Visual Basic 4 API How-To

VISUAL BASIC 4

API 程序设计

(美) Noel Jerke 等著

袁兆山 张健
袁晓辉 等译

刘宗田 审校

计算机软件开发
程序设计
系列丛书

机械工业出版社 西蒙与舒斯特国际

计算机软件开发与程序设计系列丛书

Visual Basic 4 API 程序设计

(美) Noel Jerke　Eric Brierley 著

袁兆山　张健　袁晓辉
袁晓靖　苗沛荣　等译

刘宗田　审校

机 械 工 业 出 版 社
西蒙与舒斯特国际出版公司

《VISUAL BASIC 4 API 程序设计》一书共分 10 章和 4 个附录。该书在论述了 Windows API 与 VB 的关系的基础上，完整地介绍了 API 基本编程知识，以问题、技巧、步骤、编程原理、评注的清晰层次叙述了在 VB 中应用 API 的 10 大类扩展 VB 程序功能的实例。各节提供的实例演示，引导读者把握 VB 及 API 的精髓，掌握利用 API 进行 VB 编程的技巧和新的软件设计方法，因此，无论是初学者还是具有编程经验的设计人员或专家，读了本书都会受益匪浅。

本书版权登记号：图字：01-96-1223

图书在版编目（CIP）数据

Visual Basic 4 API 程序设计/（美）泽克（Jerke, N）等著；袁兆山等译。-北京：机械工业出版社，1997. 1
书名原文：Visual Basic 4 API How-To
（计算机软件开发与程序设计系列丛书）
ISBN7-111-05460-1
1. V… Ⅱ. ①泽… ②袁… Ⅲ. Basic 语言-程序设计 Ⅳ. TP312BA
中国版本图书馆 CIP 数据核字（96）第 23784

出版人：马九荣（北京市百万庄南街 1 号　邮政编码 100037）
责任编辑：东凌　何伟新
三河永和印刷有限公司印刷　·新华书店北京发行所发行
1997年 1 月第 1 版　1997 年 4 月第 2 次印刷
787mm×1092mm 1/16 · 47.25 印张 · 1179 千字
5001-9000 册
定价：83.00 元

译者的话

Visual Basic 是 Microsoft 公司专为开发 Windows 程序而推出的优秀的计算机程序设计语言产品，具有可视化的直观工作方式和出色的性能，具有开发效率高等明显的优点，特别是大量的 VBX 控件的使用，使得软件开发真正走上了简单快速的新阶段。但是，要编写优质高效的 VB 应用程序，仅仅使用 VB 还是不够的，还必须充分利用 VB 的扩展性，调用 Windows 动态链接库中提供的大量的 API。

为帮助读者掌握利用 Windows API 更快捷地进行 VB 应用编程，在 Windows 环境下顺利地开发各类应用软件，中美合资北京华章图文信息有限公司组织我们翻译了美国 Noel. Jerke 和 Eric. Brierley 编著的《Visual Basic 4 API How—TO》一书，将其中文本奉献给国内广大读者。

本书由合肥工业大学袁兆山教授领衔翻译。参加翻译的还有袁晓辉、张健、袁晓靖、苗沛荣、王华忠、张正武。在今年这些同仁还合作翻译了《计算机网络轻松入门》(机械工业出版社，1996 年 5 月)、《Windows 95 轻松入门》(机械工业出版社，1996 年 8 月)两本著作。在本书的翻译过程中得到了华章图文信息有限公司的大力支持，还得到了合肥工业大学有关领导和同仁们的支持，在此表示谢意。

译者

1996 年 11 月　于合肥

前　言

关于本书

《Visual Basic 4 API How-To》是一本介绍 Visual Basic 4 的 win32 API（应用程序编程接口）的书。目前所有的 Windows 95 或 NT 编程都必须使用 API（包括了 Visual Basic），从而使 API 成为有史以来最常用并被证明是最有效的一套编程函数。Visual Basic 避开了 Windows 编程的复杂性，允许编程人员运用 API 函数来扩充程序功能。本书将对 API 的基本编程作一个完整的介绍，并提供大量的有关如何深入使用 API 的实例。

