



华工松联电脑丛书

武汉松联公司
北京 松岗



深入浅出

WINDOWS MFC 程序设计

侯俊杰 编著



华中理工大学出版社

· 武汉松联环球电脑信息有限公司 · 北京松岗公司 ·

深入浅出 Windows MFC 程序设计

侯俊杰 著

华中理工大学出版社

(鄂)新登字第 10 号

图书在版编目(CIP)数据

深入浅出 Windows MFC 程序设计/侯俊杰 著

武汉:华中理工大学出版社,1998.4.

ISBN 7-5609-1641-4

I. 深…

II. 侯…

III. 计算机软件-Windows 95-程序设计

IV. TP316

本书封面贴有华中理工大学出版社激光防伪标志,封底贴有台湾松岗公司防伪标志,无标志者不得销售。

版权所有 盗印必究

深入浅出 Windows MFC 程序设计

© 侯俊杰 著

责任编辑:王元珍 等

封面设计:添 艺

责任校对:张 欣

监 印:张正林

出版发行者:华中理工大学出版社

(武汉市 邮编:430074)

经 销 者:新华书店湖北发行所
武汉松联图书软件批销中心

(电话:027—7874022,7382238)

印 刷 者:华中理工大学出版社印刷厂

(邮编:430074)

本书如有破损或装订错误,请寄回出版社更换

开本:787×1092 1/16

印张:39.75

字数:972 000

版次:1998年4月第1版

印次:1998年4月第1次

印数:1—5 000

ISBN 7-5609-1641-4/TP·250

定价:68.00 元

出 版 说 明

本书中文繁体字版由台湾松岗电脑图书资料股份有限公司(下简称“松岗公司”)出版。本书中文简体字版经松岗公司授权由华中理工大学出版社出版。任何单位或个人未经出版者书面允许不得用任何手段复制或抄袭本书内容。

由于海峡两岸计算机科学技术术语的译名不太相同,因此在出版中文简体字版时对正文和屏幕显示图上的术语进行了转译。转译工作是由王元珍教授、唐伟硕士、周波学士等人完成的。转译内容力求做到表述准确贴切。考虑到屏幕显示图上对中文繁体字转译工作量大且有一定难度,故对部分图上不涉及计算机科学技术术语的繁体字、词未作转译。

在中文简体字版中,对原中文繁体字版中某些仅适合台湾地区的内容经征得松岗公司驻北京代表叶权荣先生同意后作了删节,对原版书中个别错字、漏字也作了更正。原版书中附有练习范例磁盘,因其运行环境为中文繁体字版软件,与中文简体字版的正文内容不能对应,故中文简体字版均不带磁盘。

本书在中文简繁转译工作过程中得到了其他同志的大力帮助,谨此致衷心感谢。

华中理工大学出版社

1997年10月

内 容 简 介

本书是一本既简明又深入地全面系统介绍 Win95 MFC 程序设计技术的书籍。它在简介 Windows 程序概念、C++ 重要技术、MFC 六大技术简化仿真基础上,通过一个 MFC 程序的生与死过程,即从 MFC 的运用,钻入 MFC 的内部运作,进而 Application Framework 的原理,再至面向对象精神,然后回到 MFC 的运用过程,既详细解释了 MFC 程序的来龙去脉,又全面深入地探讨了各式各样的 MFC 程序设计技术及其奥秘。

本书对于广大 Win95 应用软件开发人员和想全面深入学习应用 MFC 技术的人员或高校师生,是一本最适宜和难得的读物。

候俊杰先生是蜚声海峡两岸的著名计算机专家学者,他写的书不仅内容精深、信息量大,而且文字流畅,易读好懂。深受广大读者的推崇与喜爱。本书是他的近作之一,为想要深入了解 MFC 技术的读者指引迷津,希望每位有志于计算机软件开发的读者不可失之交臂。

