

第一章 基本界面的建立

内容提要

- 基本窗口
- 控制图
- 对象(Object)
- 实例设计
- 实例执行
- 文件访问
- 制作执行文件(.EXE)
- 范 例

使用 VisualBasic(以后简称 VB)首要条件当然必须具备有 WINDOWS 操作系统 (中文、英文版皆可), 而且 VB 已经装入(Install)到 WINDOWS 的应用程序中。启动 VB 途径有二: ①在 WINDOWS 的程序管理员(Program Manager)中敲 VB 肖像(ICON)二次或②在 DOS 状态下键入“WIN VB\VB”。

当 VB 装入计算机完成后会出现如图 1-1 的基本窗口。

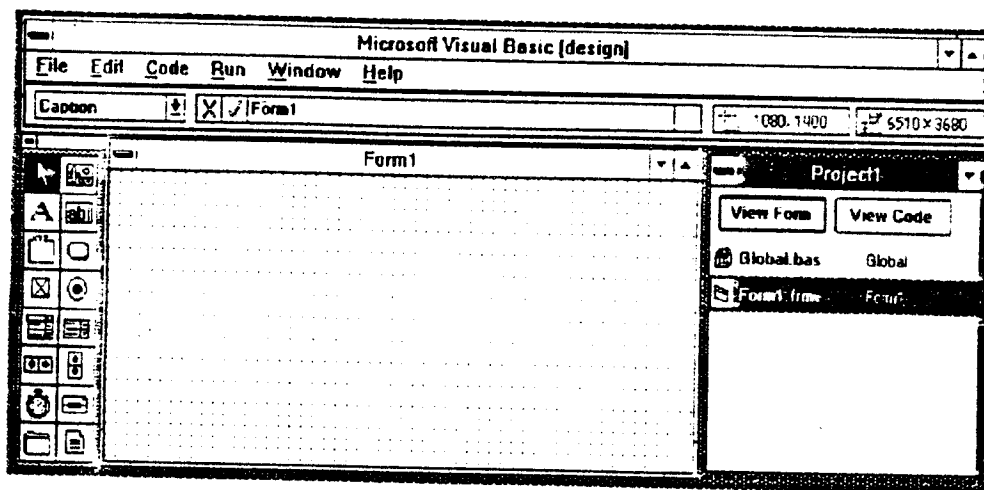


图 1-1 基本窗口

这些基本窗口包括: ① VB 设计窗口 (Design) ② 表格窗口 (Form) ③ 工具框视窗

(Tool Box)④计划窗口(Project)。现分别概略说明如下:

1. 设计窗口(Microsoft Visual Basic[design])

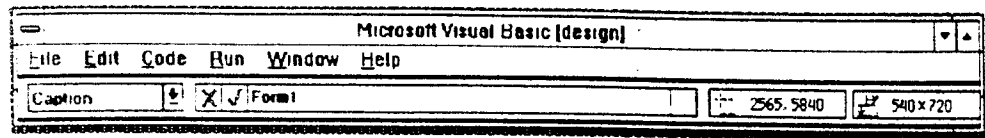


图 1-2 设计窗口

- 如上图 1-2 所示, 此窗口类似一般典型的 WINSOWS 窗口。它包括有:
- ①标题区(Title Bar)用来标示窗口的标题, 如图 1-2 上的“Micr osoft Vis ual Basic [desingn]”。
 - ②控制区(Control Menu Box)位于左上端, 它用来控制此窗口之还原(Restore), 移动(move), 大小(size), 最小化(minimize), 最大化(maximize), 开闭(close)与切换至(switch To....)等工作。
 - ③窗口大小控制区, 位于标题区的右边来放大或缩小窗口。
 - ④菜单条(Manu Bar)位于窗口上端第二列, 如上图 1-2 所显示的本窗口可用命令有: “File”, “Edit”, “Code”, “Run”, “Window”以及“Help”, 现分别说明这 6 种命令如下:

A. 文件(File)
如下图 1-3 所示:

File	Edit	Code	Run	Wind
New Project				
Open Project...				
Save Project				
Save Project As...				
New Form				
New Module				
Add File...			Ctrl+F12	
Remove File				
Save File			Shift+F12	
Save File As...			F12	
Print...				
Make EXE File...				
Exit				

