



# 第一章 基本观念

传统上,计算机的使用,都是由用户输入文本命令和数据到计算机中,计算机处理后,将文本或数字结果打印出或显示在屏幕上,人和计算机间的往来,主要是靠文本,使用者需记住命令,否则无法使用计算机。

最近计算机使用的趋势,是利用用户图形界面,简称为 GUI(Graphical User Interface);用户利用屏幕上的图形和鼠标,就可以很容易向计算机下达指令,执行所需的工作。利用 GUI 的应用软体,较易学,也较易使用,用户再也不必记住命令,计算机的使用因而更为普及,整个社会的生产力也可因而提高。

用户图形界面的研究,始自 70 年代美国全录(Xerox)公司在加州 Palo Alto 的研究中心(Xerox PARC),该中心的研究人员首先开发出利用图标(icon)和鼠标(mouse)的图形界面,全录公司的研究成果,在商业化上并不成功,其研究计划也就没有再继续开展下去。美国苹果计算机公司的创办人之一 Steve Jobs,曾到过全录的研究中心参观,全录图形界面的概念,后来就化身出现于苹果公司的麦金塔计算机上。麦金塔计算机上的用户图形界面相当受使用者的欢迎。

美国微软公司(Microsoft, Inc)开发了 Windows 图形界面,供 IBM 或与 IBM 相容的计算机使用,Windows 的第 1 版和第 2 版并不很成功,直到 1990 年 6 月推出第 3 版,Windows 才大为畅销,个人计算机上使用图形界面的大势因而确定,今天,新推出的软件,若没有使用图形界面,大概很难为市场接受。

使用 GUI 的软件,较易学、易使用,但这种软件设计起来较为困难。程序员要设计在 Windows 上使用的应用程序时,通常都用 C 语言,还要有一些程序开发的软件,如 Microsoft 公司的 SDK(software Developer's Kit),这些工具的价钱高,只有很专业的编程人员才有时间和精力去研究、使用,一般的人只能望之兴叹。

Visual Basic 的出现,使 Windows 应用程序的开发变得相当容易。过去,一个大的应用程序,大概有 90% 的程序代码是用来处理用户界面,只有 10% 的代码是真正用来处理应用工作。利用 Visual Basic 来设计程序,编程人员再也不必写用户界面的代码,Visual Basic 会自动产生这些代码,程序设计的效率因而提高数倍。此外,Visual Basic 并不难学。

Visual Basic 是用来设计 Windows 应用程序的工具,容易学习、使用。读者如果曾用 Basic 语言或其他程序语言设计过计算机程序,就可发现 Visual Basic 是一个革命性的新工具。

## 1.1 面向对象

Visual Basic 是面向对象的程序语言,面向对象的程序制作方法(OOP, Object-Orient-

ed Programming)是最新、最进步的程序制作方法,此种方法将代码和数据组成自主的对象。在一个程序内制作的对象,可以很容易的用在另外一个应用程序内,因而可节省程序开发的时间。

在一般面向对象的程序语言中,如 C++,对象是由代码和数据组成,它是一个抽象的概念。在 Visual Basic 内,对象是真正看得到、可以规定大小的东西,例如:按钮、文本框等,Visual Basic 使面向对象的程序设计向前迈进了一大步。

在面向对象的程序里,对象和程序的往来,是利用事件(event),来完成的。例如:按钮(button)是一个对象,当用户在钮上单击鼠标按钮(click)则是一事件(event)。发生此事件时需要执行的代码,就放在名为 button\_click 的过程里(在 Visual Basic 内,称为 event procedure)。也就是说,以

object\_event

为名的代码,会在对象发生了此个事件后执行。一个对象可以发生的事件有很多,例如,按钮的事件有单击、拖放、向下按键等。

因此,Visual Basic 程序的方式,和传统程序的执行方式很不一样,传统的应用程序,单独占用计算机的时间和屏幕,程序从头开始执行,一直到最后,然后终止,将计算机的控制权交回给 DOS。但用 Visual Basic 写出的程序,是 event-driven 事件驱动,也就是说,程序的执行是由事件(event)驱动。程序启动以后,就一直可使用,但除非用事件去驱动,否则程序就闲着不动,计算机的时间和屏幕都可供其他程序使用,因此,用户可同时使用许多应用程序。

Visual Basic 的每一对象都有许多属性,就像一种东西有许多属性一样,例如:一块石头有重量、大小、颜色、硬度……等属性,Visual Basic 内的对象,各有其属性,例如:命令按钮(command button)有宽、高、字体大小、名称……等属性。在设计程序时,可以规定各属性的值,例如:规定某一命令按钮的宽度为 2000 单位,高度为 600 单位。程序在执行时,也可用代码更改对象的属性。

Visual Basic 的功能很强,有很大的原因是由于程序里能随时更改物件的属性。例如:若更改对象的高度属性,则该对象大小马上就随着更改,而若更改对象的颜色属性,该对象的颜色也会马上更改。

## 1.2 Visual Basic 的由来

在个人计算机上用得最多的语言是 Basic,Basic 代表 Beginner's All-purpose Symbolic Instruction Code,它最初是在 1965 年,由美国的达特茅斯学院开发出来的。

BASIC 最初设计时,是要让它很精简,也不想要有高强的能力,其目的是让学生很快就学会一种语言,可以自己写程序来解决问题。

1974 年,Ede Roberts 使用微处理器拼凑了第一台个人计算机,并开始在市场上销售,销路比预期好很多。那时候,有位哈佛大学二年级的学生,名为 Bill Gates,打电话给 Ed Robert,说他有 Basic 语言解释程序(Intepreter),问 Ed 是否要将 Basic 语言附在机器上

