

# 第 1 版序

有一种软件名曰 `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、检查、回复过去努力的轨迹。

遗憾的是人的大脑没有 `version control` 的能力。学习过程的彷徨犹豫、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。作为私人“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，作为资讯教育体系一员的我，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，物件导向的观念与 `application framework` 的包装是其四。初学 MFC programming 时，我的脑袋犹如网目过大的筛子，什么东西都留不住；各个类别及其代表意义，过眼即忘。

初初接触 MFC 时，我对 Windows 作业系统以及 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 *Inside Visual C++ 4.0* (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 应用程序基本架构的每一个技术环节的深入探讨，其中牵扯到 MFC 本身的设计原理、物件

导向的观念, 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 骨干程序的每一个基础动作弄懂, 甚至观察其原始码, 才能实实在在掌握 MFC 这一套 application framework 的内涵, 及其物件导向的精神。我向来服膺一句名言: 原始码说明一切, 所以, 我挖 MFC 原始码给你看。

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 程序的基本观念以及 C++ 的高阶议题。“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++ 整合环境开发工具。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++ 整合环境。工具的使用虽然谈不上学问, 但在视觉化软体开发过程中扮演极重角色, 切莫小觑它。

第三篇介绍 application framework 的观念, 以及 MFC 骨干程序。所谓骨干程序, 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++ 光碟片上的一个范例程序 Scribble 为主轴, 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。

本书前身,1994/08 出版的 **Visual C++ 物件导向 MFC 程序设计 基础篇**以及 1995/04 年出版的应用篇,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明日黄花:Microsoft Systems Journal 1995/07 的一篇由 Paul Dilascia 所撰的文章 *Meandering Through the Maze of MFC Message and Command Routing*, 以及 Addison Wesley 于 1996/06 出版的 *MFC Internals* 一书,也有了相当程度的核心涉猎,即连前面提及的 *Programming Windows 95 with MFC* 以及 *Inside Visual C++ 4.0* 两本书,也都多多少少开始涉及 MFC 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,填补了上一版的缝隙。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++ 的高阶技术。至此,整个 MFC 的基础架构已经完全暴露在你的掌握之中,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,再至物件导向的精神,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?探究原始码,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巨集和指令搞清楚,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,适足作为我写作本书的心境,同时也对我与朋友之间的讨论作个总结:

“只用一样东西,不明白它的道理,实在不高明”。

祝各位:胸中丘壑自成!

侯俊杰 新竹 1996.08.15

P.S. 愈来愈多的朋友在网路上与我打招呼，闲聊谈心。有医师、盲生、北京的作家、香港的读者、从国中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，读者群只是一个想象空间，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。

## 第 2 版序

任何人看到前面的读者来函，都会感动于一本电脑书籍的读者与作者竟然能够产生如此深厚的共鸣，以及似无若有的长期情感。

何况，身为这本书的作者的我！

我写的是一本技术书籍，但是赢得未曾谋面的朋友们的信赖与感情。我知道，那是因为这本书里面也有许多我自己的感情。每当收到读友们针对这本书寄来的信件（纸张的或电子的），我总是怀着感恩的心情仔细阅读。好几位读友，针对书中的可疑处或是可以更好的地方，不吝啬地拨出时间，写满一大张一大张的信纸，一一向我指正。我们谈的不只是技术，还包括用词遣字的意境。新竹刘嘉均先生和加拿大陈宗泰先生给我非常宝贵的技术上的意见。陈先生甚至在一个月内来了五封航空信。

这些，怎不教我心怀感谢，并且更加戒慎恐惧！

感谢所有读者促成这本书的更精致化。Visual C++ 5.0 面世了，MFC 则停留在 4.2，程序设计的主轴没有什么大改变。对于新读者，本书乃全新产品自不待言，您可以从目录中细细琢磨所有的主题。对于老读者，本书所带给您的，是更精致的制作，以及数章新增的内容（请看第 0 章“与前版本之差异”）。