图 1-3 文件命令表

此命令群最主要的工作是用来访问或清除设计表格(Form)或计划(Project)的文件, 打印程序码或表格、编译(compile)程序码使成为.EXE 文件, 以便可以在 WINDOWS 环境下直接执行。

B.编辑(Edit)

如下图 1-4 所示:

Edit	Code	Run	Window	Hel
Cut		Shift+Del		
Copy		Ctrl+Ins		
Paste		Shift+Ins		
Paste Link				
Delete		Del		
Grid Settings...				
Align to Grid				

图 1-4 编辑命令表

此命令群最主要的工作是用来程序码编辑时的“剪下”(cut), “拷贝”(copy), “清除”(Delete), “贴上”(Paste)以及调整表格上面那些小点间的距离(Grid Settings)。还有一个特殊功能就是可执行 DDE 动态数据交换时的拷贝(copy)和贴上(Paste Link)工作。

C.程序码(Code)

如下图 1-5 所示:

Code	Run	Window	Help
View Code		F7	
Load Text...			
Save Text...			
Find...			
Find Next		F3	
Find Previous		Shift+F3	
Replace...			
New Procedure...			
Next Procedure		Ctrl+Down	
Previous Procedure		Ctrl+Up	
✓ Syntax Checking			

图 1-5 程序码命令表

此命令群最主要功能是可以调出编辑程序窗口(view code)以便编辑程序码, 访问文本文件(Text)。因为在 VB 所建立的文件(BAK 或.FRM)无法在其他编辑器上读出来, 如果转换成文本文件就可以在其他编辑器上进行编辑了。它又可以用来打开新的函数(New Procedure)以便建立普通函数。

D. 执行(Run)

如下图 1-6 所示:

Run	Window	Help
Start		F5
End		
Restart		Shift+F5
Single Step		F8
Procedure Step		Shift+F8
Set Next Statement		
Show Next Statement		
Toggle Breakpoint		F9
Clear All Breakpoints		
Set Startup Form...		
Modify Command\$...		

图 1-6 执行命令表

此命令群最主要的功能是当编码完成后, 可以立即以 Interpreter 方式执行(Start 或 Restart), 同时提供一些调试修改的命令(Single Step, Toggle Breakpoint...).

E. 窗口(Window)

如下图 1-7 所示:

Window	Help
Color Palette	
Immediate Window	
Menu Design Window	
Project Window	
Toolbox	

图 1-7 窗口命令表

此命令群可用来装入调色板(Color Palette), 菜单设计窗口(Menu Design Window), 计划窗口(Project Window)以及工具框(Tool Box)。通常计划窗口和工具框都

是自动装入，除非关掉它。

F.辅助说明(Help)

如下图 1-8 所示：

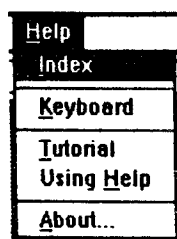


图 1-8 帮助说明命令表

此命令群是提供详细的帮助说明，当用户遇到困难时。

位于菜单条(Menu Bar)上，它用来赋予或调整每个对象(Objects,所谓 Object 是 Form 或 Color 的合称)属性的区域。

属性条左边长方格内有一个向下箭头，敲一下可以列出所有能供选择的属性（此方格内的属性会随对象的不同而提供不同的属性），如下图 1-9 所示。

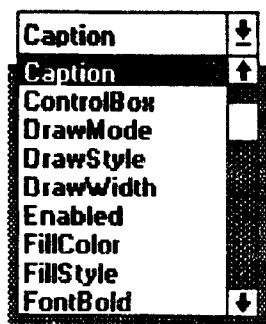


图 1-9 属性列表

兹先举三个常用的属性作一简单的说明，其他的属性留待以后在例题中再作解释：

属性名称

说明

CtlName(控制名称)