卖,Ed 说好,你拿来给我看看。Bill Gates 和他的朋友,于是连夜加班,写出一个 Basic 的解释程序,拿去给 Ed 看。到 1980 年时,Bill Gates 的 Basic 解释程序,已随附在各种个人计算机上卖出了五十万套。到今天为止,已有数千万台的个人计算机在使用,大部分的计算机都附有 BASIC,无可置疑,BASIC 语言是世界上最普遍的语言。

Bill Gates 的公司,就是今天个人计算机软件业的盟主 Microsoft 公司。Microsoft 附在 DOS 4.01 版以前的 Basic 解释程序,各版本都很类似,称之为 GW-BASIC。GW-BASIC 的能力有限,专业程序设计人员无法用它来开发大的程序。1982 年时 Microsoft 将 BASIC 大幅改进,推出 QuickBasic,让程序设计人员能用 QuickBasic 开发应用程序。

Visual Basic 虽承袭 QuickBasic,但向前迈进了非常大的一步,使专业程序设计人员可以用来开发在窗口上执行的应用程序。Visual Basic 包括了以前 GW-BASIC 里所有的功能,其程序结构又很像 QuickBasic。读者若用过 C 或 Pascal 语言,当可发现 Visual Basic 和这些语言有很多类似之处。

读者可用 Visual Basic 在二、三分钟之内设计一个简单而有用的程序,也可花较多的时间,设计一个较复杂、可以在市场上销售的程序。也就是说,Visual Basic 功能齐备,相当具有弹性,而且学起来也不难。

用 Visual Basic 设计应用程序,其方法和以前设计程序的方法不同,学用 Visual Basic 设计程序,就像进入一个新奇的世界,本书是个入门介绍,希望能引导读者轻松愉快的进入这个新的天地。



## 第二章 Visual Basic 程序的制作

本章说明如何启动 Visual Basic、Visual Basic 屏幕各个部分的功能,并制作一个简单的应用程序。

### 2.1 Visual Basic 的启动

在能使用 Visual Basic 以前,Visual Basic 必需安装,有关 Visual Basic 的安装,请参阅附录 A。

启动 Visual Basic 的方法有许多种,可以从 DOS 的提示处启动,也可以先启动 Windows,再启动 Visual Basic。Visual Basic 一般存在名为 VB 的文件目录内,因此,从 DOS 提示处启动 Visual Basic 命令为

```
Win\vb\vb
```

启动 Windows,再启动 Visual Basic 的步骤为:

1. 在 DOS 提示处,键入 Win
2. 选用程序管理员(Program Manager)的选项(O)下的自动转换图标(M)(Minimize on Use)

这个步骤使 Visual Basic 启动后,程序管理员(Program Manager)会缩小成一图标,使屏幕上不会因开启的窗口太多,而显得拥挤。这个步骤可以不执行。