自序

有一种软件名叫 Version Control,用来记录程序开发过程中的各种版本,以应不时之需,可以随时反省追寻过去努力的轨迹。

遗憾的是人的大脑没有 Version Control 的能力。学习过程的徬徨犹豫、挫折困顿、在日积月累的渐悟或刹那之间的顿悟之后,仿佛都成了遥远模糊的回忆,而屡起屡仆、大惑不解的地方,学成之后看起来则是那么“理所当然”。

学习过往的艰辛,模糊而明亮,是学成冠冕上闪亮的宝石。过程愈艰辛,宝石愈璀璨。作为自己“想当年”的绝佳话题可矣,对于后学则无甚帮助。的确,谁会在一再跌倒的地方做上记号,永志不忘?谁会把推敲再三的心得殷实详尽地记录下来,为后学铺一条红地毯?也许,没有 Version Control 正是人类的本能,空出更多的脑力心力与精力,追求更新的事物。

但是,作为信息教育体系一员的我,不能不有 Version Control。事实上我亦从来没有忘记初学 MFC 的痛苦:C++语言本身的技术问题是其一,MFC 庞大类别库的命名规则是其二,熟知的 Windows 程序基本动作统统不见了是其三,面向对象的概念与 Application Framework 的包装是其后。初学 MFC Programming 时,我的脑袋犹如网眼过大的筛子,什么东西都留不住;各个类别及其代表意义,过眼即忘。

刚刚接触 MFC 时,我对 Windows 操作系统以及 SDK 程序设计技术的掌握,实已处在众人金字塔的顶端,困顿犹复如斯。实在是因为,对传统程序员而言,Application Framework 和 MFC 的运作机制太让人陌生了。

目前市面上有不少讲解 MFC 程序设计观念的书籍,其中不乏很好的作品,包括 Programming Windows 95 with MFC(Jeff Prosise 著,Microsoft Press 出版),以及我曾经翻译过的(David J. Kruglinski 著,Microsoft Press 出版)。深入浅出 MFC 的宗旨与以上二书,以及全世界所有的 MFC 或 Visual C++ 书籍,都不相同。全世界(呵,我的确敢这么说)所有与 MFC 相关的书籍的重点,都放在如何使用各式各样的 MFC 类别上,并供应各式各样的应用实例,我却意不在此。我希望提供的是对 MFC 应用程序基本结构的每一个技术环节的深入探讨,其中牵涉到 MFC 本身的设计原理,面向对象的概念以及 C++ 语言的高级议题。有了基础面的全盘掌握,各个 MFC 类别之使用对我们而言只不过是手册查阅的功夫罢了。

本书书名已经自我说明了,这是一本既深又浅的书。深与浅是悖离的两条射

线,理不应同时存在。然而,没有深入如何浅出?不入虎穴焉得虎子?

唯有把 MFC 程序框架的每一个基础动作弄懂,甚至观察其源代码,才能实实在在掌握 MFC 这一套 Application Framework 的内涵,及其面向对象的精神。我向来服从一句名言:源代码说明一切,所以,我挖 MFC 源代码给你看。

这是我所谓的深入。

唯有掌握住 MFC 的内涵,对于各式各样的 MFC 应用才能够如履平地,面对庞大的 Application Framework 也才能够胸中自有丘壑。

这是我所谓的浅出。

本书分为四大篇。第一篇提出学习 MFC 程序设计之前的必要基础,包括 Windows 程序的基本概念以及 C++ 的高级议题。“学前基础”是相当主观的认定,不过,基于我个人的学习经验以及教学经验,我的挑选应该颇具说服力。第二篇介绍 Visual C++ 集成环境开发工具。本篇只不过是提纲挈领而已,并不企图取代 Visual C++ 使用手册。然而对于软件使用的老手,此篇或已足以让您掌握 Visual C++ 集成环境。工具的使用虽然谈不上学问,但在视觉化软件开发过程中扮演极重要角色,切莫小看它。

