



MFC与 Windows® 编程 (第2版) (影印版)

Covers
COM
and
ActiveX®

Programming
Windows®
with
MFC
Second Edition

- 来自微软的第一手技术资料，深入专业的编程技术
- 系统介绍 COM 与 ActiveX
- 关于 32 位 Windows 平台上的面向对象编程技术的权威指南
- 高级程序员必备

[美] Jeff Prosise 著

北京大学出版社

<http://cbs.pku.edu.cn>

微软编程圣典丛书(影印版)

MFC 与 Windows 编程

(第 2 版)

Microsoft 公司

江苏工业学院图书馆
藏书章

北京 大学出版社

内 容 简 介

本书是《微软编程圣典丛书(影印版)》之一,讲述如何在 32 位 Windows 平台上使用 MFC(微软基本类库)进行程序设计,内容涉及全新 COM、OLE 和 ActiveX,事件驱动型程序设计,位图,多线程等。为了增加本书的实用性,特以配套光盘的形式提供了丰富的程序实例以及本书的电子版。

本书由微软公司组织专家编写,具有相当的技术深度,是中、高级程序员必备的参考书。

Copyright (2000) by Microsoft Corporation

Original English language edition Copyright © 2000 (year of first publication by author)

By Microsoft Corporation (author)

All rights published by arrangement with the original publisher, Microsoft Press, a division of Microsoft Corporation, Redmond, Washington, U.S.A.

著作权合同登记号:图字 01-2000-3028 号

书 名: MFC 与 Windows 编程(第 2 版)(影印版)

责任著作者: Microsoft 公司 著

标准书号: ISBN 7-900629-32-7/TP·26

出 版 者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学校内 100871

网 址: <http://cbs.pku.edu.cn>

电 话: 出版部 62752015 发行部 62765127 62754140 编辑室 62765127

电 子 邮 箱: wdzh@mail.263.net.cn

印 刷 者: 中国科学院印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 85.25 印张 1200 千字

2000 年 9 月第 1 版 2000 年 9 月第 1 次印刷

定 价: 198.00 元

丛 书 序

世纪交替，IT 产业更加步履匆匆。

Microsoft 公司早已以其在编程方面的非凡成就闻名于世，并树立了在计算机软件领域和发展史上不可动摇的地位。毋庸置疑，该公司技术上的优势是其获得成功的重要因素之一。今天，它的技术不但已经变得非常强大，而且具有惊人的发展速度。尤其是 Windows 2000 技术的推出，更是展示了 Microsoft 的无穷魅力，它突然间提供了如此丰富的新特性，使我们仿佛在一瞬间便被淹没在 Windows 2000 浩瀚的技术海洋之中！

工欲善其事，必先利其器。作为 Windows 应用程序设计人员，必须紧密跟踪 Microsoft 公司的最新技术，深入 Microsoft Windows 编程的内幕，掌握关键的编程技术。这套《微软编程圣典丛书（影印版）》的推出，就是为了向有关的专业人员全面推介微软编程的核心技术，以便于他们设计高质量的 Windows 应用程序。

Microsoft 技术博大而精深，绝非某个人在短时间内所能掌握。为此，特按照技术上的逻辑关系组织成 9 个相对独立的部分，分别涉及基于服务器的应用程序、COM+ 基本服务、Windows 网络编程、国际化程序、MFC、Windows 编程、服务器端应用程序、Outlook 与 Exchange 编程、驱动程序模型等。每一部分的内容独立成册，集中讲述一组相关的编程技术。这套《微软编程圣典丛书（影印版）》共 9 本。特定编程领域的专业人员可以从中选取自己需要的一本或几本，使学习过程更加快速、省时、有效而直观。

这套丛书中的任何一本都涉及一门完整的编程技术，因此有着相当的深度，而且内容比较丰富。为了避免将其写成深奥而抽象的理论书，特在书中适当的位置穿插进许多贴切的程序实例。另外，每本书都有配套的 CD-ROM，内有书中的程序实例和本书的电子版。

本套丛书由 Microsoft 公司组织相关领域的专家编写。他们深谙 Microsoft 的编程技术内幕，具有丰富的程序开发经验，所以，这套丛书是他们智慧的结晶，是该领域极具权威性的著作，堪称独领风骚。

鉴于此，特向中、高级 Windows 应用程序设计人员郑重推荐这套佳作！

出版者
2000 年 9 月

Acknowledgments

The production of this book required the efforts of many people, but two in particular deserve to be singled out for their diligent, sustained, and unselfish efforts. Sally Stickney, the book's principal editor, navigated me through that minefield called the English language and contributed greatly to the book's readability. Marc Young, whose talents as a technical editor are nothing short of amazing, was relentless in tracking down bugs, testing sample code, and verifying facts. Sally, Marc: This book is immeasurably better because of you. Thanks.