3. 选用 Microsoft Visual Basic 程序组里的 Microsoft Visual Basic 图标(参阅图 2-1)或

选用文件(F)菜单下的执行(R)选项,然后在执行的对话框内,键入\VB\VB,再选确定

或

利用文件管理员(File Manager),使存放 Visual Basic 的文件目录内的文件名显示在一窗口内,然后在 VB.EXE 的图标上,双击鼠标按钮。

图 2-2 的屏幕出现,此屏幕可用来开发程序。

### 2.2 Visual Basic 的屏幕

图 2-2 的屏幕,看起来有点拥挤,内容很丰富,但读者很快就会熟悉此屏幕。

屏幕最上方的标题,为 Microsoft Visual Basic[design],意指目前是在设计模式,可以

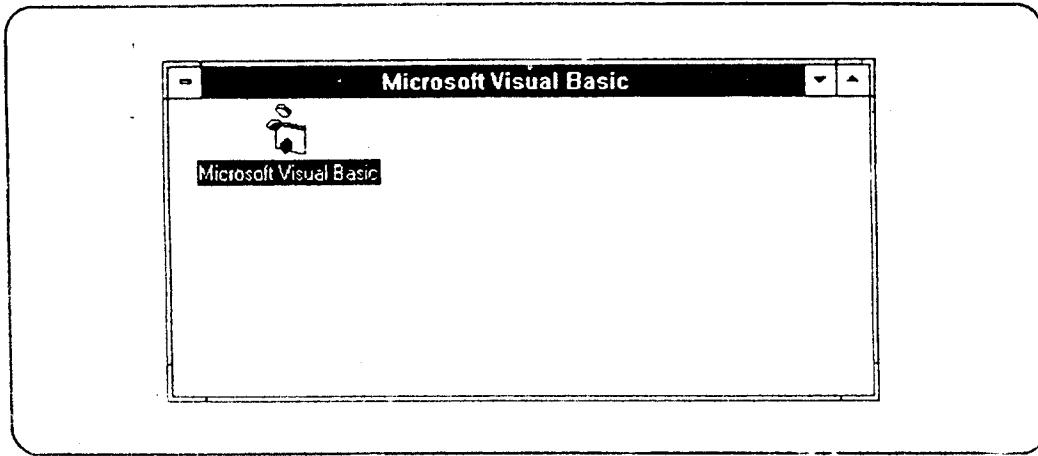


图 2-1 有 Visual Basic 图标的程序组

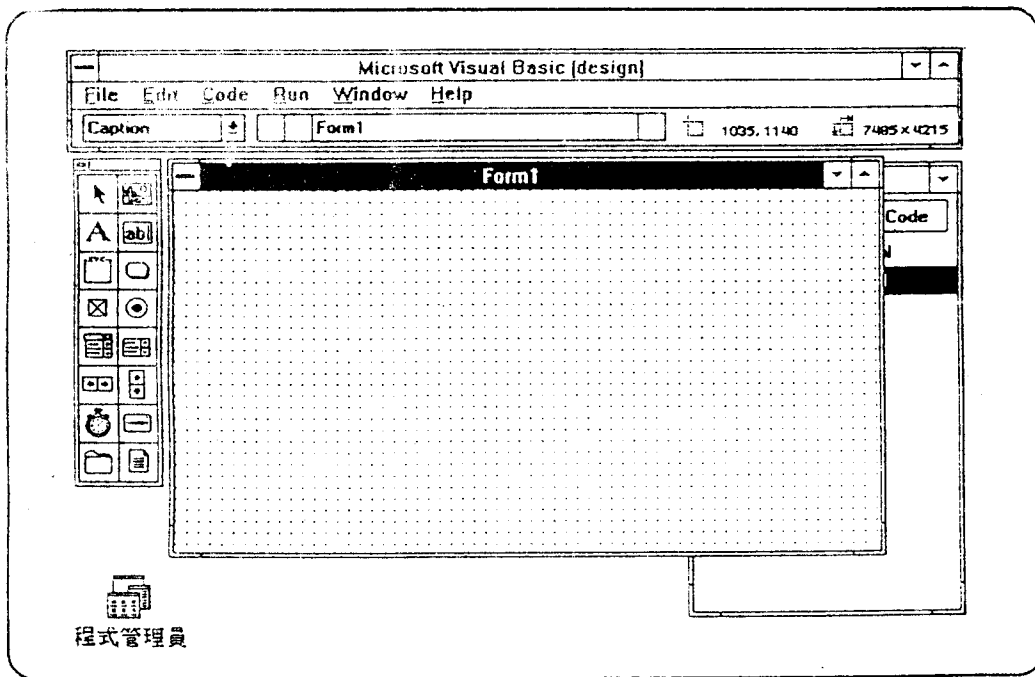


图 2-2 Visual Basic 的屏幕

用来设计应用程序。

在屏幕的中央的窗口,名为 Form1(格式)。Form 的窗口可用来设计数据的输出和输入,也就是说,Form 是应用程序的用户界面。在用 Visual Basic 设计应用程序时,通常都是先在 Form 窗口上设计用户界面。

在 form 上,可以画出一些对象,这些对象就叫做控制,例如,图 2-3 中,Form1 内有一

文本框,此框可以用来显示数据或使用户键入数据,此文本框为一控制。控制包括文本框、钮、列表框等,图 2-4 为一些控制的例子。

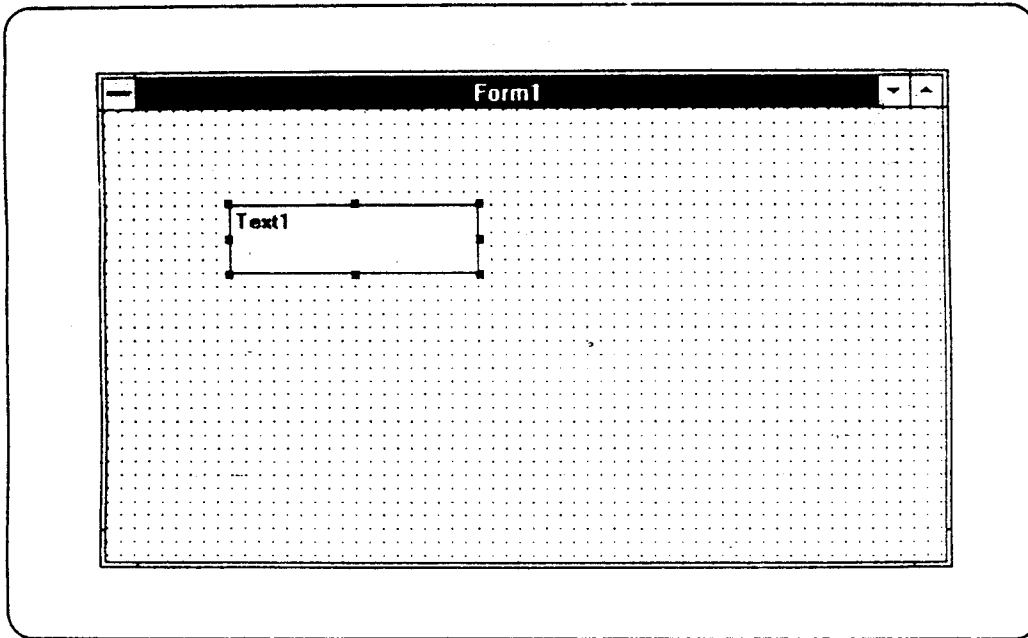


图 2-3 Form1 内有 Text 对象

用 Visual Basic 设计程序时,有一大部分的工作是在制作格式和控制。在制作格式和控制时,并不需要编写代码,Visual Basic 会自动产生这些代码,程序设计人员只需要使用屏幕上的工具画出即可。

### 2.3 工具箱(Toolbox)

屏幕左边有二栏工具(图 2-5),这些工具可用来在格式上绘制控制,其用法很像 Windows 调色盘(Paintbrush)内绘图工具的用法。这些工具的用法,本书后面会陆续讨论。本章用到两个工具,即文本框(Textbox)和命令按钮(Command button)。文本框可用来接受用户输入文本或显示文本。Command button 为命令的按钮,可让用户在钮上按鼠标按钮,代表数据要执行某件工作。