最后，我要说，我知道，这本书真的带给许多人很扎实的东西。而我之所以愿意不计代价去做些不求近利的深耕工作，除了这是身为专业作家的责任，以及个人的兴趣之外，是的，我自己是工程师，我最清楚工程师在学习 MFC 时想知道什么、在哪里触礁。

所有出自我笔下的东西，我自己受益最丰。

感谢你们。

侯俊杰 台湾.新竹 1998.09.11

[jjhou@ccca.nctu.edu.tw](mailto:jjhou@ccca.nctu.edu.tw)  
<http://www.jjhou.com>

第 0 章	你一定要知道 (导读)	/1
	这本书适合谁	/1
	你需要什么技术基础	/2
	你需要什么软硬件环境	/3
	让我们使用同一种语言	/3
	本书符号习惯	/5
	本书例程的取得	/6
	范例程序说明	/6
	与前版本之差异	/9
	如何联络作者	/10
第一篇	勿在浮砂筑高台	/11
第 1 章	Win32 程序基本概念	/13
	Win32 程序开发流程	/14
	需要什么函数库 (.LIB)	/14
	需要什么头文件 (.H)	/15
	以消息为基础, 以事件驱动之 (message based, event driven)	/15
	一个具体而微的 Win32 程序	/16
	程序进入点 WinMain	/22
	窗口类之注册与窗口之诞生	/23
	消息循环	/24
	窗口的生命中枢: 窗口函数	/24
	消息映射 (Message Map) 的雏形	/25
	对话框的运行	/27
	模块定义文件 (.DEF)	/28
	资源描文件 (.RC)	/29
	Windows 程序的生与死	/29
	空闲时间的处理: OnIdle	/30
	Console 程序	/30
	Console 程序与 DOS 程序的差别	/31
	Console 程序的编译链接	/32
	JBACKUP: Win32 Console 程序设计	/33
	MFCCON: MFC Console 程序设计	/35
	进程与线程 (Process and Thread)	/38
	核心对象	/38
	一个进程的诞生与死亡	/39
	产生子进程	/39
	一个线程的诞生与死亡	/41

