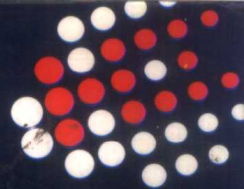
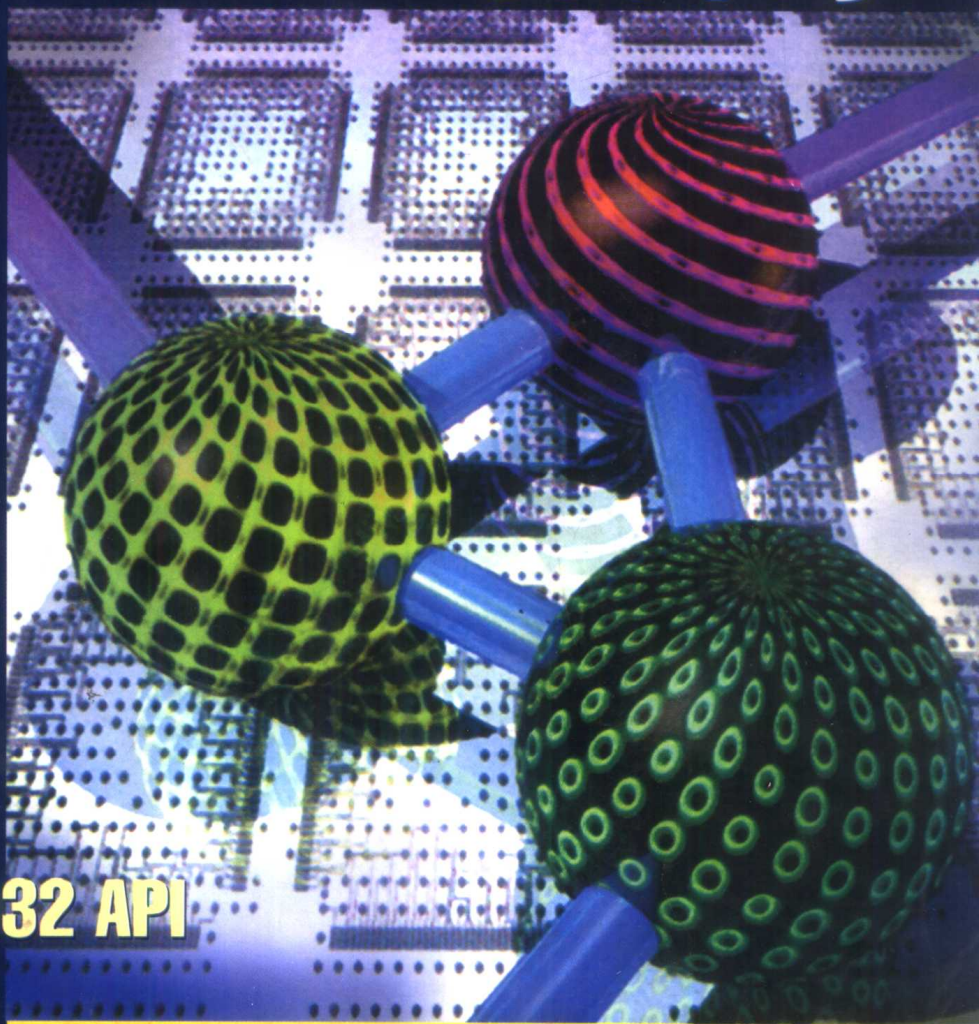


计算机软件开发与程序设计系列丛书



Visual Basic 5.0 Win32 API 开发人员 指南

(美) Dan Appleman 著
京京翻译组 译



附 CD-ROM 赠

Dan Appleman's
Visual Basic 5.0
Programmer's
Guide to the Win32 API



机械工业出版社

西蒙与舒斯特
国际出版公司



ZD Press
An Imprint of Que®

CMP

7. 11-1
计算机软件开发与程序设计系列丛书

Visual Basic 5.0 Win32 API 开发人员指南

(美) Dan Appleman 著

京京翻译组 译

机 械 工 业 出 版 社
西蒙与舒斯特国际出版公司

本书详细介绍了如何用 Visual Basic 发掘 Windows 32 位操作系统的全部潜力。作为一本世界畅销的指南性读物，它覆盖了 Win32 API 的全部要素——从窗口管理和绘图操作，从高级的过程控件到过程间通信技术。附带由 Desaware 出品的 API Toolkit 和 Desaware API 类库；通过 VB 加载项支持，提供 API 函数库；涵括 Win95、WinNT、Visual Basic 4.0/5.0 及大多数 VBA 语言。

本书配套的 CD-ROM 提供所有示例代码，方便读者检索本书电子版及其他技术文章。

本书适合计算机编程人员阅读。

Dan Appleman: Dan Appleman's Visual Basic 5.0 Programmer's Guide to the Win32 API.

Authorized translation from the English language edition published by ZD Press.

Copyright 1997 by Macmillan Computer Publishing USA.

All rights reserved For sale in Mainland China only.

