

软件工程师参考手册

# Visual C++ 6.0 MFC 类库参考手册

◎ 本书编写组 编著



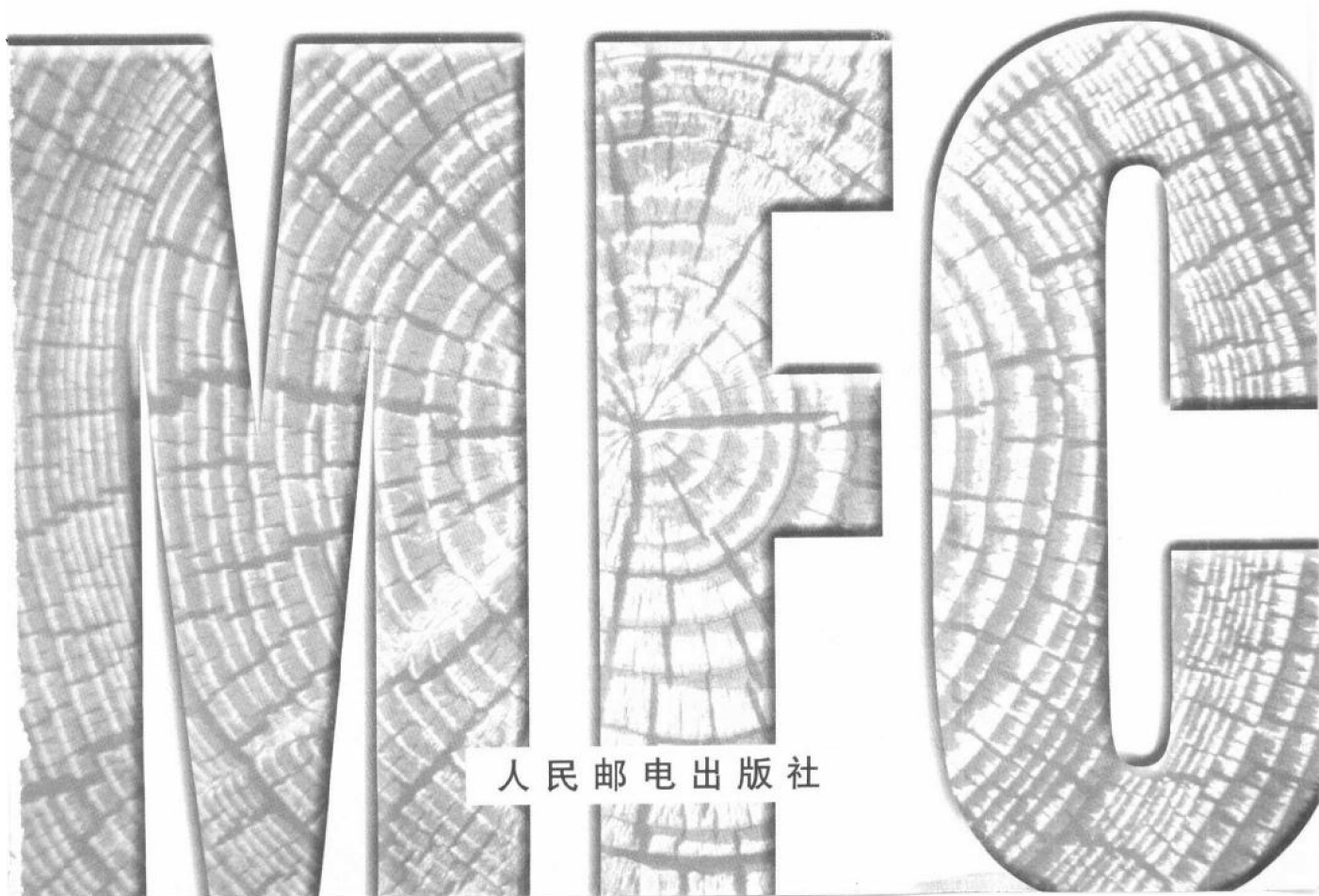
附光盘  
CD-ROM

人民邮电出版社  
POSTS & TELECOMMUNICATIONS PRESS

软件工程师参考手册

# Visual C++ 6.0 MFC 类库参考手册

◎ 本书编写组 编著



人民邮电出版社

## 图书在版编目 (CIP) 数据

Visual C++6.0 MFC 类库参考手册 / 《Visual C++6.0 MFC 类库参考手册》编写组编。

—北京：人民邮电出版社，2002.11

ISBN 7-115-10641-X

I. V... II. V... III. C 语言—程序设计—技术手册 IV. TH122

中国版本图书馆 CIP 数据核字 (2002) 第 075324 号

## 内容简介

本书是关于 Microsoft Visual C++ 基础类库 (MFC) 的完整的参考手册。MFC 库按照功能应用划分为基类、应用程序结构类、文档相关类、框架相关类、控制条相关类、属性单相关类、对话框类、视图相关类、控件类、异常类、文件类及文件管理类、绘制图像相关类、数据库类、OLE 相关类、简单数据类、基本数据结构类、套接字和网络相关类、同步化类、网络服务 API 类、支持类以及其他类。其中的每一分类都有多个相关的 MFC 类。

