中的程序有必要的 C++ 的概念。

捷径:捷径可使用户迅速简便地完成某些任务,在正文中却需用很长的形式代表。

■ 各章概述

以下内容是本书内容的简要概述。这些章节从安装 Visual C++ 开始,逐渐向读者介绍如何使用 Visual C++ 和 MFC 建立一个完整的可执行的 OLE 应用程序。

1. 第一章:引言

这是读者正在阅读的一章。它提出了本书的概述,描述了本书的内容,介绍了使用本书的方法。

2. 第二章:安装 Visual C++

本章讨论了安装问题,例如安装资料的总量、安装选项以及安装要求等。同时,本章还给出了一个简短实例,向读者介绍如何检验安装的 Visual C++ 是否正确。

3. 第三章:使用 Visual Workbench

Visual Workbench 是 Visual C++ 的核心。开始工作前,读者必须熟悉 Visual Workbench 的工具、设置和选项。本章向读者介绍如何建立一个项目以及如何访问 Visual Workbench 下的工具。

4. 第四章:使用 AppWizard

本章讨论了 AppWizard,这是在 Visual C++ 中创建的基于 MFC 的应用程序的基础。本章同时还提出了标准的 Windows 程序开发周期,并说明了 AppWizard 如何适应和缩短这种周期。在本章中,我们将引导读者学习一系列标准步骤,这是使用 AppWizard 生成一个框架应用程序所必需的知识。

5. 第五章:在文档中绘图

在本章中,介绍了在 Windows 应用程序下绘图的基础。同时还介绍了 ClassWizard,这是 MFC 类的类生成器和管理器。本章中,还向读者介绍了用于鼠标消息和 Windows 消息

的演示应用程序的基本消息处理过程。

6. 第六章:理解面向对象的编程和 MFC

至此,几乎已介绍了本书的一半内容,读者对 Visual C++ 环境已有了较深的认识。在开始增强演示应用程序以及进一步使用 Visual C++ Wizards 之前,本章向读者简单介绍面向对象的编程和 MFC(Microsoft 基础类)。本章介绍了在应用层次上面向对象编程的基本原理,讨论了诸如类等级、关键词及继承等问题。读者同时还可了解 MFC 框架应用程序,并将了解 MFC 的整体结构以及学会如何使用它来增强演示应用程序的功能。

7. 第七章:视窗与文档

大多数 Visual C++ 编程集中在视窗和文档上进行,本章描述了视窗和文档如何交互作用,以及读者如何修改各种层次以得到希望的功能。例如,读者可向演示应用程序中增加滚动功能。

8. 第八章:异常和调试

调试是任何编程者面对的事实。Visual C++ 提供了出色的工具,用于避免出错以及发现疏忽的错误。本章详细讨论了在 Visual C++ 应用程序中如何使用演示应用程序去调试和跟踪问题。

9. 第九章:使用资源

Visual C++ 备有集成资源编辑器和资源处理界面。在 Visual C++ 的早期版本中(到 1.51 版),这是一个独立的且是集成的应用程序,称为 AppStudio。在 Visual C++ 2.0 中,它被完全集成到 Visual Workbench 中。在任何情况下,读者都会从本章中了解到在资源编辑器创建资源后如何向项目文件中增加资源。本章同时还讨论了资源的主要基本类型。

10. 第十章:控制和消息

本章向读者介绍了在应用程序中如何使用资源控制,如何把这些控制连接到使自己的应用程序正确响应的消息上。例如,读者可在演示程序中增加对话框处理功能。

11. 第十一章:增强应用程序

本章包含了大量的增强功能,读者在大多数应用程序中将会用到它们。本章向读者介绍了如何为自己的应用程序存储和检索永久性信息,如何创建一个分割窗口以便在窗口中观察一个文档的几段,以及如何使用辅助的 MFC 类(例如 CTime),以在自己的应用程序中更新状态条显示。

12. 第十二章:测量与演示

本章主要讨论了由 Visual C++ 提供的工具,以及用于提供与设备无关的用于绘图的框架应用程序。读者可在演示应用程序上增加类,以便将测量系统改为与设备无关的模式。本章还介绍了如何增加新的打印功能,这依赖于与设备无关的测量。

13. 第十三章:更深入的议题

本章包含了重要的高级议题,它们是 Visual C++ 的组成部分。特别地,它展示了如何从 MFC 向一个现有应用程序中增加新的代码。例如,读者会更新演示应用程序以提供与上下文相关的帮助,然后学习如何在自己的应用程序中提供 OLE 兼容性。读者可使用同样的增强技术来向演示应用程序中增加 OLE 服务器功能。

14. 第十四章:升级到 Visual C++ 2.0

本章解释了如何把项目和程序从早期的 Visual C++ 版本(16 位)升级到 Visual C++ 2.0 中。在此,演示应用程序作为一个如何升级项目的实例使用。本章同时还讨论了在 MFC 3.0 中实现的新特点,这是 MFC 库的最新版本。

