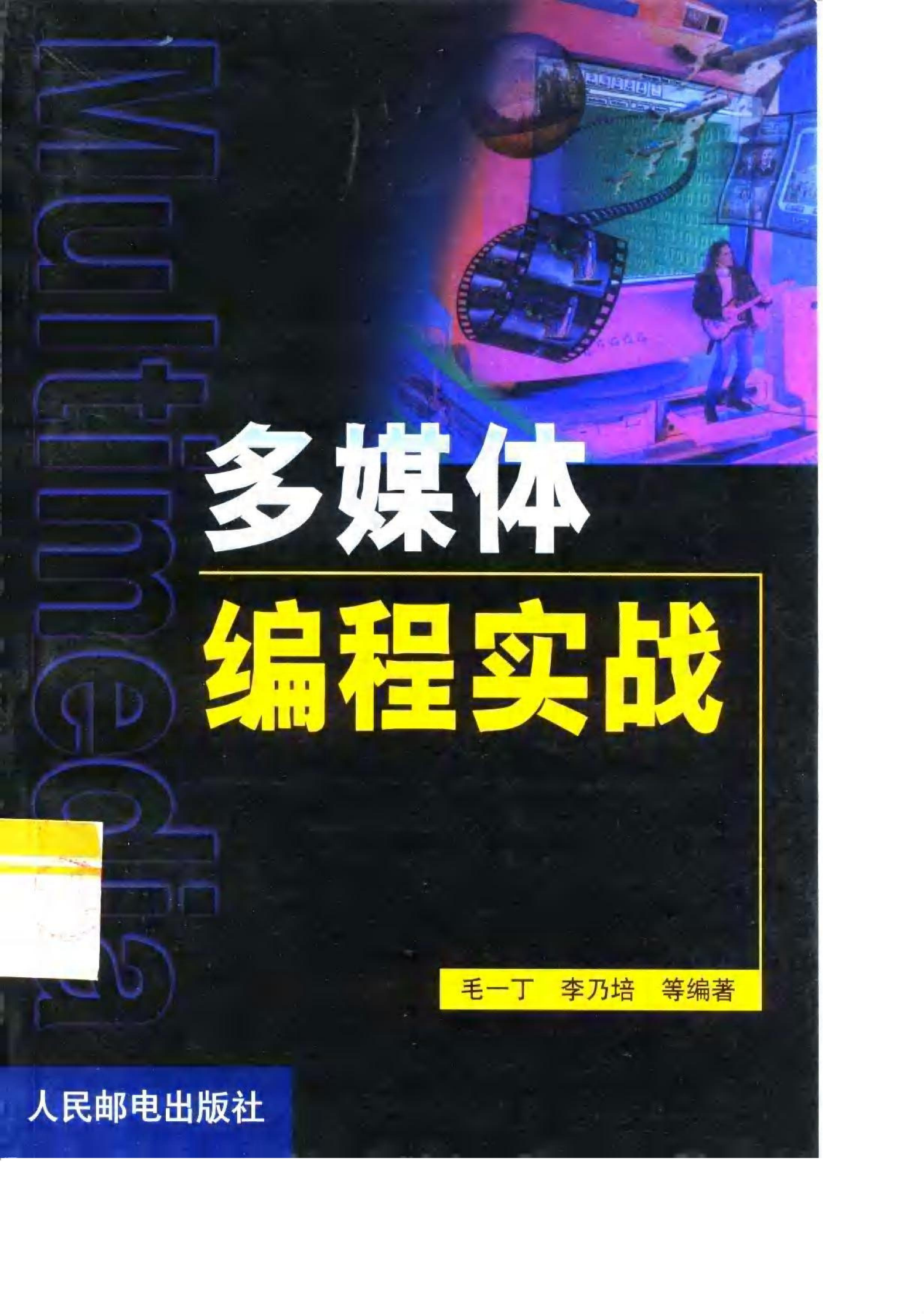




多媒体 编程实战

毛一丁 李乃培 等编著

人民邮电出版社

多媒体编程实战

毛一丁 李乃培 等编著

人民邮电出版社

内容提要

人类社会正在进入多媒体信息时代,许多编程人员对于多媒体编程知识的需要越来越强烈,为此,我们出版了本书。书中分四部分——Visual C++、Visual J++、Visual Basic 和 ActiveX——介绍了各种多媒体编程工具的使用,并通过大量实例使读者更加迅速地掌握多媒体编程的精华。相信对那些迫切需要学习多媒体编程及相应知识的读者会大有帮助。