第三篇介绍 Application Framework 的概念,以及 MFC 程序框架。所谓程序框架,是指 Visual C++ 的工具 AppWizard 所产生出来的程序码。当然,AppWizard 会根据你的选项做出不同的程序码,我所据以解说的,是大众化选项下的产品。

第四篇以微软公司附于 Visual C++ 光盘片上的一个范例程序 Scribble 为主线,一步一步加上新的功能。并在其间深入介绍 Runtime Type Information (RTTI)、Dynamic Creation、Persistence (Serialization)、Message Mapping、Command Routing 等核心技术。这些技术正是其他书籍最缺乏的部分。此篇之最后数章则脱离 Scribble 程序,另成一格。

本书的前身,是 1994 年 8 月出版的 Visual C++ 面向对象 MFC 程序设计基础篇以及 1995 年 4 月出版的应用篇,其序言之中我曾经这么说,全世界没有任何书籍文章,能够把 MFC 谈得这么深,又表现得这么浅。这些话已有一半成为昨日黄花:Microsoft Systems Journal 1995 年 7 月的一篇由 Raul Dilascia 所撰的文章 Meandering Through the Maze of MFC Message and Command Routing,以及 Addison Wesley 于 1996 年 6 月出版的 MFC Internal 一书,也有了相当程度的涉猎,即连前面提及的 Programming Windows 95 with MFC 以及 Inside Visual C++ 4.0,也都多多少少开始涉及 MFC 的核心。我有一种“德不孤,必有邻”的喜悦。

为了维护本书更多的唯一性,也由于我自己又钻研获得了新的心得,本书增加了前版未有的 Runtime Type Information、Dynamic Creation 等主题,对于 Message Mapping 与 Command Routing 的讨论也更详细得多,从而填补了上一版的不足。更值得一提的是,我把这些在 MFC 中极神秘而又极重要的机制,以简化到不能再简化的方式,在 DOS 程序中模拟出来,并且补充一章专论 C++ 的高级技术。至此,整个 MFC 的基础结构已经完全曝露在你的掌握之中,再没有任何神

秘咒语了。

本书从MFC的运用,钻入MFC的内部运作,进而Application Framework的原理,再至面向对象的精神,然后回到MFC的运用。这会是一条迢迢远路吗?

似远实近!

许多朋友曾经与我讨论过,对于MFC这类Application Framework,应该挖掘其内部机制到什么程度?探究源代码,岂不有违“黑盒子”初衷?但是,没有办法,他们也同意,不把那些奇奇怪怪的宏和指令搞清楚,只能生产出玩具来。对付MFC内部机制,态度不必像对付MFC类别一样;你只需好好走过那么一回,有个印象,足矣。至于庞大繁复的整个Application Framework技术的铺陈串接,不必人人都痛苦一次,我做这么一次也就够了。

林语堂先生在《朱门》一书中说过的一句话——“只用一样东西,不明白它的道理,实在不高明”。足以描述我写作本书时的心境,同时也对我与朋友之间的讨论做个总结。

祝各位胸中丘壑自成!

侯俊杰

新竹 1996. 8.

P. S. 愈来愈多的朋友在网络上与我打招呼,闲聊谈心。有医师、盲生、北京的作家、香港的读者、从中学到研究所的各级学生。学生的科系范围广到令我惊讶,年龄的范围也大到令我惊讶。对于深居简出的作家而言,读者群只是一个想像空间,哦,我真有这么多读者吗?! 呵呵,喜欢这种感觉。回信虽然是一种压力,不过这是个甜蜜的负担。

你们常常感谢我带给你们帮助。你们一定不知道,没有你们细心研读我的心血,并且热心写信给我,我无法忍受写作的漫漫孤寂! 我可以花三天的时间写一篇序,也可以花一个上午设计一张图。是的,我愿意! 我对拥有一群可爱可敬的读者感到骄傲。

