简 介

本书内容

您若喜爱 Visual Basic 并想用它来好好地做一番事情，本书正是专为您而作。

如果您需要的是一本指导如何产生哗哗声、如何制作一个具有三个选项的菜单、如何设置窗体标题的书籍，本书可能并不适合于您。要了解上述这些问题，我们建议您阅读 Visual Basic 的入门书或适合于初学者的手册。

充分发挥 Visual Basic 的功能

Visual Basic 问题精粹集并非改写手册，而是指导读者如何充分发挥 Visual Basic 的功能。我们认为 Visual Basic 是一套完全独立的 Windows 开发系统——它可与 C 语言程序员使用的专业 Windows SDK 相媲美，因此，我们将展示如何完成一些读者可能认为只有 SDK、C 语言才能做到的事情。本书中充满了各种技巧。

譬如，读者将学到如何自动地调整窗体的控件大小、如何使标题行闪动、如何制作能执行连续事件的按钮、如何记住窗体的大小及位置，以便下次调用时使用。读者将看到如何建立一个功能强大的编辑器来查找文本、如何制作动态的弹出式菜单，或使应用系统具有类似 NeXT 的图形界面。还有关于如何在窗口中滚动内容、文本自动对齐、仅允许数值输入到文本框中、甚至控制文本光标闪烁的速率等细节。

广泛地使用 Windows 应用程序接口(API)

本书的特色之一就是大量使用 Windows 内部的应用程序接口(API)函数。这些功能强大的函数是专业 Windows 开发人员用来制作关键应用程序的。本书将指导读者如何使用 Windows 应用程序接口来绘制透明的图标(icon)、如何制作文件对话框、如何传送消息给控件、如何在条形图上显示作业的过程、如何对复杂物体的内部进行涂色、如何制作专业的曲线图(line chart)、条形图(bar chart)及扇形图(pie chart)，甚至还有更复杂的幻灯片显示；显示 Windows 的位图及特殊的图形效果。

问答格式

问、答共分为以下几部分：控件、窗体、文本与滚动、鼠标与菜单、图形、环境与系统、外部设备、自行设计的控件、动态链接库、Windows 的应用程序接口。每个项目均包括详尽的程序解答，所有源程序、位图、字体、图标、窗体及动态链接库均存放于随书附赠的磁盘中。

附赠大量由独立软件公司提供的控件

《Visual Basic 问题精粹集》不只提供解决方案而已——随书附赠的磁盘中网罗了以下各家独立软件公司提供的控件，包括 MicroHelp、Crescent、Sheridan 软件公司及 Waite

集团软件公司．读者将找到可供多重选择的列表框、提醒注意的提醒控件、及功能类似于Excel 工具条(toolbar)的带状按钮．美观的立体按钮能以凸起或凹陷的效果显示自行设计的位图及文本．最后，语音(Sound Blaster)音效动态链接库及三个自行设计的控件也附赠给读者，以供制作数字化语音、调频合成效果、及演奏 MIDI 乐曲．

Fractal 动态链接库

"Windows 动态链接库"一章介绍如何建立 Visual Basic 与本书附赠的、用 C 语言编写的 Fractal 动态链接库之间的接口．读者将看到如何转换 Windows 动态链接库所需的参数为 Visual Basic Declare（声明）指令．本书末尾有一个详尽的附录，说明使用 Windows 的应用程序接口时，如何查阅微软(Microsoft)公司的 WINAPI.TXT 文件．此文件也包含在本书附赠的磁盘中，读者可视需要加入自己的程序中．

本书的对象

本书是针对初学者、中等程度者、及具备相当经验的程序员等不同层次的读者而编写的，每一项目的开头都标注了程序的难易程度．Visual Basic 的初学者可以只阅读"易"的部分，这些部分大多只使用纯 Visual Basic 程序代码，略具 Basic 语言基础的读者可以直接跳到"中"的部分，这些部分除了使用 Basic 程序代码外，偶而也使用简单的应用程序接口．对于极具经验的 Visual Basic 或 C 语言读者，如果想扩展程序，可以阅读"难"的部分，这些部分几乎完全使用应用程序接口及技巧来完成某些神奇的功能．

使用本书应具备的软硬件

要使用本书，读者必须有一台能运行 Windows 3.0 及 Visual Basic 软件的计算机．作者推荐使用 VGA 屏幕，但 EGA 屏幕也是可以的，因为 Visual Basic 在屏幕上只能产生 16 种颜色．还需要配置一个鼠标，可能还需要购买一本 Waite 集团出版的 Visual Basic 超级字典，该书提供超出本书范围的详尽的 Visual Basic 说明．