本书不是介绍所有 API 函数、结构以及常数的每一个使用细节的深奥的书籍，也不是对如何利用 API 从头开始构造 Windows 应用程序的解释和说明。但是，本书通过对 API 与 Visual Basic 间的关系的详细阐述，实实在在地提供了许多如何扩展 Visual Basic 程序功能的应用程序实例。

1. 使用 Visual Basic 4.0

Visual Basic 4.0 中新出现的类特征为程序设计增加了面向对象的编程方法。这种类特征被广泛地用于封装已开发出来的许多 API 技术。通过这种封装，大部分已开发出来的代码将能方便地用在各种项目中。同时，请注意本书中提供的所有代码都是为 32 位的 Windows API 编写的，因此，必须使用 Visual Basic 4.0 的 32 位版本。这些例子与 Visual Basic 4.0 的 16 位版本并不兼容。

2. 提问和回答的格式

“如何...”的格式对本书的主题特别适用。每个“如何...”都试图清晰地定义所在章节将解决的问题或“任务”。每个“如何...”都包括如何完成任务的概述，然后是对项目代码的解释说明，最后是对所用的技巧的详细介绍。每个“如何...”结尾的评注提供了如何在已完成的任务的基础上完成其他多变的任务。随书赠送的 CD-ROM 提供了所有的代码、媒体文件、自定义控件和其他各种各样的文件。

3. 读者应具备的水平

本书可用于各种层次的读者。但是，读者应具备基本的 Visual Basic 编程知识。每个“如何...”在讨论的开始给出了一个难度等级，分为初级、中级和高级。初级的“如何...”对于任何入门级别的 VB 编程人员来说都是很简单的。中级的“如何...”对 Windows API 的讨论比较广泛，但并不深入。高级的“如何...”在应用 API 方面用到了更难的技巧，它们是提供给具有丰富经验的 VB 编程人员使用的。

4. 本书的组织结构

《Visual Basic 4 API HOW-TO》分为以下 10 章：

第 1 章　Win32 API 的基础知识

本章介绍了 API 是如何工作的基础知识。前四个“如何...”是 API 基本组件的概述和如何被 Visual Basic 访问的。后三个“如何...”提供了三个项目，向读者介绍如何通

过 Visual Basic 访问 API。

第 2 章　设备、窗口、控件和信息

设备描述表、窗口和控件是使用和理解 Windows API 的中心内容。本章提供了广泛的关于如何从 Visual Basic 程序访问并与这些 API 的不同组件进行交互的示例，例如，如何扩展组合框、列表框和文本框。读者将了解到设备描述表以及它们如何与 API 编程相关联。通过运用本章介绍的技术，读者将开始发现使用 API 超越 Visual Basic 编程环境的威力。

第 3 章　内存、文件和可执行程序

运用内存、文件和可执行程序是与 Windows 环境进行交互的主要环节。在应用程序中使用扩展内存，读入并复制文件以及启动其他程序的能力将是 Visual Basic 工具中不可缺少的部分。本章介绍的技巧在全书的讨论过程中都要用到。本章还引入了新的 Visual Basic 4.0 字节 (Byte) 数据类型并将它与新的 Unicode String 数据类型相比较，同时介绍了如何将这些新的数据类型用于 API。

第 4 章　系统配置和交互作用

有时，用户的 Visual Basic 程序需要确定正在运行该程序的计算机的准确的系统配置。本章将介绍系统配置和控制的各个方面，这包括如何获得关于系统内存配置的信息以及能使用的不同类型内存的容量。同时还介绍了如何确定视频和打印系统的功能及特性，以及具体的多媒体性能。读者将学到如何确定正在运行的其他程序，在自己的项目外检测鼠标的动作，运用剪贴板和高精度定时器。

