

计算机软件开发与程序设计系列丛书

Visual C++ 5

开发人员指南

(美) David Bennett 等著
徐 军 等译

Visual C++ 5
Developer's Guide

机械工业出版社
西蒙与舒斯特
国际出版公司



SAMS
PUBLISHING

CMP



计算机软件开发与程序设计系列丛书

Visual C++ 5 开发 人员指南

(美) David Bennett 等著

徐 军 等译

机 械 工 业 出 版 社
西蒙与舒斯特国际出版公司

本书全面、深入地介绍了 Visual C++ 5.0 的各种功能和新增特性。包括数据库、多媒体以及网络等各个方面。可以说包括了 Windows 编程领域内的所有高级特性。

本书共分七大部分：第一部分主要讲基本的 Visual C++ 5.0 环境；第二部分主要讲 MFC 编程技术；第三部分主要讲 ActiveX 编程技术；第四部分主要讲 Win32 网络编程技术；第五部分主要讲数据库编程技术；第六部分主要讲多媒体编程技术；第七部分主要讲高级 Win32 编程技术。最后介绍了如何创建一个完整的应用程序及所要做的具体工作。

本书既可以作为具备 C++ 基础并且有一定 Windows 编程经验的读者的学习教材，又可以作为软件开发人员的技术参考。

David Bennett, et al: Visual C++ 5 Developer's Guide.

Authorized translation from the English language edition published by Sams Publishing.

Copyright © 1997 by Sams Publishing.

All rights reserved. For sale in Mainland China only.

本书中文简体字版由机械工业出版社和美国西蒙与舒斯特国际出版公司合作出版，未经出版者书面许可，本书的任何部分不得以任何方式复制或抄袭。

本书封面贴有 Prentice Hall 防伪标签，无标签者不得销售。

版权所有，翻印必究。

本书版权登记号：图字：01-98-0630

图书在版编目 (CIP) 数据

Visual C++ 5 开发人员指南 / (美) 本内特 (Bennett, D.) 等著；徐军等译；—北京：机械工业出版社，1998.6

(计算机软件开发与程序设计系列丛书)

书名原文：Visual C++ 5 Developer's Guide

ISBN 7-111-06337-6

I.V… II.①本… ②徐… III.C 语言-程序设计 IV.TP312

中国版本图书馆 CIP 数据核字 (98) 第 10124 号

出版人：马九荣 (北京市百万庄大街 22 号 邮政编码 10037)

责任编辑：傅豫波 张彤漫

北京第二外国语学院印刷厂印刷·新华书店北京发行所发行

1998 年 6 月第 1 版第 1 次印刷

787mm×1092mm 1/16·47.5 印张

印数：0 001—7000 册

定价：80.00 元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

译者序

Visual C++ 是汇集 Microsoft 公司技术精华的主流产品，而 Visual C++ 5.0 版又是其推出的最新版本。对于软件开发人员来说，不熟悉 Visual C++ 是不明智的。因为使用 Visual C++ 5.0 和 MFC 4.2 可以开发强大的 32 位应用程序，能为用户提供全方位的服务，其商业前景不可估量。

本书的作者都是有丰富经验且身处软件开发领域的前沿的软件开发人员，所以他们编的书值得你再三阅读。系统全面是这本书的最大特点。在章节的安排上，本书的各个部分都具有一定的独立性。因此在了解了第一部分的 Visual C++ 5.0 环境以后，你可以选择你感兴趣的部分来阅读。相信无论你是一位转向 Win32 平台的 C++ 编程人员，还是一位有 Windows 开发经验的软件开发人员，这本书对你都非常有用。

本书的翻译是集体劳动的结晶，主要翻译人员有：徐军、徐超、王子燕、王广伟、李长业、朱来文、周志祥、薛朝俊和郭文静。

限于水平和时间，翻译中出现的错误和不妥之处，敬请读者批评指正。

前 言

1. 怎样使用这本书

你可能不太想读完本书（书的结尾并不会给你一个惊喜）。大多数读者只是读他们感兴趣的章节以供参考。有的读者只是在需要的时候参阅这本书。我们已经对本书的各个章节做了安排，以便使它们相互关联。如果你不熟悉 Windows 编程，你应该通读第一部分，当在你读到具体的章节时（如数据库编程），你应该先练习几个例子。

2. 使用本书的条件

我们已假定你会使用 Visual C++ 5.0 在 Windows 95 或 Windows NT 4.0 下进行开发。要构造并运行 CD 盘上的一个示例，你需要具备下列条件：

Windows 95 或 Windows NT 4.0 (Workstation or Server)；

Visual C++ 5.0；

一个运行它们的计算机。

我知道最后一项说得有点模糊。Windows 95（和 Visual C++）能在主频为 25MHz、内存 8MB，并且有 200MB 硬盘空间的 486 机器上运行。然而，如果你每天都要做实效的开发工作，就应该升级你的机器，特别要升级内存和硬盘。

运行 Windows NT 至少需要 12MB 内存。要想有良好的工作性能，至少要 32MB 内存。另外我们推荐至少要有 1GB 的磁盘空间。Windows NT、Microsoft Office、Visual C++ 和各种的 SDK，再加上它们附带的所有例子和联机文档，需要将近 950MB 的硬盘空间。当然，你并不需要使用所有这些东西，这样可以节省一部分磁盘空间。