### 2.4 制作一简单的程序

现在,我们开始制作一个简单的程序。

1. 启动 Visual Basic

图 2-2 会出现在屏幕上。

2. 在文本框工具(即工具箱第二行、第二列的图标)上单击鼠标按钮。



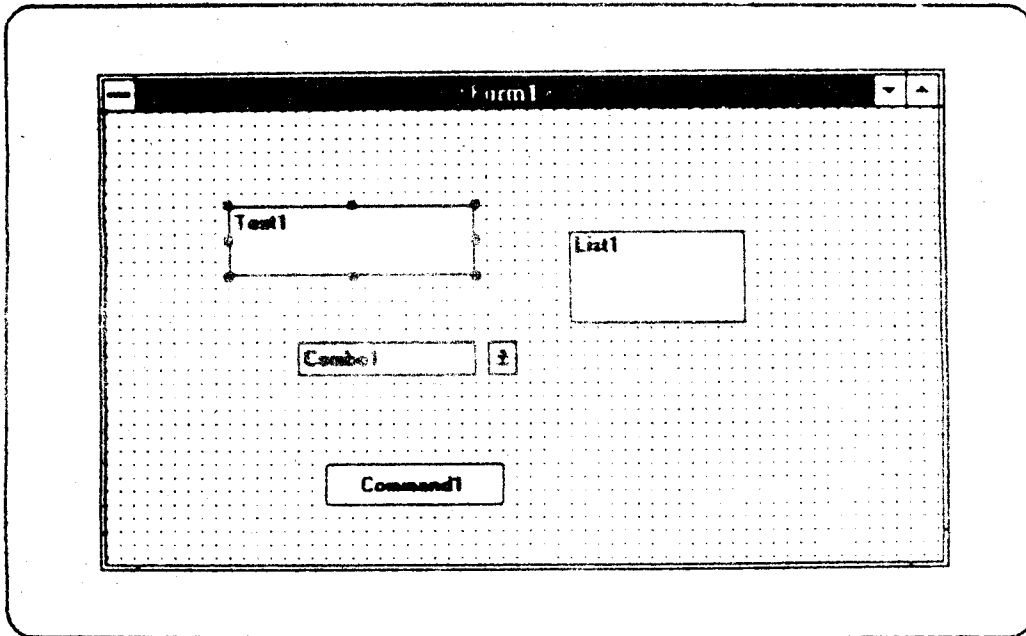


图 2-4 组控制的例子

3. 将鼠标指针移到 Form1 窗口内,单击左鼠标按钮,同时拖动鼠标,屏幕上出现一长方形的框后,释放鼠标按钮。

屏幕上的 Form1 窗口,现在如图 2-6 所示。

4. 在命令按钮工具(即工具箱第三行、第二列的图标)上单击鼠标按钮。

5. 如上面第 3 步,在 Form1 内画出一命令按钮。

屏幕现在如图 2-7 所示。

在 form 上制作了控制后,必需规定这些控制的属性(properties),属性的规定必需用属性条(properties bar),此属性条位在菜单条(menu bar)的下方,如图 2-8 所示。

Visual Basic 内的每一个对象,都有许多属性,就像每一样东西(如石头)有许多属性一样,例如石头的重量、大小、颜色、硬度等。Visual Basic 内对象的属性包括对象的名称、内容、大小、位置、颜色等。当在 form 上制作了一个对象时,Visual Basic 将此对象的属性规定为缺省值(default)。

这些值可以用属性条来更改,此外,也可以用代码来规定,当应用程序在执行时,依用户的某些动作(如单击鼠标按钮)而变动。

属性一览表内现在显示 caption,caption 为命令钮(command button)上所显示的字,缺省值为 command1,如图 2-7 所示,我们要将此字改为比较贴切的字。

