

igned for
icrosoft®
indows NT®
indows 98



D-ROM
cluded



Microsoft® 程序设计系列

企业开发人员用书

Microsoft® **Visual Basic® 6.0** 高级编程 (第2版)

适用于专业及企业开
发人员的专家技术和
解决方案

北京大学出版社

Microsoft Press

Microsoft Visual Basic 6.0

高级编程

(第 2 版)

[英]TMS(国际)有限公司 著

北京博彦科技发展有限公司 译

北京大学出版社

·北 京·

著作权合同登记 图记: 01-1999-1920 号

Advanced Microsoft Visual Basic 6.0 (Second Edition)

The Mandelbrot Set (International) Limited

本书版权为 The Mandelbrot Set (International) Limited 所有, 1998。(Copyright © 1998 By The Mandelbrot Set (International) Limited. All rights reserved.)

本书中文版由美国 Microsoft 出版社授权北京大学出版社独家出版, 2000。

本书封面贴有北京大学出版社的激光防伪标签, 无标签者不得销售。

版权所有, 翻印必究。

图书在版编目(CIP)数据

Microsoft Visual Basic 6.0 高级编程: 第 2 版/[英]TMS(国际)有限公司著; 北京博彦科技发展有限公司译. —北京: 北京大学出版社, 2000. 1

ISBN 7-301-04188-8

I. M… II. ①T… ②北… III. BASIC 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(1999)第 31643 号

书 名: Microsoft Visual Basic 6.0 高级编程(第 2 版)

著作责任者: [英]TMS(国际)有限公司 著 北京博彦科技发展有限公司 译

责任编辑: 邱淑清 熊 勇 张 青

标准书号: ISBN 7-301-04188-8/TP·464

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/cbs.htm>

电 话: 出版部 62752015 发行部 62754140 总编室 62752032

电子信箱: zpup@pup.pku.edu.cn

排 印 者: 北京大学印刷厂

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 43 印张 948 千字

2000 年元月第一版 2000 年元月第一次印刷

定 价: 91.00 元(含光盘)

前言

自 1991 年创立以来,Microsoft Visual Basic 就彻底变革了开发者创建 Microsoft Windows 应用程序的方式。通过诸如 IntelliSense、语句完成、ActiveX 控件创建、本地代码编译器,以及数据库访问的一些特性,Microsoft 已致力于在快速的应用程序开发方面让开发者的最佳配置跟上最新的创新潮流。在 Visual Basic 6 专业版和企业版中,Visual Basic 已被扩展为使开发者在他们经常要做的事情——访问数据,无论是在相关的数据库中(如 Microsoft SQL Server 7 或 Oracle 8)还是在无关的数据存储中(如目录服务)——上更有能力。认可采用 Web 作为真实的跨平台应用程序环境是一个快速的飞跃,Microsoft 已经将新技术添加到支持客户机(Dynamic HTML)和服务端(WebClass)上。这些新特性与成百的其他增强和最优化的性能一起,促进了已掌握 Visual Basic 的数百万开发者所触及到的企业范围、商业领域的应用程序的发展。另外,这些特性还提供了创建运行紧急任务方案的应用程序所需的能力和性能。

The Mandelbrot Set (International) Limited(TMS)是推行 Visual Basic 最频繁的公司。在 Visual Basic 咨询和开发方面,TMS 一向被认为是世界一流的。现在我已与 TMS 合作了一段时间,并且已经看到了他们所给予的、在这个行业中不常见的深层次的技术论文、著作,以及演示。大约每个月一次或两次,我就从 Peter Morris(TMS 的技术主管)那里获得一些问题和见解,它们是如此的高深,足以与产品本身的主要设计者对该产品所能达到的正常理解相抗衡。

大约一年前在旧金山的 VBITS,我第一次和 Peter Morris 接触。我参加了 Peter 的会议,立即被他的学术技能所吸引,并认识到将来他将是一个无价的资源。我们立刻在学术上和个人方面相处得很好,并且从那时起他在会议上为我们作了多次讲话,如去年夏天在新奥尔良的 Microsoft TechEd 会议。大家对 Peter 演讲的技术内容给予了一致好评。我热切地希望 Peter 到 Redmond 访问,那时我们将讨论 Visual Basic 线性模型或数据函数的子类的复杂性,谈论出去来一品脱 Guinness 啤酒,还有诸如与朋友一起看 Seinfeld 的最后一段之类的琐细事情。