第 5 章　绘图

在工作于象 Windows 那样的图形用户界面 (GUI) 环境中时，用户必须用到大量的绘图技巧。本章介绍了 API 提供的很多不同的绘图工具，例如：如何运用画笔、刷子、剪贴区、绘图形状等。API 提供了丰富的使用方便的绘图功能，这些功能在没有 API 时只能通过自定义来获得。最让人高兴的是这些 API 技术并不很难掌握。

第 6 章　字体和文本

在应用程序中文本是使信息能被理解的主要形式。字体是使文本具有生命力的重要的工具。本章介绍了 Windows API 如何被用于创作令人感兴趣的字体效果，如何对应用程序的文本输入获得更高的控制。读者将学到如何旋转、激活、闪烁、甚至“slash”字体文本的技巧。读者还可以发现如何对用户输入的文本框获得全部的控制。

第 7 章　位图操作

在图形用户界面中，位图图形已成为建立应用程序时的一个重要工具。利用 API，位图不仅仅可以被显示在图象框中。本章将着重说明一些不同的修改和显示图形的技巧，其中包括为一个图像增加阴影效果，用几种不同的方法淡入淡出图像，以及操纵一个位图的调色板。本章还将介绍，例如透明背景复制和动画小精灵之类的技巧。

第 8 章　菜单和鼠标

本章从几个不同的方面探讨了如何使用鼠标和菜单。Visual Basic 提供了一个完整的菜单和鼠标界面，但是通过 API，用户可以运用其他技巧来扩展对它的使用。这些技巧包括向标准菜单和弹出式菜单加入位图，以及监控菜单中的各个任务。对于鼠标来说，本章将给出如何跟踪鼠标的位置，如何控制鼠标，以及如何使用动画鼠标指针等示例。

第 9 章　多媒体

通过使用 Windows API，用户能够很容易地向应用程序加入一些多媒体功能。API 对媒体文件和设备提供了三层控制：第一层是 sndPlaySound API，第二层是 MCI Command 界面，第三层是低级多媒体 API。本章将对它们作分别介绍，并介绍如何使用这几种方法来显示声音文件以及如何浏览它的 RIFF 内容。本章还将介绍向图像加入热点和使用 MIDI 创建简单的钢琴键盘的技巧。

第 10 章　键盘、打印机和 I/O 端口

本章讨论了对键盘、打印机和通信端口进行交互、控制和监视的各种方法。API 使用户可以快速、直接、方便地从 Visual Basic 应用程序中对它们进行访问。本章还介绍了对键盘进行监视和控制，使用计算机的各个端口，以及使用打印机。

说明：

当因为书写的长度而必须将代码转入下一行时，将使用↵符号。读者在输入 Visual Basic 代码时遇到这个符号，只需接着输入代码列表的下一行即可。不要输入回车符或↵符号。

目　录

第1章　Win32 API的基础知识

Windows的32位应用程序编程接口API是一系列很复杂的函数、消息和结构，它使编程人员可以用不同类型的编程语言编制出运行在Windows 95和Windows NT操作系统上的应用程序。从例如C和C++的编程语言出发，可以访问全部Windows 32 API。Visual Basic的开发者们深有远见，预见到这种语言应能通过访问API而被方便地扩展，从而使其隐藏了大部分Windows编程的难点，同时又提供了使用Windows环境的灵活性。

具有Win32 API的Windows95和Windows NT提供了编程人员长期以来希望的一种主流操作系统应具有的特征和工具。例如，640kb内存的限制被打破了，Win32应用程序能够更好地利用Intel 386 CPU和更高级芯片所提供的潜在处理能力，对4G内存寻址的能力也不再是个问题。通过使用Windows 95，用户可以利用新的，被改进了的Windows 95界面。Visua Basic 4的动人之处在于利用32位的版本建立Windows 95应用程序变得简单易行。