本书的结构

本书共分为九章，简介如下：

第一章　控件

本章介绍鲜为人知的 Visual Basic 控件的奥秘，使用了若干个功能强大的 Windows 应用程序接口．功能最强的应用程序接口之一——SendMessage 提供了坚实的基础，可用来探索 Visual Basic 未曾被发现的许多控件特征．读者将学到如何使用真正的✓而不是×来制作自行设计的确认框、如何在程序运行时添加控件、如何使用应用程序接口来制作文件对话框，以应用程序接口为基础、具备强大功能的文本编辑程序也接制作出来，可供裁剪(cut)、拷贝(copy)、粘贴(paste)到剪贴板(clipboard)上，并可使用应用程序接口来查找列表框．还有一个项目介绍如何制作具有执行连续事件功能的按钮．此外，还介绍了如何制作旧式的文件对话框，其中目录与驱动器名称位于同一个对话框中．

第二章　窗体

读者可能想把本章介绍的技术融入自己的程序中，使自己的窗体能自动居中，并且在调整窗体的大小之后，可自动调整控件。读者也将学到如何在结束程序前确保存储好程序的设置、大小及位置，并在下次重新启动时自动恢复这些设置。本章展示引人注目的技巧，包括闪烁窗体的标题行、显示／隐藏窗体或最大化窗体、以"动画爆炸画面"启动自己的应用程序等特殊技巧。读者还将学到两种避免自己的程序被他人偷窃的技巧，其一是在键入文本时不在屏幕上显示它；其二是锁定一个窗口或文件，在想存取时必须键入密码。

第三章　文本及滚动

本章介绍多种不同的技巧，以展示如何用 Windows 应用程序接口及标准 Visual Basic 程序来滚动文本及图形。读者将学到在 Visual Basic 的自动重画(AutoRedraw)性质生效或不生效的情况下，如何逐行滚动，或像电影版权声明似的连续滚动。也将学到如何使用滚动条及命令按钮来完成这些效果。读者还将学到如何水平或垂直地排列文本（而不仅是左对齐）、如何制作 README 说明文件及能查找字符串的简单文本编辑器、如何限定只接受数值串、如何删除字符串尾部的空白字符、如何调整光标闪烁的速率、如何使用滚动条预先查看屏幕的颜色色调。同时将加入一些 Windows 应用程序接口功能，使文字处理更快速、更准确，其中包括了功能非常强大的 SendMessage 应用程序接口。

第四章　鼠标与菜单

本章介绍如何制作功能更强的鼠标捕获及选取功能，如何制作自行设计的有用的、迷人的菜单，本章将使用一些非常方便的 Windows 应用程序接口，以扩展 Visual Basic 的菜单制作功能，它们包括 GetMenu、GetSubMenu、GetMenuItem、ModifyMenu。本章也将介绍 Visual Basic 的各种定时特性。

第五章　图形

本章介绍许多范例，展示用 Visual Basic 来模拟读者想制作的任何外观是多么容易的事情。模拟对象—从最近的类似 Next 的接口（上漆的铝光泽）直到 Macintosh 式的垃圾筒均被给出。读者将学到如何使用图章和色彩来填充复杂的多边形、如何在已图标化的应用程序中画图、如何制作曲线图、条形图及扇形图（就如同 Excel 中一样）。甚至还有功能强大的幻灯片展示以及特殊简报效果的介绍。本书广泛地使用功能强大的应用程序接口，并对其作了详细的说明。

第六章　环境与系统

本章介绍与 DOS 接口的技巧和从 Visual Basic 内运行 DOS 应用程序的技巧，并判断 Windows 环境各方面的状态。读者将学到如何使用许多应用程序接口来查找剩余的内存容量、查找计算机上运行的 Windows 软件版本、找出目录名称、找出键盘类型…等等。许多重要的应用程序接口均在本章给予说明，包括可供运行 DOS 程序、但找不到应用程序时不会显示错误信息的 WinExec。还有可供确定应用程序类名(class name)的应用

程序接口，以及模拟 Windows SDK SPY 应用程序的大型程序，可提供在 Windows 中运行的所有应用程序的重要信息。

第七章　外部设备：屏幕、扬声器、及串行口

本章探讨如何使用 Visual Basic 来控制个人计算机的各项外部硬件设备，读者将学到如何计算屏幕及打印机的彩色能力、如何制作音乐并加入自己的程序中、如何制作电话机插接程序配合数据机及串行口使用、如何制作自己的 Visual Basic 通信程序，Visual Basic 缺乏像 QuickBASIC 中的 PLAY 指令，所以本章将介绍读者模拟 PLAY 指令的完整宏语言，以演奏音乐。QuickBASIC 或 DOS 中许多与硬件有关的命令在 Visual Basic 中都没有了，这是由于 Visual Basic 是针对 Windows 的多任务(multitasking)环境而设计的，所有硬件中断必须由 Windows 来处理及区分。本章利用了许多 Windows 应用程序接口来管理计算机外部设备。