多媒体编程实战

- ◆ 编 著 毛一丁 李乃培 等
责任编辑 梁 凝
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
北京汉魂图文设计有限公司制作
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本:787×1092 1/16
印张:27.75
字数:693
印数:1-5 000 册
- 2000 年 4 月第 1 版
2000 年 4 月北京第 1 次印刷

ISBN 7-115-07692-8/TP·1044

定价:41.00 元

前 言

随着计算机硬件设备性能和软件技术的快速发展，多媒体技术的应用已经越来越普及。本书的目的就是帮助读者学习如何利用各种编程工具进行多媒体编程。

为了达到这个目的，书中编写了大量的实例，每个实例都是经过精心挑选的，很有代表性。读者通过练习这些实例，可以更快地掌握书中所讲内容，增强学习兴趣。

本书分为四个部分：第一章到第三章是第一部分，主要介绍了 Visual C++ 的基础知识和如何利用 Visual C++ 进行多媒体编程；第四章到第七章是第二部分，介绍了编程工具 Visual J++ 的操作、利用 Visual J++ 编写应用程序和利用 Visual J++ 编写小程序；第八章到第十二章是第三部分，对 Visual Basic 编程进行了详细的讲解；最后一部分包括第十三章到第十六章，详细介绍了 ActiveX 多媒体编程的内容。

全书结构清晰，图文并茂，适合广大电脑爱好者和编程人员学习多媒体编程。

本书编写时间比较仓促，不妥之处在所难免，敬请批评指正。

编者
1999.12

目 录

第一章 Visual C++ 6.0 基础	1
1.1 MFC 简介	1
1.2 Visual C++ 6.0 的使用	2
1.2.1 运行 Visual C++ 6.0	2
1.2.2 自定义 Visual C++ 6.0	5
1.3 在线帮助	8
1.4 功能强大的 MSDN 库	9
1.4.1 目录选项卡	11
1.4.2 索引选项卡	12
1.4.3 搜索选项卡	12
1.4.4 书签选项卡	15
第二章 Visual C++ 编程初步	16
2.1 使用应用程序向导 AppWizard	16
2.2 应用程序的消息循环	27
2.2.1 消息的发送和获取	27
2.2.2 消息处理函数的声明	29
2.3 对话框类	31
2.4 对话框编辑器和 ClassWizard	41
第三章 Visual C++ 多媒体编程	47
3.1 两个重要概念	47
3.2 一个示例	50
3.3 使用画笔	57
3.3.1 创建画笔	57
3.3.2 使用画笔在设备上下文中进行输出	58
3.4 绘制图形	61
3.4.1 图形输出函数	61
3.4.2 使用刷子	67
3.5 设置不同字体	71
3.5.1 创建字体对象	72
3.5.2 使用字体对象	74
3.5.3 特殊的字体效果	82
3.6 坐标系统	90
3.6.1 预定义的映射模式	90
3.6.2 自定义的坐标系统	99
第四章 Java 入门	103

4.1	Java 基础	103
4.1.1	什么是 Java	103
4.1.2	Java 的出现	104
4.2	Java 语言的特点	105
4.3	Java 小程序和应用程序	108
4.4	Java 开发工具	108
4.4.1	JDK	109
4.4.2	JWS 环境	113
4.4.3	Microsoft Visual J++ 6.0	113
第五章	Java 程序设计基础	118
5.1	标识符	118
5.2	数据类型	118
5.3	运算符	120
5.3.1	单目整数运算符	120
5.3.2	双目整数运算符	120
5.3.3	布尔运算符	121
5.3.4	浮点运算符	121
5.3.5	数组运算符	121
5.3.6	对象运算符	121
5.3.7	强制和转换	121
5.4	Java 流控制	122
5.4.1	分支结构	122
5.4.2	循环结构	122
5.4.3	顺序控制	122
5.4.4	简单实例	123
5.5	类和对象	123
5.5.1	对象实例	123
5.5.2	实例变量	124
5.5.3	成员函数	124
5.5.4	new 操作符	125
5.5.5	点 (.) 操作符	126
5.5.6	成员函数调用	126
5.5.7	实例值 this	126
5.5.8	构造函数 (Constructor)	127
5.5.9	成员函数重载	127
5.5.10	继承	129
5.5.11	super	129
5.5.12	成员函数的覆盖	130
5.5.13	动态成员函数发送	131
5.5.14	final	132

