

三味创作室 编著

Visual C++

6.0

应用与提高

新版软件系列丛书

科学出版社

新版软件系列丛书

Visual C + + 6.0 应用与提高

三味创作室 编著

科学出版社

1999

前 言

在以前,为 Windows 开发应用程序是一件令人头疼的事情,数量巨大的 API 函数一方面提供了丰富的功能,但另一方面也使得程序员学习使用 API 函数进行开发变得异常艰难。自从 Microsoft 公司推出 Visual C++ 1.0 以来,这一过程变得容易了。现在,Microsoft 公司继 Visual C++ 1.0,1.5,2.0,4.x,5.0 等版本之后,又推出了 Visual C++ 6.0,其功能更强大,界面更友好,更加方便了用户对 Windows 应用程序的开发。

本书将向读者介绍如何使用 Visual C++ 6.0 的 MFC 类库(Microsoft Foundation Classes)建立 32 位应用程序,包括文档/视结构、GUI 界面、对话框与控件、打印基本技术,以及多媒体、动态链接库、ActiveX 技术、Internet 连接等高级话题。

第一章至第三章是本书的入门部分,主要目的是引导新手使用 Visual C++ 6.0 的集成开发环境,并利用 AppWizard 创建一个程序。

第四章至第十一章是本书的提高部分,重点介绍了 MFC 编程的常用方法,包括消息与命令机制、文档/视结构、绘图与文本、文档的序列化、对话框与控件、打印与打印预览、改善界面等。经过对这一部分的学习,读者可以比较熟练地编制出符合一般要求的程序。

文档/视结构将应用程序的数据与数据的显示分开处理,文档负责存储、加载数据,而视则负责显示数据和接受用户的输入。这一结构使得开发者处理数据的思路更加清晰。对话框和控件则是接受用户输入的最主要途径。绘图与文本输出则是显示数据的最基本的途径。为了永久地保存应用程序的数据,一方面可以通过文档序列化将数据存储在磁盘上,另一方面可以通过打印将数据保存在纸上。而改善程序界面将使用应用程序易于为用户接受。

第十二章至第十四章是本书的高级部分,主要介绍了 MFC 编程的一些高级话题,如数据库(ODBC)操作、ActiveX 应用、Internet 连接等。

ODBC 技术允许用户访问多种数据库文件,扩大了程序处理数据的能力。ActiveX 作为 OLE 的后继产品,促进了应用程序的通信能力。Internet 支持使得应用程序能与更大范围内的计算机进行通信。

本书提供了丰富的示例程序来对上面涉及的这些方面进行阐述,并尽可能详细、清晰地讲解开发的关键部分。但由于篇幅和时间的限制,本书只能有重点地介绍使用 Visual C++ 进行应用程序开发的一些关键的技术。我们相信读者通过本书的学习,完全可以自己研究和理解本书未涉及或未做详细介绍的其他内容。

在本书的编写过程中得到了白韬、庄永龙等人的帮助,作者在此表示感谢。

由于时间紧迫和作者水平有限,本书肯定存在一些疏漏或错误之处,希望读者提出批评和改进意见。

张龙潇 王沁磊

1999 年 1 月

目 录

第一章 Visual C++ 概述	(1)
1.1 Visual C++ 系列的发展	(1)
1.2 Visual C++ 6.0 的不同版本	(1)
1.2.1 标准版	(1)
1.2.2 专业版	(2)
1.2.3 企业版	(2)
1.3 Visual C++ 6.0 的新特性	(2)
1.3.1 MFC 基础类库的改进与增强	(2)
1.3.2 数据库支持的增强	(2)
1.3.3 实用工具	(3)
1.3.4 向导	(3)
1.3.5 编译器、调试器与连接器	(3)
1.3.6 编辑器	(3)
1.4 Windows 下的编程	(4)
1.4.1 Windows 的消息机制	(4)
1.4.2 Windows 的图形设备接口	(4)
1.4.3 Windows 的内存管理	(5)
1.4.4 Windows 的资源	(5)
1.4.5 动态连接库(DLLs)	(5)
1.5 小结	(5)
第二章 集成开发环境的使用	(6)
2.1 外观及基本操作	(6)
2.1.1 正文窗口及其操作	(6)
2.1.2 项目工作台窗口及其操作	(7)
2.1.3 输出窗口及其操作	(8)
2.2 菜单功能介绍	(8)
2.2.1 File 菜单	(8)
2.2.2 Edit 菜单	(9)
2.2.3 View 菜单	(11)
2.2.4 Insert 菜单	(11)
2.2.5 Project 菜单	(12)
2.2.6 Build 菜单	(13)
2.2.7 Tools 菜单	(13)
2.2.8 Window 菜单	(14)