本书不是规范的、所谓在几天内就能从初学者成为专业程序员的课本,它也并未提

供 Visual Basic 6 所有新特性的完整清单。本书适用于那些专业人员,以探究那些在初级读物中或者被忽略或者未被涉及到的许多深刻的主题。由许多具有各种实力的 TMS 专家撰写,它的每一章都被细分成了在日常的工程开发过程中不被仔细考虑的主题。从本书中学到的课程都是 VBITS 研讨会上或教室外和讲台外(这些都是开发者通过开放式对话增长见识的地方)的技术课程中掘取的精华。

涉及尖端技术的开发需要大量的工作和能力,TMS 已经知道这是非常艰难的道路。本书能使您得益于他们的综合经验。如果您正在使用 Visual Basic 进行工作,并试图推动其发展,那么您将从本书中发现许多有用的信息。

Sean P. Alexander
Lead Technical Product Manager, Visual Basic
Developer Tools Division
Microsoft Corporation

序言

带

着编写一本不同于其他高级 Visual Basic 图书的想法,当最初与微软出版社接触时,我们想知道我们得到的反应。我们特别热衷于以下三种综合特征:

- ◆ 本书集中了许多个人的著作和看法。这些人也许观点并不一致。
- ◆ 通过介绍有关高级现实问题的材料,我们希望正致力于将此工具推进到极限的公司管理人员和开发人员能对此书真正地感兴趣。
- ◆ 通过内容和多位作者合作的写作风格,我们希望此书极富情趣,并具有很高的可读性。

我们期望每位作者都写出自己非常感兴趣的课题。这个课题必须与实际的编程问题相关。它还必须是从实际的、现实生活经验中提取出来的,这些经验来源于实际的开发和咨询——您将发现在本书中没有描述如何上下翻转一个窗口。

微软出版社接受了我们的想法,我们也很荣幸地得知本书的第一版 Visual Basic 5 销售得非常好。现在我们已经编写了 Visual Basic 6 的更新的第二版。

在 The Mandelbrot Set (International) Limited,或简称为 TMS,从版本 1 起我们就接受了 Visual Basic——实际上,Peter Morris 向公众作了此工具在世界上的第一次演示。我们是一家英国公司,我们成功地开发了 Visual Basic 3 和 4 的程序设计员工具包,包括 Micro Help's Code Complete。在我们的 ISV 阵地中,我们已加入了 Visual Basic 的早期测试程序,并成为欧洲第一个可访问到版本 6 的站点——我们已使用 Visual Basic 6 很长时间了!

我们还能做什么?我们花费了大量的时间来开发 Visual Basic 软件,并为大多数独特的客户提供咨询。通过这些年的努力,我们已经熟悉了使用 Visual Basic 的企业级开发的所有问题。我们还不时为 Visual Basic 方面的出版物编写大量文章,作了 VBITS 和 TechEd 等活动的演讲,并主持了高级 Visual Basic 培训讲座和研讨会。

由于我们的背景和经验,所以在编写权威的 Visual Basic 6 以及最重要的版本更新方面,我们占有重要的地位。

本书适用的读者

本书主要适合两种读者。负责计划、设计和支持 Visual Basic 应用程序的技术管理人员应该阅读此书,在阅读时应注意理解正在和即将管理的小组中,他们所能和不能预料的问题,还应注意理解 Visual Basic 6 所具有的能力。在项目组中负责计划、编程设计和实现 Visual Basic 应用程序的开发者应学习此书,来找到有关专家处理相同的但经常是复杂和难以解答的问题的方法,这些问题是他们可能经常遇到的(或已经把他们难倒了)的)。

所涉及的内容

本书涉及了大量广泛的 Visual Basic 主题内容。这是讨论的课题的一些简要示例:设计服务器端对象、开发 Microsoft Windows CE 程序、生成基础代码、混合语言编程、设计分布式应用程序、调试、迁移问题、错误处理、与 Visual Basic 6 和 Visual Basic for Applications 有关的 2000 年问题、软件测试、可重用性、成功的企业开发、招聘好的开发人员、为残疾人开发、编码约定,以及大量的关于 Visual Basic 开发的其他重要内容。没有真正的开始和结束——只需浏览每一章所涉及的内容即可。读完此书,您就会明白这么多神奇技术是如何产生的。

致谢

我们真诚地感谢以下人员:

整个 TMS 创作小组。他们牺牲了大量的个人时间,并且为在紧张的时间期限内交付高质量的作品而付出了巨大的努力。这里要特别感谢 Treesje Verlinden(项目经理)严厉地鞭策!

所有支持、容忍我们的 TMS 其余的创作小组!