本书中文简体字版由机械工业出版社和美国西蒙与舒斯特国际出版公司合作出版。未经出版者书面许可，本书的任何部分不得以任何方式复制或抄袭。

本书封面贴有 Prentice Hall 防伪标签，无标签者不得销售。

版权所有，翻印必究。

本书版权登记号：图字：01-98-0722

图书在版编目 (CIP) 数据

Visual Basic 5.0 Win32 API 开发人员指南 / (美) 爱坡曼 (Appleman, D.) 著；京京翻译组译. —北京：机械工业出版社，1998

(计算机软件开发与程序设计系列丛书)

书名原文：Dan Appleman's Visual Basic 5.0 Programmer's Guide to the Win32 API

ISBN 7-111-06539-5

I. V… II. ①爱… ②京… III. BASIC 语言—程序接口，API—程序设计—指南
IV. TP312

中国版本图书馆 CIP 数据核字 (98) 第 15308 号

出 版 人：马九荣 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑：蒋 克、廉一兵

北京忠信诚胶印厂印刷·新华书店北京发行所发行

1998 年 8 月第 1 版第 1 次印刷

787mm×1092mm¹/₁₆·77 印张

印数：0 001-6 000 册

定价：135 元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

序

在决定购买本书之前，请务必阅读下述重要事项！

本书的大量内容已在《PC Magazine —— Visual Basic 之 Win32 API 编程指南》一书里出版。由于升级至 Visual Basic 5.0，所以本书的名字也作了相应调整；改名的另一个重要原因在于，这本书与 PC Magazine（个人电脑）杂志没有丝毫联系（以前亦是如此）。

这一版的大多数修订都是针对 Visual Basic 5.0 及 Windows NT 4.0 作出的。我在 CD-ROM 里新增了两章内容。提供全部源代码，同时以 Visual Basic 4.0 和 Visual Basic 5.0 格式保存——适当做了些修改，以支持新的 VB5 特性。还有另一些文章，涉及与 Win32 及 Visual Basic 有关的各个主题。然而，如果您已有原书，便可能觉得这些修改并不值得另行购买一本新书。

为解决这个问题，我已与出版商进行磋商，决定制作新的配套 CD，以便原书读者能以较低的价格“升级”。这方面的事宜由 Desaware 公司直接负责。如果读者有什么问题，可查阅位于 www.desaware.com 处的通告；或打电话给 Desaware，号码是 (408) 377-4770；亦可发电子函件给 support@desaware.com。

——您忠诚的丹尼尔·爱坡曼（苹果人）

引 言

从前，有一本为 Windows API 而写的 Visual Basic 编程指南

微软发行了 Visual Basic 1.0 后，我就决定写那本书。Basic 终于进化成了普通人也能编写 Windows 程序的一种语言——在这之前，Windows 程序的编写相当复杂，是少数“高手”才能胜任的工作。Windows 的强大功能被巧妙封装于易于理解和使用的语言结构里。同时，程序员仍能用 API 及 DLL 调用发掘 Windows 的潜力。

但事实上，我发现许多 Visual Basic 程序员并没有充分发掘出 Windows 的潜力。原因很简单，与 Windows 有关的许多帮助资料都是为 C 和 C++ 程序员编写的——它们无法帮助 Visual Basic 程序员得到本来是垂手可得的東西。那时，我打算放弃当时的工作，去开一家新公司，专为 Visual Basic 开发加载产品。但我不想做普通的加载产品，我希望的东西应能帮助 Visual Basic 程序员发掘 Windows 的全部潜力。也就是说，用它们可绕开语言本身的限制，让自己有一个更自由的空间来施展才能。因此，除创办 Desaware 公司来负责软件产品以外，我还与辛迪·哈德森（Cindy Hudson）联系，他当时已是 Ziff-Davis 的资深出版人。辛迪建议我写一本有关 Windows API 的书籍——纯粹以 Visual Basic 程序员的角度。这便是最早的《Visual Basic 之 Windows API 编程指南》一书的由来。

在平淡的心情下完成那本书后，没想到读者的反响如此热烈。来自读者的大量鼓励与支持至今仍温暖着我的心。

1995 年 6、7 月份的时候，Visual Basic 32 位版的开发正在紧锣密鼓地进行。这时，我面临着一个进退两难的局面，同时也觉得机遇在向我招手。Desaware 的客户希望我们将控件尽可能快地升级为新的 OLE 控件模型。要完成这个任务，同时保持公司的继续发展，这本身便是一个很大的挑战。同时，要写一本有关 32 位 API 的书籍，则又是另一个艰巨的挑战。说实话，此时最明智的办法便是不写书——即使成为畅销书，也不能偿还时间与金钱上的投资。由于面向的是 32 位 API，原来那本书几乎要全部改写。

但在“峰回路转疑无路”的时候，却往往“柳暗花明又一村”。为 Win32 提供 Visual Basic 特别参考的需要依旧存在，而且许多人都盼望我继续跟进。如果我没有从第一本书的读者那里获得那么多支持，那么很容易就会放弃这个任务；但事实却刚好相反。于是，周末、旅游、渡假从我的日程表上取消了——在八个月极少睡眠的情况下，这本书终于撰写完毕。

本书的内容：新与旧

编写本书时，我遵守与最初那本《Visual Basic 之 Windows API 编程指南》相同的总则与目标。

首先，这一版面向的是那些熟悉 Visual Basic、但同时不熟悉 Windows API 的人士。同样地，这也是非常透彻的一本 Windows 入门书，建立在读者现有的 Visual Basic 知识基础

上。它并不要求掌握 C 或 C++——所有代码都是用 Visual Basic 写的。在保证入门者能领会的前提下，本书进行了足够程度的深入，能满足大多数有经验的程序员的需要。

其次，我并不打算把这本书变成一堆技术文件和示范代码的组合，让人们草率地搬用。相信大多数程序员都愿意真正理解和掌握程序中使用的代码。为帮助大家达到这个目的，这本书按常规主题划分。每个主题都包含了一段入门指引、一两个示范程序以及一个“函数索引”，后者列出与当前主题有关的 Windows 函数。入门指引并非仅仅指出 Windows API 函数运作的基本原理，而且要与 Visual Basic 紧密联系在一起。