2.2.9 Help 菜单	(15)
2.2.10 Debug 菜单	(16)
2.3 工具栏的使用	(17)
2.4 定制用户界面	(18)
2.4.1 显示/隐藏工具栏	(18)
2.4.2 设计自己的工具栏和菜单	(19)
2.4.3 定制实用程序工具	(21)
2.4.4 定制加速键	(22)
2.5 文本编辑器的使用	(22)
2.5.1 文件的管理	(23)
2.5.2 在文件中定位	(24)
2.5.3 对文件进行编辑	(25)
2.5.4 查找与替换	(26)
2.6 资源与标识符	(28)
2.6.1 资源与资源编辑器	(28)
2.6.2 标识符	(29)
2.7 小结	(30)
第三章 建立第一个应用程序	(31)
3.1 利用 AppWizard 创建一个新的项目	(31)
3.1.1 创建新的项目	(31)
3.1.2 第一步:选择程序的文档支持类型	(32)
3.1.3 第二步:选择程序的数据库支持类型	(33)
3.1.4 第三步:选择程序的复合文档支持类型	(34)
3.1.5 第四步:选择程序的界面选项	(34)
3.1.6 第五步:选择程序的其他选项	(37)
3.1.7 第六步:确认文件和类名	(38)
3.1.8 第七步:生成应用程序	(38)
3.1.9 编译和运行生成的应用程序	(38)
3.2 浏览程序代码	(40)
3.2.1 应用程序类——CFirstApp 类	(40)
3.2.2 主框架类——CMainFrame 类	(46)
3.2.3 文档类——CFirstDoc 类	(48)
3.2.4 视类——CFirstView 类	(50)
3.2.5 其他文件	(52)
3.3 让程序做些工作	(52)
3.3.1 命名法则	(53)
3.3.2 向类中添加成员	(54)
3.4 理解程序的运行	(57)
3.4.1 传统 C 语言应用程序的结构	(57)

3.4.2	MFC 应用程序	(62)
3.5	小结	(63)
第四章	消息与命令	(69)
4.1	消息驱动机制	(69)
4.2	使用菜单工作	(70)
4.2.1	修改菜单资源	(71)
4.2.2	进行消息映射	(74)
4.2.3	添加代码	(76)
4.3	更新菜单状态	(77)
4.4	使用工具条	(81)
4.5	使用加速键	(84)
4.6	消息的传递	(86)
4.7	小结	(87)
第五章	绘图与文本	(88)
5.1	设备环境与图形对象	(88)
5.1.1	设备环境——CDC 类	(88)
5.1.2	图形对象——CGdiObject 类	(91)
5.2	鼠标消息的发送	(92)
5.2.1	客户区鼠标消息	(92)
5.2.2	非客户区鼠标消息	(94)
5.3	在屏幕上绘图	(95)
5.3.1	建立一个新的项目	(95)
5.3.2	添加菜单资源	(96)
5.3.3	使菜单工作	(96)
5.3.4	实现画点功能	(101)
5.3.5	实现画线功能	(103)
5.3.6	实现画矩形与画椭圆功能	(109)
5.3.7	实现填充功能	(111)
5.3.8	实现随手画功能	(113)
5.4	使用画笔和画刷	(116)
5.4.1	建立程序的框架	(116)
5.4.2	创建和使用不同的画笔	(120)
5.4.3	创建和使用不同的画刷	(125)
5.5	输出文本	(127)
5.5.1	字体	(127)
5.5.2	CDC 类的文本处理函数	(131)
5.5.3	建立程序的框架	(133)
5.5.4	在 OnDraw() 函数中实现文本输出	(138)
5.6	小结	(139)