第1章介绍有关Windows API的基本概念，前四个部分提供了API是如何设计和构造的说明性概念。接着提供一些简单的编程例示，以便大家从Visual Basic开始API编程。本章将给出用一套代码同时声明16位API和32位API的例子，使读者了解如何使应用程序支持这两种API。最后的几个部分将帮助读者开始从Visual Basic使用API。

本章所涉及的Windows API是：

FlashWindow	GetCursorPos	RegCreateKey	RegDeleteKey
RegDeleteValue	RegQueryValueEx	RegSetValueEx	Sleep
WinHelp			

1.1　如何Win32 API编程概念

介绍Win32 API编程的基础知识、Windows编程环境以及运用和开发应用程序的基本概念。

1.2　如何区分Win3.1 API和Win32 API

本节将向Visual Basic编程人员解释Visual Basic与Win3.1 API之间主要的差异和细节情况。

1.3　如何从Visual Basic访问API

本节将叙述使用Visual Basic提供的帮助文件和工具进行Win32声明的基础知识。

1.4　如何使用和引用Windows函数自变量

Windows编程环境包括了大量的API创建、操纵和撤消的对象。使用句柄来引用(reference)这些对象就变得非常重要。什么是句柄，以及它们是如何被引用和使用的讨论对帮助读者的理解来说很重要。另一个很重要的课题是理解如何将自变量从Visual Basic变量传递到Windows API的函数和结构中。本节将提供对这些方面的论述。

1.5　如何声明32位API的函数和结构

用Visual Basic声明API和进行32位API调用的例子，介绍函数和结构的声明机制。

1.6　如何读、写登录文件

32位的Windows操作系统用系统登录文件代替了INI文件，其中包括了系统配置和硬

件信息，这使登录文件扩展了其初始化文件的特征。本节将介绍如何创建关键字，存储值以及如何检索数据。

1.7 如何访问 Windows 帮助系统 Windows 为编程人员提供了一个强大的帮助系统。本节将介绍如何从 Visual Basic 控制帮助系统。

1.1 如何理解 Win32 API 编程概念

Windows 3.0 API 包含 300 多个函数，Windows 3.1 API 有 700 多种函数，最新的 Windows32 位 API 包括 1000 多个 API 调用。加上 API 附带的几百种 Windows 常量、消息和数据类型结构，使用户拥有了一系列用于 Windows 编程的复杂而又有效的工具。

为了理解 API，读者需要了解一些 Win32 系统是如何工作的基础知识。下面四个部分将完整地介绍从 Visual Basic 访问 Win32 API 的基础知识。我们首先看一下常用的 Windows API 函数类型。表 1-1 给出了这些函数类型的简单说明。

表 1-1 函数类型

类型	说明
Windows Management (User)	为应用程序创建和管理用户界面
Graphics Device Interface (GDI)	为 Windows 设备生成图形输出
System Servers (Kernel)（系统服务）	提供对计算机资源和一般操作系统的访问
Multimedia（多媒体）	对音频和视频设施的访问服务

这些函数与远程过程调用（RPC）及扩展库（Extensions Library）一起，构成了 Win32 操作系统的基础。下面将分别讨论这几种主要的函数类型。

1. Windows Management

Windows Management 提供了基本的窗口构造块，提供了建立管理程序输入显示及检索用户输入的基本函数。通过使用 Win32 API 的 Windows Management 层就可以实现应用程序的基本功能。其中包括确定应用程序如何响应鼠标和键盘输入，以及应用程序如何检索和处理送入应用程序窗口中的消息。使用 Visual Basic 进行 Windows 编程的最大优点是 Windows Management 层的大部分对编程人员来说是完全隐蔽的。Visual Basic 提供了标准窗口对象的所有动作。例如，Visual Basic 的窗体提供了很多不同的属性、事件和方法，它们允许程序员对窗体进行控制，其中包括处理鼠标和键盘输入，控制窗体的颜色，更新和移动窗体等。但是对于 C 或 C++编程人员访问 Windows 的不同消息和用法，而 Visual Basic 中却不一定全部访问到。但是读者可以通过使用 Win32 API 扩展 Visual Basic 来超越这些基本的属性、方法和事件，获得对应用程序更大的控制权。