同时，由于讲述的是 Win32，所以我面临着一些新的挑战。首先便是 Win32 API 那庞大的数量——无论怎样努力，都不可能在一本书里包罗万象。事实上，我觉得 Win32 API 的成长速度已超过学习它所达到的最高速度。而如果要把它写出来，那么速度更是远远不及。因此，我在本书中进行了如下改动：

□ 如果不能对每个 Win32 API 函数进行说明（这是显然的），备用方案是用足够的深度教会大家理解 API 编程。这样，Visual Basic 程序员便能充分理解 C 或 C++ 文件，以便对其进行转换和应用。为达到这个目标，讨论 Visual Basic→Windows 接口转换的文字便从单独的一章扩展成了两个更大的章节，另外还有散布于书中的相关论述。

□ 许多 Visual Basic 程序员仍在考虑同时支持 16 位和 32 位程序。因此，本书深入讨论了如何将 16 位代码移植为 32 位。此外，如仍在使 Visual Basic 4.0，则大量的提示和示例有助于编写同时兼容于 16 和 32 位平台的应用程序。

□ 说明文字与示范程序的数量明显增加。

□ 配套 CD-ROM 提供了本书的完全电子版，且可自由检索，更易找到所需信息。

另外，为确保本书在技术上的准确程度，我进行了大量努力。其中特别值得一提的是 Visual Basic 兼容问题，有关的大部分资料都来源于大量试验及合理的推断。

尽管如此，您仍可能发现某些形式的错误与疏漏；另外，一些较困难的概念可能有未尽其详之憾。我希望读者能无所顾忌地将意见与建议反馈给我，以便在本书的下一版改进。如欲与我联系，用因特网的读者请将电子函件发至 dan@desaware.com，或将 CompuServe 函件发至 70303, 2252。我的传真号是 (408) 371-3530。如这些办法都行不通，那就直接写信到下述地址：

c/o Ziff-Davis Press
5903 Christie Avenue
Emeryville, CA 94608

怎样使用本书（请务必一读！）

这是本大部头。

的确，是本非常大的书。

但不要被表面现象迷惑——要达到您的目的，实际不需要读完全书。

正如您已见到的那样，书中大多数章节都划分为三部分。首先是对本章主题的一个入门性讨论。这是全章的教程部分，它引入关键概念，并总结与那个主题有关的 API 函数。接下来是一些示范程序，对前面介绍的函数进行实际运用。最后通常是一个“函数索引”小节，总结如何使用那些 API 函数。

05/13/04

综上所述，下面是我提供给读者的一些建议，参考它们可更完美地利用本书。

首先，完整阅读从第 1 章到第 4 章的内容。为充分利用 Windows API ——并不是指将 Win16 程序移植为 Win32 ——这些章提供了所需的全部核心资料。

其次，阅读第 5 章到第 15 章的入门解释部分。这些章节涵括了重要的 API 函数——使用 Win32 API 时需要真正掌握的东西。第 15 章也提供了一些附加的资料，高级用户可以参考。

随后，请选择自己感兴趣的主题仔细阅读。

常见问题

☐ 我是 VB 编程的新手，该书对我有帮助吗？

这本书非常适合“新手”阅读。这句话并非违心之言，我是非常认真的！当然，这里的“新手”是指那些 Windows 初级程序员。本书确实假定了读者已经理解了 Visual Basic 编程的基本概念。

☐ 按顺序阅读各章，还是直接跳至某一特殊主题？

这并不是是一本充斥着提示、建议与技术主题的书籍。尽管它确实讨论了大量技术，但并不打算成为综合大量代码段的一本“百科全书”。我不愿意人们不暇思索地将示范代码插入自己的应用程序——尽管依然可以这样做。

本书的宗旨是教会您通过 Visual Basic 有效地使用 Win32 API ——即便以前从未用过它们。如果您是有心人，通过这本书就可以成为 Win32 专家。

为达到目标，我采用了与课本类似的组织方法：每章都假设您已掌握了前面章节的知识。因此，如能从头到尾依次阅读每一章最开始的解释部分，就能从这本书里获得最大的收获。

☐ “函数索引”小节有什么用？

许多章都在末尾提供了一个“函数索引”小节，详细总结了与本章主题有关的 API 函数。每个函数都包含下述信息：

VB 声明 Visual Basic 声明，与 API32.TXT 文件里的内容一致。

说明 对函数进行简要说明。

在 VB 里使用 在这里指出在使用该函数的时候，是否需要考虑与 VB 有关的任何问题。“没有问题”意味着在 VB 里使用该函数时，迄今为止尚未得到任何错误报告。注意，尽管我们已在 VB 里对 API 函数进行了长时间的测试，但却不能保证您使用时绝对不会遇到任何麻烦。

参数表 一张表格，列出该函数采用的参数。

返回值 说明函数的返回值。这儿常见的词语包括：

True（非零）——用这种形式注明是为了强调这样一个事实：在 Windows 里，True 的对应值通常不再是 Visual Basic 使用的“-1”（参考第 3 章）。

会设置 GetLastError ——指出函数设置了一个内部错误值，可由 API 函数 GetLastError 读取（参考第 6 章）。

适用平台 列出该函数适用的系统平台。可能包括：

Windows NT ——适用于 Windows NT 3.51 或更高版本。

Windows NT 4.0——适用于 Windows NT 或更高版本。

Windows 95——适用于 Windows 95，也有可能适用于采用了 Windows 95 外壳的 Windows NT。

Win16——该函数有对应的 Win16 函数。请参考《Visual Basic 之 Windows API 编程指南（16 位版）》，其中详细介绍了函数的 Win16 版本，API16.TXT 文件则解释了如何在 VB 里进行声明。这些资料只是在您有空的时候参阅；本书“函数索引”的所有内容都仅适用于 Win32 函数，而且一般都应假定与 Win16 函数存在着不小的差异。