第六章 文档/视结构	(140)
6.1 分析一个应用程序	(140)
6.1.1 建立一个应用程序	(141)
6.1.2 程序运行的流程	(141)
6.1.3 框架窗口类	(143)
6.1.4 文档模板	(144)
6.1.5 文档类	(144)
6.1.6 视类	(145)
6.1.7 程序员的任务	(147)
6.1.8 小结	(147)
6.2 单文档应用	(147)
6.2.1 单文档与多文档	(147)
6.2.2 在文档中加入数据变量	(147)
6.2.3 在视中处理键盘输入	(148)
6.2.4 使用视类的 GetDocument 函数	(149)
6.2.5 将用户输入的字符存入文档	(149)
6.2.6 使用设备描述表显示文本	(150)
6.2.7 处理 WM_CREATE 消息	(151)
6.2.8 在屏幕上显示插入符	(152)
6.2.9 移动插入符	(155)
6.2.10 用 DeleteContents 函数进行数据清除	(155)
6.2.11 用 OnNewDocument 函数进行初始化	(157)
6.2.12 用鼠标定位插入符	(157)
6.2.13 项目 CSingleDoc1 程序清单	(160)
6.3 多文档应用	(167)
6.3.1 建立一个多文档的应用	(168)
6.3.2 分析 AppWizard 产生的 MDI 框架程序	(169)
6.3.3 增强文本编辑器的功能	(171)
6.3.4 设置文档的修改标志	(172)
6.3.5 修改视类的 OnDraw 函数	(174)
6.4 多窗口应用	(175)
6.4.1 程序框架实现的功能	(175)
6.4.2 使文档和视保持一致	(176)
6.4.3 在 OnChar 函数中加入 UpdateAllViews 函数	(177)
6.4.4 修改视类的 OnUpdate 成员函数	(177)
6.4.5 视类的 OnInitialUpdate 函数	(179)
6.4.6 运行程序 MultiDoc1	(179)
6.4.7 程序清单	(179)
6.5 小结	(184)

第七章 文档序列化	(186)
7.1 CFile 类	(186)
7.1.1 创建一个项目 MyFile1	(186)
7.1.2 保存文件	(187)
7.1.3 载入文件	(190)
7.2 文档序列化的概念	(193)
7.3 自定义一个可序列化的类	(193)
7.3.1 从 CObject 中派生你的类	(194)
7.3.2 覆盖该类的 Serialize 成员函数	(194)
7.3.3 使用 DECLARE_SERIAL 宏	(195)
7.3.4 定义无参数的构造函数:	(195)
7.3.5 在实现文件中使用 IMPLEMENT_SERIAL 宏	(195)
7.4 序列化一个对象	(196)
7.4.1 什么是归档对象	(196)
7.4.2 创建一个归档对象	(196)
7.4.3 使用归档类的 << 和 >> 运算符	(197)
7.4.4 通过归档存储和读取对象	(199)
7.5 在应用中实现文档序列化	(201)
7.5.1 应用程序框架实现的功能	(201)
7.5.2 重新定义文档类的函数 Serialize	(202)
7.5.3 应用程序框架的工作内幕	(205)
7.5.4 定义文档类型	(207)
7.6 程序清单	(210)
7.7 小结	(219)
第八章 对话框与常用控件	(220)
8.1 有模式对话框的使用	(220)
8.1.1 建立项目 McdalDialog1	(221)
8.1.2 资源与资源管理	(222)
8.1.3 使用资源编辑器来编辑对话框	(224)
8.1.4 向对话框中加入控件	(226)
8.1.5 加入对话框类和成员变量	(233)
8.1.6 代码规整	(235)
8.2 无模式对话框	(246)
8.2.1 建立 Expl 项目	(246)
8.2.2 修改项目资源	(248)
8.2.3 为对话框指定新类和加入成员函数	(249)
8.2.4 代码规整	(249)
8.2.5 进一步理解 expl	(257)
8.3 小结	(266)