就处理器的速度而言，我的 Pentium 200MHz、48MB 内存的机器是相当好的，但我通常使用的 486/66、32MB 内存的机器也很合适。

Windows NT 和 Visual C++ 可以运行在非 Intel 平台上，包括 DEC Alpha 和 MIPS 机器，这些机器都有不同的要求。

3. Windows 95 与 Windows NT 4.0

本书讲的是开发 Win 32 应用程序，包括 Windows 95 和 Windows NT 4.0 的应用程序。Windows NT 4.0 发行以后，虽然 WinNT 和 Win95 的用户界面一致了，但是它们底层的操作系统却有一些重要的差别。

通常来说，Windows 95 是面向工业应用的 NT 和传统的 DOS/Windows 3.1 的一种折衷。其中，NT 是一种面向工业的操作系统，其要求工业应用级别的资源；而 DOS/Windows 3.1 对硬件要求低，几乎能运行在所有的机器上，但不是一个全功能、多进程、32 位的操作系统（Win 3.1 并不是一个真正的操作系统，它只是一个基于 DOS 的外壳程序）。

然而，Windows 95 不仅仅是 NT 的一个子集，它也包括一些 NT 中没有的元素。与 Windows 95 不同，Windows NT 不再支持与 16 位应用程序的兼容。确实，你可以在 NT 上运行许多 DOS 或 16 位的 Windows 应用程序，但是任何违反 NT 安全规则的程序将不被执行。这包括大多数直接访问硬件的 DOS 和 16 位 Windows 游戏。

另外,因为 Windows 95 主要是定位于家庭市场的,所以它首先是为 DirectX 和游戏 SDK 开发的操作系统。虽然这些组件中的大多数已经移植到 NT 中,但当我们谈到 DirectX API 时,有些问题仍需讨论。

这本书讨论的大部分主题和 API 对于 Win95 和 WinNT 同样适用。在一些特殊的地方,我会阐述两者之间的差别。

4. Visual C++ 5.0 新增部分

Visual C++ 5.0 版本与上一个版本 (4.0) 相比较,新增加了许多特性。其新增的内容有以下几方面:

(1) 通用部分

1) 标准 C++ 库 Visual C++ 5.0 版本包含了 ANSI 标准 C++ 扩展的头文件和库文件,包括标准模板库。

2) 小块堆内存管理器 C 运行库更改了 Win 32 应用程序的内存管理方式。它使用一种更有效的小块堆管理器来管理小块堆内存的分配。

3) 不再支持 Win32s 从 4.2 版本开始,Visual C++ 不再支持 Win32s。

(2) MFC

1) Internet 编程支持 MFC 现在封装了 ActiveX 和 Win32 Internet (WinInet) 技术,以简化 Internet 编程。

2) 新的 MFC 类和函数 4.2 版本的 MFC 增加了 22 个类和 40 个成员函数。

3) MFC 浏览信息 所包含的浏览信息文件 (MFC.BSC) 可以简化 MFC 资源的浏览。

(3) Internet

1) Internet Server API 类 MFC 为 Internet Server API (ISAPI) 增加了五个类和一个 AppWizard。

2) 访问 Word Wide Web Visual C++ 开发平台能够在开发平台中直接访问 Word Wide Web (万维网)。

(4) ActiveX

向导的改变 Visual C++ 为向导增加了一些新特性,以简化 ActiveX 编程。

(5) 数据库

1) ODBC 的增强 MFC ODBC 类现在支持多线程和多行读取。事务支持也得到了增强。

2) 远程数据捆绑 Visual C++ 包含了能捆绑本地或远程数据资源 (包括 ActiveX 数据资源) 的控件。

(6) 图形

Open Inventor 3D 工具和 IVF AppWizard 包含 Template Graphics Software 公司的工具。

(7) 其他方面

1) 增强的图形编辑器 Visual C++ 图形编辑器现在可以处理 GIF 和 JPEG 文件。

2) 新增的示例 4.2 版本新增加了 13 个 MFC 示例。

3) Win 32 SDK 文档资料 Win 32 文档资料已作了修改,包含了 NT 4.0 下的内容。

5. 与主要作者联系

如果你对这本书有什么问题或意见，或无法使用本书所给的例子，你可以使用 `bennettd@wavefront.com` 这个网址来与我联系。请记住我可不是 Microsoft 的技术支持，如果你的问题是有关 Visual C++ 的工作原理或是自己编写的一些程序的，请不要向我发送 e-mail。你可以利用附录 A（“其他资源”）中列出的各种新闻组和其他资源，来获得这些问题的答复。另外，可通过下列站点下载本书中所有的源代码和示例：WWW.Sampublishing.com。

6. 告诉我们您的想法

作为一个读者，您是这本书最重要的批评家和评论家。我们会慎重考虑您的意见，并且想知道我们做的哪些地方是对的、哪些地方我们可以做得更好、您希望我们出版哪些方面的书籍，以及您的任何好的想法。您能帮助我们编写更能满足您的需要的书籍。