第八章　独立软件公司设计的控件及功能

本章介绍如何使用独立软件公司设计的控件。MicroHelp、Crescent、Sheridan 及 Waite 集团软件公司等四家独立软件公司热心地制作了其控件的特别演示版本。除了不能编译成 EXE 执行程序外，这些控件的功能与发售的商业版本完全一样。本章包括三个由 MicroHelp 公司提供的关于控件的项目，指导读者如何制作含有位图的命令按钮、如何制作可供选择多重文件名的文件对话框、如何制作能发出电话声或蜂鸣声的精巧提醒装置。Crescent 公司提供了三个利用混合语言程序模拟的、源自其 QuickPak 产品的项目，包括迅速获取窗口句柄(handle)的方法、监视并显示系统剩余资源数量、及在辅助屏幕上显示信息等。同时也介绍利用 Sheridan 公司设计的 3D Wigets 软件包来制作美观的立体界面的方法。该产品拥有与众不同的控件装置，文本具有不同程序的阴影及凹凸，能深深吸引人们的注意。最后三个项目介绍如何制作具有多媒体功能的 Visual Basic 应用程序。Waite 集团软件公司提供了处理语音卡(Sound Blaster audio board)的控件。读者将看到演奏标准 MIDI 音乐文件的控件装置、控制 11 种声音的调频合成器的控件、演奏并录制数字化声音甚至语言的控件装置等，使用户的应用程序能与终端用户对话。所有控件均存于随本书赠送的磁盘中。

第九章　Visual Basic 与动态链接库

Visual Basic 最强大的功能之一即在于其使用自行设计的动态链接库(DLL)的能力，这可大幅度地增强其功能。本章介绍如何获取其它语言（如 C 语言）编写的动态链接库，以供 Visual Basic 使用；并指出动态链接库的优点，指出它们如何像黑盒子一样工作，并指出 Visual Basic 作为完全与动态链接库操作（源程序）无关的接口这一能力。此外，读者也将学到如何转换 Visual Basic、C 语言、及 Windows 的数据类型。

附录 A　加注解的 WINAPI.TXT 文件

全书均使用 Windows 应用程序接口函数来完成 Visual Basic 内部功能所无法完成的工作。而在 Visual Basic 应用程序能使用应用程序接口之前，函数参数、特定顺序及类型

均必须在应用程序中事先声明。微软(Microsoft)公司提供了 WINAPI.TXT 文件，它提供了使用应用程序接口所需的一切信息。但该文件加密处理过，且为 Visual Basic 软件包的一部分。本书附赠的磁盘中包括 WINAPI.TXT 文件，附录 A 详细说明此文件，以协助读者更容易地使用应用程序接口函数。读者可以使用剪贴功能把 WINAPI.TXT 文件的一部分拷贝到自己的 GLOBAL.BAS 文件中。

附录 B　Visual Basic 问题精粹集磁盘

本附录详细说明随书附赠的磁盘的组织结构及内容，并详细列出目录、子目录及文件。

磁盘内容

随书附赠的磁盘中含有窗体、模块、控件、图标、位图、及音乐。所有文件均已除错并可立即运行。磁盘中也包含许多功能强大的附加程序，可增强读者的 Visual Basic 应用程序的性能。读者将找到第九章所提到的 Fractal 动态链接库及全书中均提及的 WINAPI.TXT 文件。磁盘中也包括由许多独立软件公司所设计的控件的演示版本，可供读者学习第八章的项目。读者也可找到由 MicroHelp 公司、Crescent 软件公司、Sheridan 软件公司及 Waite 集团软件公司所提供的辅助 Visual Basic Toolbox 的附加软件。其中包括一些功能十分强大的控件，以添加语音卡及与 MIDI 兼容的声音文件到自己的应用程序中，还有"华丽"的立体按钮…等等。总而言之，读者将找到 1MB 以上的 Visual Basic 源程序及附加程序。附录 B 详细说明磁盘的组织结构及内容。

如何安装随书附赠的磁盘

如附录 B 所示，随书附赠的磁盘是按章节及问题顺序而组织的。所以，如果读者想运行第二章的项目 2.2 所制作的程序（自动调整窗体的控件大小），必须找到 CHAPTER2 目录，再找到 2.2 子目录。Visual Basic 在特定层次的目录中查找文件名，以便链接文件。亦即，窗体可能在同一个子目录或上一、两层目录中查找所需的文件。因此，拷贝磁盘内容到硬盘时，请务必维持原磁盘中的组织结构。例如，读者将注意到有一个 VBHOWTO.BAS 文件放在磁盘的根目录中。每个项目的程序中均指定在上两层目录中查找该.BAS 文件。因此，读者必须使用 DOS 的 XCOPY 命令来拷贝磁盘内容到硬盘上。

必须具备剩余 1.5MB 空间的硬盘才能安装随书附赠的磁盘，也需要一个 5.25 英寸的软驱。下列说明将假设用户个人计算机上的硬盘为 C:驱动器，而 5.25 英寸的软驱为 A:驱动器。请依实际情况略作更正。例如，可能用户硬盘做过分割，而又想拷贝这些文件到 D:驱动器中，或者 5.25 英寸的软驱是 B:盘。

使用 Windows 文件管理员或 DOS 在硬盘的根目录上生成一个新的目录，以便存放磁盘中的文件。本书建议将这个新目录取名为 VBHT，但读者也可依目录命名原则而取别的名称。