本书详细介绍了每一个 MFC 类的成员，包括成员函数和数据成员。对成员函数着重介绍了其原型表示、参数说明、返回值说明和功能简介，并且对其中重要的成员函数给出了应用实例。在每一个分类的最后部分，结合前面介绍的 MFC 类，给出了一个或多个综合应用实例，每个实例都力求做到短小精练，把本章介绍的类尽量包含进去，界面友好，可读性强，让读者能从实际的编程操作中体会各种 MFC 类的用途。

本书是从事 Microsoft Visual C++ 基础类库 (MFC) 开发和应用人员必备参考书，也可作为大专院校相关专业师生自学、教学参考用书。

软件工程师参考手册

### Visual C++ 6.0 MFC 类库参考手册

- ◆ 编 著 本书编写组  
责任编辑 张立科
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子函件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
读者热线 010-67132692  
北京汉魂图文设计有限公司制作  
北京鸿佳印刷厂印刷  
新华书店总店北京发行所经销
- ◆ 开本：787×1092 1/16  
印张：55.25 插页：1  
字数：1723 千字 2002 年 11 月第 1 版  
印数：1-4 000 册 2002 年 11 月北京第 1 次印刷

ISBN7-115-10641-X/TP • 3095

定价：88.00 元 (附光盘)

本书如有印装质量问题，请与本社联系 电话：(010) 67129223

# 前言

MFC 是指 Microsoft Foundation Class (Microsoft 基础类), 它封装了 Windows API (应用程序接口) 的数据结构、函数和宏, 以面向对象的类的形式供程序开发者编写各种类型的程序, 包括 Internet、数据库和多媒体方面的程序。MFC 6.0 版本封装了大约 200 个类, 其中的一些可以直接使用, 而另一些则主要作为用户自己的类的基础类。MFC 也是一个应用程序的框架结构。MFC 不仅仅是一个类的集合, 它还帮助定义了应用程序的结构并为应用程序处理许多杂务。

本书按功能和类别对 MFC 进行了归纳和总结, 具体分析和介绍它们的功能、用法、所在头文件、成员函数的返回值以及参数调用等信息。本书内容如下:

第 1 章介绍了 MFC 的大致特点、分层结构、通用类的设计规则等, 并提供了第一个简单的 MFC 应用程序

第 2 章介绍了根类、消息映射类和窗口类, 强调了窗口和消息的关系。

第 3 章介绍了应用程序结构类, 包括文档模板类、单文档和多文档模板类、OLE 类等。

第 4 章介绍了文档相关类, 包括基本文档类、活动文档类、文档对象服务等。

第 5 章介绍了框架相关类, 包括框架窗口类、编辑窗口类等。

第 6 章介绍了控制条相关类, 包括对话框条相关类、工具条相关类等。

第 7 章介绍了属性单相关类, 包括属性单类和属性扩展类。

第 8 章介绍了对话框类, 包括通用对话框类、颜色对话框类和文件对话框类等。

第 9 章介绍了视图相关类, 包括控制视图类、编辑视图类、滚动条视图类等。

第 10 章介绍了控件类, 包括按钮、下拉框、列表框、滚动条等可由 Visual C++ 开发环境提供的各种控件。

第 11 章介绍了异常类, 包括数据库异常类、文件异常类、资源异常类等

第 12 章介绍了文件类以及文件管理类, 包括共享文件类、流式文件类、网络文件类等。

第 13 章介绍了绘制图象相关类, 包括图象表类、位图类、画刷类、字体类等。

第 14 章介绍了数据库类, 包括数据库连接类、数据库的记录类等。

第 15 章介绍了 OLE 相关类, 包括字体封装类、OLE 自动化执行类等。

第 16 章介绍了数据库类, 包括数据库连接类、数据库的记录类等。

第 17 章介绍了基本数据结构类, 包括数组类、列表类、指针数组类、链表类等。

第 18 章介绍了套接字和网络相关类, 包括网络通信类、文件查找类、网络连接描述类等。

第 19 章介绍了同步化类, 包括临界区类、事件类、互斥类等。

第 20 章介绍了网络服务 API 类, 包括 HTTP 服务器数据处理类、功能扩展类等。

第 21 章介绍了支持类, 包括菜单显示管理类、数据交换类、在线等待的光标类等。

第 22 章介绍了其他类, 包括数组模板类、单对象资源访问控制类等。

本书为迫切需要掌握 MFC 编程技术的人提供了一条有效的捷径, 对于不同层次的 MFC 编程人员来说, 本书极具参考价值, 是一本不可多得的参考书。

参与本书的主要编写人员有付俊明、严东明、张双民等。由于 MFC 所涉及的知识面极为广泛, 而作者的知识又有限, 所以尽管我们对本书中所涉及的内容一再推敲和仔细调试, 仍有可能出现错误和纰漏, 希望广大读者批评指正。