Introduction

Like many of my colleagues in this industry, I learned Windows programming from Charles Petzold's *Programming Windows*—a classic programming text that is the bible to an entire generation of Windows programmers. When I set out to become an MFC programmer in 1994, I went shopping for an MFC equivalent to *Programming Windows*. After searching in vain for such a book and spending a year learning MFC the old-fashioned way, I decided to write one myself. It's the book you hold in your hands. And it's the book I would like to have had when I was learning to program Windows the MFC way.

MFC, as you probably already know, is Microsoft's C++ class library for Windows programming. *Programming Windows with MFC* isn't a book about C++; rather, it's a book about writing 32-bit Windows applications in C++ using MFC rather than the Windows API as the chief means of accessing the operating system's essential features and services. It was written with two kinds of people in mind:

- Windows API programmers who want to learn MFC
- Programmers who have never before programmed Windows

Whichever camp you fall into, I assume that you know the C++ programming language already and are comfortable with basic C++ idioms such as derived classes and virtual functions. If these assumptions are true, you're ready to begin climbing the hill that is MFC programming.

Even veteran Windows programmers frequently find MFC code confusing the first time they see it, in part because of the presence of code created by the MFC code-generating wizards in Visual C++ and in part because of the countless lines of code hidden away in MFC classes such as *CFrameWnd*, *CDocument*, and *CView*. That's why this book takes a rather unusual approach to teaching MFC. It begins by having you write MFC code by hand (without the wizards) and by utilizing MFC 1.0–style application architectures—that is, applications that use neither documents nor views. Only after you've mastered the fundamentals and become acquainted with basic MFC

classes such as *CWnd* and *CWinApp* do I introduce the wizards and teach you how to take advantage of MFC's document/view architecture. Along the way, you build an understanding from the ground up of the message-oriented nature of Windows and of key components of Windows itself, such as the Graphics Device Interface (GDI). I believe that this approach makes learning MFC not only less intimidating but also more enjoyable. I think you'll agree once you've worked through the book and can look back on the learning experience from the standpoint of a knowledgeable Windows programmer.

Programming Windows with MFC is divided into four parts. Part I introduces the core tenets of MFC and Windows programming, beginning with a simple "Hello, MFC" application and introducing, one by one, menus, controls, dialog boxes, and other application building blocks. Part II builds on the foundation laid in Part I with a detailed look at the document/view architecture. In particular, Chapters 9, 10, and 11 reveal much of the "magic" behind documents and views and explain not only how to write basic document/view applications but also how to implement some not so basic features such as split-window views of a document and print previews. Part III covers some of the more advanced features of Windows and MFC—features such as color palettes, bitmap handling, and multiple threads of execution. In Part IV, you'll learn how MFC wraps its arms around COM, OLE, and ActiveX and how to write COM-enabled applications and software components. By the time you're finished with Chapter 21, you'll be well versed in the art of 32-bit Windows programming using MFC. And you'll have prodigious amounts of sample code to draw from when it's time to strike out on your own and write your first great Windows application.

WHAT'S NEW IN THE SECOND EDITION

Those of you who read the first edition of this book will notice two rather obvious changes in the second edition. First, this edition contains seven new chapters. One is devoted to the MFC view classes; another covers the MFC collection classes; one introduces MFC file I/O and serialization mechanisms; and four cover the relationship between MFC and COM. MFC is not the general-purpose COM framework that the Active Template Library (ATL) is, but MFC makes certain types of COM programming exceptionally easy. For example, MFC greatly simplifies the task of writing ActiveX controls, and it makes writing Automation servers—programs that use COM to expose their functionality to scripting clients—a breeze.

The second major change in this edition has to do with wizards. The first edition didn't cover the MFC wizards at all. The second edition uses hand-generated code in Chapters 1 through 3 but then shifts gears and begins using AppWizard and ClassWizard in Chapter 4. Why the change of heart? I still believe that code-generating wizards are an impediment to learning and should be used only by knowledgeable

programmers, but I've also come to realize that in the real world, MFC programmers use the wizards. For certain tasks—writing ActiveX controls, for example—it doesn't make sense *not* to use the wizards. So after much deliberation, I decided I would be remiss not to cover them.