您能访问 Word Wide Web 吗？那么请记下我们的网址 <http://www.mcp.com>

注释 如果你有关于这本书的技术问题，请打技术支持电话：317-581-3833 或向 `support@mcp.com` 发送电子邮件。

作为这本书的主要负责人，我欢迎你提出意见。你可以发传真、电子邮件或直接给我写信，让我知道你是否喜欢这本书，以及如何使我们出版的书更完善。联系方式：

传真：317-581-4669

电子邮件地址：`programming-mgr@sams.mcp.com`

通信地址：Greg Wiegand

Comments Department

Sams Publishing

201 W.103rd Street

Indianapolis, IN 46290

目 录

译者序

前言

第一部分 Visual C++ 5.0 环境

第 1 章 Visual C++ 环境	1
1.1 开发平台	1
1.1.1 观察窗口	2
1.1.2 工具条	3
1.2 项目工作区	3
1.3 工作区窗口	4
1.3.1 FileView	4
1.3.2 Class View	5
1.4 项目配置	6
1.5 转换旧的项目	7
1.6 同时进行多个项目	8
1.7 资源管理	9
1.7.1 资源浏览	9
1.7.2 导入资源	10
1.7.3 管理资源 ID	10
1.7.4 资源模板	11
1.8 编辑资源	12
1.8.1 对话框	12
1.8.2 编辑菜单	14
1.8.3 其他资源	15
1.9 应用程序向导	15
1.9.1 AppWizard 的作用	15
1.9.2 启动 AppWizard	16
1.9.3 项目类型	16
1.10 ClassWizard	17
1.10.1 消息映射	17
1.10.2 成员变量	17
1.10.3 自动化	18
1.10.4 ActiveX 事件	18
1.10.5 类信息	19
1.10.6 增加一个类	19
1.11 组件廊	20
1.12 集成调试器	21

1.12.1 观察数据	21
1.12.2 出现异常后的调试	21
1.13 命令行工具	22
1.14 其他工具	22
1.14.1 Spy++	22
1.14.2 MFC Tracer	22
1.14.3 Register Control	22
1.14.4 ActiveX Control Test Container	22
1.14.5 OLE/COM Object Viewer	22
1.14.6 Error Lookup	23
1.14.7 WinDiff	23
1.14.8 Help Workshop	23
1.15 小结	23

第二部分 通用 MFC 编程

第 2 章 MFC 概览	25
2.1 应用程序框架	25
2.2 用 AppWizard 创建 MFC 应用程序	25
2.2.1 启动 AppWizard	25
2.2.2 了解你的应用程序	30
2.2.3 AppWizard 创建的类	30
2.2.4 CDocument, CView 及 Frame 类	31
2.3 CWinApp 类	31
2.4 CWinThread 类	32
2.5 CCmdTarget 类	32
2.6 CObject 类	32
2.7 运行时类型信息	35
2.8 调试支持	36
2.8.1 AssertValid ()	36
2.8.2 ASSERT 宏	37
2.8.3 Dump ()	37
2.8.4 TRACE () 宏	38
2.9 完整的应用程序	39
2.9.1 WinMain ()	39
2.9.2 InitApplication () 和 InitInsta- tance ()	39
2.9.3 LoadStdProfileSettings ()	41

2.9.4 用命令行工作	41	4.5.1 为文档创建另外一个视	62
2.9.5 创建主窗口	42	4.5.2 CDocument::OnChange ViewList ()	62
2.10 消息循环	42	4.5.3 UpdateAllViews ()	62
2.11 小结	42	4.5.4 从文档访问视	63
第 3 章 MFC 消息和命令	43	4.6 处理框架	63
3.1 消息的种类	43	4.6.1 状态条	63
3.2 MFC 中的消息处理	43	4.6.2 增加一个工具条	65
3.3 消息发送	44	4.6.3 浮动式和固定式工具条	66
3.4 用 ClassWizard 进行消息处理	45	4.7 再谈菜单	67
3.5 创建消息映射	46	4.7.1 更新用户界面	67
3.5.1 填充消息映射	47	4.7.2 弹出式菜单	67
3.5.2 消息映射的内部	49	4.7.3 动态创建菜单	68
3.5.3 实现处理函数	49	4.8 在应用程序中加入分割线	69
3.5.4 PreTranslate Message ()	50	4.8.1 动态分割线	69
3.6 小结	50	4.8.2 创建不同的视	70
第 4 章 框架、文档和视	51	4.8.3 静态分割线	70
4.1 文档/视体系	51	4.9 在应用程序中加入拖放功能	71
4.1.1 单文档与多文档应用程序	51	4.9.1 使拖放有效	71
4.1.2 文档类	51	4.9.2 处理 WM_DROPFILES 消息	71
4.1.3 视类	52	4.10 小结	72
4.1.4 框架	52	第 5 章 对话框和控件	73
4.1.5 文档模板	52	5.1 使用自己的对话框	73
4.2 创建应用程序	52	5.1.1 构造对话框对象	73
4.2.1 创建文档类	52	5.1.2 显示对话框	74
4.2.2 文档数据	53	5.1.3 无模式对话框	74
4.2.3 Serialize ()	54	5.2 用 ClassWizard 创建对话框类	75
4.2.4 OnNewDocument ()	54	5.3 在对话框中使用控件	77
4.2.5 DeleteContents ()	55	5.4 Windows 控件类	78
4.2.6 OnOpenDocument ()	55	5.4.1 CEdit	78
4.2.7 OnSaveDocument ()	56	5.4.2 静态控件	79
4.2.8 OnCloseDocument ()	56	5.4.3 按钮控件	80
4.2.9 ReportSaveLoadException ()	56	5.4.4 列表框控件	81
4.2.10 “脏”标志	56	5.4.5 组合框	82
4.2.11 访问文档	56	5.5 对话框数据交换	83
4.3 视	57	5.5.1 创建自己的 DDX 函数	83
4.4 视类	57	5.5.2 数据有效化	84
4.4.1 CScrollView 类	57	5.6 Windows 通用对话框	85
4.4.2 CFormView 和 CRecordView 类	58	5.6.1 CFileDialog	85
4.4.3 控件视	58	5.6.2 CPrintDialog	85
4.4.4 CEditView	59	5.6.3 CPageSetupDialog	86
4.4.5 使用 CRichEditView	60	5.6.4 CColorDialog	86
4.4.6 CListView 和 CTreeView	60	5.6.5 CFontDialog	86
4.5 文档模板	61		