编者

2002 年 10 月

# 目 录

第1章 MFC 基础.....	1
1.1 MFC 简介.....	1
1.2 使用 C++和 MFC 的好处.....	1
1.3 MFC 类的分层结构.....	2
1.4 通用类设计规则.....	2
1.5 综合实例——第一个 MFC 应用程序.....	3
第2章 基类.....	5
2.1 根类 CObject.....	5
2.2 消息映射类 CCmdTarget.....	8
2.3 窗口类 CWnd.....	9
2.4 综合实例——窗口和消息.....	41
第3章 应用程序结构类.....	50
3.1 应用程序线程类 CWinThread.....	50
3.2 应用程序对象类 CWinApp.....	56
3.3 OLE 控制模板类 COleControlModule.....	63
3.4 文档模板类 CDocTemplate.....	63
3.5 单文档模板类 CSingleDocTemplate.....	66
3.6 多文档模板类 CMultiDocTemplate.....	68
3.7 OLE 类工厂 COleObjectFactory.....	69
3.8 OLE 服务器类 COleTemplateServer.....	71
3.9 OLE 数据类 COleDropSource.....	72
3.10 OLE 通信类 COleDropTarget.....	74
3.11 OLE 并行类 COleMessageFilter.....	76
3.12 通信接口类 CConnectionPoint.....	78
3.13 综合实例.....	80
实例 1——基本应用程序.....	80
实例 2——线程.....	83
实例 3——OLE 拖放.....	85
第4章 文档相关类.....	91
4.1 基本文档类 CDocument.....	91
4.2 OLE 文档类 COleDocument.....	96
4.3 OLE 包装程序文档类 COleLinkingDoc.....	99
4.4 带格式编辑控件文档 CRichEditDoc.....	101
4.5 文档项类 CDocItem.....	102
4.6 OLE 包装程序接口类 COleClientItem.....	103

4.7	活动文档类 COleDocObjectItem .....	111
4.8	CRichEditCntrlItem .....	112
4.9	OLE 服务器程序接口类 COleServerItem .....	113
4.10	文档对象服务项目类 CDocObjectServerItemItem .....	118
4.11	文档对象服务类 CDocObjectServer .....	119
4.12	综合实例 .....	121
	实例 1——OLE 服务器 .....	121
	实例 2——OLE 容器 .....	123
<b>第 5 章</b>	<b>框架相关类</b> .....	<b>126</b>
5.1	框架窗口类 CFrameWnd .....	126
5.2	多文档界面窗口类 CMDIChildWnd .....	131
5.3	多文档界面框架类 CMDIFrameWnd .....	133
5.4	最小框架窗口类 CMiniFrameWnd .....	136
5.5	编辑窗口类 COleIPFrameWnd .....	137
5.6	分隔器窗口类 CSplitterWnd .....	139
5.7	综合实例 .....	142
	实例 1——多文档界面 .....	142
	实例 2——窗口风格 .....	146
<b>第 6 章</b>	<b>控制条相关类</b> .....	<b>152</b>
6.1	控制条相关类 CControlBar .....	152
6.2	对话框条相关类 CDialogBar .....	155
6.3	ReBar 控制相关类 CReBar .....	157
6.4	状态条相关类 CStatusBar .....	159
6.5	工具条相关类 CToolBar .....	164
6.6	综合实例——可以自己订制的工具条 .....	167
<b>第 7 章</b>	<b>属性单相关类</b> .....	<b>175</b>
7.1	属性单类 CPropertySheet .....	175
7.2	属性单扩展类 CPropertySheetEx .....	182
7.3	综合实例——使子窗口带有属性单 .....	186
<b>第 8 章</b>	<b>对话框类</b> .....	<b>189</b>
8.1	对话框类 CDialog .....	189
8.2	通用对话框类 CCommonDialog .....	196
8.3	颜色对话框类 CColorDialog .....	197
8.4	文件对话框类 CFileDialog .....	200
8.5	查找替换对话框类 CFindReplaceDialog .....	205
8.6	字体对话框类 CFontDialog .....	209
8.7	对话框类 CPropertyPage .....	214
8.8	属性页扩展类 CPropertyPageEx .....	219
8.9	综合实例——一个动画对话框 .....	221

第9章 视图相关类.....	230
9.1 视图类 CView .....	230
9.2 控制视图类 CCtrlView.....	235
9.3 编辑视图类 CEditView .....	236
9.4 视图菜单类 CListView.....	239
9.5 带格式编辑控件视图类 CRichEditView.....	240
9.6 树视图类 CTreeView .....	244
9.7 滚动条类 CScrollView.....	245
9.8 表单类 CFormView.....	248
9.9 数据库记录类 CDaoRecordView.....	249
9.10 网页视图类 CHtmlView .....	251
9.11 Ole 数据库记录视图类 COleDbRecordView .....	258
9.12 记录视图类 CRecordView.....	260
9.13 综合实例.....	262
实例 1——滚动视图 .....	262
实例 2——窗体视图 .....	264
实例 3——列表视图 .....	267
实例 4——动态分割一个视图 .....	269
第10章 控件类.....	272
10.1 动画控件类 CAnimateCtrl.....	272
10.2 按钮类 CButton.....	275
10.3 位图按钮类 CBitmapButton.....	278
10.4 组合框类 CComboBox .....	280
10.5 扩展组合框类 CComboBoxEx .....	290
10.6 时间日期类 CDateTimeCtrl .....	291
10.7 编辑框类 CEdit.....	295
10.8 眉控件类 CHeaderCtrl.....	300
10.9 热键控件类 CHotKeyCtrl.....	304
10.10 IP 地址类 CIPAddressCtrl.....	306
10.11 列表框类 CListBox.....	308
10.12 复选列表框类 CCheckListBox.....	319
10.13 拖放列表类 CDragListBox .....	322
10.14 列表视图控件类 CListCtrl.....	325
10.15 月历控件类 CMonthCalCtrl.....	331
10.16 OLE 控制类 COleControl.....	334
10.17 进度条控件类 CProgressCtrl .....	342
10.18 Rebar 控制类 CRebarCtrl .....	343
10.19 RichEdit 控件类 CRichEditCtrl .....	346
10.20 滚动条类 CScrollBar .....	351
10.21 滑动杆类 CSliderCtrl.....	353