以 `_beginthreadex` 取代 `CreateThread` /42

线程优先级 (Priority) /44

多线程程序设计实例 /45

## 第 2 章 C++ 的重要性质 /49

类及其成员——谈封装 (encapsulation) /49

基类与派生类：谈继承 (Inheritance) /50

`this` 指针 /53

虚函数与多态 (Polymorphism) /53

类与对象大解剖 /64

Object slicing 与虚函数 /68

静态成员 (变量与函数) /70

C++ 程序的生与死：兼谈构造函数与析构函数 /72

四种不同的对象生存方式 (in stack、in heap、global、local static) /74

运行时类型识别类型信息 (RTTI) /75

动态创建 (Dynamic Creation) /77

异常处理 (Exception Handling) /78

Template /81

Template Functions /82

Template Classes /83

Templates 的编译与链接 /85

## 第 3 章 MFC 六大关键技术之仿真 /87

MFC 类层次结构 /88

Frame1 范例程序 /89

MFC 程序的初始化过程 /91

Frame2 范例程序 /93

RTTI (运行时类型识别) /96

类别型录网与 `CRuntimeClass` /97

`DECLARE_DYNAMIC` / `IMPLEMENT_DYNAMIC` 宏 /98

Frame3 范例程序 /103

`IsKindOf` (类型识别) /109

Frame4 范例程序 /110

Dynamic Creation (动态创建) /111

`DECLARE_DYNCREATE` / `IMPLEMENT_DYNCREATE` 宏 /112

Frame6 范例程序 /117

Persistence (永久保存) 机制 /124

`Serialize` (数据读写) /124

---

	DECLARE_SERIAL / IMPLEMENT_SERIAL 宏 /129
	没有范例程序 /131
	Message Mapping (消息映射) /132
	Frame7 范例程序 /139
	Command Routing (命令传递) /147
	Frame8 范例程序 /155
	本章回顾 /165
<b>第二篇</b>	<b>欲善工事先利其器 /167</b>
<b>第 4 章</b>	<b>Visual C++ 集成开发环境 /169</b>
	安装与组成 /169
	四个重要的工具 /179
	内务府总管: Visual C++ 集成开发环境 /180
	关于 project /182
	关于工具设定 /185
	Source Browser /186
	Online Help /189
	调试工具 /191
	VC++调试器 /192
	Exception Handling /195
	程序代码产生器: AppWizard /196
	东圈西点完成 MFC 程序骨干 /197
	威力强大的资源编辑器 /224
	Icon 编辑器 /225
	Cursor 编辑器 /226
	Bitmap 编辑器 /226
	工具栏 (Toolbar) 编辑器 /227
	VERSIONINFO 资源编辑器 /228
	字符串表格 (String Table) 编辑器 /229
	菜单 (Menu) 编辑器 /230
	加速键 (Accelerator) 编辑器 /231
	对话框 (Dialog) 编辑器 /231
	Console 程序的项目管理 /232
<b>第三篇</b>	<b>浅出 MFC 程序设计 /235</b>
<b>第 5 章</b>	<b>总观 Application Framework /237</b>
	什么是 Application Framework? /237
	侯捷怎么说 /238

---

我说 /239
别人怎么说 /241
为什么使用 Application Framework /244
Microsoft Foundation Classes (MFC) /246
白头宫女话天宝: Visual C++ 与 MFC /248
纵览 MFC /250
General Purpose classes /250
CObject /251
数据处理类 (collection classes) 251
杂项类 /251
异常处理类 (exception handling classes) /252
Windows API classes /252
Application framework classes /253
High level abstractions /254
Afx 全局函数 /254
MFC 宏 (macros) /255
MFC 数据类型 (data types) /256
<b>第 6 章 MFC 程序的生死因果 /259</b>
不二法门: 熟记 MFC 类的层次结构 /261
需要什么函数库? /262
需要什么头文件? /263
简化的 MFC 程序结构——以 Hello MFC 为例 /264
Hello 程序程序代码 /265
MFC 程序的来龙去脉 (causal relations) /270
我只借用两个类: CWinApp 和 CFrameWnd /270
CWinApp——取代 WinMain 的地位 /270
CFrameWnd——取代 WndProc 的地位 /273
引爆器——Application object /274
隐晦不明的 WinMain /275
AfxWinInit——AFX 内部初始化操作 /278
CWinApp::InitApplication /279
CMyWinApp::InitInstance /280
CFrameWnd::Create 产生主窗口 (并先注册窗口类) /281
奇怪的窗口类名称 Afx:b:14ae:6:3e8f /289
窗口显示与更新 /291
CWinApp::Run——程序生命的活水源头 /291
把消息与处理函数连接在一起: Message Map 机制 /294

---

来龙去脉总整理 /296
Callback 函数 /297
空闲时间 (idle time) 的处理: OnIdle /300
Dialog 与 Control /302
通用对话框 (Common Dialogs) /303
本章回顾 /304
<b>第 7 章 简单而完整: MFC 骨干程序 /307</b>
不二法门: 熟记 MFC 类层次结构 /307
MFC 程序的 UI 新风貌 /307
Document/View 支撑你的应用程序 /312
利用 Visual C++ 工具完成 Scribble step0 /314
骨干程序使用哪些 MFC 类? /315
Document Template 的意义 /320
Scribble 的 Document/View 设计 /324
主窗口的诞生 /325
工具栏和状态栏的诞生 (Toolbar & Status bar) /327
鼠标拖放 (Drag and Drop) /329
消息映射 (Message Map) /331
标准菜单 File / Edit / View / Window / Help /331
对话框 /333
改用 CEditView /334
<b>第四篇 深入 MFC 程序设计 /337</b>
<b>第 8 章 Document-View 深入探讨 /339</b>
为什么需要 Document-View (形而上) /339
Document /340
View /341
Document Frame (View Frame) /342
Document Template /342
CDocTemplate 管理 CDocument / CView / CFrameWnd /342
Scribble Step1 的 Document——数据结构设计 /349
MFC Collection Classes 的选用 /349
CScribbleDoc 的修改 /352
文件: 一连串的线条 /359
线条与坐标点 /361
Scribble Step1 的 View: 数据重绘与编辑 /363
CScribbleView 的修改 /364