5.5.15 静态	132
5.5.16 抽象	133
5.6 接口和包	134
5.6.1 接口 (Interface)	134
5.6.2 包 (Package)	135
5.7 异常	136
5.7.1 定局语句 finally	137
5.7.2 常见异常	137
第六章 编写 Java 应用程序	140
6.1 应用程序向导	140
6.2 手动创建应用程序	143
6.2.1 新建项目	143
6.2.2 添加控件	144
6.2.3 设置控件属性	146
6.2.4 添加事件处理程序	147
6.2.5 项目存盘并运行应用程序	147
6.2.6 进一步探讨事件处理	148
6.3 改进应用程序	148
6.3.1 添加决策逻辑 (Decision Logic)	148
6.3.2 改变窗体背景颜色	149
6.3.3 添加消息框 (MessageBox)	150
6.3.4 使用 PictureBox 控件显示图片	154
6.3.5 在窗体中加入菜单	154
6.3.6 插入注释	158
6.4 程序调试	160
6.4.1 Visual J++ 6.0 调试器	160
6.4.2 设置断点	160
6.4.3 设置观察变量	162
6.4.4 单步执行	162
6.4.5 Immediate 窗口	163
6.4.6 Output 窗口	163
第七章 创建 Java 小程序	165
7.1 使用 Applet 模板创建小程序	165
7.2 手动创建一个小程序	168
7.2.1 创建小程序框架	168
7.2.2 增添画图功能	170
7.3 绘制图形	173
7.3.1 绘制直线	173
7.3.2 绘制矩形	173
7.3.3 绘制椭圆	174

7.3.4 绘制圆弧	174
7.3.5 绘制多边形	175
7.3.6 复制与清除图形	176
7.4 用户接口组件	177
7.4.1 标签	177
7.4.2 按钮	177
7.4.3 列表框	178
7.4.4 文本域	179
7.4.5 文本区	180
7.4.6 画板	181
7.4.7 滚动条	182
7.5 面板和布局	183
7.6 接收键盘输入	189
7.7 增强小程序	191
7.7.1 加入图形文件	191
7.7.2 播放声音	192
7.7.3 向小程序传递参数	193
7.7.4 用 JScript 与 Applet 通信	195
7.7.5 使用 HTML 控件	198
7.7.6 HTML Outline	199
7.7.7 编写 JScript	201
7.8 多线程与异常处理	203
7.8.1 多线程	203
7.8.2 同步	205
7.8.3 使用 java.lang.Runnable 接口创建线程	206
7.8.4 异常处理	206
7.8.5 异常的处理	208
7.8.6 异常的传递	208
7.8.7 错误	209
第八章 Visual Basic 初探	210
8.1 Visual Basic 简介	210
8.1.1 什么是 Visual Basic	210
8.1.2 Visual Basic 版本简介	211
8.1.3 Visual Basic 企业版的特性	211
8.2 Visual Basic 6.0 的新特性	212
8.2.1 数据访问的新内容	212
8.2.2 Internet 特性的新内容	214
8.2.3 控件的新内容	215
8.2.4 部件创建的新内容	217
8.2.5 语言特性的新内容	218