注解 这儿列出可能有帮助价值的其他所有注意事项。

移植问题 这个小节探讨了将函数从 Win16 移植过来时可能遇到的问题。但我们并不打算深入解释如何对函数进行移植——只列举一些敏感问题，以及在 Win32 与 Win16 函数不一致时指出易于混淆的地方。

示例 这个小节列出相应的示范程序，它们演示了具体如何运用本函数及相关函数。并不在这里说明函数在本书所有示范程序中的应用情况。有些函数则提供了简短的代码段作为示例。

没有例子的大多数函数都非常简单，即使初学者不要例子也可以理解——只需掌握本书的头三章内容即可。

☐ 微软已随同 VB 提供了一个不错的 WIN32API.TXT 文件，为什么你还要自己提供一个 API32.TXT 呢？

坦白地说，这是由于 WIN32API.TXT 并没有想象中的那么出色。事实上，该文件根本不完整——尽管加起来有一万多行。它是用微软的一个文件解析工具自动生成的，但并未进行多方面的测试和调校。我的 API32.TXT 包含了对 WIN32API.TXT 多达数百处的修订。它的基础不仅包括 WIN32API.TXT，而且还包括许多实际的输出表格。这些表格来源于各种 Win32 动态链接库、微软的 Win32 软件开发包（SDK）API 数据库、微软开发者网络 CD-ROM。而且最重要的是，Desaware 的软件工程师和我自己都进行了数百小时的测试与使用。

尽管进行了这么多工作，但仍不能保证 API32.TXT 完全准确。毋庸置疑，无论读者还是我自己，在以后的使用过程中都可能找出更多的错误。但无论如何，书中使用的正是这个文件，而且要求读者也用它——在发现书中某个例子不能工作时，务必核实使用的是否为 API32.TXT。

☐ 为什么你在书中省略了一些内容？

尽管本书的容量有大幅度增加，但仍然没有地盘可容下我想介绍给大家的全部东西。添加了说明文字和讨论了新的核心 API 函数后，有些东西就必须放弃。大多数 NT 专用的机制被省略了，同样省略的还有大多数扩展库。通信问题被略去了（尽管部分资料在配套 CD 里得以保存），这是由于许多人都直接用控件来实现通信。由于 16 和 24 位显卡已成为市场主流（大多数都装备于支持 Win32 的高档系统），所以有关调色板的资料已不再重要。注意，在最初那本《Visual Basic 之 Windows API 编程指南》里，我提供的大多数调色板资料依然适用。

通过多篇文章及 Desaware 公司的产品，我逐步提供了有关这些主题的一些附加资讯。请特别留意位于 www.desaware.com 的主页，那儿经常都有新的信息出台。此外，随着您越

来越精通 Win32，会发现在 Visual Basic 下使用 C/C++ 文档要比想象的容易得多。

□ 针对回调、子类和挂接，你提供了一些演示控件，它们有什么用？在实际程序里，它们为什么不能象 cbk.vbx 控件在原书里那样使用？

一个无法回避的事实是：Win32 API 提供的许多功能都要求使用回调、子类、挂接或者一个接口 DLL。

□ 这样一来，问题就变成了：怎样运用这些控件？

在原来那本为 16 位编程写的书里，我提供了 cbk.vbx 回调控件一个限制（使用）版本，它的功能却是完整的。问题在于：随着该控件商业版的广泛使用，应用程序会（正确地）删除随书提供的老版本控件。这样一来，读者便不能装载示范程序——它们尚未针对新控件进行使用授权。要么运行示范程序，要么运行以新控件为基础的程序，但不能同时进行——这种情况是不允许的。

从理想的角度看，我最好能免费提供所有这些控件，同时不断地进行技术支持与升级。但非常不幸，我实际无力这样做。即便一本成功的专业书籍，它的作者也只能获得少得可怜的收入。

当然，我也可以省略有关控件的内容，避免讨论要求用到它们的任何 API 函数。但这样做意味着将一部分重要的 API 技术排斥在外，并可能造成读者的疑惑。因为如不涉及这些问题，对 Win32 API 和 Visual Basic 的讨论根本就是不完整的。

所以到底该怎么办呢？唯一可行的方案是创建这些控件的全功能演示版，让它们只能在 Visual Basic 设计环境下运行。这便提供了试验与学习技术所需的工具。

我知道，现在许多专业书都向读者允诺在配套光（磁）盘上提供大量“免费”软件。但读者最终得到的只是一些“发育不全”或仅供演示的程序。这些东西的价值极低，有的根本没有技术支持。亲爱的读者，如果您购买这本书只是为了获得许多免费软件，那么只好让您失望了。但是，如果您购买这本书是考虑到其中的说明与资料有帮助价值，那么我将特别高兴，因为您发现了真正的金矿。正如我一贯坚持的那样，本书并不等同于免费软件——只是希望您提供大量有价值的信息。希望您觉得它们物有所值。

下面是一条重要声明：作为一名专业的软件开发者，我绝对不会让自己的应用程序完全依赖这本书中用于演示用途的免费控件，因为它们是没有技术支持的。当然，我也不会建议读者去做我自己都不会做的事情。

如确实想在自己的应用程序里使用这些演示控件的功能，必须通过其他途径获得控件。您可以购买 SpyWorks 控件。我不大会谦虚，但就我个人的观点看，SpyWorks 控件确是整个业界最棒的。当然，您亦可选购其他公司的产品。但无论怎样，一定要从声望卓著的公司购买技术支持完善的产品。子类处理很容易就会走上歧路，它被公认为是最难对付的（即使用上了 Visual Basic 5.0 的地址运算符）。Visual Basic 4.0 的回调功能也有类似的麻烦。所谓“免费”控件的后期费用可能相当巨大，所以在挑选控件时请务必慎重。