目 录

第 0 章 你一定要知道(导读)	(1)
0.1 这本书适合谁	(1)
0.2 你需要什么技术基础	(2)
0.3 你需要什么样的软硬件环境	(2)
0.4 让我们使用同一种语言	(2)
0.5 本书符号习惯	(5)
0.6 碟片内容与安装	(5)
0.7 范例程序说明	(5)
0.8 如何与作者联系	(7)
0.9 与前版本之差异	(7)

第一篇 勿在浮砂筑高台——本书技术前提

第 1 章 Win32 程序基本概念	(10)
1.1 Win32 程序开发流程	(10)
1.2 需要什么函数库(.LIB)	(11)
1.3 需要什么头文件(.H)	(12)
1.4 以信息为基础,以事件驱动之	(12)
1.5 一个具体而微的 Win32 程序	(13)
程序入口 WinMain	(19)
窗口类的注册与窗口的诞生	(20)
信息循环	(21)
窗口的生命中枢——窗口函数	(22)
信息映射(Message Map)的雏形	(22)
对话框的运用	(24)
模块定义文件(.DEF)	(25)
资源定义文件(.RC)	(26)
1.6 Windows 程序的生与死	(26)
1.7 闲置时间的处理:OnIdle	(27)
1.8 进程与线程(Process and Thread)	(28)
核心对象	(28)
一个进程的诞生与死亡	(29)

产生子进程.....	(29)
一个线程的诞生与死亡.....	(31)
线程的优先权(Priority)	(32)
第2章 C++的重要性质	(35)
2.1 类及其成员——谈封装(encapsulation)	(35)
2.2 基类与派生类——谈继承(Inheritance)	(36)
2.3 this 指针	(39)
2.4 虚函数与多态性(Polymorphism)	(39)
从薪水说起.....	(43)
晴天霹雳.....	(45)
三个结论.....	(45)
虚函数与一般化.....	(47)
多态性(Polymorphism)	(50)
2.5 类与对象大剖析.....	(51)
2.6 Object slicing 与虚函数	(55)
2.7 静态成员(变量与函数).....	(58)
2.8 C++程序的生与死:兼谈构造函数与析构函数	(60)
2.9 动态类型识别(RTTI)	(61)
2.10 动态生成(Dynamic Creation)	(64)
2.11 异样处理(Exception Handling).....	(64)
Template	(68)
Template Functions	(69)
Template Classes	(70)
Templates 的编译与链接	(72)
第3章 MFC 六大关键技术之仿真.....	(74)
3.1 MFC 类的层次结构	(75)
Frame1 范例程序	(75)
3.2 MFC 程序的初始化过程	(78)
Frame2 范例程序	(81)
3.3 自制 RTTI(动态类型识别)	(85)
类别型录网与 CRuntimeClass	(86)
DECLARE_DYNAMIC/IMPLEMENT_DYNAMIC 宏.....	(87)
Frame3 范例程序	(93)
IsKindOf(类型识别)	(101)
Frame4 范例程序.....	(102)
3.4 自制 Dynamic Creation(动态生成)	(103)
DECLARE_DYNCREATE/IMPLEMENT_DYNCREATE 宏	(104)
Frame5 范例程序.....	(110)
MFC 4.0 对动态生成的作法	(119)
Frame6 范例程序.....	(121)

3.5	自制 Persistence(永久保存)机制	(121)
	Serialize(数据读写)	(121)
	DECLARE_SERIAL/IMPLEMENT_SERIAL 宏	(126)
	没有范例程序	(129)
3.6	自制 Message Mapping(信息映射)	(129)
	Frame7 范例程序	(138)
3.7	自制 Message Routing(信息传递)	(148)
	Frame8 范例程序	(157)