8.2.6 向导的新内容	219
8.3 Visual Basic 概念	219
8.3.1 Windows 的工作方式: 窗口、事件和消息	220
8.3.2 事件驱动模型	220
8.3.3 交互式开发	220
8.4 Visual Basic 的安装	221
8.4.1 运行安装之前	221
8.4.2 安装 Visual Basic	222
8.5 开发过程中的联机帮助	222
8.5.1 上下文相关帮助	222
8.5.2 运行“帮助”中的代码示例	223
第九章 用 Visual Basic 编程	224
9.1 Windows 编程的基本概念	224
9.1.1 几个关键概念	224
9.1.2 事件驱动的编程方式	224
9.1.3 Visual Basic 进行程序设计的特点	224
9.2 开始使用 Visual Basic	225
9.2.1 启动 Visual Basic	225
9.2.2 认识 Visual Basic 集成开发环境	225
9.3 编写第一个程序	227
9.3.1 工程的概念	227
9.3.2 运行应用程序	227
9.3.3 设计用户界面	228
9.3.4 编写代码	230
9.3.5 生成可执行文件	234
9.4 使用向导	234
第十章 多媒体界面设计	239
10.1 若干基本概念	239
10.1.1 界面样式	239
10.1.2 窗体	240
10.2 使用 Visual Basic 标准控件	245
10.2.1 控件数组	246
10.2.2 命令按钮 (CommandButton)	249
10.2.3 标签控件 (Label)	250
10.2.4 文本框 (TextBox)	252
10.2.5 框架控件 (Frame)	254
10.2.6 单选按钮控件 (OptionButton)	254
10.2.7 复选框 (CheckBox)	255
10.2.8 图像控件 (Image)	258
10.2.9 图片框控件 (PictureBox)	259

10.2.10 线段控件 (Line)	259
10.2.11 形状控件 (Shape)	260
10.2.12 组合框控件 (ComboBox)	262
10.2.13 文件系统控件 (DirListBox、DriveListBox、FileListBox)	263
10.3 窗体对象和多文档界面	265
10.3.1 窗体对象	265
10.3.2 多文档界面 (MID)	266
10.4 改变焦点和 Tab 键顺序	268
10.5 为窗体添加菜单和工具栏	270
10.5.1 菜单	270
10.5.2 工具栏	277
10.6 使用对话框	278
第十一章 VB 下的文本与图形	280
11.1 Windows 坐标系统	280
11.2 添加打印支持	282
11.3 显示和打印文本	283
11.3.1 字体	283
11.3.2 输出文本	284
11.4 显示和打印图形	284
11.4.1 颜色	284
11.4.2 图形方法	286
11.5 程序范例	287
第十二章 VB 中使用多媒体控件	299
12.1 使用 Multimedia MCI 控件集成音乐和视频	299
12.1.1 Multimedia MCI 控件介绍	299
12.1.2 播放 WAV 音频文件	302
12.1.3 播放 avi 视频文件	306
12.1.4 播放音频 CD 中的音乐	310
12.2 ActiveMovie Control 的使用	312
12.3 屏幕保护程序设计	317
12.4 调用 API 播放声音文件	325
第十三章 ActiveX 介绍	329
13.1 ActiveX 的由来	329
13.2 ActiveX 技术的组成	330
13.3 ActiveX 的优势、技术与发展	331
13.4 ActiveX 的构筑工具	334
13.4.1 基本概况	334
13.4.2 开发工具	334
13.5 ActiveX 文档技术	334
13.5.1 ActiveX 文档特性	335

13.5.2 ActiveX 文档服务程序的实现	336
13.6 用 ActiveX 开发多媒体 Web 页	339
13.7 ASP 中 ActiveX 控件的内嵌及调用	340
13.7.1 内嵌各种 ActiveX 控件的方法和属性	341
13.7.2 如何调用用户编写的 ActiveX 控件	345
第十四章 VB 中的 ActiveX 编程	347
14.1 ActiveX 控件的使用	347
14.1.1 播放媒体文件	350
14.1.2 使应用程序具有资源管理器的外观	356
14.1.3 RTF 文本编辑器	364
14.1.4 小结	389
14.2 创建 ActiveX 控件	390
第十五章 VC 下的 ActiveX 编程	398
15.1 什么是 ActiveX 控件	398
15.2 使用 ActiveXMovie 控件的视频播放器	399
第十六章 ActiveX Scripting 技术	419
16.1 ActiveX Scripting 背景	419
16.2 ActiveX Scripting 结构	420
16.3 ActiveX Scripting 实现	423
16.4 ActiveX Scripting 实例	429
16.5 结束语	432

第一章 Visual C++ 6.0 基础

Visual C++是在 Windows 95/98 和 Windows NT 平台下进行 32 位应用程序开发的重要工具。相比其它的编程工具而言, Visual C++在提供可视化的编程方法的同时, 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, 也要优于其它很多的开发工具。随 Visual C++所提供的 Microsoft 基础类库 MFC(Microsoft Foundation Class Library), 对 Windows 95/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, 从而大量地节省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解放出来。Visual C++使 Windows 编程不再深奥和晦涩, 而成为一件有意义并且有趣的事情。