Windows Management 还支持动态数据交换 DDE (Dynamic Data Exchange) 之类的系统功能和剪贴板函数。在进行图形界面的输出时，Windows Management 子系统是与 Graphics Device Interface (GDI) 配合工作的。系统管理着所有输出，所以输出被置于相应的窗口中，而且被设置成标准的图形用户界面。

2. Graphics Device Interface（图形设备接口）

图形设备接口 (GDI) 为应用程序提供了支持系统中已安装设备的功能。这些设备包括监视器和联机打印机。GDI 允许用户定义不同的绘图对象，如画笔、刷子和字体。它还提供了画线、画圆和绘制其他形状的功能，提供了进行位图操作的有力支持。

使用不同设备时可以指定不同的绘图模式。其中包括修改图形的坐标系统或利用绘图和

光栅操作以组合颜色。除了打印机和显示器外，还有其他可以利用的，例如 metafile 和内存设备描述表之类的逻辑设备。内存设备描述表对于在 Visual Basic 中使用图形是十分重要的。

3. System Server（系统服务）

System Server 系统服务为计算机资源和操作系统提供的基本工具提供了访问的途径，其中包括内存、文件系统、运行处理等。令人高兴的是 Visual Basic 的应用程序可以访问和管理这些系统服务，例如为应用程序分配内存，以及用操作文件系统的函数等。VB 提供的重要服务之一是访问和使用动态连接库 DLL（Dynamic Link Libraries），例如 API 和自定义 DLL。系统服务的工具可以确定正在运行操作系统的硬件的组成，包括是否安装有鼠标或键盘，确定屏幕分辨率等。

4. Multimedia（多媒体）

在 Windows 3.1 下，通过 Multimedia Extensions 和使用诸如 Video for Windows 之类的工具可在操作系统中加入多媒体的功能。有了 Win32 API，这些工具都被作为系统的基本部件提供给用户。通过多媒体功能，用户可以增加音频，MIDI 音乐，AVI Video，游戏杆支持和定时器。

多媒体子系统提供了几种不同的层次便于应用程序使用多媒功能。MCI Command String 和 Command Message Interface 对运行不同类型的媒体文件提供了不同层次的支持。它还提供了一个用于与 VCR 和 MIDI 序列发生器通信的接口。

低层的文件 I/O 函数可以用来操作不同文件格式的文件信息，包括波形和 AVI 数据。这些文件以资源交换文件格式 RIFF（Resource Interchange File Format）存储起来，可以通过使用这些低层函数来操作这些文件。

游戏杆接口是用来使用和操作游戏杆活动的。同样，高精度的定时器函数可以用于满足定时动画制作任务对定时功能的要求。

1.2 如何区分 Win3.1 API 和 Win32 API

对于 Visual Basic 的用户来说，Windows 16 位 API 编程与 Windows 32 位 API 编程之间的差异并不是一个很大的障碍。用户必须学会一些新的技巧，如获得登录用户标识或转入另一个新的进程，以及使用新的 Unicode 数据格式的技术等。Visual Basic 编程人员应注意的一个主要变动是 API 函数参数的数据类型的变化。在大多数情况下，所有整型（Integer）参数都被升为长型（Long）数据类型以支持 32 位的处理能力。表 1-2 列出了两种版本在数据类型上的变化。

表 1-2　16 位 Windows 与 32 位 Windows 在数据类型上的变化

VB 4 类型	长度	WINDOWS 16 位类型	WIN32 数据类型
Integer	2 Bytes	Integer，Short，Word，Hwnd Handle，WChar	Short，WChar
Long	4 Bytes	Long，LPSTR	Integer，Long，Handle Hwnd，LPSTR