第二篇 欲善工事先利其器——Visual C++v4.0 开发工具

第4章	Visual C++ 软件集成开发环境	(172)
4.1	安装与组成	(172)
4.2	最重要的四种工具	(178)
4.3	内务部总管: Visual C++ 集成开发环境	(179)
	关于 project	(180)
	关于工具设定	(181)
	Source Code Browser	(182)
	Online Help	(184)
	调试工具	(186)
	• VC++ 调试器	(187)
	• Exception Handling	(190)
4.4	程序编码生成器: App Wizard	(191)
	东圈西点完成程序框架	(192)
	Scribble Step0	(202)
4.5	威力强大的资源编辑器	(221)
	Icon 编辑器	(223)
	Cursor 编辑器	(223)
	Bitmap 编辑器	(224)
	Toolbar(工具栏)编辑器	(224)
	VERSIONINFO 资源编辑器	(225)
	String Table(串资源)编辑器	(226)
	Menu(菜单)编辑器	(227)
	Accelerator(加速键)编辑器	(228)
	Dialog(对话框)编辑器	(229)

第三篇 浅出 MFC 程序设计

第5章	纵观 Application Framework	(232)
5.1	什么是 Application Framework?	(232)
5.2	侯捷怎么说	(232)
5.3	我怎么说	(234)

5.4	别人怎么说	(236)
5.5	为什么使用 Application Framework	(238)
5.6	Microsoft Foundation Classes(MFC)	(240)
5.7	白头宫女话天宝: Visual C++ 与 MFC	(242)
5.8	纵观 MFC	(243)
	General Purpose classes	(243)
	Windows API classes	(245)
	Application framework classes	(246)
	High level abstractions	(246)
	Afx 全局函数	(247)
第 6 章	MFC 程序设计导论——MFC 程序的生与死	(248)
6.1	不二法门: 熟记 MFC 类的层次结构	(250)
6.2	需要什么函数库(.LIB)?	(251)
6.3	需要什么头文件(.H)?	(252)
6.4	简化的 MFC 程序结构——以 HelloMFC 为例	(253)
6.5	Hello 程序源代码	(255)
6.6	MFC 程序的来龙去脉	(259)
	我只借用两个类: CWinApp 和 CFrameWnd	(259)
	CWinApp——取代 WinMain 的地位	(260)
	CFrameWnd——取代 Wndproc 的地位	(262)
	引爆器——Application object	(263)
	隐晦不明的 WinMain	(265)
	AfxWinInit——AFX 内部初始化操作	(268)
	CWinApp::InitApplication	(270)
	CMyWinApp::InitInstance	(271)
	CFrameWnd::Create 产生主窗口(并先注册窗口类)	(272)
	窗口显示与更新	(281)
	CWinApp::Run——程序生命的源泉	(282)
	把信息与处理函数联结在一起: Message Map 机制	(284)
	来龙去脉总汇	(286)
6.7	Callback 函数	(287)
6.8	Dialog 与 Control	(290)
6.9	通用对话框(Common Dialogs)	(291)
	本章回顾	(292)
第 7 章	简单而完整: MFC 程序框架	(295)
7.1	不二法门: 熟记 MFC 类的层次结构	(293)
7.2	MFC 程序的 UI 新风貌	(294)
7.3	Document/View 支撑你的应用程序	(399)
7.4	利用 Visual C++ 工具完成 Scribble step0	(302)
7.5	程序框架使用哪些 MFC 类?	(302)

7.6	Document Template 的意义	(307)
7.7	Scribble 的 Document/View 设计	(312)
7.8	主窗口的诞生	(313)
7.9	工具栏和状态栏的诞生	(315)
7.10	拖曳鼠标(Drag and Drop)	(317)
7.11	信息映射(Message Map)	(318)
7.12	标准菜单 File/Edit/View/Window/Help	(319)
7.13	对话框	(321)
7.14	改用 CEditView	(321)