第九章 通用对话框与 Windows95 的新控件	(267)
9.1 通用对话框	(268)
9.1.1 CFileDialog 类的使用	(268)
9.1.2 CPageSetupDialog 类的使用	(277)
9.1.3 CPrintDialog 类的使用	(278)
9.1.4 CFindReplaceDialog 类的使用	(280)
9.1.5 COleDialog 类的使用	(282)
9.2 Win95 的新控件	(283)
9.2.1 一个能运行的项目	(283)
9.2.2 spin 控件(微调按钮)	(284)
9.2.3 Progress Control(进度控件)	(287)
9.2.4 Slider Control(滑块控件)	(290)
9.2.5 Hot Key(热键控件)	(293)
9.2.6 Image List Control(图像列表控件)	(296)
9.2.7 List View Control(列表视图控件)	(298)
9.2.8 Tree Control(树形视图控件)	(307)
9.2.9 Rich Edit Control(多样式文本编辑控件)	(311)
9.2.10 Month Calendar Control(月历控件)	(315)
9.2.11 Tab Controls(标签控件)	(316)
9.2.12 运行程序	(319)
9.3 小结	(331)
第十章 打印与打印预览	(332)
10.1 基本打印与打印功能	(332)
10.2 改变映射模式	(335)
10.3 打印多页	(338)
10.3.1 设置矩形的数目	(338)
10.3.2 设置页数	(340)
10.3.3 设置每页的起点	(343)
10.4 MFC 的打印进程	(345)
10.5 小结	(347)
第十一章 改进程序界面	(348)
11.1 控制条类	(348)
11.1.1 控制条	(348)
11.1.2 工具条	(349)
11.1.3 状态条	(349)
11.1.4 对话框条	(350)
11.1.5 集合条	(350)
11.2 工具条和状态条	(350)
11.2.1 缺省的工具条与状态条	(351)

11.2.2 创建自己的工具条.....	(355)
11.2.3 向状态条中添加指示器.....	(365)
11.3 对话框条.....	(370)
11.4 集合条.....	(372)
11.4.1 建立 AdvBar 程序框架	(372)
11.4.2 建立新的工具条.....	(373)
11.4.3 建立集合条.....	(378)
11.5 动态修改菜单.....	(382)
11.5.1 CD 播放控制	(382)
11.5.2 动态修改菜单.....	(386)
11.5.3 使菜单活动起来.....	(387)
11.6 添加快捷菜单.....	(389)
11.6.1 添加快捷菜单.....	(389)
11.6.2 响应 Track 下拉按钮	(390)
11.7 小结.....	(391)
第十二章 ActiveX 技术	(392)
12.1 ActiveX 基础知识	(392)
12.1.1 对象的链接与嵌入(Object Linking and Embedding)	(392)
12.1.2 容器与服务器(Container and Server)	(394)
12.1.3 复合文档(Compound Document)	(395)
12.1.4 Active 文档(Active Document)	(395)
12.1.5 ActiveX 自动化(ActiveX Automation)	(395)
12.1.6 ActiveX 控件(ActiveX Controls)	(396)
12.2 创建 ActiveX 容器应用程序	(396)
12.2.1 建立应用程序框架.....	(396)
12.2.2 框架代码的变化.....	(397)
12.2.3 移动与重设大小.....	(409)
12.2.4 处理多个对象.....	(412)
12.3 创建 ActiveX 服务器应用程序	(418)
12.3.1 建立应用程序框架.....	(418)
12.3.2 框架代码的变化.....	(419)
12.3.3 使 TestMsg1 程序工作	(428)
12.4 ActiveX 控件	(431)
12.4.1 建立 ActiveX 控件框架	(432)
12.4.2 ControlWizard 产生的代码	(433)
12.4.3 添加属性.....	(437)
12.4.4 添加方法.....	(442)
12.4.5 添加事件.....	(444)
12.5 小结.....	(446)

第十三章 数据库编程	(447)
13.1 关系数据库模型.....	(447)
13.1.1 数据结构.....	(447)
13.1.2 完整性规则.....	(448)
13.1.3 数据操作.....	(449)
13.1.4 结构化查询语言 SQL(Structured Query Language)	(449)
13.2 使用 ODBC	(452)
13.2.1 ODBC 概述	(452)
13.2.2 ODBC 的工作原理	(453)
13.2.3 MFC ODBC 类	(454)
13.2.4 创建 ODBC 数据库应用程序	(455)
13.2.5 建立 ODBC 数据源	(456)
13.2.6 用 AppWizard 创建数据库应用程序 School 的框架	(457)
13.2.7 为数据库应用程序 School 创建视图	(459)
13.2.8 应用程序是如何工作的.....	(462)
13.2.9 遍历、添加、修改和删除记录.....	(465)
13.2.10 数据库异常	(467)
13.2.11 记录的筛选和排序	(468)
13.2.12 编程举例	(468)
13.2.13 统计函数的使用	(474)
13.2.14 多表的连接	(44)
13.2.15 统计函数、多表连接使用举例.....	(475)
13.2.16 直接使用 SQL 语句.....	(480)
13.2.17 使用 CDatabase 进行事务处理	(481)
13.3 使用 DAO	(482)
13.3.1 DAO 概述	(482)
13.3.2 MFC DAO 类	(482)
13.3.3 创建 DAO 数据库应用程序	(483)
13.3.4 理解从 CDaoRecordset 派生的记录集类	(486)
13.3.5 DAO 类的进一步介绍和使用	(489)
13.3.6 如何在 ODBC 和 DAO 中选择	(500)
13.4 其他数据库编程技术简介.....	(500)
13.4.1 OLE DB	(500)
13.4.2 ADO	(501)
13.5 小结.....	(501)
第十四章 Internet 编程	(502)
14.1 WinInet (HTTP,FTP,Gopher)类编程简介	(503)
14.1.1 WinInet 的特色	(503)
14.1.2 WinInet 的优点	(503)