- View 的重绘操作: GetDocument 和 OnDraw /368
- ClassWizard 的辅佐 /370
- WizardBar 的辅佐 /372
- Serialize: 对象的文件读写 /372
  - Serialization 以外的文件读写操作 /372
  - 台面上的 Serialize 操作 /374
  - 台面下的 Serialize 写文件奥秘 /379
  - 台面下的 Serialize 读文件奥秘 /383
- DYNAMIC / DYNCREATE / SERIAL 三宏 /389
- Serializable 的必要条件 /394
- CObject 类 /395
  - IsKindOf /395
  - IsSerializable /396
  - CObject::Serialize /397
- CArchive 类 /397
  - operator<< 和 operator>> /398
  - 效率考虑 /401
- 自定义 SERIAL 宏给抽象类使用 /401
- 在 CObList 中加入 CStroke 以外的类 /402
- Document 与 View 交流——为 Step4 做准备 /406
  
- 第 9 章 消息映射与命令传递 /409**
  - 到底要解决什么 /409
  - 消息分类 /410
  - 万流归宗 Command Target (CCmdTarget) /411
  - 三个奇怪的宏, 一张巨大的网 /412
    - DECLARE\_MESSAGE\_MAP 宏 /413
    - 消息映射网的形成: BEGIN.../ON.../END... 宏 /414
  - 米诺托斯 (Minotaurus) 与西修斯 (Theseus) /418
  - 二万五千里长征——消息的传递 /422
    - 直线上溯 (一般 Windows 消息) /423
    - 拐弯上溯 (WM\_COMMAND 命令消息) /426
    - 罗塞达碑石: AfxSig\_xx 的奥秘 /432
  - Scribble Step2: UI 对象的变化 /436
    - 改变菜单 /436
    - 改变工具栏 /438
    - 利用 ClassWizard 连接命令项识别代码与命令处理函数 /440
    - 维护 UI 对象状态 (UPDATE\_COMMAND\_UI) /443

- 
- 本章回顾 /446
  
  - 第 10 章 MFC 与对话框 /447**
    - 对话框编辑器 /448
    - 利用 ClassWizard 连接对话框与其专用类 /451
    - 对话框的消息处理函数 /456
    - 对话框数据交换与校验 (DDX & DDV) /458
    - 如何调用对话框 /462
    - 本章回顾 /464
  
  - 第 11 章 View 功能的加强与重绘效率的提高 /467**
    - 同时修改多个 Views: UpdateAllViews 和 OnUpdate /468
      - 在 View 中定义一个 hint /470
      - 把 hint 传给 OnUpdate /473
      - 利用 hint 增加重绘效率 /474
    - 可滚动的窗口: CScrollView /476
    - 大窗口中的小窗口: Splitter /484
      - 拆分窗口的功能 /484
      - 拆分窗口的程序概念 /484
      - 拆分窗口的实现 /486
    - 本章回顾 /488
  
  - 第 12 章 打印与预览 /491**
    - 概述 /491
    - 打印操作的后台原理 /494
    - MFC 默认的打印机制 /498
    - 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 /511
      - 尺寸与方向: 关于映射方式 (坐标系统) /511
      - 分页 /514
      - 页眉与页脚 /516
      - 动态计算页码 /517
    - 打印预览 (Print Preview) /517
    - 本章回顾 /518
  