“我遇到了麻烦，需要您的帮助！”

我几乎每天都能收到这样或内容类似的电子邮件。真希望能为所有这些人作出回答，但这显然是不可能的。由于工作安排紧，所以无法抽出许多时间——作答。但我又确实不愿简单地拒绝。因为如果一个关键项目卡壳，主管人员肯定会急得团团乱转；而有时为了得到正

确答案，其间要经历多少辛苦！我对此有深切体会。

所以，当一个问题摆在我面前时，它通常会经历下面这些程序。

如果能迅速看出问题，能用短短几行说明，那么我通常都会立即作答——即便有时只是指出正确的努力方向。

如果问题来自 Desaware 的一位客户，我会特别加以注意——即使它并不直接关系到我们产品的使用。我或者会转交给技术人员具体处理。

如果我发现一个有趣的问题，而且有闲余时间，便会坐下来好好想想，看看它在教育、研究方面有什么价值。

有时，我可能以顾问的身份解决一些问题，但这通常都不是最优先考虑的（估计读者也不想知道我为这种咨询工作收取多少费用）。

如果您的问题完全与其他公司的产品有关，便请直接与那家公司联系——除了 Desaware 的产品和 Visual Basic 本身之外，我没有其他方面的资料，不能回答您的任何问题，也不能提供那些方面的支持。如果那家公司不能提供足够的支持，我也只好爱莫能助了。

另外，不要用电子邮件发来很长的程序清单，并要求我调试——对此我是不予理睬的。

最后，Desaware 公司在网上开办了一个 Web 网站和一台邮寄列表服务器。随着时间的推移，有关这本书的更新、信息及程序注释都会不断更新。如果您想成为邮寄列表的一员，只需向 listserv@desaware.com 发一封电子函件，将主题写成“Subscribe”即可。欲知更进一步的信息，请将请求信发至 Desaware 公司，地址是 support@desaware.com，Go Desaware (CompuServe) 或 74431.3534@compuserve.com。

目 录

序
引言

第一部分 Windows API

第 1 章 DLL 和 API	1
1.1 从 DOS 到 Windows	1
1.2 动态链接库 (DLL)	2
1.2.1 静态链接	2
1.2.2 动态链接	4
1.2.3 Visual Basic 和 DLL	5
1.3 应用程序编程接口 (API)	5
1.4 五花八门的 Windows	7
1.4.1 主要的 Windows DLLs	8
1.4.2 无穷无尽的扩展库	8
1.5 变化中的 Visual Basic	9
第 2 章 初步认识 Windows	10
2.1 Visual Basic 与 Windows 的联系	10
2.1.1 Visual Basic 综述	10
2.1.2 Windows 综述	11
2.1.3 将 Visual Basic 的概念转换到 Windows 里	11
2.2 深入 Windows 内部	12
2.2.1 什么是窗口?	12
2.2.2 窗口属于不同的类	12
2.2.3 窗口有样式	14
2.2.4 其他窗口属性	15
2.2.5 窗口函数	16
2.2.6 超类和 Visual Basic 控件	17
2.2.7 Visual Basic 与 Windows 的关系 回顾	22
2.3 Windows 里的图形输出	22
2.3.1 图形设备接口	23
2.3.2 设备场景	23
2.3.3 内存设备场景	25
2.4 对象的句柄	25
2.5 接下去做什么	27
第 3 章 Visual Basic - Windows 接口	28

3.1 声明语句	28
3.2 Unicode、ANSI 和别名	29
3.2.1 Visual Basic、ANSI 和 Unicode	30
3.2.2 访问适当的函数	32
3.2.3 API 调用结果	33
3.3 DLL 参数	33
3.3.1 参数列表	33
3.3.2 DLL 参数类型和记号法	35
3.3.3 参数类型入门	37
3.3.4 8 位数值参数	37
3.3.5 16 位数值参数	38
3.3.6 32 位数值参数	39
3.3.7 货币参数	40
3.3.8 浮点参数	40
3.3.9 变体	41
3.3.10 句柄 (Windows 对象)	41
3.3.11 对象 (Visual Basic 和 OLE)	41
3.3.12 旗标与域域	42
3.3.13 字符串	45
3.3.14 数值指针	46
3.3.15 结构指针	47
3.3.16 数组指针	48
3.3.17 函数指针	49
3.3.18 能接收多种类型的参数	50
3.4 API/DLL 调用是如何工作的	51
3.4.1 堆栈帧	51
3.4.2 返回值	54
3.4.3 理论之外的东西	55
第 4 章 实际的 API 编程	56
4.1 API 编程技术	56
4.1.1 “类型安全”声明	56
4.1.2 发送消息	57
4.1.3 模块化编程	58
4.1.4 API 声明文件及工具	58
4.2 将 API 文件转换到 Visual Basic (第 1 部分——函数)	60
4.3 为 16 和 32 位平台编写程序 (仅适用于	

VB 4)	65
4.3.1 条件编译	66
4.3.2 要区分平台的编码	68
4.3.3 示例: 专用初始化文件 (修订版)	68
4.3.4 条件编译无效时怎么办	77
4.4 移植现有的 16 位代码	78
4.4.1 发生了变化的函数	78
4.4.2 弃之不用函数	79
4.4.3 消息的移植	80
4.4.4 与向 Windows 体系的过渡有关的 移植问题	82
4.5 将 API 文件转换到 Visual Basic (第 2 部分——文件与结构)	82
4.5.1 示例: TrueType 字体浏览器	82
4.5.2 移植时要注意的问题	96
4.6 API 代码的调试	96
4.7 接下去做什么	99

