

万水计算机技术实用大全系列

MFC

窗口程序设计

姚领田 高守传 等编著

涵盖**MFC窗口**基类CWnd、框架窗口、文档-视图窗口、分割窗口、控制栏窗口、对话框窗口、属性表窗口、控件窗口等，并对程序设计中的调试技术进行简要介绍，以帮助读者对程序进行调试。

工程
实践



中国水利水电出版社
www.waterpub.com.cn

TP312 VC++
419
1:

万水计算机技术实用大全系列

MFC 窗口程序设计

姚领田 高守传 等编著

中国水利水电出版社

内 容 提 要

窗口是 Windows GUI 程序的基础, 在 Windows 程序设计中, 窗口编程具有举足轻重的地位。本书深入浅出, 全面地介绍了 MFC 中窗口程序的设计方法, 内容主要包括 MFC 窗口基类 CWnd、框架窗口、文档一视图窗口、分割窗口、控制栏窗口、对话框、属性表窗口、控件窗口等, 并对程序设计中的调试技术进行了简要介绍, 以帮助读者对程序进行调试。

本书内容丰富详实、讲解细致, 对涉及 MFC 实现理论的部分均进行了较为深入的剖析, 并且结合工程实践给出了相应的实现代码, 具有较强的针对性, 力求让读者通过亲自动手掌握使用 MFC 进行窗口程序设计的方法, 学习尽可能多的知识。同时, 对于开发中可能会遇到的问题也都给出了相应的提示或说明。

本书主要定位于初、中级读者, 同时也可作为培训班以及高校师生的参考教材。

本书附带的程序源代码可从中国水利水电出版社网站免费下载, 网址为:
<http://www.waterpub.com.cn/softdown/>。

图书在版编目(CIP)数据

MFC 窗口程序设计 / 姚领田等编著. —北京: 中国水利水电出版社, 2007

(万水计算机技术实用大全系列)

ISBN 978-7-5084-4606-6

I. M… II. 姚… III. ①C 语言—程序设计②窗口软件, Windows—程序设计 IV. TP312 TP316.7

中国版本图书馆 CIP 数据核字 (2007) 第 067989 号

书 名	MFC 窗口程序设计
作 者	姚领田 高守传 等编著
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787mm×1092mm 16 开本 29.75 印张 720 千字
版 次	2007 年 5 月第 1 版 2007 年 5 月第 1 次印刷
印 数	0001—4000 册
定 价	58.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换
版权所有·侵权必究

前 言

Visual C++是 Microsoft 公司推出的一种高度综合性能的开发环境，它是一套面向对象的可视化集成编程系统。它以其强大的功能和非常友好的人机界面成为目前最受欢迎的开发工具之一，使用它开发的程序除了有运行速度快、可移植能力强的特点外，还有一个最重要的原因——它拥有一个庞大且功能齐全的 MFC 库，并且提供了一套相应的编程机制。这些都是它的优点，但同时也正是这些优点使得对它的掌握变得十分困难。因此，在学习 Visual C++和 MFC 的过程中，始终伴随着矛盾的取舍。

不过，由于 Visual C++使用的广泛性和 MFC 学习过程的复杂性，使得很多读者在开始学习时容易陷于迷茫而不知从何处着手。笔者根据知识特点和学习内容，结合窗口在 Windows 系统中的地位、窗口编程在 MFC 中的重要性、学习 MFC 的心得体会以及多年从事工程开发的经验，精心编写了此书，以帮助准备学习使用 MFC 进行开发、正在学习 MFC 进行工程开发以及正在使用 MFC 进行工程开发的读者少走弯路，提高学习和使用的效率。

全书按照 MFC 中窗口类在应用框架中的地位和读者的学习规律循序渐进、由浅入深地进行讲述。对于书中每个知识点的讲解，都力求使读者不但“知其然”，而且“知其所以然”。只有这样，才能真正让读者领会掌握知识的精髓，从而提高工程实践的能力。最后，对于书中的大部分内容都结合工程实践给出了详细的实现代码，并且对于实践中需要注意的事项也都给出了相应的说明或提示。