要拷贝文件，必须使用 DOS 的 XCOPY 命令。XCOPY 命令可拷贝磁盘的目录及子目录结构到硬盘中。若是运行 Windows，必须切换到 DOS 提示符。务必确保命令提示符

出现的是预备拷贝文件的磁盘；例如，若要安装文件到C:驱动器中，则DOS提示符应该是：

C:＼＞

在此提示符后键入：

XCOPY A:＼VBHT ／S 【ENTER】

其中A:为软驱代码，＼VBHT为拟拷贝文件的目录，／S代表拷贝所有子目录。

请依实际情况指定软驱及硬盘目录。

用户可在计算机屏幕上看到拷贝过程的报告。一旦拷贝完毕，屏幕上将出现以下信息：

265 files copied

其中，数值代表本书付印时的文件数，磁盘中的最后文件数可能略有变化，请参阅README文件或直接查阅磁盘内容。

拷贝完成后，利用DIR命令来检查目录、子目录及文件，确保拷贝正确无误。

移动动态链接库文件

磁盘中有许多动态链接库文件，要使Windows能查找到这些文件，必须将其放到WINDOWS或WINDOWS＼SYSTEM目录中。

操作注意事项

所有文件都经过测试，可以在B:驱动器的磁盘上直接运行。如果使用其它的驱动器，或拷贝文件到其它磁盘上，可能需要重新链接部分程序并更改路径名才能正常运行。尤其在用到VBHOWTO.BAS文件（位于磁盘的根目录中）或FILEDLG.FRM文件时更需要如此。在运行任何程序时，若获得PATH NOT FOUND:（路径找不到）提示信息，就先按一下OK（确定）按钮，然后使用Visual Basic的File（文件）菜单，指定Add File…（增加文件）选项，针对该文件指定新的目录后，双击鼠标按钮。一旦重新链接完毕，务必重新存盘。

因为真正的路径名随每个人硬盘的规划情况而异，作者尽量避免在每个项目的程序中指定路径名。在要运行某个项目的程序前，必须先移动到存放该文件的目录位置。

如果都遵照以上安装程序行事，则从Visual Basic的菜单中指定Open Project（打开项目）后，必须在Open Project对话框中指定VBHT目录，然后再指定CHAPTER子目录（例如[CHAPTER]），最后再指定各项目的子目录（例如[1.1]）。此时，.MAK文件将显示在Files（文件）菜单下方，只要双击鼠标按钮即可完成操作。

程序注意事项

本书采用"＜＝"符号来表示超过一行以上的程序行。亦即"＜＝"符号标记的那几行程序事实上属于同一行，键入时不可中断，也不要键入"＜＝"符号。本书中特别将"＜＝"符号标记的那行文本向右缩进，以提醒读者注意。但是，Windows应用程序接口的声明部分则不向右缩进，因为传统上这些声明都很长。

第一章　控件

本章介绍鲜为人知的 Visual Basic 控件的奥秘，其中使用了若干个功能强大的 Windows 应用程序接口。功能最强的应用程序接口之一——SendMessage 提供了坚实的基础，借助它可以探索到 Visual Basic 中许多未被发掘出来的控件特性。读者将会学到如何使用真正的"√"而非"×"来制作自行设计的复选框、如何在程序运行时添加控件、如何使用应用程序接口(API)来制作文件对话框。也将制作一个使用应用程序接口的、具有强大功能的文本编辑程序，可将其裁剪、拷贝及粘贴到剪接板中，并可使用应用程序接口来查找列表框。还有一个项目介绍如何制作具有执行连续事件功能的按钮。此外，还介绍如何制作旧式的文件对话框，其中目录与驱动器名称位于同一个对话框中。

本章涉及的 Windows 应用程序接口

ControlhWnd	GetModuleHandle	SendMessage
GetFocus	GetWindow	SetWindow
GetModuleUsage	PutFocus	SetWindowText

1.1 制作自行设计的复选框

读者将开始学习如何制作 Option（选项）复选框，使在指定时能显示真正的确认符号(√)，而不只是简单的"×"符号。此简单技巧亦适用于将任何点阵图作为复选框的确认符号。本项目未使用任何应用程序接口。

1.2 在程序运行时添加控件

通常，在程序运行之前并不知道应加上多少控件到窗体中。读者将看到如何使用控件

数组来完成以上目的。

1.3 制作包括驱动器、目录、文件三项的列表框

1.4 迅速传送 Windows 消息给控件

欲在 Visual Basic 的控件中"尽情地"使用 Windows 的应用程序接口，必须传送消息给控件。本项目介绍如何使用功能强大的 SendMessage 应用程序接口，该应用程序接口在本书的许多项目中也将用到。

1.5 迅速清除列表框

读者将使用前一个项目提过的 SendMessage 应用程序接口来制作列表框清除程序，该程序比使用 Visual Basic 的 RemoveItem（清除项）程序要快得多。