  - 第 13 章 多重文件与多重视图 /519**
    - MDI 和 SDI /519

- 多重视图 (Multiple Views) /520
- 窗口的动态拆分 /521
- 窗口的静态拆分 /523
  - CreateStatic 和 CreateView /524
- 窗口的静态三叉拆分 /526
  - Graph 范例程序 /527
  - 静态拆分窗口之观念整理 /537
- 同源子窗口 /537
  - CMDIFrameWnd::OnWindowNew /538
  - Text 范例程序 /539
  - 非标准做法的缺点 /544
- 多重文件 /545
  - 新的 Document 类 /545
  - 新的 Document Template /547
  - 新的 UI 系统 /548
  - 新文件的读写操作 /549
- 第 14 章 MFC 多线程程序设计 /553**
  - 从操作系统层面看线程 /553
    - 三个观念：模块、进程和线程 /553
    - 线程优先级 (Priority) /555
    - 线程调度 (Scheduling) /557
    - Thread Context /557
  - 从程序设计层面看线程 /558
    - Worker Threads 和 UI Threads /559
    - 错误观念 /559
    - 正确态度 /559
  - MFC 多线程程序设计 /560
    - 探索 CWinThread /560
    - 产生一个 Worker Thread /563
    - 产生一个 UI Thread /564
    - 线程的结束 /565
    - 线程与同步控制 /565
  - MFC 多线程程序例程 /568
- 第 15 章 定制一个 AppWizard /571**
  - 到底 Wizard 是什么? /573
  - Custom AppWizard 的基本操作 /573

- 剖析 AppWizard Components /577
  - Dialog Templates 和 Dialog Classes /578
  - Macros /579
  - Directives /580
- 动手修改 Top Studio AppWizard /581
  - 利用资源编辑器修改 IDD\_CUSTOM1 对话框画面 /581
  - 利用 ClassWizard 修改 IDD\_CUSTOM1 对话框的对应类 CCustom1Dlg /582
  - 改写 OnDismiss 虚函数，在其中定义 macros /583
  - 修改 text template /584
  - Top Studio AppWizard 执行结果 /584
  - 更多的信息 /585

## 第 16 章 站上众人的肩膀——使用 Components & ActiveX Controls /587

- 什么是 Component Gallery /587
- 使用 Components /590
  - Splash screen /590
  - System Info for About Dlg /592
  - Tip of the Day /593
  - Components 实际运用：ComTest 程序 /594
  - 修改 ComTest 程序内容 /608
- 使用 ActiveX Controls /611
  - ActiveX Control 基础观念：Properties、Methods、Events /611
  - ActiveX Controls 的五大使用步骤 /613
  - 使用 Grid ActiveX Control：OcxTest 程序 /614

## 第五篇 附录 /627

- 附录 A 无责任书评：从摇篮到坟墓 Windows 的完全学习 /629
  - 无责任书评：MFC 四大天王 /637
- 附录 B Scribble Step 5 完整原始码 /651
- 附录 C Visual C++ 5.0 MFC 范例程序一览 /683
- 附录 D 以 MFC 重建 DBWIN /689

## 第 0 章

# 你一定要知道 (导读)

## 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 Windows 应用软件的开发感到兴趣, 并欲使用 Visual C++ 可视化集成开发工具, 以 MFC 为程序基础的人, 都可以从此书中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 (Object Oriented) 观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 如何实作, 本书也能够充分满足你的需要。事实上, 依我之见, 这些核心技术与彻底学会操控 MFC 乃同一件事情。

全书分为四篇:

第一篇【勿在浮砂筑高台】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 程序观念: message based, event driven, multitasking, multithreading, console programming。
- 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、模板 (template) 类、异常处理 (exception handling)。
- MFC 六大技术之简化仿真 (Console 程序)。

第二篇【欲善工事先利其器】提供给对 Visual C++ 集成开发环境全然陌生的朋友一个导引。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盘性的介绍, 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】介绍一个 MFC 程序的生死因果。已经有 MFC 程序经验的朋友, 不见得不会对本篇感到惊艳。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 程序之来龙去脉, 为初入 MFC 领域的读者奠定扎实的基础。说不定本篇会让你有醍醐灌顶之感。