接下来，笔者结合自己的学习经历，给读者提供一些有益的建议。

第一，要树立必胜的信念，不要在开始学习之前就强加给自己一个“VC 或 MFC 非常难学”的观念。做任何事情都一样，在开始的时候都要充满信心，否则是走不远的。

第二，要有持之以恒的精神。没有必胜的信念，也就不会有这种精神。你要相信自己的能力，在学习的道路上，你肯定曾经克服过不少的困难，这一次和那些困难类似，那些困难你都能克服，现在的困难你更能克服。毕竟，你已经积累了不少的经验，这些经验有助于你克服现在的困难，这是一个人成长的标志。

第三，良好的基础知识。C++基本知识是必不可少的，尤其是要深刻理解面向对象的概念，它具有封装、继承、多态三大特性，是学习 MFC 的基础。

第四，对 Windows 消息机制有一定的理解。消息机制是 Windows 操作系统的核心，因此，在学习 MFC 之前，如果具有 Win32 SDK 的编程经验是最好的。即使没有也不用担心，可以进行适当的学习，不需要太深入地了解，在熟悉 MFC 的编程思路以后，还可以回过头来弥补这一欠缺。

第五，好的学习教材和学习方法。好的学习教材可以使复杂的问题简单化，使人循序渐进地掌握知识。而不好的教材却相反，甚至误人子弟。此外，虽然曲径可以通幽，但若

有通幽之捷径，则还是选择后者。MFC 包含内容甚广，不可能完全从几本书中找到问题的答案，只要有几本好的教材足矣，在掌握了 MFC 的整体框架以后，可以慢慢地进行具体深入的学习。学习时针对性要强，不能全面铺展开来，那样容易陷入烂泥潭，或者被 MFC 的庞大繁杂而吓怕。

第六，多泡论坛，多研究别人的代码。论坛是初学者最好的“教室”，只要你热爱学习或者有“打破沙锅问到底”的精神，那里有无数热心的“老师”乐意帮助你，而且还有很多优秀的代码供你享用。此外，研究别人的代码是另一个好的习惯，在研究的基础上进行修改以使其满足自己的需要，“他山之石，可以攻玉”。

第七，学会跟踪调试技术。跟踪调试的重要性非言语所能描述。在我们面前的是一个庞大的 MFC 库，它最重要的特点就是对 Win32 API 的封装。从概念上理解，封装就是使用包裹将另外一种东西包装起来，以使被包装的东西不可见或部分可见。但是，可以使用一种工具，通过这种工具，我们可以进入包裹内了解被包装的东西。以 MFC 库为例，它就是用来包装的包裹，而 Win32 API 就是被包装的函数。因此，在开发过程中，Win32 API 对于开发人员来说就是不可见的，而跟踪调试技术就是可以让我们进入包裹从而了解包裹内容的工具，它的重要性可见一斑。MFC 的抽象性是使读者望而却步的一个重要因素，因此，跟踪调试是我们学习 MFC 的一项必备技术。

第八，必要的英文水平。在学习 MFC 时，有一个最好的教材就是 MSDN，但它是英文版的。如果 MSDN 是中文的，MFC 的学习者不但会更多，而且要比现在容易得多。此外，还有很多优秀的论坛也是英文的。因此，若想真正地学好 MFC，一定的英文水平是必需具有的。不过，MSDN 中的文章都比较简单，如果能够坚持看上两个月，你就会离不开英文的 MSDN 了，我当初就是这样。

本书主要由姚领田、高守传等编著，其中姚领田编写了第 1、10 章和附录，高守传编写了第 2、5 章，古丽编写了第 3、8 章，徐锦拓编写了第 4、9 章，刘挺编写了第 6 章，古磊编写了第 7 章，参与编写的其他人员还有李洪峻、陶志杰、姚晓玉等。