6. 键入 Click Button

所键入的字,出现在属性条上,也同时出现在命令按钮上。

7. 在属性条左方的向下箭头上,单击鼠标按钮,然后选用 CtlName。

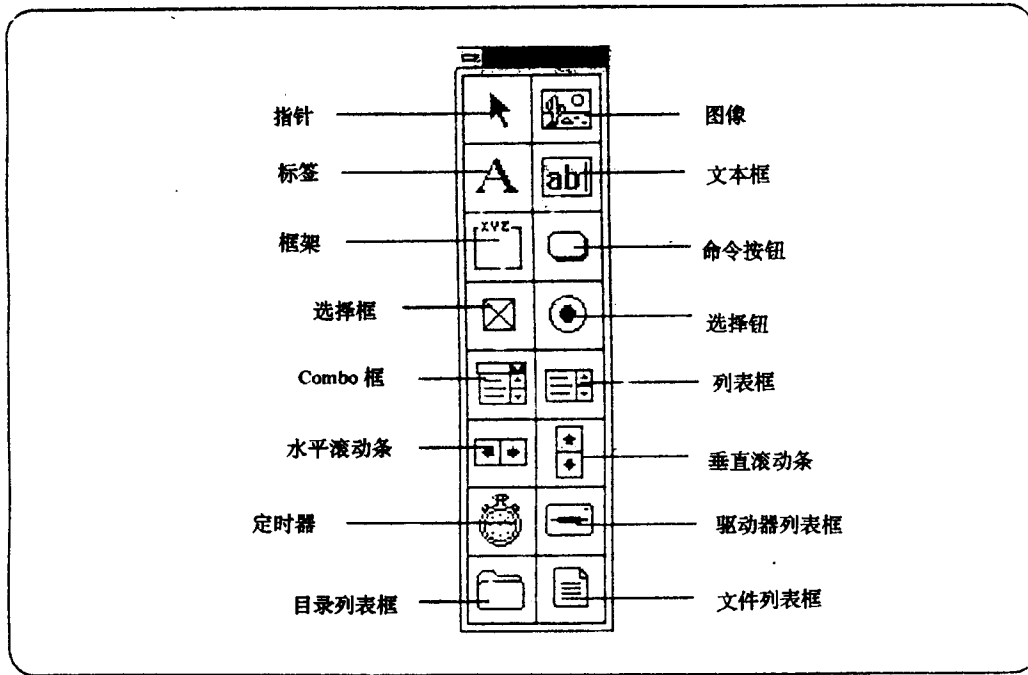


图 2-5 工具箱内的工具

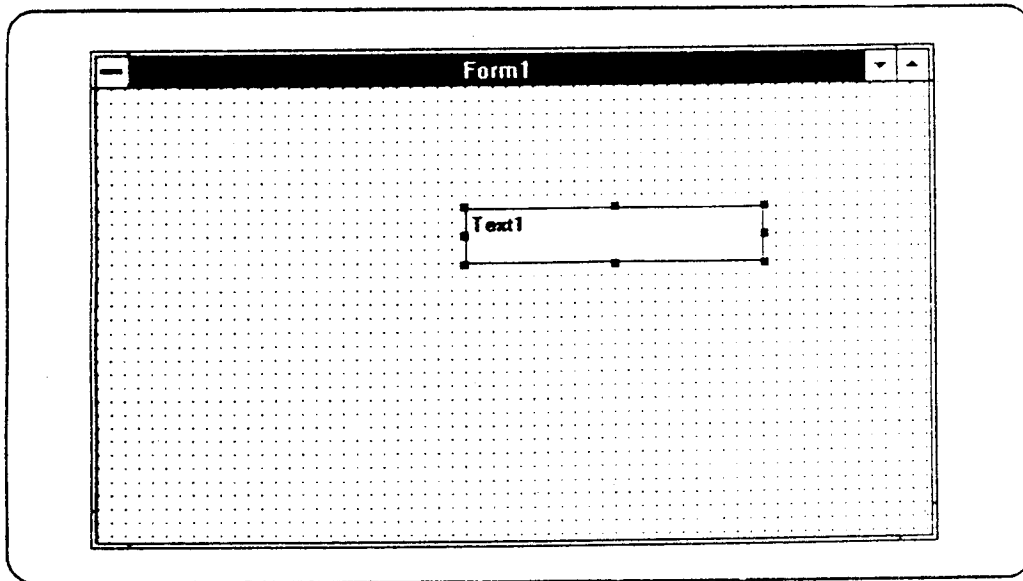


图 2-6 有文本框的 Form

CtlName 意指 Control Name, 表示对象的名称, Visual Basic 设定命令的 Control Name

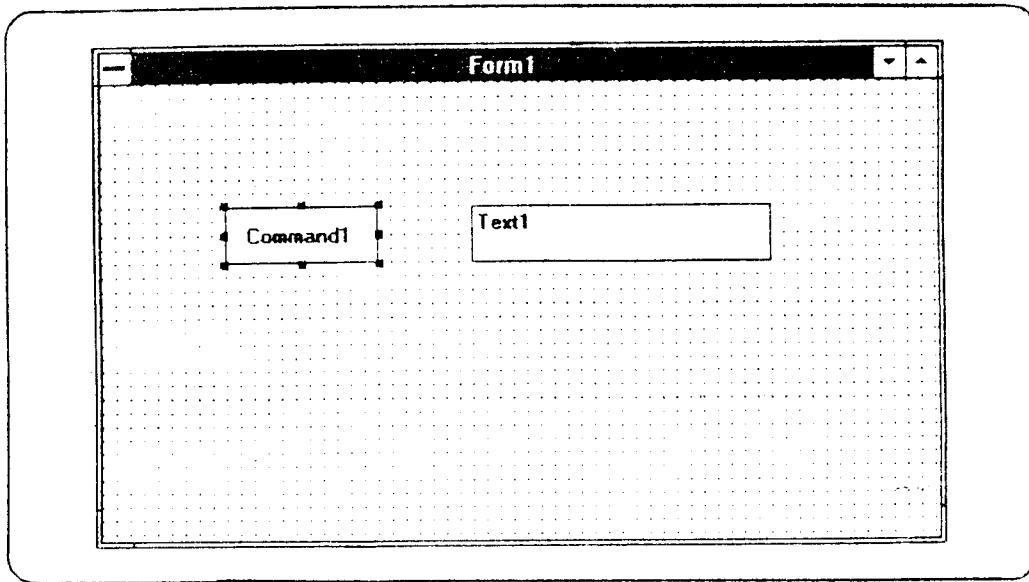


图 2-7 有文本框和命令按钮的 Form

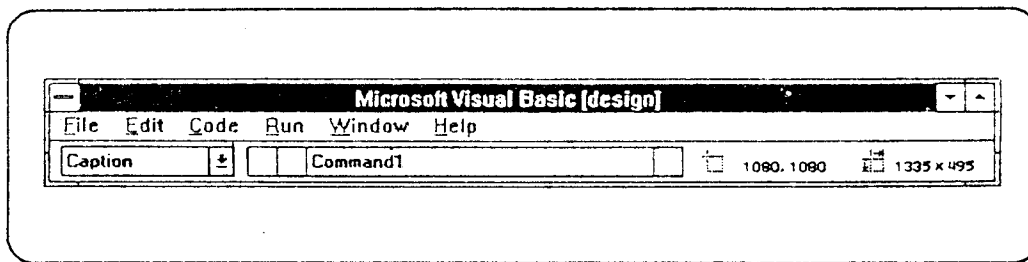


图 2-8 属性条和菜单条

为 command,我们要将此名称改为 HiButton。

8. 键入 HiButton

按钮上的字和按钮的名称(CtlName)已规定好,下面我们规定文本框(text box)的名称(CtlName)和所显示的文本(text)。

9. 在 text1 的文本框内,单击鼠标按钮

文本框的框上,现在有点点标签,代表这是目前所选的对象,现在可以规定此物件的一些特性。屏幕现在如图 2-9 所示,此图显示目前文本框的 CtlName 为 Text1,我们要将此名称改为 Wordout,意指显示文字的框。