14.1.3 MFC 的 WinInet 类	(504)
14.2 创建一个互连网客户应用程序	(504)
14.2.1 用 AppWizard 创建应用程序 MyInternet1	(504)
14.2.2 编辑对话框资源	(505)
14.2.3 加入成员函数和成员变量	(506)
14.3 HTTP 应用	(508)
14.3.1 进行对话框的初始化	(508)
14.3.2 创建一个会话对象	(508)
14.3.3 CInternetFile 类	(510)
14.3.4 打开一个 Internet 文件	(511)
14.3.5 读取并显示文件	(514)
14.3.6 关闭文件	(516)
14.4 FTP 应用	(518)
14.4.1 进行对话框的初始化	(518)
14.4.2 创建一个会话对象	(518)
14.4.3 建立一个 FTP 连接	(518)
14.4.4 获取文件列表	(521)
14.4.5 关闭连接	(524)
14.5 Gopher 应用	(526)
14.5.1 进行对话框的初始化	(526)
14.5.2 创建一个会话对象	(526)
14.5.3 建立一个 Gopher 连接	(527)
14.5.4 使用 Gopher 定位器	(529)
14.5.5 在 Gopher 服务器上查找文件	(532)
14.5.6 使用 CGopherFile 类来操纵文件	(532)
14.5.7 关闭连接	(535)
14.5.8 清单程序	(536)
14.6 小结	(546)
附录 MFC 的常用类	(547)
1 CWinApp 类	(547)
2 CWnd 类	(548)
3 CDocTemplate 类	(554)
4 CView 类	(555)
5 CDocument 类	(556)
6 CFrameWnd 类	(557)
7 CSplitterWnd 类	(558)
8 CDC 类	(559)
9 CGdiObject 类	(564)
10 CControlBar 类	(564)

11	CObject 类	(565)
12	CFile 类	(565)
13	CArchive 类	(566)
14	CObList 类	(567)
15	CString 类	(567)
16	CMenu 类	(568)
17	CDialog 类	(569)
18	CPoint 类	(570)
19	CSize 类	(571)
20	CTime 类	(571)
21	CInternetSession 类	(572)
22	CInternetConnection 类	(572)
23	CFileFind 类	(573)
24	CGopherLocator 类	(574)

第一章 Visual C++ 概述

Visual C++ 6.0 主要是为 Windows 95 和 Windows NT 等 32 位操作系统开发应用程序用的编程工具,具有强大的功能、友好的界面。

本章要点:

- Visual C++ 系列的发展
- Visual C++ 6.0 的不同版本
- Visual C++ 6.0 的新特性
- Windows 编程的基本知识

1.1 Visual C++ 系列的发展

Visual C++ 系列由 Microsoft C/C++ 7.0 发展而来。Microsoft C/C++ 7.0 中包含了 MFC 1.0 类库(Microsoft Foundation Classes)。这一产品的发布,使得 C++ 语言成为最成功的面向对象的程序设计语言,也大大促进了面向对象的软件的开发。

随着 Windows 操作系统的升级,Microsoft 公司发布了 Visual C++ 1.0。这一产品提供了一个 Windows 风格的编辑器和调试器,深受用户欢迎。同时,Visual C++ 1.0 中包含有 MFC 2.0 类库。MFC 2.0 在 MFC1.0 的基础上,增加了动态链接库(DLL)支持、OLE 接口等新功能。