由于作者水平所限，书中难免有错误和不足之处，敬请广大读者批评指正。

作者

2007 年 1 月

目 录

前言

第 1 章 Windows 窗口与 CWnd 对象	1
1.1 窗口和 CWnd 对象	1
1.2 窗口与 CWnd 对象的映射关系	1
1.2.1 句柄的概念	1
1.2.2 句柄映射机制	2
1.2.3 关联窗口对象与 CWnd 类对象	6
1.2.4 从窗口对象获取 CWnd 类对象	9
1.2.5 解除窗口对象与 CWnd 类对象的关联	11
1.2.6 临时映射对象的清除	13
1.3 窗口的创建	15
1.3.1 传统的 Windows 程序设计中的窗口类注册	15
1.3.2 MFC 应用的窗口类注册	16
1.3.3 SDI 中的窗口创建顺序	21
1.4 窗口的销毁	22
1.4.1 Windows 和应用的交互	22
1.4.2 使用 CWnd::PostNcDestroy 自动销毁	22
1.4.3 销毁窗口对象总结	23
1.5 窗口的编程内容	24
1.5.1 窗口消息传递过程	24
1.5.2 消息映射	26
1.5.3 绘制窗口	26
1.5.4 窗口的设备环境	27
1.6 绘制窗口类 CDrawManager	29
1.6.1 CDrawManager 成员简介	29
1.6.2 绘制原理	30
1.7 小结	34
第 2 章 CWnd 类综述	35
2.1 CWnd 类基础	35
2.2 CWnd 类成员函数简介	35
2.2.1 CWnd 类成员函数分类	35
2.2.2 构造析构成员函数	36

2.2.3	初始化成员函数	37
2.2.4	对话框项成员函数	39
2.2.5	处理初始化消息的成员函数.....	41
2.2.6	窗口状态成员函数	42
2.2.7	处理系统消息的成员函数.....	43
2.2.8	窗口大小和位置有关的成员函数.....	45
2.2.9	菜单成员函数	48
2.2.10	处理一般消息的成员函数.....	48
2.2.11	访问窗口的成员函数.....	55
2.2.12	工具提示成员函数	59
2.2.13	处理控件消息的成员函数.....	60
2.2.14	处理更新/绘制的成员函数	62
2.2.15	处理输入消息的成员函数.....	66
2.2.16	处理坐标映射的成员函数.....	67
2.2.17	处理非客户区消息的成员函数.....	68
2.2.18	处理窗口文本的成员函数.....	70
2.2.19	处理窗口消息的成员函数.....	71
2.2.20	处理滚动的成员函数	72
2.2.21	剪贴板成员函数	73
2.2.22	剪贴板消息处理函数	74
2.2.23	可重载的成员函数	75
2.2.24	脱字符成员函数	77
2.2.25	其他	78
2.3	CWnd 派生类.....	80
2.3.1	框架窗口类	80
2.3.2	视图类	81
2.3.3	对话框类	82
2.3.4	控件类	83
2.3.5	控制条类	85
2.3.6	属性表类	85
2.4	编程实例	85
2.4.1	创建工程	85
2.4.2	添加代码	86
2.5	小结	88
第 3 章	框架窗口	89
3.1	CFrameWnd 成员一览.....	89
3.1.1	CFrameWnd 类数据成员	89