10.22	微调按钮控件类 CSpinButtonCtrl .....	356
10.23	静态文本控制类 CStatic .....	358
10.24	状态条控制类 CStatusBarCtrl .....	359
10.25	切换控制类 CTabCtrl .....	361
10.26	工具条控制类 CToolBarCtrl .....	364
10.27	工具提示类 CToolTipCtrl .....	379
10.28	树型控件类 CTreeCtrl .....	386
10.29	综合实例——WindowsXP 风格的按钮 .....	397
<b>第 11 章</b>	<b>异常类 .....</b>	<b>407</b>
11.1	异常类 CException .....	407
11.2	串行化异常类 CArchiveException .....	409
11.3	DAO 数据库异常类 CDaoException .....	410
11.4	数据库异常类 CDBException .....	413
11.5	文件异常类 CFileException .....	415
11.6	网络异常类 CInternetException .....	417
11.7	内存异常类 CMemoryException .....	418
11.8	不支持异常类 CNotSupportedException .....	419
11.9	OLE 异常类 COleException .....	420
11.10	OLE 界面异常类 COleDispatchException .....	420
11.11	资源异常类 CResourceException .....	421
11.12	用户端异常类 CUserException .....	422
11.13	综合实例——调试和异常处理 .....	423
<b>第 12 章</b>	<b>文件类以及文件管理类 .....</b>	<b>426</b>
12.1	文件类 CFile .....	426
12.2	驻留文件类 CMemFile .....	432
12.3	共享文件类 CSharedFile .....	434
12.4	数据流操作类 COleStreamFile .....	436
12.5	数据流类 CMonikerFile .....	438
12.6	异步标志类 CAsynCMonikerFile 类 .....	439
12.7	异步特性类 CDataPathProperty .....	442
12.8	控件属性类 CCachedDataPathProperty .....	444
12.9	CSocketFile 类 .....	445
12.10	流式文件类 CStdioFile .....	446
12.11	网络文件类 CInternetFile .....	448
12.12	Gopher 文件类 CGopherFile .....	451
12.13	http 文件类 CHttpFile .....	452
12.14	当前文件列表类 CRecentFileList .....	456
12.15	综合实例 .....	458
	实例 1——访问二进制文件 .....	458
	实例 2——访问标准 I/O 文件 .....	459



实例 3——访问内存文件 .....	460
<b>第 13 章 绘制图像相关类 .....</b>	<b>462</b>
13.1 设备上下文类 CDC .....	462
13.2 客户区设备上下文类 CClientDC .....	475
13.3 元文件设备上下文类 CMetaFileDC .....	476
13.4 绘制设备上下文类 CPaintDC .....	479
13.5 窗口设备上下文类 CWindowDC .....	480
13.6 停靠状态类 CDockState .....	481
13.7 图像表类 CImageList .....	483
10.8 图形对象类 CGdiObject .....	491
13.9 位图类 CBitmap .....	495
13.10 画刷类 CBrush .....	499
13.11 字体类 CFont .....	503
13.12 调色板类 CPalette .....	509
13.13 画笔类 CPen .....	513
13.14 区域类 CRgn .....	516
13.15 菜单类 CMenu .....	524
13.16 命令行信息类 CCommandLineInfo .....	533
13.17 综合实例 .....	536
<b>第 14 章 数据库类 .....</b>	<b>544</b>
14.1 数据库连接类 CDaoDatabase .....	544
14.2 查询定义的类 CDaoQueryDef .....	547
14.3 类 CDaoRecordset .....	550
14.4 数据库的工作管理类 CDaoWorkspace .....	576
14.5 连接数据源的对象 CDatabase .....	579
14.6 数据库中的记录类 CRecordSet .....	585
14.7 应用示例 .....	611
<b>第 15 章 OLE 相关类 .....</b>	<b>628</b>
15.1 字体的封装类 CFontHolder .....	628
15.2 CURRENCY 结构的封装类 COleCurrency .....	630
15.3 VARIANT 结构的封装类 COleVariant .....	633
15.4 客户端 OLE 自动化的执行类 COleDispatchDriver .....	635
15.5 应用示例 .....	638
<b>第 16 章 简单数据类 .....</b>	<b>649</b>
16.1 创建文档视图的类 CCreateContext .....	649
16.2 内存泄漏检测类 CMemoryState .....	649
16.3 OLE 的变量独立数组类 COleSafeArray .....	651
16.4 绘图用的坐标点类 CPoint .....	656
16.5 打印作业的信息类 CPrintInfo .....	657
16.6 绘图用的矩形类 CRect .....	660