5.6.6 CFindAndReplace	87	7.3 COleDateTime 和 COleDateTime Span	110
5.6.7 OLE 通用对话框	88	7.4 MFC 的文件类	111
5.7 创建属性表	88	7.4.1 打开文件	111
5.7.1 建立属性页	89	7.4.2 使用 CFile 类进行读写	112
5.7.2 创建属性表	89	7.4.3 使用 CStdioFile 类进行读写	112
5.7.3 用属性表进行对话框数据交换	90	7.4.4 得到有关文件的信息	112
5.7.4 加入应用按钮	90	7.4.5 关闭文件	112
5.7.5 创建自己的向导	91	7.4.6 CMemFile 类	112
5.8 对话框	92	7.4.7 CSharedFile 类	113
5.9 小结	92	7.5 CArchive 类	113
第 6 章 用 MFC 进行绘图和打印	93	7.6 MFC 的集合类	113
6.1 绘制窗口	93	7.6.1 MFC 非模板集合类	113
6.2 CDC 类	93	7.6.2 MFC 的模板集合类	114
6.2.1 GDI 封装的类	93	7.6.3 声明简单集合类	114
6.2.2 映像模式	94	7.6.4 使用数组	115
6.2.3 使用 GDI 对象	95	7.6.5 模板基集合类的帮助函数	115
6.2.4 用 CDC 绘图	99	7.6.6 使用 CList 类	118
6.2.5 其他热门绘图话题	100	7.6.7 使用映像类	119
6.3 打印	101	7.7 异常处理	119
6.3.1 打印过程	101	7.7.1 用 C++ 进行异常处理	120
6.3.2 打印循环	102	7.7.2 发出异常	121
6.4 打印预览	103	7.8 MFC 异常类	121
6.5 小结	103	7.8.1 CMemoryException	122
第 7 章 通用类	104	7.8.2 CNotSupportedException	122
7.1 CString	104	7.8.3 CFileException	122
7.1.1 CString 和 C 字符串	104	7.8.4 CArchiveException	123
7.1.2 CString 和 Unicode	105	7.8.5 CResourceException	124
7.1.3 CString 的内存分配问题	105	7.8.6 COleException	124
7.1.4 创建 CString 对象	105	7.8.7 CUserException	124
7.1.5 CString 表达式	106	7.8.8 CDBException	124
7.1.6 处理单个的字符	106	7.8.9 CDaoException	124
7.1.7 比较 CString 的值	106	7.8.10 CInternetException	125
7.1.8 格式化 CString	107	7.9 Win32 系统级异常	125
7.1.9 其他 CString 函数	107	7.10 小结	126
7.1.10 什么时候不用 CString 类	108	第 8 章 内存管理	127
7.2 CTime 类和 CTimeSpan 类	108	8.1 Win 32 内存模式	127
7.2.1 构造 CTime 对象	108	8.1.1 Win 32 内存地址空间	127
7.2.2 CTime 和时区	108	8.1.2 Win 32 保留地址	128
7.2.3 GetCurrentTime ()	109	8.1.3 16 位与 32 位的区别	128
7.2.4 格式化时间	109	8.2 简单内存分配	129
7.2.5 提取时间值	110	8.2.1 栈结构分配	130
7.2.6 CTime 运算	110	8.2.2 堆内分配	130
7.2.7 CTimeSpan 类	110		

8.2.3 GlobalMemoryStatus ()	132	11.1 OLE 历史的简要回顾	161
8.3 虚拟内存	132	11.2 最终用户对 OLE 的见解	163
8.4 共享内存	133	11.3 程序开发人员对 OLE 的看法	167
8.5 小结	135	11.3.1 组件对象模型	168
第 9 章 用 MFC 进行多线程编程	137	11.3.2 结构化存储	169
9.1 在应用程序中使用多线程	137	11.3.3 Monikers	171
9.1.1 MFC 对象和线程	137	11.3.4 统一数据传输	172
9.1.2 线程的类型	137	11.3.5 OLE 文档	173
9.2 使用工作线程	138	11.3.6 OLE 自动化	174
9.2.1 开始线程	138	11.3.7 OLE 控件	175
9.2.2 编写线程函数	139	11.4 OLE 技术扩展到 ActiveX	176
9.2.3 访问线程的返回码	139	11.4.1 ActiveX 文档	176
9.3 用户界面线程	140	11.4.2 ActiveX 控件	176
9.3.1 创建线程	140	11.4.3 COM	177
9.3.2 创建线程类	140	11.4.4 Internet Monikers	178
9.3.3 在线程中处理消息	141	11.5 ActiveX 技术	178
9.3.4 中止线程	142	11.5.1 ActiveX 超级链接	178
9.3.5 线程局部存储区	142	11.5.2 ActiveX 协议	178
9.4 线程同步	143	11.5.3 ActiveX 服务器扩展	179
9.4.1 潜在的错误	143	11.5.4 ActiveX Script	179
9.4.2 发送事件	146	11.5.5 代码签名	179
9.5 创建一个新进程	148	11.5.6 HTML 扩展	179
9.6 小结	149	11.5.7 ActiveMovie	179
第 10 章 动态链接库	150	11.6 小结	179
10.1 使用动态链接库	150	第 12 章 ActiveX 文档	180
10.1.1 输入库	150	12.1 什么是 ActiveX 文档	180
10.1.2 调用约定	150	12.2 MFC 有些什么新内容	181
10.1.3 加载 DLL	151	12.3 有关 ActiveX 文档的一些细节	181
10.1.4 动态加载 DLL	152	12.4 COM 接口	184
10.2 创建自己的 DLL	153	12.4.1 IOleObject	184
10.2.1 从 DLL 中输出函数	154	12.4.2 IDataObject	185
10.2.2 DLL 内存	155	12.4.3 IPersistStorage	186
10.2.3 建立 DLL	156	12.4.4 IPersistFile	187
10.3 DLL 和 MFC	156	12.4.5 IOleDocument	187
10.3.1 普通型 MFC DLL	156	12.4.6 IOleInPlaceObject	187
10.3.2 扩展 MFC DLL	157	12.4.7 IOle InPlaceActiveObject	188
10.3.3 调试 DLL	158	12.4.8 IOleDocumentView	189
10.3.4 Dumpbin.exe	158	12.4.9 Iprint	189
10.4 小结	159	12.4.10 IOleCommandTarget	190
第三部分 ActiveX 编程		12.5 Active 模板库 (ATL)	190
第 11 章 ActiveX 概览	161	12.6 ACTIVEDOC 程序	193
		12.6.1 activect1.h	195
		12.6.2 oledocument.h	200