3.1.2	CFrameWnd 类构造函数.....	89
3.1.3	CFrameWnd 类初始化函数.....	90
3.1.4	CFrameWnd 类操作函数.....	91
3.1.5	CFrameWnd 类可重载函数.....	93
3.1.6	CFrameWnd 类命令处理函数.....	94
3.2	SDI 框架窗口的创建过程.....	94
3.2.1	框架窗口的创建.....	94
3.2.2	使用 LoadFrame 创建框架窗口.....	95
3.2.3	使用 AfxEndDeferRegisterClass 注册窗口类.....	96
3.2.4	使用 GetIconWndClass 获取窗口类.....	97
3.2.5	使用 PreCreateWindow 注册窗口类.....	98
3.2.6	使用 CFrameWnd::Create 创建框架窗口.....	99
3.2.7	框架窗口子窗口的初始化.....	101
3.3	SDI 框架窗口的销毁.....	104
3.3.1	响应 WM_CLOSE 消息.....	104
3.3.2	销毁 Windows 窗口对象.....	106
3.3.3	销毁 C++对象.....	107
3.4	SDI 框架窗口对命令消息及控件通知的处理.....	108
3.4.1	命令消息.....	108
3.4.2	控件通知.....	108
3.4.3	框架对 WM_COMMAND 的处理.....	109
3.4.4	框架对 WM_NOTIFY 的处理.....	114
3.5	MDI 框架窗口的创建及销毁.....	114
3.5.1	SDI 与 MDI 的主要区别.....	115
3.5.2	主 MDI 框架窗口的创建.....	115
3.5.3	MDI 客户窗口及其创建.....	116
3.5.4	MDI 子框架窗口的创建.....	118
3.5.5	框架窗口与 MDI 客户窗口之间的关系.....	120
3.5.6	MDI 框架窗口的销毁.....	121
3.6	MDI 框架窗口对命令消息的处理.....	121
3.6.1	主 MDI 框架窗口的命令处理.....	121
3.6.2	主 MDI 框架窗口的命令分发.....	122
3.6.3	MDI 框架窗口的窗口过程.....	123
3.7	框架窗口的更新机制.....	124
3.7.1	命令 UI 类.....	124
3.7.2	用于更新操作的成员 CCmdUI::DoUpdate.....	125
3.8	框架窗口的客户区管理机制.....	126

3.9 小结	127
第4章 文档—视图窗口	129
4.1 文档—视图模型	129
4.1.1 文档—视图模型简介	129
4.1.2 文档—视图类	130
4.1.3 CView 的派生类	130
4.1.4 创建工程	131
4.2 文档—视图重要成员	131
4.2.1 文档成员	131
4.2.2 视图成员	133
4.3 对象的创建	135
4.3.1 创建模板对象	135
4.3.2 创建文档对象	136
4.3.3 创建框架对象	139
4.3.4 创建视图对象	140
4.3.5 各对象间的关系	143
4.4 文档—视图的初始化和清除	145
4.4.1 初始化和清除的时机	145
4.4.2 初始化和清除的过程	146
4.5 命令消息在视图窗口中的分发过程	146
4.5.1 视图窗口对命令消息的处理	146
4.5.2 文档类对命令消息的处理	147
4.5.3 命令消息的处理函数的调用过程	148
4.6 视图窗口的使用	150
4.6.1 在视图中绘制	150
4.6.2 通过视图解释用户输入	151
4.6.3 多视图窗口	151
4.7 定制控件视图	152
4.7.1 指导思想	152
4.7.2 具体实现	152
4.8 编程实例	153
4.8.1 资源编辑	154
4.8.2 为工程添加新类	154
4.8.3 主框架实现	155
4.8.4 列表视图实现	158
4.8.5 表单视图实现	159
4.8.6 地址栏实现	161

4.9	小结	163
第 5 章	分割窗口	164
5.1	窗口分割	164
5.2	CSplitterWnd 类	165
5.2.1	CSplitterWnd 类成员一览	165
5.2.2	创建动态分割窗口	167
5.2.3	创建静态分割窗口	170
5.2.4	创建分割窗口的窗格	171
5.2.5	设置窗格信息	173
5.2.6	绘制分割窗口的组件	174
5.2.7	绘制拖动分割条	176
5.3	分割窗口中的通信机制	177
5.3.1	有文档相连的视图之间的通信	178
5.3.2	无文档视图与文档关联视图之间的通信	178
5.3.3	无文档关联视图之间的通信	178
5.3.4	任意视图之间的通信	179
5.4	分割窗口中的命令路由	179
5.4.1	通过文档对象路由视图窗口	179
5.4.2	通过分割窗口路由视图窗口	180
5.4.3	通用路由方法	181
5.5	编程实例	181
5.5.1	工程的创建	181
5.5.2	分割窗口的创建	182
5.5.3	自定义分割窗口类 CSplitterWndEx	184
5.5.4	位图信息窗格	189
5.5.5	原始位图窗格	196
5.5.6	处理后位图窗格	197
5.6	位图类	199
5.6.1	位图	199
5.6.2	位图类	199
5.7	小结	202
第 6 章	控制栏窗口	203
6.1	控制栏综述	203
6.1.1	控制栏简介	203
6.1.2	CControlBar 类成员	203
6.1.3	控制栏布局算法	206
6.1.4	控制栏对消息的处理	209