Jim Brown、Steve Guty、Wendy Zucker、Marc Young、Rebecca McKay、Jean Ross、Bill Teel、Michael Victor、Travis Beaven,以及微软出版社的每一个人,感谢他们完成了如此高质量的工作。

感谢 Visual Basic 小组、Sean Alexander 和支持我们的每一个人。他们制作了真正伟大的产品——做得非常好!

Scott Swanson,感谢他最初建议写作这本书。

最后,但并不意味着最少,感谢 Helen、William、Katie、Amy、Fiona 和 Daniel 等人所有的关爱与支持。

最后

我们希望您能喜欢这本书,并从中获得知识。如果您想和某个章节的作者联系,可以按照该章开始处的电子邮件地址给他发送电子邮件。

Peter J. Morris 和 Mark Sewell
TMS 主管和创始人

目录

前言	1
序言	3
第一章 发生错误就下地狱	2
1.1 提示 1: 错误是如此反复,尽可能地模仿 VB 的错误处理方案	4
1.2 提示 2: 在源代码中使用行号	10
1.3 提示 3: 因为返回值将被忽略,所以应尽可能产生异常	12
1.4 提示 4: 自动记录关键的 MsgBox 错误	15
1.5 提示 5: 在每个例程中都有一个错误处理程序	17
1.6 提示 6: 编写有效的错误日志(如果可能,记录到集中的位置)	19
1.7 提示 7: 使用断言	25
1.8 提示 8: 不要盲目改进错误处理程序	26
1.9 提示 9: 跟踪堆栈	26
1.10 提示 10: 使用 ROOS(Resource Only Object Server)	27
1.11 提示 11: 用您自己的对象来替换有用的内部对象	27
1.12 提示 12: 检查 DLL 版本错误	32
1.13 提示 13: 尽可能使用 Microsoft 系统信息(MSINFO32.EXE)	35
1.14 提示 14: 像事务处理那样对待错误处理	39
1.15 提示 15: 不要测试自己的软件或者编写自己的测试计划	40
1.16 提示 16: 加强测试应用程序	40
1.17 提示 17: 使用自动测试工具	41
1.18 提示 18: 考虑错误值	42
1.19 提示 19: 加强 Visual Basic 的类型检查	43
1.20 提示 20: 用 TypeLib 或 Enum 定义常量	55
1.21 提示 21: 在资源文件中保存错误文本	55
1.22 提示 22: 总是在(您所构造的)控件和组件中处理错误	58

1.23 提示 23: 使用符号调试信息	61
第二章 关心商业(对象)	64
2.1 数据访问层	66
2.1.1 数据访问层的细节	66
2.1.2 记录集	67
2.1.3 检索记录集	68
2.1.4 串行化记录集	69
2.1.5 锁定记录集	70
2.1.6 更新记录集	70
2.1.7 同步化记录集	70
2.2 工厂-工作者对象	73
2.2.1 工厂-工作者模式	73
2.2.2 工厂对象	75
2.2.3 暂留	80
2.2.4 删除	81
2.2.5 了解记录集	82
2.2.6 确定工厂参数	83
2.2.7 创建工作者	84
2.2.8 商业对象浏览器	88
2.2.9 小结	89
2.3 操作对象	89
2.3.1 操作对象接口	90
2.3.2 操作中的事务	91
2.3.3 小结	92
2.4 客户程序	92
2.4.1 小结	93
第三章 在我面前看到的 IIS 是一个模板	94
3.1 Web 应用程序是什么?	96
3.1.1 IIS 还是 ASP?	96
3.2 使用 Visual Basic 6 开发 Web 应用程序	97
3.2.1 基础: 免费的素材	97
3.2.2 建立自己的主页	100
3.2.3 将您的主页扩展到客户	102
3.2.4 将您的主页转化为商业用途	104
3.3 结论	112

第四章 使用 Variant 进行程序设计	114
4.1 Variant 概述	115
4.2 内部结构	116
4.3 使用 Variant 代替简单数据类型	117
4.3.1 性能不是问题	117
4.3.2 内存也不成问题	118
4.3.3 类型安全性	118
4.3.4 灵活性	124
4.3.5 保护性代码编写	125
4.3.6 使用 Variant 作为普通数字数据类型	126
4.4 使用变量代替对象	128
4.5 其他 Variant 子类型	129
4.5.1 Empty 和 Null	129
4.5.2 数组	130
4.5.3 用户定义类型	135
 第五章 在 Windows CE 环境下开发应用程序	140
5.1 Windows CE 是什么?	142
5.1.1 适用的读者	143
5.1.2 构建 Windows CE 设备	145
5.2 内部揭示	147
5.2.1 支持的体系结构	147
5.2.2 Win32 API	148
5.2.3 对象存储	148
5.2.4 动态同步	150
5.2.5 进程和线程	150
5.2.6 实时功能	152
5.2.7 开发环境	154
5.3 Windows CE 及 IT 部门	155
5.4 Visual Basic 开发	156
5.4.1 开发环境	156
5.4.2 Windows CE 应用程序设计思想	157
5.4.3 开发第一个 Windows CE 应用程序	158
5.4.4 设计中要考虑的一般事项	159
5.4.5 错误处理	172
5.4.6 Windows CE 桌面模拟器	173