第四篇 深入 MFC 程序设计

第 8 章	Document-View 深入探讨	(326)
8.1	为什么需要 Document-View(形式上)	(326)
	Document	(327)
	View	(328)
	Document Frame(View Frame)	(328)
	Document Template	(329)
	CDocTemplate 管理 CDocument/CView/CFrameWnd	(329)
8.2	Scribble Step1 的 Document——数据结构设计	(335)
	MFC Collection Classes 的选用	(336)
	• Template-Based Classes	(337)
	• Template-Based Classes 的使用方法	(338)
	C ScribbleDoc 的修改	(339)
	• SCRIBBLEDOC.H(阴影表示与 Step0 的差异)	(340)
	• SCRIBBLEDOC.CPP(阴影表示与 Step0 的差异)	(342)
	文件:一连串的线条	(346)
	• CScribbleDoc 的成员变量	(346)
	• CObList	(347)
	• CScribbleDoc 的成员函数	(347)
	线条与坐标点	(349)
	• CStroke 的成员变量	(349)
	• CArray<CPoint,CPoint>	(349)
	• CStroke 的成员函数	(349)
8.3	Scribble Step1 的 View:文件重绘与编辑	(351)
	C ScribbleView 的修改	(352)
	• SCRIBBLEVIEW.H(阴影表示与 Step0 的差异)	(352)
	• SCRIBBLEVIEW.CPP(阴影表示与 Step0 的差异)	(353)
	View 的重绘动作:GetDocument 和 OnDraw	(357)
	• CScribbleView 的成员变量	(357)
	• CScribbleView 的成员函数	(357)

View 与用户的会话(鼠标信息处理实例).....	(358)
ClassWizard 的帮助	(359)
WizardBar 的帮助	(361)
8.4 Serialize:对象的文件读写	(362)
Serialization 以外的文件读写动作.....	(362)
台面上的 Serialize 动作	(364)
台面下的 Serialize 写文件奥秘	(368)
台面下的 Serialize 读文件奥秘	(374)
8.5 DYNAMIC/DYNCREATE/SERIAL 三组宏定义	(381)
8.6 Serializable 的必要条件	(385)
8.7 CObject 类	(387)
IsKindOf	(387)
IsSerializable	(388)
CObject::Serialize	(388)
8.8 CArchive 类	(389)
operator<<和 operator>>.....	(390)
效率考虑	(394)
8.9 自定义 SERIAL 宏给抽象类使用	(394)
8.10 在 CObList 中加入 CStroke 以外的类	(395)
8.11 Document 与 View 交流-为 Scribble Step4 做准备	(399)
第 9 章 信息映射与命令路由	(402)
9.1 到底要解决什么	(402)
9.2 信息分类	(403)
9.3 万流归宗 Command Target(CCcmdTarget)	(404)
9.4 三个奇怪的宏,一张巨大的网.....	(404)
DECLARE_MESSAGE_MAP 宏	(405)
信息映射网的形成:BEGIN./ON./END. 宏	(47)
9.5 米诺托斯(Minotaurus)与西修斯(Theseus).....	(411)
9.6 两万五千里长征——信息路由(Message Routing).....	(415)
直线上溯(一般 Windows 信息).....	(417)
拐弯上溯(WM_COMMAND 命令信息)	(421)
9.7 罗塞达碑石 Rosetta stone	(427)
9.8 Scribble Step2:UI 对象的变化	(432)
改变菜单	(432)
改变工具栏	(435)
利用 ClassWizard 连接命令项识别码与命令处理函数	(437)
维护 UI 对象状态(UPDATE_COMMAND_UI)	(440)
本章回顾	(444)
第 10 章 MFC 与对话框	(445)
10.1 对话框编辑器.....	(446)