早在 1989 年, Microsoft 的程序员们开始试图将 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努力之后诞生, 却未能被大多数 Windows 程序员所接受。AFX 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, 但是 AFX API 库根本不兼容于现有的 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 的第一个正式版本于 1992 年 3 月随 Microsoft C/C++ 7.0 一起推出。在 MFC 1.0 中还没有文档/视结构, 但有类 CObject 和 CArchive。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 3.51 直到 1993 年 7 月才问世。在 MFC 2.0 中, 增加了对文档/视结构、OLE 1.0、Windows 3.1 公用对话框的支持和消息映射等。在 Windows NT 3.51 面世一个月以后, Microsoft 推出了 32 位的 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。1998 年 Microsoft 又推出了 Visual C++ 6.0, 为我们的编程提供了强大的支持。

1.1 MFC 简介

Visual C++的核心是 Microsoft 基础类库, 即通常所说的 MFC。尽管使用 Visual C++进

行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，当你看到关于 Visual C++ 或是 MFC 的资料时，在绝大多数情况下，它们都是指同一样东西。

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）中的功能，它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，并也可简化其它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。

1.2 Visual C++ 6.0 的使用

1.2.1 运行 Visual C++ 6.0

从开始菜单中运行 Visual C++ 6.0，屏幕显示如图 1.1 所示的界面。

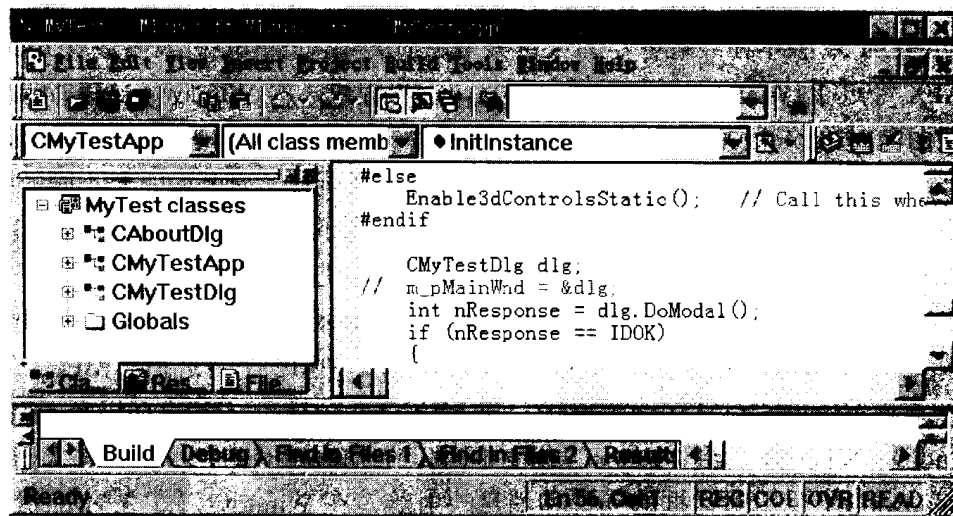


图 1.1 Microsoft Visual C++ 6.0 集成开发环境

当然，你所看到的内容和工具条等也许会有少许的不同，这取决于你的设置。下面我们来讲述其中的主要部分。我们看到，在 Visual C++ 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，你所见到的 Visual C++ 组件和相互之间的位置也会不一样。

在 Visual C++ 中，我们是以工作空间（workspace）和项目（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：

第一种方法是创建一个空白的工作空间，然后向工作空间中添加项目。这时，我们从 Visual C++ 6.0 的 File 菜单下选择 New...菜单项，出现如图 1.2 所示的对话框。选中其中的 Workspaces 选项卡。

在 Workspace name 处键入工作空间的名字，这里假设为 MySpace，则 Visual Studio 将在 Location 所指定的目录下创建名为 MySpace 的子目录，然后以 MySpace.dsw 的文件名将该工作空间保存到这个目录下。