第二部分 Windows API 函数

第 5 章 Windows 控件与信息函数	101
5.1 矩形和画点函数	101
5.1.1 RECT 结构	101
5.1.2 POINTAPI 结构	102
5.1.3 矩形函数与 RectPlay 示例	105
5.2 Windows 控件与信息函数	116
5.2.1 窗口级别与标识函数	116
5.2.2 窗口位置与大小函数	119
5.2.3 窗口信息函数	121
5.2.4 输入焦点、前台与输入状态	121
5.2.5 Statevw 示例	124
5.3 其他窗口函数	130
5.4 窗口样式索引	153
5.4.1 对话框样式	160
5.4.2 对进一步实践的建议	160
5.5 函数索引	160
第 6 章 硬件及系统函数	195
6.1 鼠标、指针和插入符函数	195
6.1.1 指针剪切	195
6.1.2 指针位置	196
6.2 其他鼠标及指针函数	196
6.3 键盘和其他输入函数	198

6.3.1 字符集重述	198
6.3.2 扫描码和虚拟键	199
6.3.3 虚拟键码	201
6.3.4 地方	203
6.3.5 输入控制函数	204
6.4 时间函数	205
6.5 系统信息和控制函数	206
6.6 示例: SysInfo——一个系统 信息观察器	210
6.6.1 SysInfo 的使用	210
6.6.2 项目说明	211
6.7 模拟击键和鼠标事件	227
6.8 函数索引	238
第 7 章 设备场景	286
7.1 设备场景入门	286
7.1.1 设备场景背后的机理	286
7.1.2 剪裁	287
7.1.3 坐标转换	288
7.1.4 深入设备场景	288
7.2 设备场景的使用	290
7.2.1 获得设备场景	291
7.2.2 设备场景属性	292
7.2.3 在 Visual Basic 中使用设备 场景	292
7.3 设备场景信息函数	295
7.4 缩放和坐标系统	299
7.4.1 逻辑坐标和设备坐标	299
7.4.2 映射模式	301
7.4.3 窗口和视口——范围和起点	301
7.4.4 世界转换	304
7.4.5 Windows 坐标系统与 Visual Basic	307
7.4.6 平台间的差异	307
7.4.7 Viewport 示例	307
7.5 剪裁、区域和其他设备场景函数	326
7.5.1 区域	327
7.5.2 剪裁	327
7.5.3 校验和其他设备场景函数	328
7.5.4 对性能的考虑	329
7.5.5 示例: Region	329
7.6 函数索引	342
第 8 章 绘图函数	373
8.1 图形输出综述	373

8.2 GDI 绘图对象	374	9.5.4 DIBSect —— 演示如何使用 DIBSection 对象	550
8.2.1 绘图对象的创建	374	9.6 函数索引	558
8.2.2 GDI 对象的选定和删除规则	375	第 10 章 菜单的使用	580
8.2.3 画笔	376	10.1 深入菜单系统	580
8.2.4 刷子	376	10.1.1 Windows 菜单的工作原理	580
8.2.5 GDI 对象和 Visual Basic	377	10.1.2 标准和扩展菜单	583
8.3 绘图属性	378	10.1.3 Visual Basic 菜单是如何 工作的	584
8.3.1 线光栅操作	379	10.2 在 VB 里使用菜单 API 函数	587
8.3.2 背景模式	379	10.2.1 创建定制的复选符号	587
8.3.3 当前位置	379	10.2.2 用位图定制菜单	587
8.4 绘图函数	380	10.2.3 追踪弹出式菜单	588
8.5 路径	381	10.2.4 创建 VB 菜单控件池	588
8.6 综合运用: 按例学习	383	10.2.5 菜单、系统菜单和子类	589
8.6.1 示例: Path.vbp	383	10.2.6 获取与 VB 菜单结构有关的 信息	589
8.6.2 示例: ExtPen.vbp	391	10.3 示例	589
8.6.3 示例: Bezier.vbp	402	10.3.1 MenuLook —— 菜单结构 观察器	589
8.7 图元文件	408	10.3.2 SysMenu —— 演示系统和上下文 关联菜单	613
8.8 综合运用 II: 其他例子	410	10.4 函数索引	618
8.8.1 示例: QuikDraw —— 一个简单的 绘图程序	411	第 11 章 文本和字体	635
8.8.2 使用 QuikDraw	411	11.1 字体的使用	635
8.8.3 示例: Analyze —— 深入图元文件 内部	442	11.1.1 字体入门	635
8.9 函数索引	450	11.1.2 理解字体属性	637
第 9 章 位图、图标和光栅运算	496	11.2 字体宽度	637
9.1 位图	496	11.3 字体大小	638
9.1.1 像素的显示	497	11.4 字体家族	640
9.1.2 颜色与设备调色板	497	11.5 字符集	641
9.1.3 与设备有关的位图	501	11.6 发音符、连字和 Kashida	642
9.1.4 颜色转换	504	11.7 创建逻辑字体	643
9.1.5 在 Visual Basic 里使用位图	504	11.8 GDI 字体映射	643
9.2 与设备无关的位图	505	11.9 字体信息函数	644
9.2.1 动态调整大小的结构	505	11.10 添加和删除字体	644
9.2.2 DIB 结构	507	11.11 TEXTMETRIC 和 NEWTEXTMETRIC 结构字段	646
9.2.3 DIBSection	509	11.12 文字的描绘	648
9.3 图标与指针	511	11.12.1 文字的描绘过程	648
9.4 光栅运算	513	11.12.2 文本范围及对齐	649
9.5 示例	515	11.13 示例	649
9.5.1 StockBMs —— 固有位图和图标 观察器	515	11.13.1 FontView 示例	649
9.5.2 ROPTest —— 光栅运算试验 程序	521		
9.5.3 Puzzle —— 分裂位图还原程序	532		