12.6.3	activedoc.htm	204	14.5	设计 MFC 自动化服务器	257
12.7	小结	205	14.5.1	使用 AppWizard	257
第 13 章	ActiveX 容器	206	14.5.2	细看 MFCAuto 类	257
13.1	ActiveX 容器	206	14.5.3	在 MFCAuto 样例框架程序中 加入自己的代码	258
13.2	MFC 中新加入了些什么	207	14.5.4	测试 MFCAuto 例子	261
13.3	关于 ActiveX 容器的细节	207	14.6	使用 ActiveX 模板库来设计自动化 服务器	263
13.3.1	结构化存储	208	14.6.1	使用 ATL COM AppWizard	263
13.3.2	Monikers	208	14.6.2	定制 ATLAuto 例子	264
13.3.3	统一数据传输	209	14.6.3	测试 ATLAuto 例子	267
13.3.4	嵌入式对象	209	14.7	小结	268
13.3.5	链接式对象	209	第 15 章	ActiveX 控件	269
13.3.6	拖放	209	15.1	简要回顾	269
13.3.7	在位激活	209	15.2	什么是 ActiveX/OLE 控件	271
13.3.8	ActiveX 文档	210	15.3	ActiveX/OLE 控件结构体系	272
13.4	COM 接口	210	15.3.1	属性	272
13.4.1	IOleInPlaceFrame	211	15.3.2	事件	276
13.4.2	IOleCommandTarget	212	15.3.3	方法	278
13.4.3	IOleInPlaceUIWindow	212	15.4	OLE 控件接口	280
13.4.4	IOleContainer	212	15.5	ActiveX 控件	282
13.4.5	IOleClientSite	213	15.5.1	支持 IUnknown 接口	282
13.4.6	IAadviseSink	213	15.5.2	ActiveX 必须自注册	284
13.4.7	IOleDocumentSite	214	15.5.3	组件类型	288
13.4.8	IOleInPlaceSite	214	15.5.4	组件类型以及互操作性	290
13.4.9	IContinueCallback	215	15.5.5	代码签名	290
13.5	建立一个 ActiveX 容器	216	15.5.6	性能考虑	292
13.5.1	XContainer 程序	216	15.6	重用代码	295
13.5.2	生成 XContainer	216	15.6.1	Visual C++ ActiveX 控件	296
13.5.3	回顾代码	220	15.6.2	Internet Explorer 库存控件	296
13.5.4	加入新的接口	221	15.7	测试 ActiveX/OLE 控件	297
13.6	小结	229	15.7.1	在网页上显示控件	297
第 14 章	ActiveX 服务器	230	15.7.2	ActiveX Control Pad	301
14.1	ActiveX 服务器	230	15.7.3	ActiveX/OLE 控件测试容器	302
14.2	用服务器组件进行三层开发	231	15.7.4	开发工具中的 ActiveX/OLE 控 件	302
14.3	MFC 对 ActiveX 模板库	232	15.8	创建 ActiveX/OLE 控件的方法	303
14.4	设计一个 MFC 小型服务器	232	15.9	ActiveX 模板库 (ATL)	303
14.4.1	AppWizard 的操作步骤	232	15.10	ActiveX 开发人员工具箱 (BaseCtl Framework)	305
14.4.2	详细了解 GLServer 类	234	15.10.1	获得 BaseCtl Framework	306
14.4.3	在激活过程中合并容器和服务 器的菜单	243	15.10.2	建立 BaseCtl Framework	306
14.4.4	测试 GLServer 框架	243	15.11	用 Visual C++ 和 MFC 创建一个	
14.4.5	往 GLServer 框架中添加定制部 分	244			
14.4.6	测试 GLServer 的例子	256			