10.2	利用 ClassWizard 连接对话框与其专属类	(448)
	PENDLG.H	(451)
	PENDLG.CPP	(452)
10.3	对话框的信息处理函数	(454)
	MFC 中各式各样的 MAP	(456)
10.4	对话框数据交换与检查(DDX&DDV)	(457)
	MFC 中各式各样的 DDx_函数	(461)
10.5	如何使用对话框	(462)
	本章回顾	(464)
第 11 章	View 功能的加强与重绘效率的提高	(465)
11.1	同时修改多个 Views-UpdateAllViews 和 OnUpdate	(466)
	在 View 中定义一个 hint	(468)
	把 hint 传给 OnUpdate	(471)
	利用 hint 提高重绘效率	(473)
11.2	可滚动的窗口: CScrollView	(475)
11.3	大窗口中的小窗口: Splitter	(483)
	划分窗口的功能	(483)
	划分窗口的程序概念	(484)
	划分窗口的实作	(485)
	本章回顾	(488)
第 12 章	打印与预览	(489)
12.1	概述	(489)
12.2	打印动作的原理	(492)
12.3	MFC 预设的打印机制	(497)
12.4	Scribble 打印机制的增强	(509)
	打印机的页和文件的页	(509)
	配置 GDI 绘图工具	(510)
	尺寸与方向: 关于映像模式(坐标系统)	(511)
	分页	(514)
	表头与页尾	(516)
	动态计算页码	(517)
12.5	打印预览(Print Preview)	(518)
	本章回顾	(518)
第 13 章	多重文件与多重视图	(520)
13.1	MDI 和 SDI	(520)
13.2	多重视图(Multiple Views)	(521)
13.3	窗口的动态划分	(522)
13.4	窗口的静态划分	(524)
	CreateStatic 和 CreateView	(526)
13.5	窗口的静态三叉划分	(527)

Graph 范例程序	(529)
静态划分窗口的概念整理	(539)
13.6 同源子窗口	(540)
CMDIFrameWnd::OnWindowNew	(540)
Text 范例程序	(542)
非制式作法的缺点	(548)
13.7 多重文件	(550)
新的 Document 类	(550)
新的 Document Template	(552)
新的 UI 系统	(553)
新文件的文件读写操作	(555)

第五篇 附 录

附录 A Scribble Step5 完整源代码	(558)
附录 B Visual C++ 范例程序一览	(601)
附录 C OWL 程序设计一览	(605)

第 0 章

你一定要知道(导读)

0.1 这本书适合谁

深入浅出 MFC 是一本介绍 MFC(Microsoft Foundation Classes)程序设计技术的书籍。对于对 Windows 应用软件的开发感到有兴趣,想使用 Visual C++ 集成环境的可视开发工具,并以 MFC 为程序基础的人,都可以从此书获得最根本最重要的知识与实例。

如果你是一位对 Application Framework 和面向对象概念感兴趣的技术狂热分子,想知道神秘的 Runtime Type Information、Dynamic Creation、Persistence、Message Mapping 以及 Command Routing 如何实现,本书也能够满足你。事实上,依我之见,这些核心技术与彻底学会使用 MFC 是同一件事情。

全书共分为四篇:

第一篇 勿在浮砂筑高台——本书技术前提:它提供进入 MFC 核心技术以及应用技术之前的所有技术基础,包括:

- Win32 程序概念:Message Based、Event Driven、Multi-Tasking、Multi-Threading。
- C++ 重要技术:类与对象、this 指针与继承、静态成员、虚函数与多态性、模板类别、异常处理。
- MFC 六大技术之简化仿真(Console 程序)

第二篇 欲善工事先利其器——Visual C++ v4.0 开发工具:它提供给对 Visual C++ 集成环境全然陌生的朋友。这一篇当然不能取代 Visual C++ User's Guide 的地位,但对整个软件开发环境有全面以及概括性的介绍,可以让初学者迅速了解手上掌握的工具,以及它们的主要功能。

第三篇 浅出 MFC 程序设计:它介绍一个 MFC 程序的生与死。已经有 MFC 程序设计经验的朋友,不见得不会对本篇感到惊奇。根据我的了解,太多人使用 MFC 是“只知道这么做,不知道为什么”;本篇详细解释 MFC 程序的来龙去脉,为初入 MFC 领域的读者奠定扎实的基础。说不定本篇会让你醍醐灌顶。

第四篇 深入 MFC 程序设计:它介绍各式各样 MFC 技术。“只知其然,不知其所以然”的不良副作用,在程序设计的意图进一步展开之后,会愈来愈严重。最终会行不通!那些最困扰我们的 MFC 宏、MFC 常量定义,不得一窥真面目的 MFC 黑箱作业在本篇将陆续曝光。本篇将使您高喊:Eureca!