随后,Microsoft 公司发布了其 Visual C++ 系列的 1.5、2.0、4.x、5.0 等版本。其中,Visual C++ 2.0 版首次将资源设计集成到开发工作台(Workbench)中,使界面设计变得容易起来,而且它支持 Windows 3.1、Windows NT 3.5 和 Windows 95 的应用程序开发,十分方便。Visual C++ 4.x 系列功能更为强大,支持多重平台和平台间的开发,也简化了应用程序在多重平台间的移植。Visual C++ 5.0 增加了许多重要功能,如 ActiveX 编程,包括建立 ActiveX 控件、ActiveX 容器和服务器等;如 Internet 编程,包括建立 Internet 插口、Internet 下的 ActiveX 控件开发等等。

现在,Visual C++ 6.0 提供了更强大的功能和更丰富的 MFC 类库,能更好的帮助用户建立所需的应用程序。

1.2 Visual C++ 6.0 的不同版本

Visual C++ 6.0 一共有三个不同的版本:标准版、专业版和企业版。

1.2.1 标准版

标准版适合于利用专业的 Visual C++ 工具集来学习 C++ 语言。标准版与专业版大致相同,但不包括以下特性:静态链接到 MFC 类库、代码优化、源代码配置文件、远程

数据控件、定制 AppWizard、安装工具 InstallShield 等。

1.2.2 专业版

专业版提供了优秀的性能来满足用户开发具有商业水准的应用程序的需要。利用专业版可以开发基于 Win32 平台的应用程序、服务器或控件等。读者可以浏览一下 Visual C++ 的联机文档以获得更多的信息。

1.2.3 企业版

企业版在专业版的基础上提供了许多额外的工具和组件,满足用户开发企业水平的应用程序的需要。利用企业版,可以开发为全球互联网或企业内部网用的客户机/服务器大型应用程序,或是进行 SQL 数据库访问。Visual SourceSafe 源代码控制系统还大大方便了小组的共同开发工作。

1.3 Visual C++ 6.0 的新特性

与以前的版本相比,Visual C++ 6.0 在许多方面都有改进或增强。

1.3.1 MFC 基础类库的改进与增强

- Active 文档容器:Active 文档容器为 Active 文档提供了更好的支持。在 OLE 技术支持下,用户每次只能对复合文档的某一特定的部分进行操作,Active 文档容器可以一次激活整个文档。
- CHtmlView 类:CHtmlView 类提供了在 MFC 应用程序中对动态 HTML 的支持。使用 CHtmlView 可以在应用程序的视中直接显示 HTML 页面。
- IE4.0 公共控件:MFC 通过 CComboBoxEx 类、CDateTimeCtrl 类、CIPAddressCtrl 类和 CMonthCalCtrl 类提供了对最新的几种公共控件的支持。
- 增强或新增的类:MFC 对二十多个类进行了更新与增强,同时提供了十多个新的类,以满足用户的要求。
- 属性页对 Wizard97 风格的支持:Wizard97 是用于 Windows 98 和 Windows NT 5.0 的新的属性页风格,MFC 对此提供了支持。

1.3.2 数据库支持的增强

- ADO 数据绑定对话框向导:该向导引导用户完成利用 ActiveX 数据对象(ADO: ActiveX Data Object)建立数据绑定对话框的全过程,并提供了强大的帮助。
- 支持 ADO 和 OLE DB 的数据控件:这是新增的数据源控件,使得用户可以处理任何由 OLE DB 建立的数据库资源。
- 支持 Oracle 的数据工具:提供了更强大、更方便的数据处理方式。在新的版本中有关 Oracle 的功能得到了增强。

1.3.3 实用工具

- 组件管理器:使得用户可以更方便地存储和共享软件工具、定制组件等。组件管理器的目的就是为代码或资源共享、小组开发、数据资源管理等提供更好的支持。
- HTML 格式的帮助:提供了 HTML Help 1.1 程序供用户开发基于 HTML 规范的帮助文件,MSDN 就是 HTML 格式帮助文件的一个典型的例子。
- 增强的安装工具(Install Shield):提供了专为 Visual C++ 6.0 开发的安装程序生成工具。
- 增强的测试容器(Test Container):新的测试容器更稳定,同时也提供了 OCX96 新特性的支持。

1.3.4 向导

- 增强的 AppWizard:新版本的 AppWizard 为用户开发应用程序提供了更多的选择,如支持非文档/视结构的应用程序、IE4 风格的工具条、Active 文档容器等等。
- 删除成员函数功能:这是以前的任何版本都不具有的,虽然是很小的改进,但很方便。
- ClassWizard 对 IE4 控件的支持:ClassWizard 可以对 IE4 提供的四种新控件进行消息映射或添加成员变量,用户也可以从这四种新控件类中派生自己的类。
- 新的项目类型:定制的 AppWizard 现在支持普通应用程序、控制台应用程序、Active 控件等项目。
- 非 MFC 项目类型的向导:新的版本为非 MFC 项目也提供了向导,方便用户的工作。