10. 键入 Wordout,并按 Enter 键

11. 在属性条(properties bar)左边的箭头上,单击鼠标键,然后按数次 t 按钮,使 text 出现在上方,然后在箭头上单击鼠标按钮。

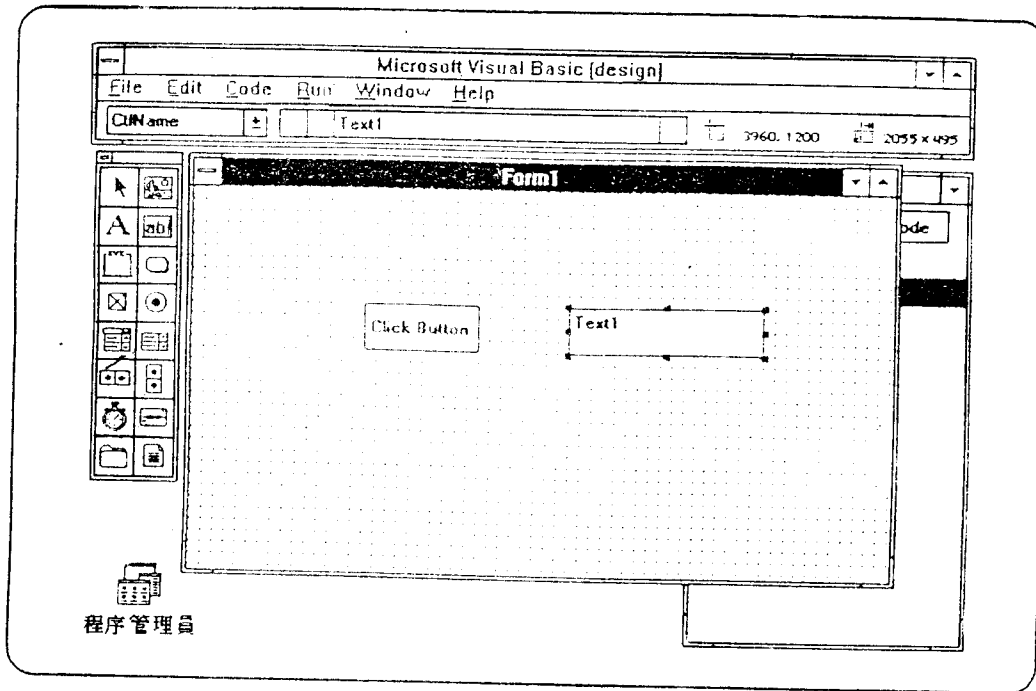


图 2-9 用黑点标示的控制为目前所选用的控制

上面这个步骤,其目的是要选用 Text 属性(property)。

12. 在属性条(properties bar)中间的 Setting box(设定框)上,单击鼠标按钮。

13. 单击数次 Delete 按钮,将框的字删除

对象内文本框内的 text1,现在也被删除了。

对象的名称(CtlName),并不会显示在 form 内,这些名称只有在编码时才会用到。

Text 和 Caption 两个属性的值,会显示在 form 上。前面按钮的 Caption 规定为 Click Button,此字显示在钮上,文本框的 text 属性,其值为空白,因此文本框内没有显示任何数据。

由于控制的名称(CtlName),并没有显示出来,若要看某一控制的名称,则必需打开该控制的代码窗口(Code Windows)。例如,假设我们要看 Click Button 钮的名称(CtlName),

14. 在 Click Button 钮上,双击鼠标按钮

图 2-10 的窗口出现,此窗口上的第一行代码为 Sub Hibutton-Click(),Sub 意指子程序(Subroutine),Hibutton 为此钮的 CtlName,Click(单击鼠标按钮)代表击一次鼠标按钮,因此,若在图 2-10 中的两行命令中,键入一些代码,则程序执行时,若在 Hibutton 钮上单击鼠标按钮,就会执行这些代码。

对于每一对象,都有许多事先规定好的事件(例如:Click),读者若要查看按钮所能执行的事件,可以在过程框(图 2-10 中用 Proc:代表)右边的箭头上,单击鼠标按钮,下拉式菜单会出现,

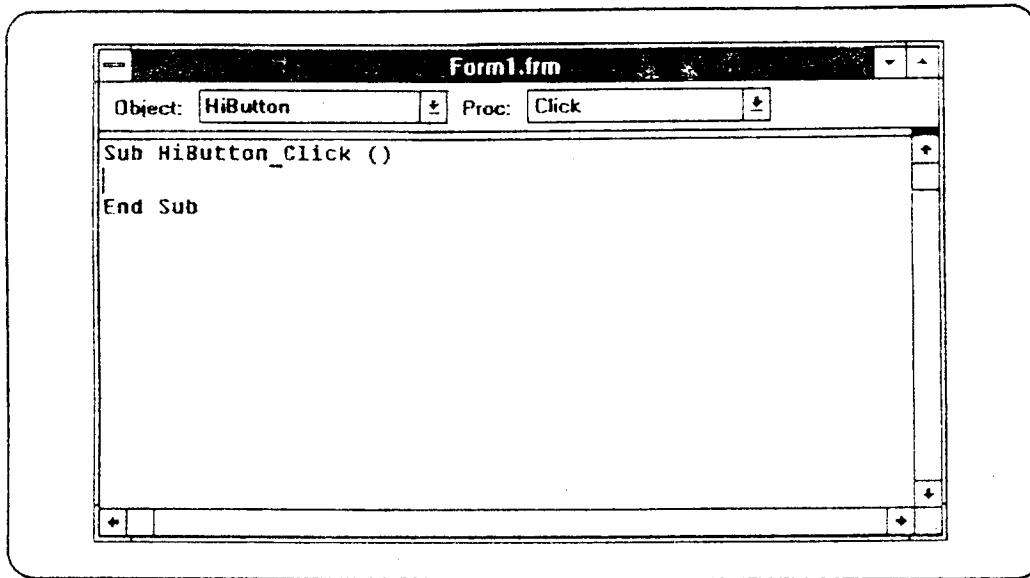


图 2-10 代码的窗口

15. 在 Proc 右边的箭头上,单击鼠标按钮

窗口现在如图 2-11 所示。

16. 选用 GotFocus

窗口内的第一行,现在出现 Sub HiButton-GotFocus(),GotFocus 为一个事件,现在可以规定 HiButton 钮若发生 GotFocus 事件(event),应用程序需如何反应。由上述的练习可知,子程序的名称,用对象的名称和事件名称来代表,中间加一线,如下所示:

Objectname \_ eventname

objectname 为对象(object)的名称,如前面的 HiButton,eventname 为事件(event)的名称,如 Click。

此时规定若在 HiButton 的钮上单击鼠标按钮(Click),应用程序需有何反应,因此 Proc 内需选用 Click。

17. 使 Proc 的框内出现 Click

18. 键入下列的代码:

```
IF Wordout.Text = "" Then
    Wordout.Text = "Good Morning"
Else
    Wordout.Text = ""
End IF
```

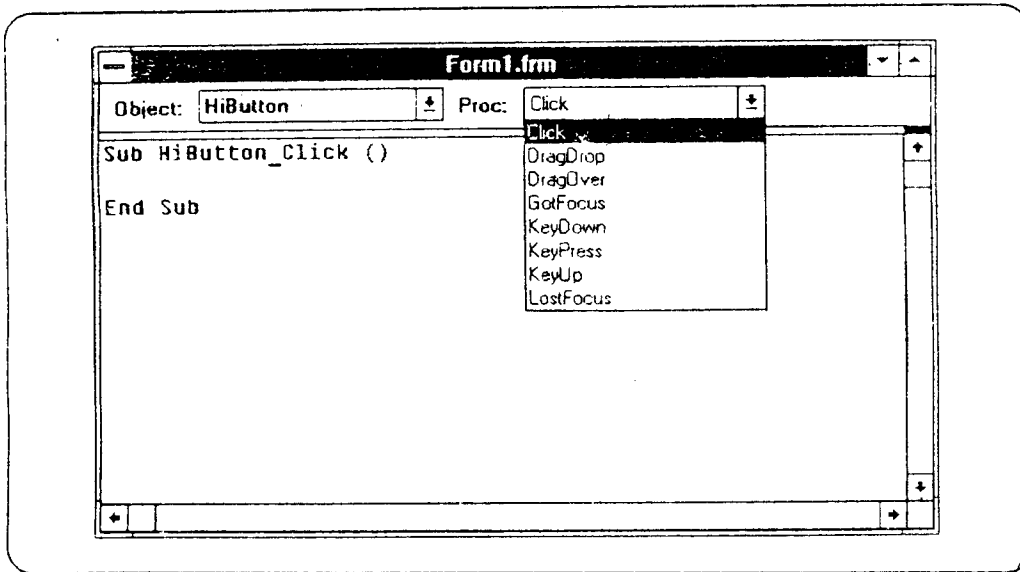


图 2-11 下拉式菜单列出可以选用的事件

窗口现在应如图 2-12 所示。此代码的意义为：若 Wordout 文本框内的文本(Text)(用 Wordout.Text 表示)没有任何东西，则 Wordout 的文本变为 good Morning, Good Morning 会出现在文本框内，若 Wordout 文本框内已有数据(即 Else)，则 Wordout 文本框内的文本变为无任何东西。也就是说，当文本框内有数据时，在钮上单击鼠标按钮，则数据会消失，再击一次鼠标按钮，Good morning 又会出现。

读者需注意的一点，就是对象(object)的属性(properties)，其初值可以用属性条来更改，当程序在执行时，可以用下列的语句来更改：

```
objectname. propertyname = setting
```

objectname 为对象的名称，propertyname 为对象的某一属性，setting 为属性的值，例如，前面的 Wordout. text = "Good Morning"，此处 Wordout 为对象名称，Text 为此对象的一个属性(property)，而 Good Morning 为其值。

程序现在已制作好，可以执行了。

19. 按 F5 键(或选用 Run 菜单下的 Start)