Despite the new material regarding wizards, however, this is not—and never will be—a book about clicking buttons in AppWizard. After introducing a fundamental skill, such as how to write a message handler with ClassWizard, I thereafter let the source code do the talking and assume that you can figure out how the source code was created. Keep in mind that the wizards never do anything you can't do yourself, so it's perfectly feasible to type in every source code listing by hand if you'd like to.

The downside to using wizards in a book that teaches MFC programming is that the code they produce isn't fit to publish. The first edition of this book included printed listings for each and every source code file. This one does not. It contains printed copies of *relevant* source code files and provides the others on CD. Why? Because printing a source code file that's 50 percent meat and 50 percent fat adds bulk to a book without adding content. Some of the code produced by the MFC AppWizard in Visual C++ 6.0 won't even compile. (For details, see Chapter 4.) I'm not very proud of the parts of my book that the wizards created, because those portions are littered with arbitrary blank lines, comments that lack consistent style, and unnecessary functions. For someone who takes pride in writing concise, readable sample code, wizard output is a bitter pill to swallow.

Nevertheless, wizards represent the new world order in Windows programming, and they're something that you, I, and everyone else must get used to. It's a shame that the Visual C++ team won't give us real wizards to play with instead of the toys that they pass off as wizards today. Until they do, we must make do with what we have.

WHAT'S ON THE CD

The companion CD contains source code and executables for all the sample programs presented in the book. All samples were written and compiled with Visual C++ 6.0 and MFC 6.0 and tested on various Win32 platforms. Unless otherwise noted, all are compatible with Windows 98, Windows NT 4.0, and Windows 2000. Most are also compatible with Windows 95 and Windows NT 3.51.

You can copy the contents of the CD to your hard disk by running the setup program found in the CD's root directory, or you can retrieve the files directly from the CD's \Code directory. The \Code directory contains one subdirectory for each chapter of the book—Chap01, Chap02, and so on. Inside these subdirectories you'll find the sample programs. Each set of source code files is accompanied by a release-build EXE as well as a Visual C++ workspace (DSW) file that you can open with Visual C++'s Open Workspace command.

FROM ME TO YOU (AND YOU TO ME)

From the day in 1995 when I began writing the first edition of *Programming Windows with MFC*, my goal has been to provide C++ programmers with the same kind of timeless, irreplaceable resource that *Programming Windows* is to C programmers. Whether I've achieved that goal, I'll let you be the judge.

I want to know what you think about *Programming Windows with MFC*, and I particularly want to hear from you if you find mistakes. You can reach me by sending mail to jeffpro@msn.com or by visiting my Web site at www.prosise.com. At that site you'll find up-to-date information regarding the book, a list of errata, and information about other projects that I'm working on. Later this year, I plan to post a brand new chapter on MFC DLLs that you can read and comment on online.

Given the huge volume of computer books vying for buyers' attention in bookstores today, I know that you could have chosen any number of MFC books besides this one. I thank you for purchasing *Programming Windows with MFC*, and I sincerely hope you conclude that your money was well spent. Enjoy!

Jeff Prosise
March 12, 1999

Contents

Acknowledgments

xxi

Introduction

xxiii

Part I Fundamentals of Windows and MFC

<i>Chapter 1</i>	Hello, MFC	3
	THE WINDOWS PROGRAMMING MODEL	4
	Messages, Messages, and More Messages	6
	Windows Programming, SDK-Style	7
	Hungarian Notation and Windows Data Types	10
	SDK Programming in Perspective	11
	INTRODUCING MFC	12
	The Benefits of Using C++ and MFC	12
	The MFC Design Philosophy	13
	The Document/View Architecture	14
	The MFC Class Hierarchy	15
	AFX Functions	15
	YOUR FIRST MFC APPLICATION	16
	The Application Object	19
	How MFC Uses the Application Object	21
	The Frame Window Object	22
	Painting the Window	25
	The Message Map	27
	How Message Maps Work	29
	Windows, Character Sets, and the _T Macro	31
	Building the Application	33
	The Big Picture	34