请注意大多数数据类型的长度均由 2 字节变为 4 字节，这是由基于 16 位的操作系统变为 32 位的操作系统引起的。在很多情况下，从 16 位 API 向 32 位 API 变动时需要在声明调用中作一些简单的变化，将 Visual Basic 的 Integer 数据类型变为 Long 数据类型。

另一个较小的，但又很重要的变化是 API 声明的大小写问题。函数被正确调用和拼写的

最简单方法是使用 Visual Basic 提供的 API 声明工具。这些工具将在下一个部分讨论。

最让人困惑的变化是将 Unicode 编码字符串而不是 ANSI 字符串用于基本的操作系统。Unicode 编码方案为每个字符提供 2 个字节编码，而不管这个字符是否是 ASCII 码字符。Microsoft Windows NT 平台和 Windows 95 都支持这个方案。一些 32 位的 DLL 包含稍有差别的函数版本可以适应 Unicode 编码和 ANSI 字符串。函数名结尾有 A 的是 ANSI 版，函数名结尾有 W 的是 Unicode 版本（W 代表 Wide）。Visual Basic 中的大多数情况都使用 ANSI 版本的函数调用。

1.3 如何从 Visual Basic 访问 API

这一节介绍使用 Visual Basic 提供的帮助文件和工具进行 Win32 声明的基础知识。这些工具在使用 Visual Basic 进行 API 编码的过程中将会多次使用。

Visual Basic 的专业版本（Professional）和企业（Enterprise）版本在 Visual Basic 目录的\WINAPI 子目录中，用几个文件提供了有关 API 的信息。WIN32API.TXT 文件包含了 32 位 Windows API 函数中所用到的函数和 Type 结构声明以及全局常量的值。

Visual Basic 提供了 API Viewer，如图 1-1 所示，以方便用户使用 WIN32API.TXT 文件。它允许用户将 API 函数复制和粘贴到自己的 Visual Basic 代码中。可先确定一下系统中是否安装了 API Viewer，若安装了，则可以在 WINAPI 目录中找到它。启动 Viewer 并按照下面的步骤使用这个工具。

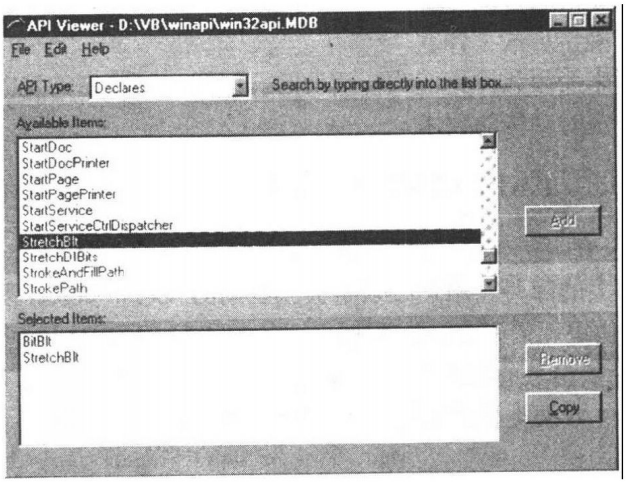

图 1-1　API 视口

1. 装载 WIN32API.TXT 文件。注意在第一次打开这个文件时，API Viewer 会询问用户是否要将文件转换成 Microsoft Access 数据库。一般情况下，这个步骤比较有用而且将提高以后装载和查找的速度。

2. 在 API Type 下拉式列表框中选定 Declares，Constants 或者 Type 以便观察 Available Items 框中的项。从 Available Items 框中选择一个你想加入到自己的 Visual Basic 项目中的项，然后按下 Add 按钮，这一项就被加入到 Selected Items 表列中。请确定一下被选定的 API 函数是否需要在 Visual Basic 中进行 Type 结构声明或者任何常量声明。用户可以检查一下声明的参数以及对于该函数的 Windows SDK 帮助(由 VB 提供)。注意，用户可以从键盘输入 API 函