1.6 使用应用程序接口来制作文件对话框

每个程序都需要打开及关闭文件，因此每个程序也都必须显示一个 File Open（打开文件）及 File Save（保存文件）对话框，以供用户处理其文件系统、检测错误及文件名拼写的正确性等。这些项目包括使用纯 Visual Basic 来完成以上功能，以及使用应用程序接口来完成同样的功能。使用应用程序接口使读者对文件的内容拥有更多的控制权，例如可以看见隐藏文件。

1.7 使用应用程序接口来制作功能强大的文本编辑程序

Microsoft 公司的手册中包含一个用 Visual Basic 编写的 Text Editor（文本编辑程序），不幸的是，源程序既冗长又复杂，整整三章篇幅。本项目介绍如何制作一个具有裁剪、拷贝、粘贴、及取消(Undo)功能的"迷你"型编辑程序，其中使用了 SendMessage 应用程序接口来处理窗体上的所有控件及对话框中光标的位置。读者将学到如何使用控件的句柄(handle)性质，还有一个简单的动态链接库附于磁盘中，可供读者取得任何 Visual Basic 控件的代码。

1.8 制作旧式的文件对话框

新式的 Windows 文件对话框将目录及驱动器名称置于不同的列表框中，而旧式（3.0 以前版本）的文件对话框则把驱动器用字母表示，例如[-c-]，且像目录名一样滚动，同时也包含(..)符号。Visual Basic 提供目录、文件及驱动器控件，希望用户同时使用这三者。如果用户偏爱旧式的目录／驱动器混在一起的对话框，本项目正用于此目的。

1.9 制作可供多重输入的任务表

比同一个 Visual Basic 应用程序同时使用多个不同数据来运行是一个不错的主意。多任务使得同时能有多个工具在运行，但是 Visual Basic 不允许在 Windows 的 Task List（任务表）对话框中使用多重输入，使得此计难行，应用程序接口(API)展示如何使用 Visual Basic 中隐藏的所有主(owner)窗口来完成任务表中多重输入的目标。

1.10 制作具有简易查找功能的列表框

1.11 使用应用程序接口来制作具有查找功能的列表框

列表框的自动查找意即当用户键入名称时，滚动的清单即把符合要求的内容显示出来。Windows 的帮助(Help)特征正是如此。许多含有清单以供从中选取的文件对话框也是一样。这两个项目介绍如何使用纯 Visual Basic 功能及使用 Windows 应用程序接口这两种方式来实现上述效果。

1.12 制作具有执行连续事件功能的按钮

本项目介绍制作按钮的技巧；使鼠标按钮按下之后，可连续执行一系列动作。介绍了使用 DoEvents 函数以供其它 Windows 程序运行并避免显示处理时间的技巧。

1.1 制作自行设计的复选框

问题: 难易程度：易

Visual Basic 的复选框(Check Box)并不是真正的复选框，而是用"×"来表示的。应如何制作一个框内具有真正确认符号(✓)的自行设计的复选框控件?

技巧:

使用 Visual Basic 功能来制作一个具有真正确认符号(✓)的复选框是可能的。通过调整 picture box（图片框）的控件数组即可模拟复选框的特点，并具有使用各种位图作为 checked（确认）、unchecked（不确认）及 pressed（按下）确认框的附加能力。

在 Visual Basic 的图标库中并没有类似复选框的图标可用，所以本书附赠的磁盘中提供了一些图标以便利用。读者也可以使用 Visual Basic 提供的 IconWorks 范例应用程序来自行绘制复选框位图。

步骤:

打开并运行 CUSTOM.MAK 文件。用户可以单击复选框并观察 checked（确认）、unchecked（不确认）、pressed（按下）等三个复选框的不同状态。

1. 建立一个新项目(project)，命名为CUSTOM.MAK。再遵照表1.1所示的对象(object)及性质(property)建立新窗体，且存入文件 CUSTOM.FRM 中。

图 1.1 制作中的自行设计的窗体

表 1.1 自行设计的项目窗体的对象与性质

对象(Object)	性质(Property)	设置(Setting)
窗体	标题(Caption)	Custom check box
	窗体名称(FormName)	CustomCheckbox
	底色(BackColor)	&HC0C0C0
标签(Label)	标题	Save changes
	控件名称(CtlName)	Label 1
图片框(Picture box)	自动重画(AutoRedraw)	False
	自动调整大小(AutoSize)	True
	控件名称	Checked
	下标(Index)	0
	图片(Picture)	CHECKOFF.BMP
	可见性(Visible)	False
图片框	自动重画	False
	自动调整大小	True
	控件名称	Checked
	下标	1
	图片	CHECKON.BMP
	可见性	True
图片框	自动重画	False
	自动调整大小	True
	控件名称	Checked
	下标	2
	图片	CHECKPRS.BMP
	可见性	False