阿基米德在洗澡时发现浮力原理,高兴得不及穿上裤子,跑到街上大喊:
Eureca(我找到了)。

范例程序方面,第三章有数个 Console 程序(俗称 DOS 程序,在 Windows 系统的 DOS Box 中执行),简化仿真 Application Framework 六大核心技术。另外,全书以一个循序渐进的 Scribble 程序(Visual C++ 所附程序),从第七章开始,分章探讨每一个 MFC 应用技术主题。最后一章另有三个程序,示范 Multi-View 和 Multi-Document 的情况。

0.2 你需要什么技术基础

从什么技术层面进入 Windows 软件开发领域? C/SDK? 抑或 C++/MFC? 这一直是个有争议的论题。就我个人观点,使用 C++/MFC 必须跨越四大技术障碍:

1. 面向对象概念与 C++ 语言。
2. Windows 程序基本概念(程序入口、信息循环、窗口函数、callback...)
3. Microsoft Foundation Classes(MFC)本身。
4. Visual C++ 集成环境与各种开发工具(难度不高,但需多练习)。

换言之,如果你从未接触 C++,千万不要阅读本书,那只会打击你学习新技术的信心而已。如果已接触过 C++ 但不十分熟悉,你可以一边复习 C++ 一边学习 MFC,这也是我所鼓励的方式。C++ 语言的继承(inheritance)特性对于我们使用 MFC 尤为重要,因为使用 MFC 就是要继承各个类并为己用。所以,你应该对 C++ 的继承特性(以及虚函数,当然)多加体会。我在第 2 章安排了一些 C++ 的必要基础。我所挑选的题目都是本书会用到的。

如果你有了 C++ 语言基础,但从未接触过 Win16(或 Win32)程序设计,只在 DOS 环境下开发过软件,我在第 1 章为你安排了一些 Win32 程序设计基础。这个基础至为重要,只会在各个 Wizard(或 Expert)上按来按去,却不懂所谓 message loop 与 window procedure 的人,不可能搞定 Windows 程序设计——不管你用的是 MFC 或 OWL 或 OpenClass,是 Visual C++ 或 Borland C++ 或 Optima++。

0.3 你需要什么样的软硬件环境

一套 Windows 95(或 Windows NT)操作系统当然是必须的,中英文皆可。此外,你需要一套 Visual C++ 32 位版。目前的最新版本是 Visual C++4.0,也是我使用的版本。

硬件方面,只要能使用上述两种软件系统的主机就算过关。内存(RAM)是影响运行速度的主要原因,多多益善。厂商宣称 16MB RAM 是一个能够使你工作舒适的内存大小,但我却怀疑“舒适”这个字眼的定义。写作本书时我的软硬件环境是:

- Pentium 133
- 48MB RAM
- 2GB 硬盘
- 17 英寸显示器。别以为显示器和程序设计没有关系。大尺寸屏幕使我们一次看多一点东西,不必在 Visual C++ 集成环境所提供的密密麻麻的画面上找来找去。
- Windows 95 (中文版)
- Visual C++4.0

0.4 让我们使用同一种语言

要在电脑书籍不可或缺的英文术语与流利顺畅的中文解说之间保持平衡,是多么不容易的一件事。我曾经以为我通过了最大的考验,但每次总有新鲜事发生。是该叫 class 好呢? 还