数的名字，Available Items 列表框会自动地查找这一项。

3. 选定需要加到代码中的项后，按下 Copy 按钮，把 Selected Items 框中的所有项复制到剪贴板上。

4. 转入用户自己的项目并将剪贴板上的声明复制到代码中。要保证送给函数的数据符合声明的要求。

Visual Basic 的 Enterprise 版本还提供了 Microsoft Developer Network (MSDN) 启动工具箱。安装它之后，用户就可以访问 Win32 Software Developer ment Kit (SDK)。它为 Win32 API 提供了文件说明并帮助用户理解每个函数、结构和常量是如何工作的。一般对于深入开发 Win32 API 来说，购买 MSDN 将是很有价值的。

大多数情况下，当一个 API 函数被声明时，它会使用一个别名 (alias) 来代替真正的函数名。别名提供了用另一个名子调用 API 函数的方法。在多数情况下，当真正的 API 函数名是 Visual Basic 中的保留字的时候，如 GetObject，用户必须使用别名。另外，正如 1.2 节所提到的，对很多函数来说都具有 ANSI 和 Unicode 版本。多数情况下 Visual Basic 使用的是 ANSI 版本，函数名具有别名时用户就不需要使用 A 标记了。下面以 Lstrcpy 函数为例，说明使用别名的情况。

Declare Function lstrcpy Lib "kernel32" Alias "lstrcpyA" (ByVal lpString1 As String, ByVal lpString2 As String) As Long

注意，Visual Basic 将用 Lstrcpy 调用这个 API，但它实际的函数名是 ANSI 版本的 lstrcpyA。

1.4 如何使用和引用 Windows 函数自变量

Windows API 是为 C 和 C++编程人员，而不是为 Visual Basic 编程人员编写的。因此理解 C 的数据类型以及它们与 Visual Basic 的数据类型的联系，对于理解如何从 Visual Basic 访问 Windows API 来说无疑是十分重要的。表 1-3 列出了 Windows API 的数据类型以及它们在 16 位和 32 位 Windows 中的字节长度。

表 1-3 用于 16 位和 32 位调用的 Win32 数据类型

WINDOWS 类型	WIN16 类型长度	WIN32 类型长度
unsigned Int, UNIT, Int	2 Bytes	4 Bytes
short	2 Bytes	2 Bytes
Long	4 Bytes	4 Bytes
Char	1 Bytes	1 Bytes
Word	2 Bytes	2 Bytes
Hwnd	2 Bytes	4 Bytes
hDC	2 Bytes	4 Bytes
LPSTR	4 Bytes	4 Bytes
Wchar	2 Bytes	2 Bytes
Tchar	1 Bytes	1 Bytes (Ansi) 2 Bytes (Unicode)
Point	4 Bytes	8 Bytes

在向 Windows API 函数传递参数和设置类型结构值时，理解如何使 Visual Basic 参数与

这些函数接口将非常重要。表 1-4 给出了 C 语言声明类型和调用中使用的 Visual Basic 数据类型。

表 1-4　C 语言声明类型和相应的 Visual Basic 数据类型

C 声明类型	Visual Basic 数据类型
Boolean	ByVal B as Boolean
String Pointer (LPSTR)	ByVal S as String
	B () as Byte (reference first element in an array)
Integer Pointer (LPINT)	I as Integer
Long Integer Pointer (LPDWORD)	I as Long
Integer	ByVal I as Integer
Handle	ByVal L as Long
Long	ByVal L as Long
Array Pointer of Integers	I () as Integer (reference first element in array)
Void Pointer	A as Any
Char	ByVal S as String