16.7	绘图用的尺寸类 CSize .....	665
16.8	字符串类 CString .....	666
16.9	绝对时间类 CTime .....	682
16.10	相对时间类 CTimeSpan .....	685
16.11	应用示例 .....	687
<b>第 17 章</b>	<b>基本数据结构类</b> .....	<b>696</b>
17.1	数组类 CArray .....	696
17.2	动态字节数组类 CByteArray .....	697
17.3	双字数组类 CDWordArray .....	698
17.4	指针数组类 CObArray .....	698
17.5	指针数组类 CPtrArray .....	702
17.6	数据对象数组类 CStringArray .....	702
17.7	无符号整数数组类 CUIntArray .....	703
17.8	字数组类 CWordArray .....	704
17.9	列表类 CList .....	704
17.10	指针列表类 CPtrList .....	705
17.11	指针列表类 CObList .....	706
17.12	数据对象列表类 CStringList .....	709
17.13	映射类 CMap .....	710
17.14	字节映射指针类 CMapWordToPtr .....	712
17.15	指针映射字节类 CMapPtrToWord .....	713
17.16	指针映射指针类 CMapPtrToPtr .....	714
17.17	数据对象映射指针类 CMapStringToPtr .....	714
17.18	字典收集类 CMapStringToOb .....	715
17.19	指针映射指针类 CMapStringToString .....	715
17.20	综合实例 .....	716
	实例 1——数组函数 .....	716
	实例 2——列表函数 .....	718
	实例 3——映射函数 .....	719
<b>第 18 章</b>	<b>套接字和网络相关类</b> .....	<b>721</b>
18.1	网络通信类 CAsyncSocket .....	721
18.2	文件查找类 CFileFind .....	726
18.3	网络服务器的连接管理类 CFtpConnection .....	730
18.4	FTP 文件查找类 CFtpFileFind .....	734
18.5	Gopher 服务器的连接管理类 CGopherConnection .....	735
18.6	Gopher 文件查找类 CGopherFileFind .....	737
18.7	Gopher 服务器的定位类 CGopherLocator .....	739
18.8	WWW 服务器的连接管理类 CHttpConnection .....	740
18.9	网络服务器的连接管理类 CInternetConnection .....	742
18.10	网络连接描述类 CInternetSession .....	743

18.11	封装 Windows Socket 的类 CSocket.....	748
18.12	应用示例.....	750
<b>第 19 章</b>	<b>同步化类.....</b>	<b>790</b>
19.1	基于临界区的同步对象类 CCriticalSection.....	790
19.2	一种基于事件的同步对象类 CEvent.....	791
19.3	基于同步对象的类 CMutex.....	793
19.4	一种基于信号的同步对象 CSemaphore.....	794
19.5	基于同步对象的新基类 CSyncObject.....	795
19.6	应用示例.....	797
<b>第 20 章</b>	<b>网络服务 API 类.....</b>	<b>809</b>
20.1	HTML 记忆文件的管理类 CHtmlStream.....	809
20.2	超链接对象的管理类 CHttpFilter.....	813
20.3	CHttpFilter 对象的工具类 CHttpFilterContext.....	817
20.4	HTTP 服务器的功能扩展类 CHttpServer.....	819
20.5	HTTP 服务器的数据处理类 CHttpServerContext.....	824
20.6	应用示例.....	828
<b>第 21 章</b>	<b>支持类.....</b>	<b>834</b>
21.1	菜单显示管理类 CCmdUI.....	834
21.2	基于 DAO 的记录域交换类 CDaoFieldExchange.....	836
21.3	数据交换类 CDataExchange.....	837
21.4	ODBC 的变量类 CDBVariant.....	839
21.5	基于 OLE 的菜单管理类 COleCmdUI.....	841
21.6	OLE 的数据对象类 COleDataObject.....	842
21.7	支持对象的显示、移动和大小改变的类 CRectTracker.....	845
21.8	在线等待的光标类 CWaitCursor.....	848
<b>第 22 章</b>	<b>其他类.....</b>	<b>856</b>
22.1	多个对象的资源访问控制类 CMultiLock.....	856
22.2	单个对象的资源访问控制类 CSingleLock.....	858
22.3	用户定义的数组模板类 CTypedPtrArray.....	859
22.4	用户定义的链表模板类 CTypedPtrList.....	861
22.5	用户定义的位图模板类 CTypedPtrMap.....	863
22.6	应用示例.....	865

# 第 1 章 MFC 基础

几年以前，编写 Windows 程序只有有限的一些编程工具可供选择。那时候，C 已经是 Windows Software Development Kit (SDK) 使用的语言，而其他 Windows 编程环境，如 Microsoft Visual Basic 等工具还没有出现。大多数应用程序是用 C 语言编写的，所以那些无经验的程序员面临着艰巨的任务，他们不仅要学习有关新操作系统的一切，还要熟悉 Windows 提供的成千上万个不同的应用软件编程接口（即 API）函数。