20. 在 Click Button 上单击鼠标按钮

Good Morning 会出现，如图 2-13 所示。

21. 在 Click Button 上单击鼠标按钮

Good Morning 现在消失了。读者可以自行多击几次鼠标按钮。

22. 选用 Run 下的 End

或

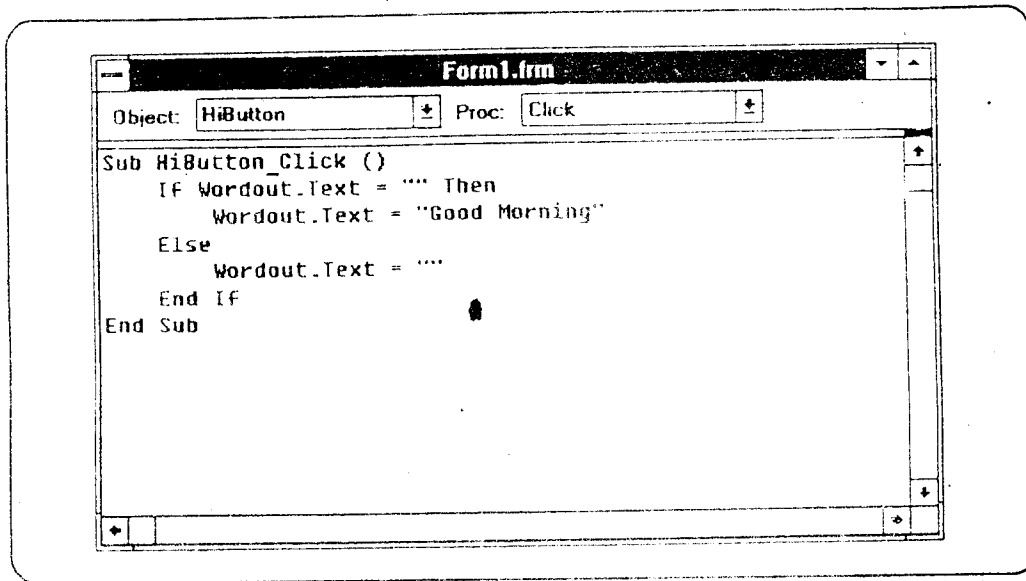


图 2-12 键入代码的窗口

在 Form1 左上角控制框上, 双击鼠标按钮  
结束程序的执行, 回到 design 的模式。

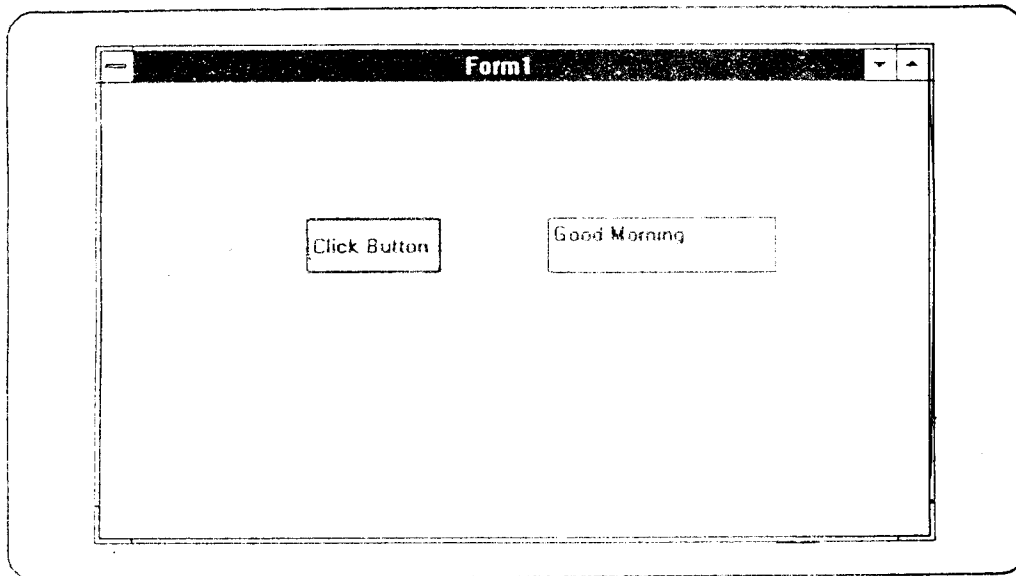


图 2-13 执行程序时的窗口

## 第三章 界面的制作和 Help

制作 Visual Basic 应用程序的第一个步骤,就是制作程序和用户之间的界面,也就是在 Form 上画出文本框、命令按钮等,并规定这些控制的属性。第二章已简单讨论如何在 Form 上画出控制,本章前半部对于在 Form 上画出控制做更进一步的论述,此外还说明如何将制作的程序存储。

本章后半部讨论 Visual Basic 的 Help(帮助)。Visual Basic 的帮助很完整,读者若有需要可以先阅读第 3.4 节,以利以后随时使用。

### 3.1 工具箱

工具箱(toolbox)内有一组工具,可用来画出、移动 form 上的控制,也可以更改控制的大小。工具箱内的工具共有 16 个,除了最左上角的 pointer(鼠标指针)外,都是用来在 form 上画出控制。指针无法画出控制,其目的是用来移动或调整 form 上已画出的控制的大小。

打开工具箱的方法为:

- 选用 Windows 菜单下的工具箱(toolbox)

要使工具箱从屏幕消失,可以

- 在工具箱左上角的钮上双击鼠标按钮

#### 3.1.1 绘制文字标示

利用工具箱内的各种工具绘制控制,其步骤都一样,都是先在工具箱内的某一工具上单击鼠标按钮,表示要使用该工具,然后在 Form 上制作所要的控制。画出文字标签的控制也不例外,其步骤如下:

1. 工具箱若没出现在屏幕上,则选用 Window 菜单下的 Toolbox。
2. 在左边一系列的第二个工具上(即以 A 表示的工具),单击鼠标按钮。
3. 将鼠标指针移到 form 上(鼠标指针变成十字)要绘制控制处的左上角。
4. 按左鼠标按钮,同进拖动鼠标。

屏幕上会出现一个方框。

5. 当方框的大小符合需要时,释放鼠标按钮。

Label 的目的是用来画出文字标签,文本框的目的是用来显示数据,或接受用户输入数据,而命令按钮是用来接受用户下达命令。请读者自行练习在 Form 上画出一些控制。



### 3.1.2 另外一种划控制的方法

除了用上面的方法制作控制外,另外有一个较简单的方法,其步骤如下:

- 在工具箱内的某一工具上,双击鼠标按钮

该工具所绘制的控制,会出现在 form 的中央。控制可移动、拷贝更改大小或删除,这是下一节的主题。

## 3.2 控制的拷贝、移动、调整大小和删除

### 3.2.1 控制的拷贝

若要制作数个相同而且大小一样的控制,可以利用拷贝命令,加速控制的制作。控制可以拷贝到同一 form 内,或不同的 form 内或 project 内。

拷贝控制的步骤如下:

1. 选用要拷贝的控制
2. 选用 Edit 菜单下的 Copy
3. 在新的控制要存放的 form 上单击鼠标键组
4. 选用 Edit 菜单下的 Paste 或按 Shift+Ins 按钮

若拷贝到同一 form,屏幕会出现一个对话框,问你是否要制作控制数组(Control array),有关控制数组的使用方法,请参阅本书后面,此处先选否(N)。

### 3.2.2 控制的移动

控制的移动相当简单

- 将鼠标指针移到控制上,然后拖动该控制至新的地点,再释放鼠标按钮。

同时移动两个以上的控制

1. 在第一个要移动的控制上,单击鼠标按钮
2. 按着 Ctrl 键,然后同时在其他要移动的控制上,依次单击鼠标按钮  
在所选的控制的边缘,会出现灰色的四方形小点,如图 3-1 所示:
3. 拖动任一控制至新的地方。

其他所选用的控制,也会随着一起移动。

### 3.2.3 删除控制

Form 上面的控制可以删除,其步骤如下:

1. 在要删除控制上,单击鼠标键。
2. 若要一起删除数个控制,则按 Ctrl 按钮,同时要在要删除的 Control 上,依次单击鼠标按钮
3. 按 Del 按钮