理解用 ByVal 或 ByRef 向 DLL 函数传递一个变量之间的差别很重要。ByVal 调用只将变量的值送给 DLL。ByRef 调用将一个引用（指针）送给 DLL 。Visual Basic 默认时为 ByRef。图 1-2 说明了它们之间的差别。

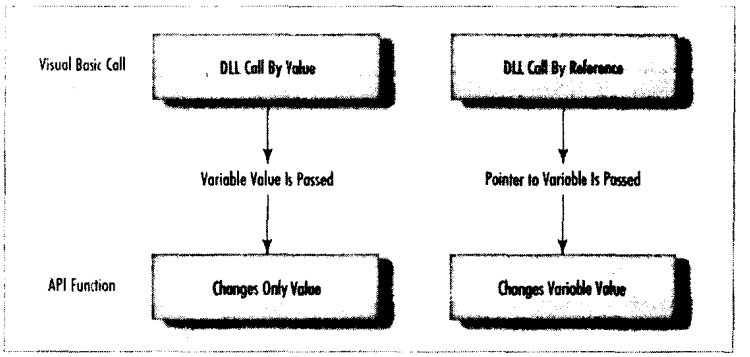

图 1-2　ByVal 与 ByRef 的比较

本书后面的编程部分将使读者看到大量的 Windows API 函数和类型声明。通过这些例子，本书将完整地介绍如何、何时和为什么要将不同的 Visual Basic 数据类型用于 API。

1.5　如何声明 32 位 API 函数和结构

难度：初级

问题：

读者需要知道如何声明和使用 32 位 API 函数。这些函数中，有一些函数的参数有一定的结构。那么应该如何声明 API 函数、API 子程序和它们使用的数据结构呢？

技巧：

声明 Win32 和 Win16API 函数的方法是基本相同的。而明显的不同点通常是在 Win16 返回整型值的地方，Win32 返回的是长整型。这一节将使用 3 个简单的 API 调用来说明结构作为参数时的用法以及在声明和调用 API 函数及子程序中的用法。

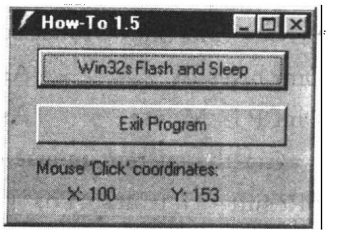

图 1-3　程序 1-5. VBP 运行中的窗体

步骤：

打开并运行 1-5. VBP，运行程序时的显示如图 1-3 所示。

按下标有 Win32S Flash and Sleep 的命令按钮，将看到窗口闪烁 3 次。每次单击窗体或任何控件，鼠标的坐标就会被显示出来。

1. 创建一个新的名为 1-5. VBP 的项目。将表 1-4 中列出的对象和属性加入到窗体中并把窗体保存为 1-5. FRM。

表 1-4　项目窗体的对象和属性

对象	属性	设置值
Form	Name	frmMain
	Caption	"How-To 1. 5"
CommandButton	Name	cmdAction
	Caption	"Win32s Flash and Sleep"
	Index	0
CommandButton	Name	cmdAction
	Caption	"Exit Program"
	Index	1
Label	Name	lblCaption
	Caption	"Mouse'Click'coordinates:"
Label	Name	lblCoordX
	Caption	"X:"
Label	Name	lblCoordY
	Caption	"Y:"

2. 将下面的代码加到窗体的 General Declarations 段。这些语句组成的程序定义了将要使用的结构以及 API 调用的函数原型。注意，窗体模块中的所有类型定义和函数声明都必须使用关键字 PRIVATE。

```
Option Explicit
Private Type POINTAPE
    x           As Long
    y           As Long
End Type
Private Declare Function GetCursorPos Lib"user32"_
    (lpPoint As POINTAPI) As Long
Private Declare Function FlashWindow Lib"user32"_
    (ByVal hwnd As Long, _
     ByVal bInvert As Long) As Long
Private Declare Sub Sleep Lib"kernell32"_
```