现在，许多 Windows 程序仍然用 C 语言编写。但是多种 Windows 编程环境使具有商业品质的 Windows 程序可以使用 C、C++、Pascal、Basic 以及其他的许多语言来编写，而且，由于 Windows 的复杂性和 Windows API 所涉及领域的广泛性，这就急需有一种面向对象的编程语言；因此，C++ 就取代了 C，而成为了专业 Windows 程序员所选用的语言。许多 Windows 程序员都认为 C++ 是 C 的强劲对手，它带有一个类库来抽象 API，并将窗口和其他对象的基本操作封装在可重复使用的类中，这样就使得 Windows 编程变得简单了。而且，非常多的 C++ 程序员已经选择了 Microsoft Foundation Class（即 MFC，微软基础类库）。虽然也可以使用其他类库，但是只有 MFC 是由编写操作系统的公司编制的。MFC 被不断的更新以适应 Windows 操作系统最新的变动，并且它提供了一组内容全面的类，完整的包括了从窗口到 ActiveX 控件的所有类，使编写 Windows 应用程序更加容易。

## 1.1 MFC 简介

Microsoft 提供的 MFC 封装了 Windows API，是一个面向对象的封装的 C++ 类库。MFC 6.0 版本封装了大约 200 个类，其中的一些可以被用户直接使用，而另一些则主要作为用户自己的类的基础类。一些 MFC 类极其简单，例如 CPoint 类，它代表一个点（一个由  $x$  和  $y$  坐标定义的位置）。而有些类则很复杂，例如 CWnd 类，它封装了窗口的功能。在 MFC 程序中，您并不经常直接调用 Windows API，而是从 MFC 类创建对象并调用属于这些对象的成员函数。在类库中定义的成员函数有几百个，其中许多是 Windows API 的简单封装，甚至与相应的 API 具有同样的名字。这种命名约定的一个明显好处就是它加速了 C 程序员转变为 MFC 程序员的过程。想要移动一个窗口的话，一个 C 程序员可能会调用 SetWindowsPos API 函数。而 MFC 也支持 SetWindowsPos，它是 CWnd 类的一个成员，这可以理解为将一个窗口作为一个对象，而 SetWindowsPos 是希望在该对象上的操作。

MFC 也是一个应用程序的框架结构。MFC 不仅仅是一个类的集合，它还帮助定义了应用程序的结构并为应用程序处理许多杂务。以 CWinApp 类为例，CWinApp 类就代表应用程序自身，MFC 几乎封装了程序操作的所有方面。框架结构提供 WinMain 函数，而 WinMain 反过来调用应用程序对象的成员函数使程序运行下去。Run 是 WinMain 调用的一个 CWinApp 成员函数，它提供了一个消息循环，将消息送到应用程序的窗口。框架结构还提供了抽象功能，它远远的超出了 Windows API 的功能。例如 MFC 的文档/视图体系结构在 API 上建造了一个功能强大的基础结构，它把程序中数据的图形表示（或成为视图）与数据分开。这种抽象对 API 而言完全是陌生的，而且在 MFC 框架结构之外或相似类库中也不存在。

## 1.2 使用 C++ 和 MFC 的好处

阅读本书就说明您已经听到过对面向对象程序设计方法的传统的赞誉了：可重用性、代码和数据更紧密的捆绑，等等。您也应该对常用的面向对象程序设计（OOP）的术语很熟悉了，比如对象、继承和封装等，它们属于 C++ 语言。但是如果没有一个好的类库作为出发点，OOP 可能不会减少您编写代码的数量。

而这，正是 MFC 成功的地方。要给应用程序添加一个工具栏，使它能够被放置在窗口的各个边上或浮在它自己的窗口上吗？没有问题：MFC 提供了一个 CToolBar 类，它可以为您做大量的工作。如果需要一个连接的列表或可调整尺寸的数组呢？这也容易：CList、CArray 以及其他的 MFC 集合类可以为您的数据提供

封装的容器。另外，还有一些如 COM、OLE 和 ActiveX 等方面的类。可以这样说，很少有人希望了解或知道如何从头开始编写 ActiveX，但是很多程序员都可以号称 ActiveX 高手，因为在 COleControl 和 COlePropertyPage 等类中，MFC 提供了您所需的大量代码，从而简化了 ActiveX 控件的开发。

使用 MFC 的另一个优点是框架结构使用了很多技巧，使 Windows 对象（如窗口、对话框以及控件等）变得如同 C++ 中的对象了。假定您想编写一个可重复使用的列表框类，用来显示一个可浏览的、能显示 PC 主机中的驱动器和目录的列表。除非创建一个自定义控件来进行此工作，否则您无法用 C 来实现这样的列表框，这是因为在列表框中单击某一项会发送一个通知给列表框的父亲（列表框出现的对话框或窗口），而父亲将处理那份通知。换句话说，列表框控件控制不了自己的命运；当驱动器或目录被改变时，是父亲来更新列表框中的内容。