ActiveX/OLE 控件	307	16.7.6 接受与服务器的连接	333
15.11.1 获得 ActiveX SDK	308	16.7.7 发送并接收流式数据	333
15.11.2 使用合适的 Visual C++ 和 MFC 版本	308	16.7.8 关闭套接字	334
15.11.3 使用 Visual C++ 和 MFC 开发 ActiveX 和 OLE 控件	309	16.8 使用 CSocket 对象	334
15.11.4 MFC 对 ActiveX 和 OLE 控件的 封装	309	16.8.1 创建 CSocket 对象	334
15.11.5 OLE Control Wizard	313	16.8.2 建立连接	334
15.12 小结	316	16.8.3 发送和接收数据	335
第四部分 Win 32 网络编程		16.8.4 与 CSocket 一起使用 CArchive ...	335
第 16 章 Windows 套接字	317	16.8.5 利用 CArchive 对象来发送和接 收数据	335
16.1 什么时候使用套接字	317	16.8.6 清除对象	336
16.2 什么是套接字	317	16.9 Windows Sockets 2	336
16.2.1 数据报套接字	317	16.9.1 WinSock 2 的新函数	336
16.2.2 流式套接字	318	16.9.2 使用多种协议	338
16.3 使用 WinSock API	318	16.9.3 多协议名分辨	339
16.3.1 初始化 WinSock	318	16.9.4 重叠和分散/聚集 I/O	340
16.3.2 创建套接字	319	16.9.5 服务质量	341
16.4 使用数据报套接字	320	16.9.6 多点套接字	341
16.4.1 bind ()	320	16.9.7 共享的套接字	341
16.4.2 创建服务器套接字	323	16.9.8 扩展的连接操作	341
16.4.3 字节顺序转换	324	16.9.9 具体协议扩展	342
16.4.4 从数据报套接字接收数据	324	16.10 小结	342
16.4.5 向数据报套接字发送数据	324	第 17 章 管道和邮槽	343
16.5 使用流式套接字	325	17.1 匿名管道	343
16.5.1 流式套接字和服务端	325	17.1.1 创建一个匿名管道	343
16.5.2 侦听连接	326	17.1.2 继承的句柄	343
16.5.3 接受连接	326	17.1.3 使用管道改变标准输入和输出 的方向	344
16.5.4 请求连接	327	17.1.4 向管道写入数据	344
16.5.5 使用流式套接字来发送和接收 数据	328	17.1.5 由管道读出数据	344
16.6 等待事件	329	17.1.6 为标准输出使用匿名管道	345
16.6.1 使用 Select () 函数	329	17.2 命名管道	347
16.6.2 使用 WSAAsyncSelect ()	329	17.2.1 创建一个命名管道	347
16.6.3 关闭套接字	331	17.2.2 创建额外的实例	348
16.7 MFC 和 WinSock API	331	17.2.3 侦听连接	349
16.7.1 创建 CAsyncSocket 对象	331	17.2.4 重叠 I/O	350
16.7.2 CAsyncSocket 对象的错误处理 ...	332	17.2.5 与命名管道相连接	351
16.7.3 套接字选项和 CAsyncSocket ...	332	17.2.6 向命名管道写入	351
16.7.4 发送和接收数据报	332	17.2.7 由命名管道读取数据	352
16.7.5 套接字与服务器连接	332	17.2.8 命名管道的操作	353
		17.2.9 关闭管道	353
		17.3 邮槽	353
		17.3.1 创建邮槽	354

17.3.2 从邮槽读数据	354	19.3.2 用 CMC 发送信报	381
17.3.3 向邮槽写入数据	355	19.3.3 用 CMC 分辨信报的地址	383
17.3.4 关闭邮槽	355	19.3.4 用 CMC 接收信报	385
17.4 小结	355	19.3.5 CMC 数据扩展	387
第 18 章 远程调用 (RPC)	356	19.4 简单 MAPI	389
18.1 开发 RPC 应用程序	357	19.5 扩展 MAPI	389
18.2 定义接口	358	19.6 OLE 信报库	390
18.2.1 通用唯一标识符 (UUID)	358	19.7 小结	390
18.2.2 端点	359	第 20 章 电话 API (TAPI)	391
18.2.3 接口正文	359	20.1 辅助电话服务提供程序	392
18.2.4 IDL 属性	360	20.1.1 发出一次呼叫	392
18.2.5 应用程序配置文件 (ACF)	360	20.1.2 获取位置信息	392
18.2.6 使用 MIDL 编译器	360	20.2 基本电话程序	393
18.2.7 MIDL 编译器和 Visual C++	361	20.2.1 初始化 TAPI	393
18.3 开发 RPC 服务器	361	20.2.2 异步 TAPI 函数	394
18.3.1 基础函数	361	20.2.3 协商 API 版本	396
18.3.2 RPC 服务器实体	362	20.2.4 设备功能	396
18.3.3 协议序列	362	20.2.5 发出出站呼叫	398
18.3.4 注册接口	362	20.2.6 接收 TAPI 消息	400
18.3.5 将捆绑输出到名称服务程序	363	20.2.7 结束一次呼叫	401
18.3.6 侦听请求	363	20.2.8 接收入站呼叫	402
18.3.7 建立服务器	365	20.2.9 媒体流设备	403
18.4 RPC 客户机	365	20.2.10 TAPI 地址处理	404
18.5 RPC 中的参数传递	366	20.2.11 以辅助 TAPI 服务提供程序的 形式注册	404
18.5.1 数组和字符串	366	20.3 增补电话 API	404
18.5.2 联合	367	20.3.1 附加行设备函数	405
18.5.3 指针	367	20.3.2 电话设备	406
18.5.4 存储分配	367	20.4 扩展电话 API	407
18.6 捆绑到 RPC 服务器	368	20.5 小结	407
18.6.1 自动捆绑	368	第 21 章 加密 API (CAPI)	408
18.6.2 手工捆绑	369	21.1 加密 101	408
18.7 RPC 的异常处理	373	21.1.1 数据加密	408
18.8 小结	373	21.1.2 数字签名	408
第 19 章 信报 API	375	21.1.3 对话密钥	409
19.1 MAPI 结构	375	21.1.4 公共和私人密钥	409
19.1.1 客户应用程序	375	21.1.5 编码类型	409
19.1.2 服务提供程序	375	21.2 CryptoAPI 体系结构	410
19.1.3 MAPI 假脱机程序	376	21.2.1 密钥库	410
19.1.4 MAPI API	376	21.2.2 CryptoAPI 配置	410
19.1.5 MAPI 接口的可用性	377	21.3 与服务提供程序连接	411
19.2 在 MFC 中支持 MAPI	377	21.4 密钥处理	412
19.3 通用信报调用	378	21.4.1 创建一个密钥	412
19.3.1 开始对话	379		