5.4.7 应用程序的测试及调试	175
5.4.8 配置应用程序	176
5.4.9 使用 COM DLL 扩展 Visual Basic	177

第六章 停留在控制阶段 178

6.1 昂贵的故事	180
6.2 我们要实现什么?	181
6.2.1 漫游指导	181
6.3 策略问题	183
6.3.1 优先权:四球魔术表演	183
6.3.2 进步可能更危险	184
6.3.3 按部起舞	185
6.4 态度问题	187
6.4.1 吞下旁边的犀牛	187
6.4.2 循环往复	188
6.4.3 重返校园	189
6.4.4 还需更多的功课	190
6.4.5 谨小慎微	191
6.4.6 跳出圈子	191
6.5 着手进行工作	192
6.5.1 飞行测试	193
6.5.2 如何健全您的程序?	197
6.5.3 查看存储过程的内幕	199
6.5.4 这儿有龙	200
6.5.5 有害的类型强制	204
6.5.6 捕获错误	208
6.5.7 声明意图	212
6.5.8 ActiveX 文档	216
6.6 一些 Visual Basic 6 工具	217
6.6.1 注册三种 Sourcerer	217
6.6.2 断言自身:Assertion Sourcerer	217
6.6.3 大小问题:Metrics Sourcerer	219
6.6.4 黑匣子:Instrumentation Sourcerer	220
6.7 总结	221
6.8 需要的读物文献	221

第七章 细节	222
7.1 关于编译器	223
7.1.1 p 代码简介	224
7.1.2 生成代码	225
7.1.3 记录器	228
7.1.4 连接器	230
7.1.5 使用编译器最优化代码	232
7.1.6 高级最优化	233
7.1.7 记录器代码	238
7.2 有关最优化的资料	242
7.2.1 选择合适的编程人员	242
7.2.2 使用混合语言编程	242
7.2.3 控制代码的速度	243
7.2.4 多动脑筋	243
7.2.5 抓住要点	243
7.2.6 “借用”代码	244
7.2.7 利用所有解决问题的技巧	246
7.2.8 使用烟雾和反射	248
7.3 关于对象、类型和数据结构	249
7.3.1 Visual Basic 是面向对象的语言	249
7.3.2 使用集合来扩展类型系统	254
7.3.3 添加 VarType	255
7.3.4 指针	259
7.4 关于类型库	265
7.4.1 删除 Declare 语句	266
7.5 关于 Smartie	272
7.6 其他资料	275
7.6.1 通过外出吃饭理解软件开发	275
第八章 Visual Basic 程序员指南：如何正确表示日期	278
8.1 历法简介	279
8.2 如何用 Visual Basic 正确表示日期？	280
8.2.1 Date 数据类型	280
8.2.2 在 Visual Basic 中使用日期数据	281
8.3 指定日期	288
8.3.1 有时您必须变得严格	302

8.4 公众面前的日期：用户界面问题	305
8.4.1 显示日期信息	305
8.4.2 日期输入	305
8.5 从哪里得到日期：存储问题	308
8.5.1 数据库的操作	308
8.6 什么时候可以继续：移植问题	310
8.6.1 当改动老代码时需要注意些什么	311
8.6.2 测试	313
8.6.3 测试数据和条件	313
8.6.4 改变您的系统日期	315
8.6.5 考虑第三方工具	316
8.7 找到日期的其他背景材料	316
8.8 结论	322
第九章 至少编译是成功的！	324
9.1 测试目的	328
9.2 正式的测试周期	328
9.2.1 单元/组件测试	328
9.2.2 综合测试	329
9.2.3 系统测试	330
9.2.4 用户接受测试	330
9.2.5 回归测试	331
9.2.6 代码复审	331
9.3 测试 Visual Basic 代码	332
9.3.1 与其他开发人员合作	332
9.3.2 在运行时测试	333
9.3.3 定期创建 Build	333
9.3.4 在编写代码的同时写测试脚本	334
9.3.5 决定放置测试代码的位置	334
9.3.6 在测试过程中确保源代码的覆盖率	335
9.3.7 理解测试数据	335
9.3.8 获得有关的用户	335
9.3.9 跟踪错误	336
9.4 测试计划	337
9.5 测试脚本	338
9.5.1 占位程序和驱动程序	338
9.6 计划代码组件	339