第二种创建工作空间的办法是直接创建一个项目。创建一个新的项目同样是选择 File 菜单下的 New...菜单项，单击 Project 选项卡，如图 1.3 所示。然后在该对话框中选择 Create new workspace 单选钮。这样，Visual C++ 6.0 将创建一个同名的工作空间，然后将所创建的项目添加到该工作空间中。

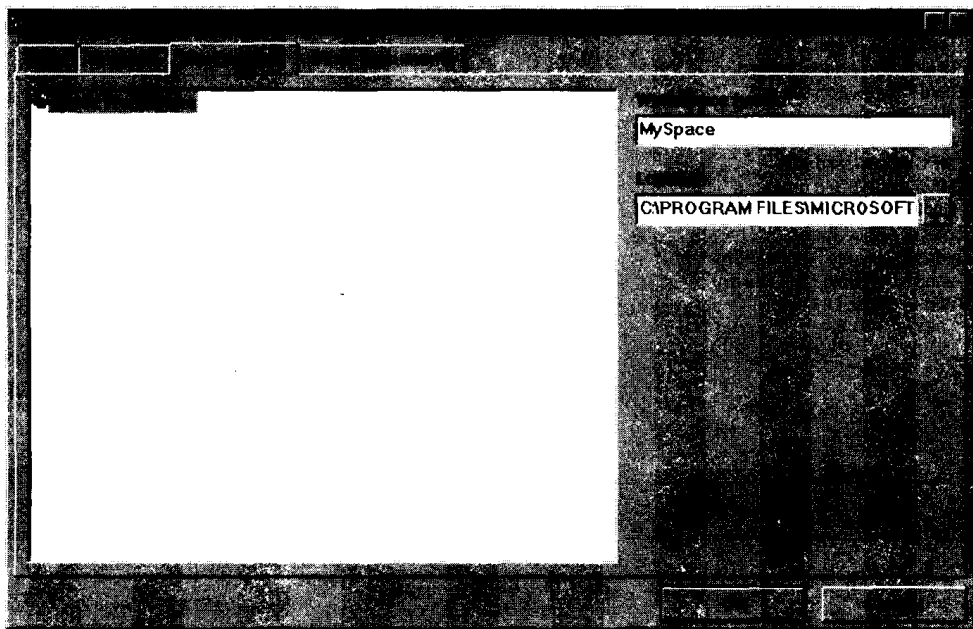


图 1.2 创建空白工作空间

如果仅在工作空间中包括一个项目的话，使用第二种方式显然是很合理的，而且，第二种方式比第一种方式简洁。在今后的很多情况下，我们都将使用第二种方式来创建项目和包括项目的工作空间，但并不是说第一种方式就没有用处了。事实上，在第二种方法中，Visual C++ 将工作空间和项目保存到 Location 所指定的同一个目录下，这对于单个项目的工作空间是合理的。但如果你希望在工作空间中包括多于一个的项目的话，那么你也许希望在保存工作空间的目录下新建子目录来保存这些项目，因为这样更有条理，更利于文件的管理。这时，我们就需要使用第一种方式来创建空白工作空间，然后再在这个工作空间中新建和添加项目。

在工作空间中新建项目的方法和上面的第二种方式几乎一样。只不过这时我们应该在图 1.3 所示的对话框中选择 Add to current workspace（在图 1.3 中，这个单选钮是灰的，这是因为当前并没有打开的工作空间的缘故）。要注意这时 Location 处的目录名是基于当前工作空间所在的目录的。单击 OK 后，Visual C++ 根据在 Project name 处所键入的项目名以 .dsp 的扩展名来保存该项目文件。

对于由 ActiveX 部件（最常见的 ActiveX 部件有 Microsoft Word 和 Excel 等，但是，这

里所指的 ActiveX 部件并不限于 Microsoft 的产品，其它任何符合 ActiveX 部件标准的应用程序都是 ActiveX 部件）创建的文档，你可以在 Visual Studio 窗口内部打开并编辑它们，这时，由该部件提供的菜单项融合进了 Visual Studio 原有的菜单项，由该部件所提供的工具条取代了 Visual Studio 原有的工具条。并且，所打开的文档显示于原有的 InfoViewer Topic 窗口所在位置（如图 1.4 所示，在这幅图中，我们向项目 Project1 中添加了一个新建的 Microsoft Word 文档，在 Visual Studio 内部打开并编辑该文档）。这样，你无需离开 Visual Studio 就可以查看和修改这些文档，这就是 ActiveX 技术所带来的巨大方便之处。