21.4.2 从口令中派生出密钥	413	22.4.3 获取响应信息	437
21.4.3 密钥参数	414	22.4.4 对 Cookie Jar 初步了解	439
21.5 密钥交换	414	22.4.5 使用高速缓存	439
21.5.1 输出一个密钥	414	22.5 Gopher 客户机函数	440
21.5.2 输入一个密钥	416	22.6 MFC WinInet 类	440
21.6 数据加密	416	22.6.1 CInternetSession	440
21.6.1 多块数据加密	417	22.6.2 连接类	440
21.6.2 加密时进行散列编码	418	22.6.3 文件类	441
21.6.3 数据解密	418	22.6.4 CInternetException	441
21.7 数字签名	418	22.6.5 使用 CInternetSession	441
21.7.1 创建数字签名	419	22.6.6 FTP 处理	445
21.7.2 验证数字签名	421	22.6.7 HTTP 处理	446
21.8 小结	422	22.6.8 Gopher 处理	448
第 22 章 WinInet API	423	22.7 小结	449
22.1 WinInet C API 的使用	423	第 23 章 Internet Server API (ISAPI)	450
22.1.1 句柄	423	23.1 开发服务器扩展应用程序	450
22.1.2 错误处理	423	23.1.1 ISA 入口点	450
22.1.3 传送缓冲区参数	424	23.1.2 扩展控制块 (ECB)	453
22.1.4 异步 I/O	424	23.1.3 文件发送	457
22.2 普通 Internet 函数	424	23.1.4 异步操作	458
22.2.1 开始一个 WinInet 对话	424	23.1.5 写服务器日志	459
22.2.2 设置句柄选项	425	23.1.6 ISA 异常处理	459
22.2.3 查询句柄选项	426	23.1.7 调试 ISA	460
22.2.4 确认 Internet 的连接	428	23.1.8 从 CGI 到 ISAPI 的转换	460
22.2.5 与服务器连接	428	23.2 ISAPI 过滤程序	461
22.2.6 增加一个回调函数	429	23.2.1 安装过滤程序	461
22.2.7 URL 处理	430	23.2.2 ISAPI 过滤程序结构	461
22.2.8 基本文件操作	430	23.2.3 HTTP_FILTER_CONTEXT 结构	464
22.2.9 查询数据可用性	431	23.2.4 过滤程序通知处理	467
22.2.10 数据读取	431	23.3 MFC 对 ISAPI 的支持	470
22.2.11 移动文件指针	431	23.4 Internet 服务管理器 API	474
22.2.12 向 Internet 文件写数据	432	23.5 小结	474
22.3 FTP 客户机函数	432		
22.3.1 当前目录	432		
22.3.2 创建和更换目录	433		
22.3.3 查找文件	433		
22.3.4 获取文件	433		
22.3.5 发送文件	434		
22.3.6 在 FTP 服务器上打开一个文件	434		
22.3.7 其他 FTP 文件操作	435		
22.4 HTTP 客户机函数	435		
22.4.1 HTTP 请求	435		
22.4.2 HttpSendRequest () 错误处理	437		

第五部分 数据库编程

第 24 章 数据库概览	477
24.1 ODBC 3.0	477
24.2 MFC 的数据库类	478
24.3 OLE DB	478
24.4 ActiveX 数据对象 (ADO)	478
24.5 结构化查询语言	478
24.6 数据定义语言 (DDL)	479
24.6.1 表	479