6.1.5	布局算法应用实例	214
6.2	对话框窗口	215
6.2.1	工程的创建	215
6.2.2	对话框窗口的创建	216
6.2.3	对话框窗口的 UI 对象更新	219
6.2.4	接收来自主菜单或工具栏的命令消息	219
6.2.5	对话框实例分析	220
6.3	伸缩栏窗口	222
6.3.1	类 CReBar 和 CReBarCtrl	223
6.3.2	CReBar 成员介绍	223
6.3.3	在 Rebar 控件中使用对话框	224
6.3.4	伸缩栏的 UI 对象更新	225
6.3.5	伸缩栏实例分析	225
6.4	工具栏窗口	226
6.4.1	工具栏窗口简介	226
6.4.2	工具栏成员一览	227
6.4.3	工具栏窗口的创建	227
6.4.4	在工具栏中使用图像列表	230
6.4.5	在工具栏中使用下拉按钮	230
6.4.6	工具栏窗口的 UI 对象更新	232
6.4.7	在工具栏上显示文本	233
6.4.8	在工具栏上创建控件	235
6.4.9	工具栏热点按钮的切换	239
6.5	状态栏窗口	239
6.5.1	状态栏窗口简介	240
6.5.2	状态栏成员一览	240
6.5.3	状态栏窗口的创建	241
6.5.4	为状态栏添加窗格	242
6.5.5	状态栏窗口的 UI 对象更新	243
6.5.6	在状态栏上创建控件	244
6.5.7	状态栏中的消息响应	246
6.5.8	改变状态栏的大小	247
6.6	主框架对控制栏的控制	248
6.6.1	工具箱的显隐	248
6.6.2	工具箱的布局	248
6.6.3	工具箱和对话框的通信	249
6.7	小结	250

第 7 章 对话框	251
7.1 对话框及 MFC 相关类	251
7.1.1 对话框与 CDialog 类	251
7.1.2 工程的创建	252
7.2 模态对话框	254
7.2.1 添加对话框及其类	254
7.2.2 创建模态对话框	256
7.2.3 模态对话框消息循环	264
7.2.4 模态的终结	268
7.3 非模态对话框	270
7.3.1 添加对话框及其类	270
7.3.2 创建非模态对话框	271
7.3.3 模态对话框与非模态对话框的区别	273
7.4 对话框数据交换和验证	273
7.4.1 CDataExchange 类简介	273
7.4.2 数据交换和验证	274
7.5 对话框中更新机制的实现	277
7.5.1 CWnd::UpdateDialogControls 成员	278
7.5.2 使用 WM_KICKIDLE 钩子实现 UI 更新	280
7.5.3 在模态检测 ContinueModal 中实现 UI 更新	281
7.5.4 按钮和编辑控件更新的实现	281
7.6 对话框背景及背景透明的控件	282
7.6.1 设置对话框的背景颜色	283
7.6.2 设置对话框背景位图	283
7.6.3 绘制对话框背景	285
7.6.4 对话框控件透明背景的实现	287
7.6.5 MFC 处理对话框及其控件颜色和背景的实现	288
7.7 为对话框添加工具栏	289
7.7.1 创建工具栏	290
7.7.2 为工具栏添加提示信息	290
7.7.3 控制工具栏的显隐状态	292
7.8 为对话框添加状态栏	292
7.8.1 创建状态栏	293
7.8.2 设置状态栏窗格信息	293
7.8.3 控制状态栏的显隐状态	296
7.8.4 状态栏中系统时间的更新方法	296
7.9 为对话框添加菜单	297