用 MFC 则不会这样。在 MFC 应用程序中，窗口和对话框将发送给他们的未处理的通知回送给发送这个通知的控件。通过从 CListBox 派生出您自己的列表框类，您可以创建一个自含的且高度可重用的列表框类来响应通知。结果列表框实现了自己的行为，并可以通过仅仅在源代码文件中添加一条 #include 语句来把它移植到另外一个应用程序中。这就是可重用性的实质。

### 1.3 MFC类的分层结构

MFC 提供了许多预先设计好的类来满足广泛的需要，它的类按本书目录所示的种类给出。

附录中图 A 所示为 MFC 6.0 类分层结构图。由图可知，大多数 MFC 类都是从 CObject 类中直接或间接的派生出来的。

CObject 类给那些继承它的类提供了 3 个重要的特性。

- 支持串行化。
- 运行时类的信息。
- 诊断和调试支持。

串行化是对象的永久数据流出或流入存储介质（如磁盘文件）的进程。把 CObject 作为基类，可以创建可串行化的类，其实例容易存储和重新创建。运行时类信息（RTCI）允许您在运行时检索对象的类名称以及对象的其他信息。RTCI 的执行不同于 C++ 中的运行时类型信息（RTTI）机制，因为它比 RTTI 要早出现好多年。嵌入在 CObject 中的诊断和调试支持允许您对 CObject 派生类的实例执行有效性检查，并将状态信息转储到一个调试窗口。

CObject 对它的派生类还提供了别的好处。例如重载 new 和 delete 运算符防止内存泄漏。如果从 CObject 派生类创建了一个对象，而没有在应用程序结束前删除它，那么 MFC 将会在调试输出窗口吐一条警告消息。随着您对 MFC 越来说熟悉，这个 MFC 类基本特点的重要性也会变得越来越清晰。

### 1.4 通用类设计规则

在 C++ 语言开始流行以前很久，Microsoft Windows 就被设计出来了。因为数以千计的应用程序都使用 C 语言 Windows 应用程序编程界面（API），所以此界面还要保持下去。因此，任何一个 C++ Windows 界面必须建立在过程化的 C 语言 API 之上。这就保证了 C++ 应用将能够与 C 应用共存。

Microsoft 基本类库是一个面向对象的 Windows 界面。它必须符合下列设计目标：

- 有效简化编写 Windows 应用的难度；
- 与 C 语言 API 相当的执行速度使经常使用的代码量最小化；
- 能够直接调用任何一个 Windows C 函数的能力；
- 使已有的 C 应用向 C++ 转换更容易；
- 可以更好地运用已有的 C 语言 Windows 编程经验基础的能力；
- 使得 C++ 的 Windows API 比 C 的 Windows API 更易使用；
- 更容易使用一些强大的复杂功能的抽象概念，如 ActiveX、数据库支持、打印、工具条以及状态条；
- 能够更有效使用 C++ 语言功能的 C++ Windows API。

## 1.5 综合实例——第一个MFC应用程序

这里我们使用经典的“Hello”例子程序为基础，来讲述 Windows 程序的一般结构以及使用 MFC 编写 Windows 应用程序所涉及到的基本原则。

### 1. 功能与设计

本示例创建一个简单的 Windows 窗口，响应一定的 Windows 消息。

### 2. 主要步骤与核心代码

- ① 使用 MFC APPWizard 产生一个 MFC 应用程序。
- ② MFC 的 AfxWinMain 函数调用该应用程序对象的 InitInstance 函数，用以构造一个窗口对象，同时该窗口对象的构造函数创建了出现在屏幕上的窗口。
- ③ 调用 ShowWindow 函数是窗口可见，并调用 UpdateWindow 函数发送第一个 WM\_PAINT 消息。
- ④ 调用 CmainWindow::OnPaint 函数，用来创建一个 CpaintDC 对象和调用其 DrawText 函数，在窗口的客户区绘制文本“Hello,MFC”。源代码如下（完整代码参见光盘 Exam1-1）：

```
//Hello.h 文件
class CMyApp : public CWinApp //创建一个类 CMyApp
{
public:
    virtual BOOL InitInstance ();
};

class CMainWindow : public CFrameWnd //创建一个类 CMainWindow
{
public:
    CMainWindow ();
protected:
    afx_msg void OnPaint ();
    DECLARE_MESSAGE_MAP ()
};

//Hello.cpp
#include <afxwin.h>
#include "Hello.h"

CMyApp myApp;
// CMyApp 成员函数
BOOL CMyApp::InitInstance () //初始化窗口
{
    m_pMainWnd = new CMainWindow;
    //调用 ShowWindow 和 UpdateWindow 函数来显示窗口
    m_pMainWnd->ShowWindow (m_nCmdShow);
    m_pMainWnd->UpdateWindow ();
    return TRUE;
}
```

//定义 CMainWindow 类的消息和使用的成员函数

```
BEGIN_MESSAGE_MAP (CMainWindow, CFrameWnd)
    ON_WM_PAINT ()
END_MESSAGE_MAP ()
```

CMainWindow::CMainWindow ()

```
{
    Create (NULL, _T ("The Hello Application")); //从 CFrameWnd 类继承了 Create 成员函数
}
```