第二章 安装 Visual C++

本章讨论了安装问题,例如要安装的推荐软件、安装选项以及安装要求等。本章同时亦给出了一个简单实例,以检验读者是否正确地安装了 Visual C++。

本章包含的重要议题如下:

- * Visual C++的 CD-ROM 版本
- * Visual C++ 1.5 和 Visual C++ 2.0
- * 硬件和软件要求
- * 安装过程
- * 检验 Visual C++的安装

2.1 Visual C++的 CD-ROM 版本

Visual C++ 2.0 制作在 CD-ROM 上,它包含要安装的所有的 Visual C++ 组件的资料。CD-ROM 含有以下目录:

- msv15: 本目录包含 Visual C++ 1.51 的开发环境。如果用户正在使用 Windows 3.1, Windows for Workgroups 或 Windows NT 3.1 或更早的版本,则可以安装这个版本。这个环境可使用户开发 16 位应用程序,并在该环境下运行。
- msvc20: 本目录包含 Visual C++ 2.0 的开发环境,只有当用户正在运行 32 位的操作系统(例如 Windows NT 3.5 或 Windows 95)时,才可以安装。
- msvccdk: 本目录包含了 Microsoft OLE Custom Control Developer's Kit (CDK) 的文件。这可使用户为 MFC 和其它应用程序开发新格式的 OLE 定制控制。这是用 32 位取代过时的 16 位的 Visual Basic 控制,它在开发人员中间曾经十分流行。
- win32s: 本目录包含了 Microsoft Win32s 的库文件,它允许用户的 32 位应用程序可在 16 位操作系统上运行,例如 Windows 3.1。
- wincim: 本目录包含了 CompuServe Information Manager 应用程序和支持文件的 Windows 版的免费拷贝,这是为方便用户而提供的。它可使用户与 CompuServe 联网或签约。Microsoft Developer Support 提供了许多有关 CompuServe 的特别讨论,在此可使用户从 Microsoft 专业人员和其它开发人员那里得到一些帮助和建议。

尽管 CDK, Win32s 以及 CompuServe 工具很有价值且很吸引人,但它们并不是本书着重讨论的问题。这里将不再深入讨论它们或讨论如何使用或安装这些库。如果读者打算使用它们,可查阅 Visual C++ CD-ROM 上的联机文献集,以进一步了解有关情况。

2.2 Visual C++ 1.5 和 Visual C++ 2.0

这是一个技巧问题。当用户的 CD-ROM 上有两种不同版本的 Visual C++ 时,在开始设置和使用时,用户不会有太多的选择。

Visual C++ 2.0 是为创建 32 位应用程序而设计的开发环境,在 Windows NT 3.5 或 Windows 95 操作系统下运行。而且,Visual C++ 2.0 本身也是 32 位的应用程序。如果用户没有在 Windows NT 3.5 或 Windows 95 下运行,就不能在自己的系统上安装 Visual C++ 2.0。下面将要描述的标准安装步骤明确了当用户没有在 32 位操作系统上运行时,将会使得用于安装 Visual C++ 2.0 的按钮无效。

尽管用户不能在 16 位操作系统(例如 Windows 3.1)上安装 Visual C++ 2.0,但却可在 Windows NT 3.5 下安装 Visual C++ 1.51。如果这样做,表明用户是在 Windows 兼容方式下运行它(Windows 下的 Windows,即 WOW)。用户选择这样做有两种理由,首先,如果用户正在开发 16 位的应用程序,或靠它们自己,或与 32 位的应用程序相连,则必须在 Visual C++ 1.51 上才能完成;这是产生 16 位应用程序的唯一开发环境。其次,用户可选择使用 Windows NT 或 Windows 95 作为开发平台,因为它比 Windows 3.1 或另外的 16 位操作系统更强大、更安全。这种方法的主要缺点是这些环境中的每一种都需要大量的磁盘空间。

因此,当读者阅读本章时,要牢记 Visual C++ 2.0 的选项和工具只有在 32 位操作系统上才行得通。

2.3 硬件和软件要求

从 CD-ROM 中安装 Visual C++ 的要求,依据以下两种情况而变:用户正在使用的操作系统和安装时用户选择的选项。正如用户所预料到的,运行 Visual C++ 2.0 的要求比运行 Visual C++ 1.51 要更精确。

2.3.1 Visual C++ 1.51

下面是在用户系统上安装 Visual C++ 1.51 的最低要求:

- * 一台 IBM 个人计算机,或 100% 与 80386 兼容运行的、或更新的处理器
- * 一台 VGA 或更好的显示器
- * 4MB 可用内存(推荐 8M)。如果用户正在运行 Windows NT,则至少需要 16MB
- * 一个与读者的操作系统兼容的 CD-ROM 驱动器
- * 运行于增强模式下的 Microsoft Windows 3.1, Windows for Workgroups 3.1 或 Windows 95, 或 Windows NT。Visual C++ 提供了命令行去访问某些工具和函数,但在安装过程中需要 Windows 支持。
- * 一个硬盘驱动器,应具有足够大的空间安装用户选择的选项。最小的安装必须具有 13M 的硬盘空间。