7.10	对话框中特定键的处理	298
7.11	对话框扩展与收缩	300
7.12	控件在对话框中的应用	302
7.12.1	“关于”对话框的滚动字幕控件	302
7.12.2	显示 Gif 图片的控件	302
7.13	其他实现	303
7.14	小结	303
第 8 章	属性表窗口	304
8.1	属性表相关类	304
8.2	属性表类	304
8.2.1	属性表类简介	304
8.2.2	属性表类成员	305
8.3	属性页类	309
8.3.1	属性页类简介	309
8.3.2	重要成员函数介绍	310
8.4	创建属性表	312
8.4.1	创建模态属性表	313
8.4.2	创建非模态属性表	313
8.5	处理“应用”按钮	314
8.5.1	属性表处理	314
8.5.2	属性页处理	315
8.6	在对话框中使用属性表	316
8.6.1	工程的创建	316
8.6.2	工程资源的添加	318
8.6.3	属性表类实现	319
8.6.4	“一般信息”页编程	319
8.6.5	“类信息”页编程	325
8.6.6	主对话框窗口编程	326
8.7	树形属性表的实现	332
8.7.1	工程的创建	333
8.7.2	CPropPageFrame 类	334
8.7.3	CTreePropSheet 类	334
8.7.4	属性表“应用”按钮的编程	335
8.7.5	属性页编程	336
8.7.6	属性表接口编程	336
8.8	扩展属性表类——CTreePropSheet	338
8.8.1	属性表的初始化	338

8.8.2	树形控件中属性页项的创建和填充.....	342
8.8.3	结束当前属性页	345
8.8.4	响应树形控件中属性页项的变化.....	346
8.8.5	更新属性页的标题	347
8.9	小结	349
第9章	控件窗口	350
9.1	控件窗口	350
9.1.1	控件窗口的创建	352
9.1.2	派生公共控件窗口	353
9.2	动画控件	354
9.2.1	CAnimateCtrl 类.....	354
9.2.2	动画控件的使用	355
9.2.3	动画控件的局限	357
9.3	按钮控件	358
9.3.1	按钮控件的风格类型	358
9.3.2	CButton 类.....	358
9.3.3	CBitmapButton 类.....	360
9.3.4	按钮控件的通知消息	360
9.4	组合框/扩展组合框控件	361
9.4.1	组合框/扩展组合框的风格类型	361
9.4.2	CcomboBox 类	362
9.4.3	CComboBoxEx 类.....	366
9.4.4	扩展组合框与图像列表	367
9.5	时间提取器/月历控件	368
9.5.1	时间提取器/月历控件的风格类型	368
9.5.2	CDateTimeCtrl 类.....	369
9.5.3	CMonthCalCtrl 类	371
9.6	图像列表控件	373
9.6.1	CImageList 类.....	373
9.6.2	图像列表的类型	374
9.6.3	图像列表的编程	374
9.7	编辑控件/静态控件	375
9.7.1	编辑控件/静态控件的风格类型	376
9.7.2	CEdit 类.....	377
9.7.3	CStatic 类.....	379
9.8	Rich Edit 控件	381
9.8.1	Rich Edit 控件基础	381

9.8.2	CRichEditCtrl 类	381
9.8.3	Rich Edit 控件实例	384
9.9	列表框/进度条控件	388
9.9.1	列表框/进度条控件的风格类型	389
9.9.2	CListBox 类	390
9.9.3	CProgressCtrl 类	392
9.9.4	编程实例	393
9.10	滑块/调节钮控件	394
9.10.1	滑块/调节钮控件的风格类型	394
9.10.2	CSliderCtrl 类	395
9.10.3	CSpinButtonCtrl 类	397
9.11	滚动条控件	399
9.11.1	滚动条控件的风格类型	399
9.11.2	CScrollBar 类	400
9.12	列表/表头控件	401
9.12.1	列表/表头控件的风格类型	402
9.12.2	CListCtrl 类	404
9.12.3	CHeaderCtrl 类	412
9.13	树形控件	416
9.13.1	树形控件的风格类型	416
9.13.2	CTreeCtrl 类	416
9.14	标签/IP 控件	423
9.14.1	标签控件的风格类型	423
9.14.2	CTabCtrl 类	424
9.14.3	CIPAddressCtrl 类	428
9.15	小结	429
第 10 章	程序调试	430
10.1	基础知识	430
10.1.1	基本概念	432
10.1.2	Debug 版本调试环境配置	432
10.1.3	Release 版本调试环境配置	433
10.2	设置/取消断点	433
10.2.1	位置断点的设置	434
10.2.2	数据断点的设置	436
10.2.3	消息断点的设置	439
10.2.4	断点的取消	439
10.3	跟踪程序执行	440