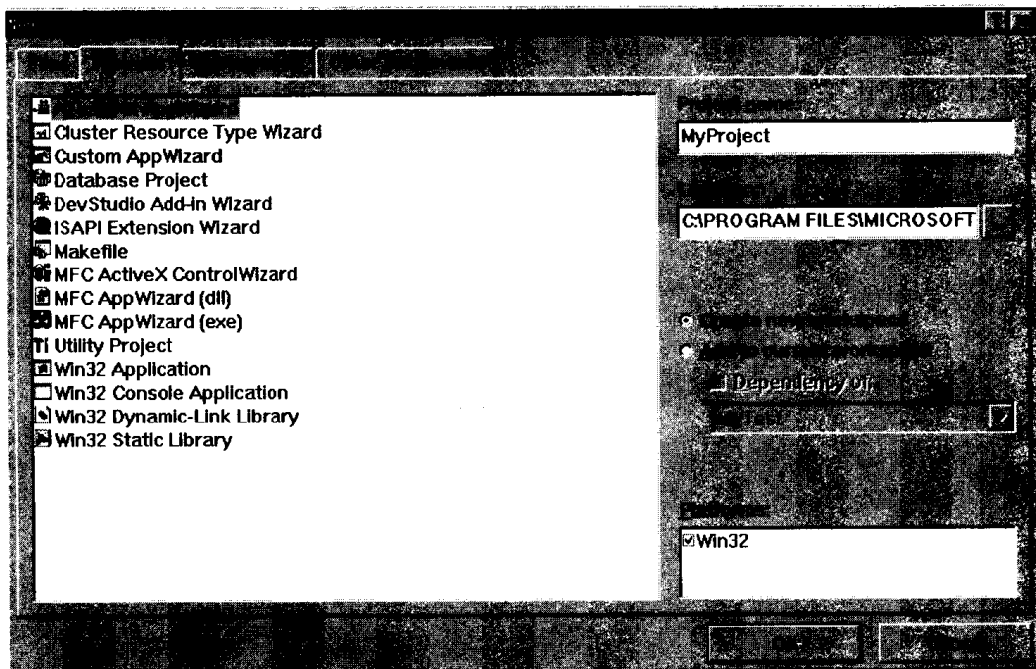


图 1.3 新建一个项目

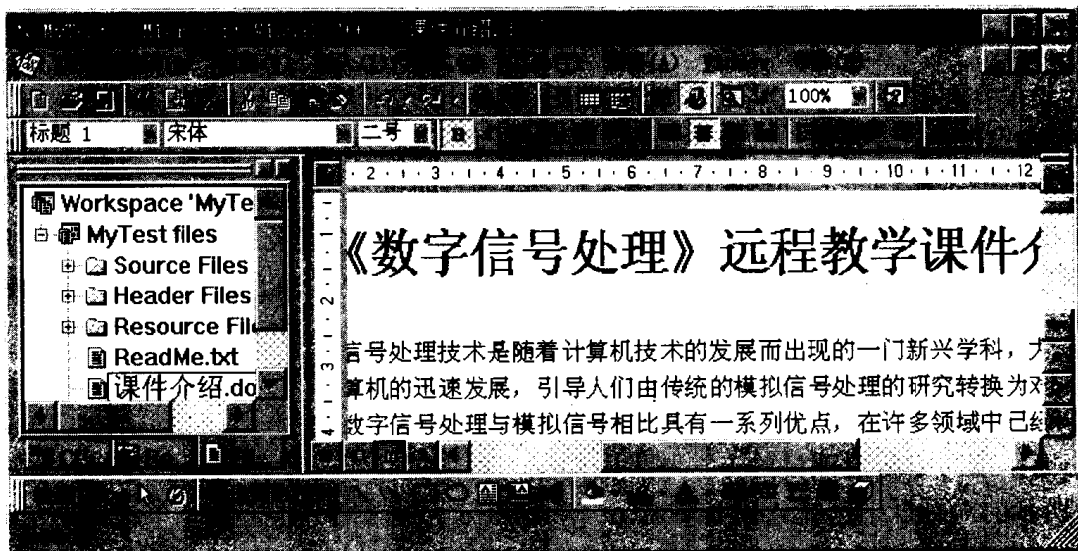


图 1.4 向项目中添加并编辑 Word 文档

向项目中新建这类文档只需在图 1.2 和图 1.3 所示的对话框中选择 Other Document 选项卡，然后指定新建的文档的类型，并给出文档文件名即可。如果是在项目中添加已有的文