一旦完成制作过程，窗体应该如图 1.1 如示。注意这三个图片框是如何重叠显示成一个的。这三个图片框仅在下标值及其位图方面有所不同，图 1.2、1.3、1.4 显示了这三个位图的放大图像。

图 1.2 OFF 位图

图 1.3 ON 位图

图 1.4 PRESSED 位图

2. 在任一图片框的Click事件中放入下列源程序。由于图片是控件数组的一部分，当单击任一图片框时，Visual Basic 就运行放在 Checked_Click 事件子程序中的该程序（Visual Basic 传递的 Index 参数为可见的图片框的下标值。因此，如果下标值非零，就意味着可见的图片框为 checked（确认）图片框；所以 unchecked（不确认）图片框应变为可见的）。否则，若下标为零，可见的图片框即为 unchecked

(不确认) 图片框；所以 checked (确认) 图片框应该变为可见的。

```
Sub Checked_Click (Index As Integer)
    If Index Then
        Checked(0).Visible = -1
        Checked(1).Visible = 0
        Checked(2).Visible = 0
        Print "Changes will not be saved!"
    Else
        Checked(0).Visible = 0
        Checked(1).Visible = -1
        Checked(2).Visible = 0
        Print "Changes will be saved when you exit."
    End If
End Sub
```

3. 当按下鼠标按钮时，Checked_MouseDown事件子程序中的下列程序使pressed (按下) 复选框显示出来。

```
Sub Checked_MouseDown (Index As Integer, Button As Integer, Shift As <=
    Integer, X As Single, Y As Single)
    Checked(0).Visible = 0
    Checked(1).Visible = 0
    Checked(2).Visible = -1
End Sub
```

4. 当鼠标按钮松开后，Visual Basic即运行在Checked_MouseUp事件子程序中的下列程序。该程序仅调用 Click 事件子程序，所以适当的图片框变为可见，而 pressed (按下) 复选框则变为不可见。

```
Sub Checked_MouseUp (Index As Integer, Button As Integer, Shift As <=
    Integer, X As Single, Y As Single)
    Checked_Click (Index)
End Sub
```

处理说明：

控件数组是一个"救命良方"。控件数组有许多用途，经常可无意中解决一些问题。在制作控件数组的三种确认 (确认、不确认、按下) 控件时，可避免分别编写程序供三者使用。

通常，当鼠标按钮被按下并被松开后，Visual Basic 的 Click 事件就会发生。由于当鼠标按钮被按下时，控件必须经常作出响应，所以，Visual Basic 提供了 MouseDown 事件。

Checked_MouseDown 获取鼠标按钮被按下的消息，并使得只有 pressed (按下) 复选框可见。然后，当鼠标按钮松开后，Visual Basic 即执行 Checked_MouseUp，调用 Checked_Click 来显示适当的复选框位图。

如果控件能有 MouseDown 或 MouseUp 事件子程序，它们即可消除 Click 事件。当鼠标按钮松开后，Visual Basic 将不会自动为该控件运行 Click 事件子程序，而是由

Checked_MouseUp 负责完成这一工作.

Checked_Click 确定所按下的图片框的下标，并使相应的图片框版本成为唯一可见的图片框. 在此，我们只在窗体中显示一些信息，而在实际应用时，若 checked 图片框可见，则可设置一个全程变量为真；若 unchecked 图片框可见，则可设置此全程变量为假.

评论:

如果不想提供 pressed 复选框，只要删除下标为 2 的图片框，并删除 Checked_MouseDown 及 Checked_MouseUp 事件子程序即可，当鼠标按假被按下并被松开后（针对图片框），Visual Basic 即调用 Checked_Click 事件子程序.

1.2 在程序运行时添加控件

问题: 难易程序：易

在设计窗体时，并不是每次都知道运行时需要多少控件. 当应用程序正在运行时，该如何增加、删除控件?

技巧:

Visual Basic 提供控件数组可供使用. 当控件声明为控件数组的一个对象时，可视需要在程序的控制下拷贝多份控件.

步骤:

打开并运行 ADDCTL.MAK 文件. 使用 Command-Button（命令按假）菜单来建立或删除 Command1 控件，Option-Button（选项按假）菜单增加、删除 Option1 控件. 图 1.5 显示添加了若干个新控件后的窗体. 单击任何命令按钮或选项按钮，都可按标签来识别指定的控件.

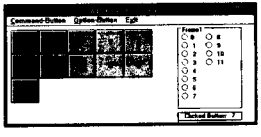

图 1.5 制作中的 Addctl 窗体

依下列步骤输入对象及程序来建立 Addctl 项目.

1. 建立新项目，命名为ADDCTL.MAK，遵照表1.2所示的内容建立新窗体，并存入 ADDCTL.FRM 文件中，确保框架(frame)在窗体中的右边，如图 1.5 所示.