10.4 状态查看工具	440
10.4.1 变量窗口	441
10.4.2 查看窗口	441
10.4.3 寄存器窗口	442
10.4.4 内存窗口	442
10.4.5 调用堆栈	442
10.4.6 反汇编窗口	443
10.4.7 调试信息泡泡窗口	444
10.4.8 快速查看变量窗口	444
10.5 调试宏与函数	445
10.5.1 断言宏	445
10.5.2 跟踪宏	446
10.5.3 打印对象	447
10.6 小结	448
附录 SDI 应用的消息循环	449
A.1 主循环过程	449
A.2 空闲处理	451
A.3 消息的处理	453
A.3.1 消息的获取与分发	453
A.3.2 消息的预处理	454
A.3.3 空闲处理状态的维护	455
A.4 结束主循环	456
参考文献	458

第 1 章 Windows 窗口与 CWnd 对象

本章从介绍 Windows 窗口和 CWnd 对象的关系入手，详细介绍二者的映射关系、窗口的创建及销毁过程，此外，还对与窗口编程相关的内容，如消息处理、绘制窗口以及设备环境进行了讨论，最后给出了一个用于绘制窗口的 C++ 类，并对其绘制原理进行了剖析，同时也给出了该类的使用方法。

1.1 窗口和 CWnd 对象

在 C 语言时代，当人们使用纯粹的 C 开发 Windows 应用程序时，只需要通过窗口句柄 HWND 即可完成有关窗口的各项操作。这里 HWND 是 WND 的句柄，而 WND 是 Windows 的一个内部私有数据结构，它存储着操作系统中窗口的相关信息，如尺寸信息、标题信息等。一般来说，开发人员无法访问这个数据结构，不过，可以以 HWND 为参数调用诸如 GetWindowRect 和 GetWindowText 的 Windows API 函数。从开发者的角度来看，HWND 只是 Windows 中窗口的惟一标识，而实际上，它是 WND 结构的内存句柄。

Windows 窗口与 CWnd 对象不同，但是两者有紧密联系。CWnd 对象是由 CWnd 的构造函数创建，由析构函数负责销毁，而 Windows 的窗口是由 CWnd 的 Create 成员函数创建，由 CWnd 的虚拟析构函数负责销毁。

CWnd 类和消息映射机制隐藏了 WndProc 函数。窗口接收到的 Windows 通知消息通过消息映射，被自动发送到适当的 CWnd 类的 OnMessage 成员函数。通常在派生类中重载 OnMessage 成员函数，以处理成员的特定消息。

1.2 窗口与 CWnd 对象的映射关系

在 C++ 中，类是表示模型的重要概念，同样，它也使用 C++ 类来表示窗口。在 MFC 中，该类即为 CWnd。此时，在程序设计中不再以 HWND 为主要处理对象，而是以 CWnd 及其各个派生类为主要处理对象。CWnd 类有一个 HWND 类型的数据成员 m_hWnd，MFC 使用该成员将 MFC 调用转换为 C 语言的 API 调用。大部分 Win32 API 函数在 MFC 中都有相对应的内联函数，这些函数的实现方式都很类似，即以 CWnd 类的成员 m_hWnd 为第一个参数，调用相对应的 Win32 API 函数。图 1-1 显示了 CWnd 类对象和 Windows 窗口对象之间的关系。

本节主要介绍实现上述机制的原理。

1.2.1 句柄的概念

句柄是 Windows 用来标识被应用程序所建立或使用的对象的惟一整数，而这些对象都占