24.6.2 索引	481	25.6.3 SQLFetch ()	515
24.6.3 视	482	25.6.4 关闭光标	516
24.7 数据控制语言 (DCL)	483	25.6.5 重复使用语句句柄	516
24.7.1 授与优先权	483	25.6.6 SQLGetData ()	516
24.7.2 取消优先权	483	25.6.7 列的信息	517
24.8 数据处理语言 (DML)	483	25.7 一次取回多个行	517
24.8.1 SELECT 语句	484	25.7.1 块光标	517
24.8.2 INSERT 语句	493	25.7.2 可滚动光标	518
24.8.3 DELETE 语句	494	25.7.3 ODBC 光标库	518
24.8.4 UPDATE 语句	494	25.7.4 使用块光标	518
24.8.5 调用过程	494	25.7.5 用可滚动光标	520
24.9 小结	495	25.8 插入、更新和删除行	522
第 25 章 开放数据库连接 (ODBC)	496	25.9 异步操作	526
25.1 ODBC 体系结构	496	25.10 事务处理	526
25.1.1 ODBC 驱动程序	497	25.10.1 ODBC 提交模式	526
25.1.2 ODBC 驱动管理器	498	25.10.2 事务隔离层次	527
25.1.3 数据源	498	25.10.3 光标并发类型	528
25.1.4 ODBC 数据源管理器	498	25.11 函数目录	529
25.1.5 ODBC 安装和设置编程	499	25.12 小结	529
25.2 ODBC API 基础	499	第 26 章 MFC 数据库类	530
25.2.1 ODBC 句柄	499	26.1 CDatabase 类	530
25.2.2 ODBC 数据类型	500	26.1.1 打开一个数据库连接	530
25.2.3 ODBC 诊断	502	26.1.2 连接信息	532
25.3 创建 ODBC 应用程序	503	26.1.3 用 CDatabase 执行 SQL 语句	532
25.3.1 分配 ODBC 环境	504	26.1.4 用 CDatabase 进行事务处理	533
25.3.2 分配连接句柄	504	26.1.5 直接使用 ODBC API	534
25.3.3 连接选项	504	26.2 CRecordset 类	534
25.4 与一个数据资源连接	505	26.2.1 从 CRecordset 中派生类	534
25.4.1 SQLConnect ()	505	26.2.2 记录字段交换 (RFX)	536
25.4.2 SQLDataSources ()	507	26.2.3 打开记录集	537
25.4.3 SQLDriverConnect ()	508	26.2.4 刷新记录集	539
25.4.4 SQLBrowseConnect ()	508	26.2.5 关闭记录集	539
25.4.5 获取连接信息	509	26.2.6 在记录集中移动	539
25.4.6 切断连接	509	26.2.7 成批取回行	540
25.5 执行 SQL 语句	510	26.2.8 改变记录集中的数据	544
25.5.1 语句句柄	510	26.2.9 事务处理	545
25.5.2 SQLExecDirect ()	510	26.2.10 使用语句参数	545
25.5.3 准备 SQL 语句	511	26.2.11 记录集错误处理	546
25.5.4 SQLExecute ()	511	26.3 CRecordView 类	546
25.5.5 参数处理	511	26.3.1 创建一个 CRecordView	546
25.6 处理结果集	514	26.3.2 用 CRecordView 进行对话框数 据交换	547
25.6.1 捆绑列	514	26.3.3 OnGetRecordset ()	547
25.6.2 SQLBindCol ()	514		

26.3.4 CRecordView:: OnMove ()	547
26.4 用 AppWizard 创建数据库应用程序	548
26.5 小结	548
第 27 章 使用 OLE DB	549
27.1 OLE DB 组件	549
27.1.1 数据源	549
27.1.2 会话	550
27.1.3 事务对象	550
27.1.4 命令	551
27.1.5 行集	551
27.1.6 枚举器	552
27.1.7 错误	552
27.2 OLE DB 数据使用者应用程序	552
27.2.1 应用程序结构	552
27.2.2 初始化 COM	552
27.3 连接数据提供程序	553
27.3.1 创建一个提供程序实例	553
27.3.2 设置提供程序属性	554
27.3.3 使用 IDBInitialize	557
27.4 创建会话	558
27.5 打开行集	558
27.6 执行命令	561
27.6.1 创建命令	561
27.6.2 设置命令文本	562
27.6.3 执行命令	562
27.6.4 准备命令	563
27.6.5 使用带参数的命令	563
27.7 操作行集	566
27.7.1 获得列信息	566
27.7.2 操作访问器	568
27.7.3 获取行	570
27.7.4 使用 IRowsetLocate	572
27.8 更新数据源	573
27.8.1 IRowsetChange	573
27.8.2 IRowsetUpdate 和延迟更新模式	573
27.9 事务	575
27.9.1 开始一个事务	575
27.9.2 提交一个事务	576
27.9.3 终止一个事务	576
27.10 使用枚举器	577
27.10.1 实例化根枚举器	577
27.10.2 ISourcesRowset::GetSources Rowset ()	577
27.10.3 从源行集中实例化对象	578
27.11 扩展的 OLE DB 错误处理	580
27.12 小结	582
第 28 章 使用 ActiveX 数据对象 (ADO) 编程	583
28.1 ADO 数据源	583
28.2 ADO 对象	583
28.3 Connection 对象	584
28.3.1 Connection 属性	584
28.3.2 Connection 方法	584
28.4 Command 对象	584
28.4.1 Command 属性	585
28.4.2 Command 方法	585
28.5 Recordset 对象	585
28.5.1 Recordset 属性	585
28.5.2 Recordset 方法	586
28.6 Fields 集合和 Fields 对象	587
28.6.1 Fields 集合成员	587
28.6.2 Fields 对象属性	587
28.6.3 Field 对象方法	587
28.7 Parameter 对象和 Parameter 集合	587
28.7.1 Parameter 集合成员	587
28.7.2 Parameter 对象属性	588
28.7.3 Parameter 对象方法	588
28.8 Property 对象和 Properties 集合	588
28.8.1 Properties 集合成员	588
28.8.2 Property 对象属性	588
28.9 Error 对象和 Errors 集合	589
28.9.1 Errors 集合成员	589
28.9.2 Error 对象属性	589
28.10 在 VC++ 中使用 ADO 对象	589
28.10.1 初始化 COM	589
28.10.2 使用 #import	590
28.10.3 COM 支持类	590
28.10.4 智能指针和 ADO	591
28.11 处理 ADO 错误	593
28.11.1 ADO C++ 异常	593
28.11.2 ADO HRESULT 的值	594
28.11.3 Error 集合	594
28.12 打开一个连接	596
28.12.1 连接超时	596
28.12.2 连接访问模式	597
28.12.3 缺省数据库	597