表 1.2 Addctl 项目窗体的对象及性质

对象	性质	设置
窗体	窗体名称	Addctl
	标题	Add and Delete Controls
命令按钮(Command button)	控件名称	Command1
	标题	0
	下标(Index)	0
	高度(Height)	735
	宽度(Width)	855
框架(Frame)	控件名称	Frame1
标签(Label)	控件名称	Label1
	排列(Alignment)	2–Center
选项按钮(Option button)	控件名称	Option1
	标题	0
	下标	0
	高度	255

2. 使用Menu Design windows（菜单设计窗口）为窗体建立表1.3所示的菜单。

表 1.3 Addctl 菜单

标题	控件名称	下标	加速键
&Command–Button	CB		
–&Add	MenuCmdButton	0	Ctl+A
–&Delete	MenuCmdButton	1	Ctl+D
&Option–Button	OB		
–&Add	MenuOptButton	0	Shift+F1
–&Delete	MenuOptButton	1	Shift+F2
E&xit	Quit		

3. 放置下列程序到MenuCmdButton_Click事件子程序中。该子程序添加一个新的Command–Button（命令按钮）控件到窗体中，或删除最近加入的控件，要依AddDelete参数而定。

```
Sub MenuCmdButton_Click (AddDelete As Integer)
Static MaxIndex As Integer              ' # of Command Buttons
Dim CTop As Integer, CWidth As Integer, CLeft As Integer

' AddDelete = 1, Delete the last Command button.
' When adding a new Command button, place to right of previous one.
' If it would overlap frame control, place it at beginning of
'      next line.

' Finally, make control visible; new controls are not visible by default.
   If AddDelete = 1 Then
```

```
        If MaxIndex = 0 Then
           MsgBox "Unable to Delete Original Control", 48
           Exit Sub
        End If
'       -------------
        Unload Command1(MaxIndex)       ' Remove the control
'       -------------

        MaxIndex = MaxIndex - 1
     Else
        MaxIndex = MaxIndex + 1
'       ---------- ---
        Load Command1(MaxIndex)         ' Add the control
'       -------------
        CTop = Command1(MaxIndex - 1).Top
        CWidth = Command1(MaxIndex) Width
        Cleft = Command1(MaxIndex   1).Left + CWidth
        If Cleft + CWidth > Frame1.Left Then
           Cleft = Command1(0).Left
           CTop = CTop + Command1(MaxIndex).Height
        End If
        Command1(MaxIndex).Top = CTop
        Command1(MaxIndex).Left = Cleft
        Command1(MaxIndex).Caption = Str$(MaxIndex)
        Command1(MaxIndex).Visible = -1
     End If
End Sub
```

4. 放置下列程序到MenuOptButton_Click事件子程序中。该子程序向(从)Frame1中增加(删除)Option-Button（选项按钮)。

```
Sub MenuOptButton_Click (AddDelete As Integer)
Static MaxIndex As Integer
Dim CHeight As Integer, CTop As Integer, CWidth As Integer, <=
    Cleft As Integer

' AddDelete = 1, Delete last option button added.
' When adding new option button, place directly below previous one.
' If new button overlaps bottom edge of frame, start a new column.
' Finally, make button visible; new controls are not initially visible.
   If AddDelete = 1 Then
      If MaxIndex = 0 Then
         MsgBox "Unable to Delete Original Control", 48
         Exit Sub
      End If
'     ------------
      Unload Option1(MaxIndex)       ' Remove control
'     ------------
      MaxIndex = MaxIndex - 1
   Else
      MaxIndex = MaxIndex + 1
'     ------------
      Load Option1(MaxIndex)         ' Add control
'     ------------
      CHeight = Option1(MaxIndex).Height
      CTop = Option1(MaxIndex - 1).Top + CHeight

      CWidth = Option1(MaxIndex).Width
      Cleft = Option1(MaxIndex - 1).Left
```

```
        If CTop + CHeight >= Frame1.Height Then
            CLeft = CLeft + CWidth
            CTop = Option1(0).Top
        End If
        Option1(MaxIndex).Top = CTop
        Option1(MaxIndex).Left = CLeft
        Option1(MaxIndex).Caption = Str$(MaxIndex)
        Option1(MaxIndex).Visible = -1
    End If
End Sub
```

5. 放置下列程序到Command1_Click事件子程序中，在任何时候，只要窗体上的Command-Button（命令按钮）被按下，该子程序即识别控件中的哪一个按钮被指定。

```
Sub Command1_Click (Index As Integer)
    Label1.Caption = "Clicked Button: " + Str$(Index)
End Sub
```

6. 放置下列程序到Option1_Click事件子程序中，在任何时候，只要窗体上的Command-Button（命令按钮）被按下，该子程序即识别控件中的哪一个按钮被指定。

```
Sub Option1_Click (Index As Integer)
    Label1.Caption = "Clicked Option " + Str$(Index)
End Sub
```

7. 放置下列程序到Quit_Click菜单事件中。该程序可结束应用程序。

```
Sub Quit_Click()
    End         ' Exit the Application
End Sub
```