11.13.2 TextDemo 示例	661	13.6.1 版本标记与 Visual Basic	787
11.13.3 DrawText.vbp 示例	667	13.6.2 版本资源数据结构	787
11.13.4 EnumFontX 示例	669	13.7 示例	791
11.14 函数索引	676	13.7.1 FileDemo——初始化文件、注册表 和版本标记演示程序	792
第 12 章 打印	696	13.7.2 CkServe/ CkClient——协作式应用 程序	802
12.1 在 Windows 里打印	696	13.7.3 Disk.vbp——磁盘剩余空间 工具	814
12.1.1 打印机设备场景	696	13.8 函数索引	820
12.1.2 Win32 打印子系统	697	第 14 章 进程与线程	881
12.1.3 打印机的配置	699	14.1 进程、线程与实例	881
12.1.4 打印顺序	699	14.1.1 有关多任务的情况	881
12.2 打印和 Visual Basic	702	14.1.2 任务、进程与同步	883
12.2.1 兼容问题	702	14.1.3 线程	884
12.2.2 打印机配置与 Visual Basic	703	14.1.4 实例	885
12.2.3 驱动程序的问题	703	14.2 系统与同步对象	886
12.3 打印机设置及 DEVMODE 结构	703	14.2.1 使用系统对象	886
12.4 对后台打印程序的控制	706	14.2.2 Mutex、信号机、事件和侍服式 计时器	888
12.5 示例	710	14.2.3 同步与 Visual Basic	891
12.5.1 PicPrint——打印一幅位图， 并显示打印机的配置	710	14.3 应用程序的启动	892
12.5.2 DocJob.vbp——将输出导向至 打印机	729	14.3.1 进程与线程对象	892
12.5.3 Spooler.vbp——使用 API 函数的 另一种形式	732	14.3.2 判断其他应用程序何时关闭	894
12.6 函数索引	741	14.4 进程间通信	895
第 13 章 文件处理	773	14.4.1 管道	895
13.1 应该用什么类型的文件访问？	773	14.4.2 邮路	897
13.1.1 专用的初始化文件	773	14.5 示例	898
13.1.2 系统注册表	773	14.5.1 ckServe/ ckClient: 协作式应用 程序重述	898
13.1.3 专用文件格式	774	14.5.2 Launch.vbp: 启动应用程序	902
13.1.4 数据库	774	14.5.3 dwWatch.vbp: 与 OLE 回调 同步	909
13.1.5 OLE 结构化存储	774	14.5.4 PipeSrc.vbp/ PipeDest.vbp: 匿名 管道示例	916
13.2 初始化文件	774	14.5.5 PipeClnt.vbp/ PipeSrv.vbp: 命名 管道示例	923
13.3 系统注册表	776	14.5.6 PipeClnt.vbp 示例	929
13.4 文件和目录处理	779	14.6 函数索引	932
13.4.1 Win32 文件函数	780	第 15 章 内存、字符串、结构及 资源	967
13.4.2 对压缩文件的操作	783	15.1 内存和字符串	967
13.5 文件映射 (是文件, 还是 内存?)	784		
13.5.1 内存映射文件的重要性 何在?	785		
13.5.2 内存映射文件到底重要 在哪里?	786		
13.6 版本标记	787		

15.1.1 用 Visual Basic 创建缓冲区	967
15.1.2 用 Windows 创建缓冲区	968
15.1.3 在 Visual Basic 里使用全局 内存块	973
15.1.4 字符串函数	975
15.2 Visual Basic 和 Windows 接口: 核心 版本	976
15.2.1 高级结构技术	977
15.2.2 结构中的固定串	980
15.2.3 结构中的动态串	980
15.3 资源	985
15.3.1 资源与 Windows 应用程序	985
15.3.2 资源与 Visual Basic	987
15.3.3 资源 API 函数	987
15.4 原子与属性	989
15.4.1 原子	989
15.4.2 属性函数	990
15.5 函数索引	991

第三部分 Windows 消息

第 16 章 常见的 Windows 消息	1027
16.1 消息控制	1027
16.1.1 Windows 事件消息	1027
16.1.2 执行操作的消息	1028
16.1.3 消息组织	1028
16.1.4 发送消息	1029
16.1.5 消息参数	1030
16.1.6 能返回值的消息	1030
16.1.7 注册消息	1030
16.1.8 WM_COMMAND 消息	1031
16.2 对 Visual Basic 有用的消息	1031
16.3 Visual Basic 和子类	1032
16.4 函数索引	1034
16.5 消息索引	1039
第 17 章 编辑控件消息	1073
17.1 文本控件	1073
17.1.1 撤消功能	1073
17.1.2 文本格式化	1073
17.1.3 选定与显示	1074
17.2 示例: TextMsgs——编辑控件消息 的演示	1075
17.2.1 使用 TextMsgs	1076

17.2.2 项目说明	1076
17.2.3 窗体说明	1076
17.2.4 TextMsgs 程序清单	1079
17.3 通告消息 (编辑控件)	1083
17.4 编辑控件消息	1085
第 18 章 列表框、组合框和按钮 消息	1094
18.1 列表控件	1094
18.1.1 选定和数据消息	1094
18.1.2 显示消息	1095
18.2 组合框控件	1096
18.3 按钮控件	1098
18.4 通知消息 (列表、组合框、 按钮)	1098
18.5 组合框控件消息	1101
18.6 列表框控件消息	1105
18.7 按钮控件消息	1115