Contents

<i>Chapter 2</i>	Drawing in a Window	37
	THE WINDOWS GDI	38
	The MFC Device Context Classes	39
	Device Context Attributes	42
	The Drawing Mode	44
	The Mapping Mode	45
	Programmable Mapping Modes	48
	Coordinate Conversions	49
	Moving the Origin	50
	A Final Word on Coordinate Systems	51
	Getting Information About a Device	52
	DRAWING WITH THE GDI	54
	Drawing Lines and Curves	54
	Drawing Ellipses, Polygons, and Other Shapes	57
	GDI Pens and the <i>CPen</i> Class	60
	GDI Brushes and the <i>CBrush</i> Class	64
	Drawing Text	67
	GDI Fonts and the <i>CFont</i> Class	69
	Raster Fonts vs. TrueType Fonts	71
	Rotated Text	72
	Stock Objects	73
	Deleting GDI Objects	75
	Deselecting GDI Objects	76
	The Ruler Application	78
	SEEING WHAT YOU'VE DRAWN	81
	Adding a Scroll Bar to a Window	81
	Setting a Scroll Bar's Range, Position, and Page Size	82
	Synchronizing the Thumb Size and the Window Size	85
	Processing Scroll Bar Messages	85
	Scrolling a Window	87
	The Accel Application	88
	LOOSE ENDS	98

<i>Chapter 3</i>	The Mouse and the Keyboard	101
	GETTING INPUT FROM THE MOUSE	102
	Client-Area Mouse Messages	103
	The TicTac Application	106
	Nonclient-Area Mouse Messages	124
	The WM_NCHITTEST Message	127
	The WM_MOUSELEAVE and WM_MOUSEHOVER Messages	127
	The Mouse Wheel	130
	Capturing the Mouse	132
	Mouse Capturing in Action	133
	The Cursor	139
	The Hourglass Cursor	141
	Mouse Miscellanea	142
	GETTING INPUT FROM THE KEYBOARD	144
	The Input Focus	144
	Keystroke Messages	145
	Virtual Key Codes	148
	Shift States and Toggles	150
	Character Messages	151
	Dead-Key Messages	154
	The Caret	154
	THE VISUALKB APPLICATION	158
	Handling the Caret	171
	Entering and Editing Text	174
	Other Points of Interest	175
<i>Chapter 4</i>	Menus	177
	MENU BASICS	178
	Creating a Menu	179
	Loading and Displaying a Menu	182
	Responding to Menu Commands	183
	Command Ranges	185
	Updating the Items in a Menu	186
	Update Ranges	189
	Keyboard Accelerators	190

Contents

THE SHAPES APPLICATION	193
Running the MFC AppWizard	213
Analyzing AppWizard's Output	216
Beyond AppWizard	220
The Process in Review	225
MENU MAGIC	226
Creating Menus Programmatically	226
Modifying Menus Programmatically	227
The System Menu	229
Owner-Draw Menus	231
Cascading Menus	235
Context Menus	237
THE COLORS APPLICATION	240
The Context Menu	267
On Your Own	269
<i>Chapter 5</i> The MFC Collection Classes	271
ARRAYS	271
The MFC Array Classes	273
Dynamic Array Sizing	275
Creating Type-Safe Array Classes with <i>CArray</i>	278
LISTS	279
The MFC List Classes	281
Creating Type-Safe List Classes with <i>CList</i>	283
MAPS	285
The MFC Map Classes	285
How Maps Work	287
Optimizing Lookup Efficiency	288
Creating Type-Safe Map Classes with <i>CMap</i>	290
THE TYPED POINTER CLASSES	291
<i>Chapter 6</i> File I/O and Serialization	293
THE CFILE CLASS	294
Opening, Closing, and Creating Files	294
Reading and Writing	297
CFile Derivatives	299
Enumerating Files and Folders	300

SERIALIZATION AND THE CARCHIVE CLASS	302
Serialization Basics	303
Writing Serializable Classes	304
Versioning Serializable Classes: Versionable Schemas	306
How Serialization Works	309
Serializing CObjects	313
<i>Chapter 7</i> Controls	315
THE CLASSIC CONTROLS	316
The CButton Class	319
The CListBox Class	323
The CStatic Class	331
The FontView Application	333
The CEdit Class	342
Presto! Instant Notepad	348
The CComboBox Class	357
The CScrollBar Class	361
ADVANCED CONTROL PROGRAMMING	362
Numeric Edit Controls	362
Owner-Draw List Boxes	363
Graphical Push Buttons	374
Customizing a Control's Colors	375
Message Reflection	382
<i>Chapter 8</i> Dialog Boxes and Property Sheets	385
MODAL DIALOG BOXES AND THE CDIALOG CLASS	386
The Dialog Box Template	386
The CDialog Class	392
Creating a Modal Dialog Box	395
Dialog Data Exchange and Dialog Data Validation	397
Interacting with the Controls in a Dialog	403
The DlgDemo1 Application	405
MODELESS DIALOG BOXES	417
The DlgDemo2 Application	418
USING A DIALOG BOX AS A MAIN WINDOW	432