处理说明：

控件数组为 Visual Basic 的核心之一，它是许多特性的基础，其中包括程序运行时动态添加控件的能力。单一的控件在设计时通过设置其 Index（下标）性质，即可转变为控件数组。在本项目中，我们先从下标为零的两个控件数组着手。在程序运行时，要加入新的控件，需要使用 Load CtlName(Index)指令。CtlName（控件名称）必须已存在于窗体中，且成为一个控件数组，而 Index（下标）值必须尚未被该控件名称所使用。在程序运行过程中，当加入新的控件时，其可见性预置为 Off，即使其位置及大小均符合缺省值。

在 MenuCmdButton_Click 事件子程序中，AddDelete 标志(flag)表示是否要增加或删除一个控件，MaxIndex 静态变量追踪以往加入的命令按钮的总数，由于是静态变量，因此，在子程序调用过程中其值保持不变。一旦加入新的控件，就检查是否能将其插入到 Frame1 的左边，如果不行，就插入到前一行的第一个控件下方。

删除控件可使用 Unload CtlName(Index)指令，并减少 MaxIndex 数目。

MenuOptButton_Click 事件子程序可增加、删除 Option button（选项按钮），MaxIndex 变量同样用来追踪 Option button（选项按钮）的数目。由于 MaxIndex 是在子程序中声明的，每个子程序都有各自的备份，因而不能共享。Command button（命令

按钮）从左到右加入，选项按钮从上到下加入，这表明，为新加入的控件安排位置是多么容易的一件事情。

Command1_Click 及 Option1_Click 这两个事件子程序演示 Visual Basic 如何让用户知道究竟选择了数组中的哪一个控件。Index 参数则用以识别控件数组的各个元素。

评论：

本项目演示如何在程序运行时加上两个 Visual Basic 控件、命令按钮及选项按钮。然而，任何标准控件均可加入控件数组中，一个窗体最多可有 255 个控件。换言之，窗体并没有下标性质，不能作为控件数组。

需特别注意的是，当控件已包括在某"容器"中时（例如本例中的 Option1 控件），加入该控件数组的控件也将放于相同的容器中。

1.3 制作包括驱动器、目录、文件三项的列表框

难易程度：易

问题：

现在希望合并文件选择框，以方便用户指定及打开某个文件来操作。应如何取得各种相关的控件以实现以上目的？

技巧：

文件选择这一基本功能是驱动器、目录及文件列表框的内部功能。我们所需做的就是输入一小段程序到适当的地方，以便当用户更改时，各部分能相互沟通，确保顺利运行。此外，我们还要加上一个模式(pattern)字符串，以供用户输入要显示的文件类型。我们也将加入一些错误检测功能，以便在用户忘记了插入磁盘时，程序不会无所适从。

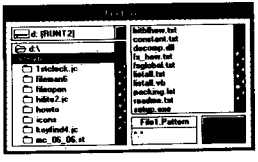

图 1.6 FileBox 项目窗体及其设计过程中的控件

步骤：

打开并运行 FILEBOX.MAK，FILEBOX 项目可依下列步骤来制作：

1. 建立一个新项目，命名为FILEBOX.MAK。建立如图1.6所示的新窗体，其中含有控件；依表 1.4 所示建立所有性质(property)。未特别注明的性质请采用其缺省值。完成后存入文件 FILEBOX.FRM 中。

表 1.4 FileBox 项目窗体的对象及性质

对象	性质	设置
窗体	窗体名称	MainForm
	标题	FileBox
驱动器列表框(Drive list box)	所有性质	default
目录列表框(Direcory list box)	所有性质	default
文件列表框(File list box)	所有性质	default
标签	边框风格(BorderStyle)	1—Fixed Single
	标题	File1.Pattern
文本框(Text box)	边框风格	1—Fixed Single
	文本(Text)	*.*
命令按钮(Command Button)	控件名称	ExitDemo
	标题	Exit

在设计时首次拖放到窗体中时，驱动器、目录及文件列表框将反映当前目录，正如 CurDir$ 所返回的值一样。图 1.6 所示的文件及目录反映了本范例的当前目录。

2. 放置下列程序到Drive_Change事件子程序中:

```
Sub Drive1_Change ()
    On Error GoTo Drive1Error
    dir1.path = Drive1.Drive
Exit Sub
Drive1Error:
    Beep
    If Err = 68 Or Err = 71 Then
        Msg$ = "Error #" + Str$(Err) + " No Floppy in the Drive!"
        MsgBox Msg$, 48
    Else
        Msg$ = "Error #" + Str$(Err)
    End If
    Resume
End Sub
```

3. 放置下列程序到Dir1_Change事件子程序中：

```
Sub Dir1_Change ()
    File1.path = dir1.path
End Sub
```

4. 前置下列程序到Text_KeyDown事件子程序中:

```
Sub Text1_KeyDown (KeyCode As Integer, Shift As Integer)
    If KeyCode = 13 Then      'If Enter Key is hit
        File1.Pattern = Text1.Text
    End If
End Sub
```

5. 放置下列程序到ExitDemo_Click事件子程序中:

```
Sub ExitDemo_Click ()
```