第四部分 API 编程实践

第 19 章 在 VB 里使用 API 调用——软件 设计中的权衡	1117
19.1 用 Windows API 改善性能	1117
19.1.1 性能评测	1117
19.1.2 性能评测示例: Visual Basic 与基于 API 的图形	1122
19.2 使用 Windows API 时要功能与安全 兼顾	1124
19.3 设计选择	1129
19.3.1 比较	1130
19.3.2 问题	1131
19.4 在实践中学习	1132
第 20 章 构建 API 类库	1134
20.1 类库设计	1134
20.1.1 API 类库的设计标准	1134
20.1.2 类的选择	1135
20.2 类库的实现	1137
20.2.1 函数名和参数	1137
20.2.2 结构的封装	1138
20.2.3 对象的封装	1140
20.2.4 常数的运用	1143
20.2.5 出错控制	1145

20.2.6 综合运用: dwDeviceContext 类	1145
20.3 使用类库	1148
20.3.1 使用 dwRECT	1148
20.3.2 使用 dwWindow	1149
20.3.3 使用 dwSystem 类	1149
20.3.4 使用 dwDeviceContext	1150
第 21 章 串行通信	1152
21.1 串行通信入门	1152
21.1.1 串行数据格式	1153
21.1.2 The RS-232 Standard	1154
21.1.3 DTE and DCE	1154
21.1.4 数据信号	1155
21.1.5 流控制	1156
21.2 Windows 串行通信驱动程序	1156
21.2.1 串行驱动程序的结构	1156
21.2.2 使用串行驱动程序	1157
21.2.3 串行驱动程序的事件	1158
21.2.4 “中断”状态	1158
21.2.5 通信驱动程序和并行端口	1159
21.2.6 配置串行端口和设备控制块	1159
21.2.7 了解串口状态	1159
21.2.8 串行驱动程序 API 函数	1159
21.3 示例: CommDemo——一个简单的终端程序	1160
21.4 函数索引	1160
第 22 章 网络函数	1169

22.1 Windows 与网络	1169
22.2 WNet 函数和类	1170
22.2.1 类结构	1170
22.2.2 网络错误控制	1171
22.3 枚举网络资源	1172
22.3.1 dwNetResource 类	1174
22.3.2 声明、数据和初始化	1174
22.4 示例: Treenet.vbp——一个网络浏览实用程序	1178
22.5 函数索引	1178
第 23 章 API 类型库	1186
23.1 什么是类型库?	1186
23.2 ODL 文件语法	1188
23.2.1 结构声明	1189
23.2.3 枚举定义	1190
23.2.4 常数和函数定义	1190
23.3 类型库的优缺点	1191
23.4 Desaware API 类型库	1191

第五部分 附录

附录 A 使用回调、APIGID32.DLL、dwcbk32d.ocx 和 desbc32d.ocx	1193
附录 B Windows 数据结构	1205
附录 C Windows 文件格式	1206
附录 D Visual Basic→DLL 调用转换: 技术性总结	1210
附录 E 光栅运算表	1216

第一部分 Windows API

第 1 章 DLL 和 API

1991 年，微软发行了一种新语言的 1.0 版，这便是后来大名鼎鼎的 Visual Basic。它从根本上解放了 Windows 编程，并从多方面改善了编程技术。据估计，微软总共售出了约 100 万份 Visual Basic 1.0 拷贝。这为它后来成为最流行的编程语言之一打下了坚实的基础。人们之所以喜欢用 Visual Basic，最重要的或许是由于它真正方便了人们编写 Windows 应用程序。即便不是 Windows 专家，亦能编出让自己高兴好一阵子的程序。

1.1 从 DOS 到 Windows

Visual Basic 是一种全方位的编程语言，可用它编出各种类型的应用程序。但是，有些问题也必须注意。Visual Basic 有一些尚未直接支持的领域，而读者迟早都会接触到这些领域。例如，声音支持、I/O 端口地址以及许多位图运算都不是 Visual Basic 软件包的一个集成部分。

但这并不能成为 Visual Basic 的缺点。没有任何一种语言能包容每名程序员可能需要的全部命令与函数。在 DOS 环境下，这个问题的解决办法是创建和使用函数库，从而对语言本身的功能进行扩展。可在需要的时候将这些函数链接到自己的程序中。

而在 Windows 的世界里，情况却稍微有些不同。首先，运行环境明显比 DOS 复杂——即便在屏幕上显示一条线段这样简单的行动，也要涉及使用象 Windows 句柄、显示场景和画笔这样的对象。外部函数里的链接方式也发生了变化——Windows 应用程序通常用一种名为“动态链接”的技术，用它代替 DOS 下常见的静态链接（有关链接的问题本章稍后会详述）。

如使用的一直都是 Visual Basic 函数与命令，上述复杂性就会永远隐藏于幕后。但是，一旦准备扩展语言本身的功能，就注定要亲自面临这些问题。

但不要紧张。

像所有正规的编程语言那样，Visual Basic 也提供了强大的扩展能力。尽管其他公司投入很大精力为 VB 编写加载项与函数库，但最好的加载库仍然自动包含于 VB 语言的每一份拷贝里。那便是 Windows——这个操作系统的本身。

曾有许多宣传都说 Windows 编程是如何如何难——这一点不假，但那是纯粹用 C++、C 或 Pascal 编程的人而言的。作为 Visual Basic 程序员，您实际已拥有了一个很好的知识储备，那就是知道 Windows 的基本运作原理（尽管您可能尚未意识到这点）。这是由于作为微软公司的产品，Visual Basic 已与 Windows 非常牢靠地结合在一起。VB 的“事件”与 Windows 的“消息”存在着紧密的联系。VB 里的“属性”通常都与 Windows 的“样式”与“属性”相关。从某种程度上，可认为 Visual Basic 就是 Windows 的一个可编程外壳。