第四篇【深入 MFC 程序设计】介绍各式各样的 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开展之后，愈来愈严重，最终会行不得也！那些最困扰我们的 MFC 宏、MFC 常数定义，难得一窥奥妙的 MFC 黑箱操作，将在本篇陆续曝光。本篇将使您高喊：Eureka！

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

范例程序方面，第 3 章有数个 Console 程序（DOS-like 程序，在 Windows 系统的 DOS Box 中执行），仿真并简化 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++ 所附范例），从第 7 章开始，分章探讨每一个 MFC 应用技术主题。第 13 章另有三个程序，示范 Multi-View 和 Multi-Document 的情况。14 章~16 章是第二版新增内容，主题分别是 MFC 多线程程序设计、Custom AppWizard，以及如何使用 Component Gallery 提供的 ActiveX controls 和 components。

## 你需要什么技术基础

从什么技术层面切入 Windows 软件开发领域？C/SDK？抑或 C++/MFC？这一直是个引起争议的论题。就我个人观点，C++/MFC 程序设计必须跨越四大技术障碍：

1. 面向对象观念与 C++ 语言。
2. Windows 程序基本观念（程序进入点、消息传递、窗口函数、callback...）。
3. Microsoft Foundation Classes (MFC) 本身。
4. Visual C++ 集成开发环境与各种开发工具（难度不高，但需熟练）。

换言之，如果你从未接触 C++，则千万不要阅读本书，那只会打击你学习新技术的信心。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（很多人是为了使用 MFC 而去学习 C++ 的）。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技术，而其深度你不见得能够在一般的 C++ 书籍中发现。

如果你有 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 Wizards 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、OWL 或 Open Class Library，不管你用的是 Visual C++ 或 Borland C++ 或 VisualAge C++。

## 你需要什么软硬件环境

一套 Windows 9x (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。本书配套的版本是 Visual C++ 5.0, 也是我使用的版本。

硬件方面, 只要能运行上述两种操作系统就算过关。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数字, 但我因此怀疑“舒适”这个字眼的定义。写作本书时我的软硬件环境是:

- Pentium 133
- 96MB RAM
- 2GB 硬盘
- 17 吋显示器。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++ 集成开发环境所提供的密密麻麻的画面上滚来滚去。
- Windows 9x (中文版)
- Visual C++ 5.0

## 让我们使用同一种语言

要在计算机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, 但每次总有新鲜事儿发生。是该叫 class 好呢? 还是叫“类”好? 该叫 object 好呢? 还是叫“对象”好? framework 难道该译为“框架”吗? Document 译为“文件”, 可也, 可 View 是什么东西? 我很伤脑筋耶。考虑了这本书的潜在读者所具备的技术基础与教育背景之后, 原谅我, 不喜欢在中文书中看到太多英文字的朋友, 你终究还是在这本书上看到了不少原文名词。只有已统一化、没有异议、可以望文生义的中文名词, 我才使用。

虽然许多名词已经耳熟能详, 我想我还是有必要把它们界定一下:

**API : Application Programming Interface.** 系统开放出来, 给程序员使用的接口, 就是 API。一般人的观念中 API 是指像 C 函数那样的东西, 不尽然! DOS 的中断向量 (interrupt vector) 也可以说是一种 API, OLE Interface (以 C++ 类的形式出现) 也可以说是一种 API。不是有人这么说吗: MFC 势将成为 Windows 环境上标准的 C++ API (我个人认为这句话已成为事实)。

**SDK : Software Development Kit,** 原指软件开发工具。每一套环境都可能有自己的 SDK, 例如 Phar Lap 的 386/DOS Extender 也有自己的 SDK。在 Windows 这一领域, SDK 原是指 Microsoft 的软件开发工具, 但现在已经变成一个专有名词。凡以 Windows raw API 撰写的程序我们通常也称为 SDK 程序。也有人把 Windows API 称为 SDK API。