//在屏幕上绘制窗口

void CMainWindow::OnPaint ()

```
{
    CPaintDC dc (this);
    //输出矩形窗口
    CRect rect;
    GetClientRect (&rect);
    //输出文本
    dc.DrawText (_T ("Hello, MFC"), -1, &rect,
        DT_SINGLELINE | DT_CENTER | DT_VCENTER);
}
```

3. 程序执行结果 (如图 1-1 所示)



图 1-1 Hello 程序运行结果

# 第 2 章 基类

Microsoft 基础类库中的类大多数派生于基类 CObject。由 CObject 类派生的 CCmdTarget 类和 CWnd 类又是很多类的父类，本章中将主要介绍这三个类。

## 2.1 根类CObject

### 1. 来龙去脉

CObject 是 Microsoft 基础类库中主要的基类。它不光是一些库类（如 CFile 类和 COblist 类）的根，也是用户自己所编写的所有类的根。

### 2. 功能描述

CObject 类提供了基本的服务，其中包括：

- 串行化支持。
- 运行时类的信息。
- 对象诊断的输出。
- 与收集类之间兼容。

### 3. 调用方法

为成功调用 CObject 类，需在程序中包含 afx.h 头文件。

CObject 类不支持多重继承。用户派生的类只能拥有一个 CObject 基类，并且 CObject 在继承体系中必须位于最开始的位置。

如果在类的执行与声明过程中使用了可选的宏，那么就能发现 CObject 派生的主要优点。对于第一层次的宏，如 DECLARE\_DYNAMIC 和 IMPLEMENT\_DYNAMIC，允许运行时访问类名以及它在整个继承体系中的位置。这样，就可以依次地进行有意义的诊断转储。对于第二层次的宏，如 DECLARE\_SERIAL 和 IMPLEMENT\_SERIAL，包含了第一层次宏的所有功能，并且能够使对象有效地串行化成一个“档案”或者从一个“档案”串行化而来。

### 4. 成员简表（见表 2-1）

表 2-1 CObject 类的成员函数

成员函数	功能简述
CObject	缺省的构造函数
operator new	特别的 new 操作
operator delete	特别的 delete 操作
operator=	赋值操作
AssertValid	证实该对象的完整性
Dump	进行该对象的诊断转储
IsSerializable	测试该对象是否被串行化
Serialize	从档案文件中装载或向档案文件中存储某对象
GetRuntimeClass	返回对应该对象类的 CRuntimeClass 结构
IsKindOf	测试该对象是否与指定类相关联



## 5. 用法详述

其主要成员函数和变量的用法如下:

### (1) 构造函数——CObject::CObject

#### ① 函数原型

```
CObject();
CObject( constCObject&objectSrc );
```

#### ② 功能简介

本函数为标准的 CObject 构造函数。派生类的构造函数自动调用该函数的缺省形式。

如果用户的类可串行化(它引入了 IMPLEMENT\_SERIAL 宏),那么在类的派生中必须使用缺省的构造函数(即没有参数的构造函数)。若不需要缺省的构造函数,请在派生类中事先声明私有的和受保护的“空”构造函数。

#### ③ 主要参数

objectSrc 另一个 CObject 对象的参考。

### (2) 诊断函数——CObject::AssertValid

#### ① 函数原型

```
virtual void AssertValid() const;
```

#### ② 功能简介

AssertValid 通过检测内部状态,对对象进行有效性检查。在 MFC 的调试版本中,AssertValid 将进行诊断,并发出诊断消息(列出断言失败的行数和文件名并终止程序)。

当编写了自己的类后,应当调用重载的 AssertValid 函数,为读者自己和使用类的用户提供诊断服务。被重载的 AssertValid 在检测对应于派生类的数据成员之前,通常调用它的基类的 AssertValid 函数。

由于 AssertValid 为一个常量函数,所以在测试期间,不允许改变对象的状态。自己派生类的 AssertValid 函数不应当产生异常,而应当断言是否检测到无效的对象数据。

“有效性”的诊断依赖于对象的类。作为规则,函数应当执行“浅度检测”(“shallow check”),也就是说,如果一个对象包含了其他对象的指针,那么它应当进行检测,以查看指针是否为空,而不是对指针指向的对象进行有效性测试。

#### ③ 主要参数(无)

#### ④ 调用方法示例

```
// CObject::AssertValid 举例
void CAge::AssertValid() const{
    CObject::AssertValid();
    ASSERT( m_years > 0 );
    ASSERT( m_years < 105 );
}
```

### (3) 诊断函数——CObject::Dump

#### ① 函数原型

```
virtual void Dump( CDumpContext& dc ) const;
```

#### ② 功能简介

将对象的内容转储到 CDumpContext 对象中。

当编写了自己的类后,应当重载 Dump 函数,为读者自己和使用类的用户提供诊断服务。被重载的 Dump 在打印对应于派生类的数据成员之前,通常调用它的基类的 Dump 函数。如果类使用了 IMPLEMENT\_DYNAMIC 和 IMPLEMENT\_SERIAL 宏,那么 CObject::Dump 将打印出类名。

#### ③ 主要参数

dc 用于转储的诊断性转储文本,通常该值为 afxDump。