档,则必须保证这些文件的扩展名与文档的类型相符合,因为 Visual Studio 是根据相应的文件扩展名来判断文档的类型与寻找创建和维护该文档的 ActiveX 部件的。如果添加由其它非 ActiveX 部件的软件创建和维护的文档,也必须遵从这个约定。对于非 ActiveX 部件的软件创建和维护的文档,在 Visual Studio 选择打开时,Visual Studio 将在另一个单独的窗口中打开该文档以供用户进行编辑。

在 Visual C++ 中的另一个重要的部分是 Workspace 窗口,它包括四个选项卡:ClassView、ResourceView、FileView 和 InfoView,分别用来查看项目的类信息、资源信息、文件信息和在线帮助文档。不论是否存在当前打开项目,InfoView 总是存在的,并且,可以通过 InfoView 得到当前所有安装的开发工具包的在线文档。比如你当前正在开发 Visual C++ 程序,但你一样可以查阅 Visual J++ 的在线文档。

Output 窗口用来显示各种输出信息,如编译和链接信息、调试信息以及查找信息等。

1.2.2 自定义 Visual C++ 6.0

通过对 Microsoft Visual C++ 进行各种自定义设置,可以更有效地使用这个开发工具。可以进行自定义的功能包括:

- 自定义工具条和菜单;
- 自定义快捷键;
- 使用宏。

下面我们分别讲述以上几个方面的内容。

1. 自定义工具条和菜单

在 Tools 菜单中选择 Customize 菜单项,单击 Commands 选项卡,在如图 1.5 所示的对话框中修改 Visual C++ 的菜单命令的默认设置。单击 Toolbars 选项卡可修改工具栏的默认设置。

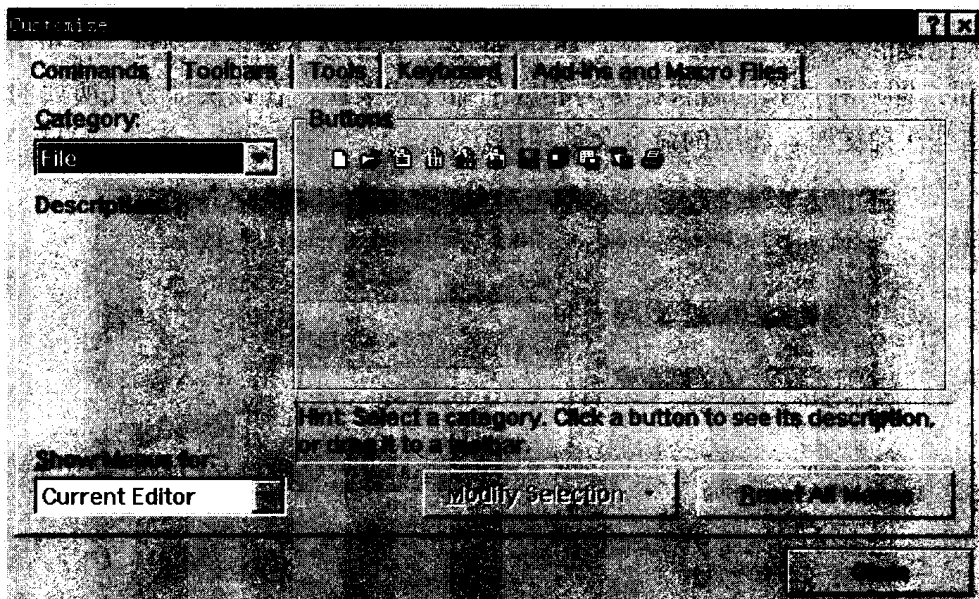


图 1.5 自定义工具条和菜单

我们假定已经编写了一个宏 NewMacro,在 Category 下拉列表框中选择 Macro,然后将宏 NewMacro 直接用鼠标拖放到 Visual C++ 的菜单条中的适当位置,就可以为宏 NewMacro