9.6.1 功能规范	340
9.6.2 测试脚本规范	342
9.7 性能测试	344
9.8 准备适当的测试环境	344
9.8.1 测试机器配置	345
9.9 警告	347

第十章 使用已装入的基础代码开始新工程 348

10.1 为什么使用基础代码?	349
10.2 模板中的内容	351
10.2.1 帮助程序函数	352
10.2.2 通用窗体	352
10.2.3 ActiveX 组件	352
10.2.4 子分类	352
10.3 子分类函数和子例程	353
10.3.1 子分类的益处	353
10.3.2 有关子分类的问题	356
10.3.3 子分类对象	357
10.4 通用窗体	358
10.5 隐藏 API 调用	359
10.6 注册表工具	360
10.7 资源文件	361
10.8 自定义控件	362
10.9 文档编制	362
10.9.1 对象浏览器(Object Browser)	363
10.9.2 帮助文件	363
10.9.3 HTML	363
10.10 构建您自己的基础代码	364
10.10.1 从哪里开始?	364
10.10.2 谁来构建基础代码?	365
10.10.3 源代码还是可执行代码?	365
10.10.4 商业实例	365
10.11 更改基础代码	366
10.12 模板	366
10.13 小结	367

第十一章 用 Visual Studio 进行混合语言编程	368
11.1 什么是混合语言编程?	369
11.1.1 典型的 MLP 情形	370
11.1.2 使用 MLP 有些什么要求?	371
11.1.3 在 Windows 中进行 MLP 的可能性	371
11.1.4 混合语言编程方法	376
11.1.5 使用 COM 及 ActiveX 与使用 DLL	378
11.2 使用单一入口	383
11.3 利用 COM 取代常规 DLL 入口	385
11.3.1 连接实例	385
11.3.2 从 Visual B++ 中调用 C++ DLL	387
11.3.3 将 Visual B++ 对象传递至 Visual C++	388
11.3.4 使用 ActiveX 控件进行 MLP	390
11.3.5 在汇编语言中混合	392
11.3.6 其他语言	392
11.3.7 定义组件版本	393
11.4 结论	395
 第十二章 数据库	396
12.1 一旦做,就一次做好,然后将之抛到脑后	397
12.1.1 数据	398
12.1.2 层和数据	400
12.1.3 取回数据	401
12.1.4 构建两层体系结构	402
12.2 关于客户/服务器交互作用的三种方法	407
12.2.1 典型的客户/服务器计算	407
12.2.2 日志/临时条目	408
12.2.3 多状态/事务/会话服务器	409
12.2.4 锁定角度解锁	409
12.2.5 查找到数百条信息时可能引发的问题	410
12.2.6 何时打开和关闭连接	412
12.2.7 决定事务作用域的放置位置	413
12.2.8 获取层	414
12.2.9 关于 SQL	416
12.2.10 静态查找数据	419
12.2.11 层和数据	419
12.2.12 对象、状态和数据: Buxom Server	420

12.2.13 依然不能确定哪些数据是对象所关注的？	423
12.2.14 数据——对象系统的“货币”	423
12.2.15 远程传输	424
12.3 将所有事情组件化	433
12.4 结束语	433

第十三章 定向程序设计 434

13.1 窗口管理	436
13.1.1 模式的还是非模式的？	437
13.1.2 一个通用模式类	439
13.1.3 扩展 CFormAttributes 类	440
13.2 复杂性处理	444
13.2.1 状态处理方面的技巧	446
13.2.2 使用 FSM 构建 GUI 模型	447
13.2.3 简洁的结束	449
13.2.4 实现 FSM	450
13.2.5 数据驱动编码	458
13.3 尘埃落定之后	467

第十四章 我上星期编写过那个函数吗？ 468

14.1 影响代码重用性的问题	469
14.1.1 最终期限的压力	470
14.1.2 关于当前技术的知识	470
14.1.3 开发队伍的素质	470
14.2 重用的商业情形	471
14.3 重用的关键	472
14.4 有效地满足重用的要求	473
14.4.1 使用对象的代码重用	474
14.4.2 将窗体作为可重用的组件	485
14.4.3 创建您自己的控件	492
14.4.4 使用 ROOS	504
14.5 程序员重用的问题	511
14.5.1 做几个假定	513
14.5.2 开发耦合策略	513
14.5.3 组功能	518
14.5.4 为您的代码编写文档	518
14.5.5 改善您的习惯	518