1.3.5 编译器、调试器与连接器

新版本中的编译器、调试器和连接器也得到了增强:

- 增强的编译器:包括 _assume 关键字,增强的内联控制,改进并增加了警告,运行中的错误检测等等。
- 增强的调试器:包括改进的反汇编输出、改进的函数指针显示、进程中的远程过程调用、MMX 寄存器显示、线程信息块等等。
- 增强的连接器:包括延迟装载技术、新的连接选项、改进了的库格式等等。

1.3.6 编辑器

- 自动完成语句功能:该功能包括自动显示对象的成员函数、自动显示函数原型、自动完成标识符的剩余部分等等,使用起来很方便。
- 快速宏录制:用户可以记录任何在集成开发环境中的操作,并在适当的时候再次使用。
- 支持 IE4 新控件的资源编辑器:新版本的资源编辑器支持 IE4 提供的四种新控件,在资源编辑器中可以方便地将这四种新控件添加到工具条或对话框中。

Visual C++ 6.0 提供的新特性远不止上面提到的这些,有关的具体信息请用户参阅

Visual C++ 的联机文档中 What's new 部分的内容。

1.4 Windows 下的编程

开发 Windows 应用程序与开发 MS-DOS 程序有很大的不同。虽然我们假定读者已经对 Windows 操作系统本身和 Windows 应用程序比较熟悉,但为了开发自己的 Windows 应用程序,读者应该对 Windows 操作系统了解得更多一些。

1.4.1 Windows 的消息机制

当用 C 语言来编写一个 MS-DOS 程序的时候,肯定会有一个 `main()` 函数,`main()` 函数是程序的入口和出口。在 `main()` 函数中,可以调用适当的函数或使用适当的程序结构(典型的如顺序、分支、循环结构)来完成所需的功能。如果程序需要获得用户的键盘或鼠标输入,就需要调用诸如 `getchar()` 之类的函数,通过操作系统才能得到用户输入。程序员需要精心地设计和处理何时去获取用户输入、得到用户的输入后如何处理所获得的信息。

Windows 的应用程序中,也有一个类似于 `main()` 的函数,这就是 `WinMain()` 函数。同样地,`WinMain()` 函数也是程序的入口和出口。在 `WinMain()` 函数中,要完成以下工作:

- (1)注册应用程序的窗口类;
- (2)创建和实现窗口,进行必要的初始化工作;
- (3)创建和启动应用程序的消息循环;
- (4)如果接收到 `WM_QUIT` 消息,则退出应用程序。

Windows 操作系统接受用户的输入,并用消息的方式通知应用程序。Windows 应用程序接受并处理由操作系统发送来的消息。Windows 消息本身是一个整数值,但被定义成容易理解的常量,如上述 `WM_QUIT` 消息。比如说,在窗口被创建的时候,会发送 `WM_CREATE` 消息;在鼠标左键被按下的时候,会发送 `WM_LBUTTONDOWN` 消息;在用户敲击键盘的时候,会发送 `WM_CHAR` 消息;在窗口被关闭的时候,会发送 `WM_CLOSE` 消息;在用户选择菜单的时候,也会发送消息。当然,也可以在程序中定义自己的消息,即所谓“用户消息”。

利用 MFC 编程很大程度上就是处理各种各样的消息,协调它们工作,并完成所需要的功能。幸运的是,应用程序框架已经为我们完成了不少的工作,这使得我们的开发工作变得容易了许多,但理解 Windows 的消息机制对我们彻底地理解程序是很有帮助的。

1.4.2 Windows 的图形设备接口

为 MS-DOS 程序编制显示接口或打印接口简直就是一场恶梦,每一种显示卡或打印机都有其独特的驱动程序,让某个程序支持所有的显示卡或者打印机是不可能的。在 Windows 中,提供了一个抽象的接口,即图像设备接口(GDI:Graphics Device Interface)。Windows 操作系统本身提供了对不同种类的显示卡或打印机的支持,即提供了图形系统的设备无关性。应用程序通过调用 GDI 函数来与硬件打交道,而 Windows 会自动将设备