Contents

PROPERTY SHEETS	449
The PropDemo Application	452
THE COMMON DIALOGS	469
Modifying the Common Dialogs	470
The Phones Application	472

Part II **The Document/View Architecture**

Chapter 9 **Documents, Views, and the Single Document Interface** **493**

DOCUMENT/VIEW FUNDAMENTALS	494
The <i>InitInstance</i> Function Revisited	496
The Document Object	498
The View Object	503
The Frame Window Object	506
Dynamic Object Creation	506
More on the SDI Document Template	508
Registering Document Types with the Operating System Shell	510
Command Routing	511
Predefined Command IDs and Command Handlers	513
YOUR FIRST DOCUMENT/VIEW APPLICATION	515
The SdiSquares Application	516
SdiSquares Step by Step	537
DOC + VIEW = LESS WORK FOR YOU	539

Chapter 10 **Scroll Views, HTML Views, and Other View Types** **541**

SCROLL VIEWS	542
CScrollView Basics	542
CScrollView Operations	546
Optimizing Scrolling Performance	547
The ScrollDemo Application	548
Converting an Ordinary View into a Scroll View	557
HTML VIEWS	557
CHtmlView Operations	558
CHtmlView Overridables	560
Utilizing DHTML in CHtmlView-Based Applications	562
Tree Views	568

Initializing a Tree View	569
Tree View Member Functions and Notifications	571
The DriveTree Application	573
LIST VIEWS	586
Initializing a List View	586
Changing the Presentation Style	589
Sorting in a List View	590
Hit-Testing in a List View	591
The WinDir Application	592
DO-IT-YOURSELF CONTROL VIEWS	607
<i>Chapter 11</i> Multiple Documents and Multiple Views	609
MFC AND THE MULTIPLE DOCUMENT INTERFACE	610
Synchronizing Multiple Views of a Document	612
The MdiSquares Application	615
Supporting Multiple Document Types	637
Alternatives to MDI	639
SPLITTER WINDOWS	639
Dynamic Splitter Windows	641
The Sketch Application	642
Static Splitter Windows	663
The Wanderer Application	665
Custom Command Routing	698
Three-Way Splitter Windows	699
Dynamic Splitter Windows with Multiple View Types	702
<i>Chapter 12</i> Toolbars, Status Bars, and Rebars	703
TOOLBARS	704
Creating and Initializing a Toolbar	704
Docking and Floating	709
Controlling a Toolbar's Visibility	712
Keeping Toolbar Buttons in Sync with Your Application	713
Adding ToolTips and Flyby Text	715
Adding Non-Push-Button Controls to a Toolbar	718
Updating Non-Push-Button Controls	719
Making Toolbar Settings Persistent	720
Toolbar Support in AppWizard	721

Contents

STATUS BARS	722
Creating and Initializing a Status Bar	722
Providing Context-Sensitive Help for Menu Items	725
Creating Custom Status Bar Panes	725
Status Bar Support in AppWizard	728
PUTTING IT ALL TOGETHER: THE MYWORD APPLICATION	729
The Main Toolbar	751
The Style Bar	752
More About <i>CRichEditView</i>	755
REBARS	757
<i>Chapter 13</i> Printing and Print Previewing	759
PRINTING WITH DOCUMENTS AND VIEWS	760
The Windows Print Architecture	760
The MFC Print Architecture	765
Print Previewing	772
A BARE-BONES PRINTING APPLICATION	774
Black-and-White Print Previews	780
A MORE COMPLEX PRINTING APPLICATION	781
A Unique Approach to Serialization	795
PRINTING TIPS AND TRICKS	796
Using the Print Dialog's Selection Button	796
Assume Nothing—And Test Thoroughly!	797
Adding Default Pagination Support	799
Enumerating Printers	800
 Part III Beyond the Basics	
<i>Chapter 14</i> Timers and Idle Processing	805
TIMERS	806
Setting a Timer: Method 1	806
Responding to WM_TIMER Messages	808
Setting a Timer: Method 2	811
Stopping a Timer	813
THE CLOCK APPLICATION	813
Processing Timer Messages	825
Getting the Current Time: The <i>CTime</i> Class	826