每一个对象（表格、控制图）在程序码中的名称。
每一个对象都有其缺省的控制名称，但是最好将它
改为有意义的名称以便识别。

Caption(标题)	每一个对象的外在标题名称, 它也有缺省的标题名称。
Text(文本)	文本框、列示框和混合框等控制图上所显示的内容。

属性条中央空白处称之为设置区。它可用来改变属性值(例如图 1-1 上中央表格的 Caption 属性值为"Form1", 如果想将此 Form1 改为其他名称, 则必须在此区域键入新标题名称), 新设置属性的方式按属性的不同分成三种不同的类型:

- a. 第一种: 自行键入新属性值。例如标题(Caption)名称, 或文本框的文本内容(Text)等都必须自行键入。
- b. 第二种: 有任选的属性。例如"BorderStyle"、"ControlBox"、"DrawStyle"等属性, 当你选择这些任选的属性时, 在此中央区的右边会出现一个向下箭头, 敲一下该箭头, 所有可能的选择会被列示出来, 当选完其中之一属性后, 敲一下它左边的"V"记号以示确定。下面图 1-10 列出"BorderStyle"的 4 种选择。

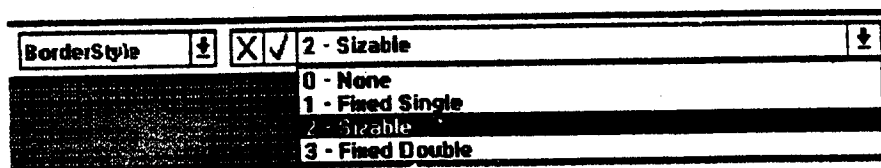


图 1-10 属性选择列表

- c. 第三种: 当属性是有关图像(Picture)或肖像(Icon)时, 中央区右边会出现三个小点(...), 如下面图 1-11 所示。敲一下此三小点, 画面上会出现一个叫"Load Picture"的对话框窗口。它会引导你如何从外面装入图像到像框(Picture Box)或表格(Form)上。

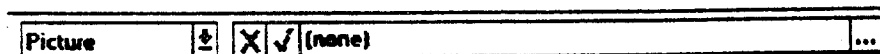


图 1-11 图像属性

属性条最右边有二个长方形方框。左边一个是显示活动的对象(Active object)的位置(左边数字表示距右边的距离, 右边数字显示距上面的距离)。右边一个是显示活动对象的大小(左边数字显示该活动对象的长, 右边数字显示该活动对象的高)。这些数字都是以 twip 作为单位(1440twips=1 英寸; 567twips=1 厘米)。此处的长度单位(厘米或英

寸) 是以输出打印机上的长度为准。

2. 表格窗口(Form)

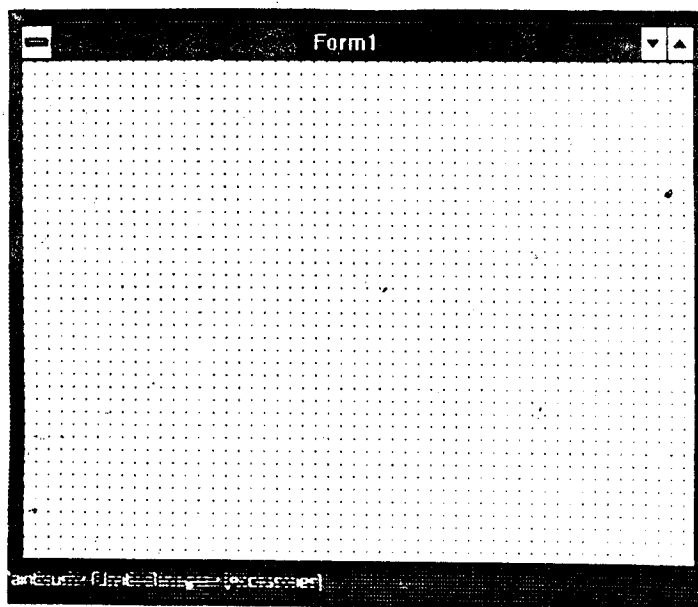


图 1-12 表格窗口

如上面图 1-12 所示, 表格窗口提供设计规划用户界面(Interface)的窗口或对话框(Dialog Box)。每一个表格窗口最多可容纳 2551 个控制图(control)。

3. 工具框窗口(Tool Box)

工具框是在设计时用来将控制图移置到表格上的一套工具。如下面图 1-13 所示, 它位于基本窗口的左边。

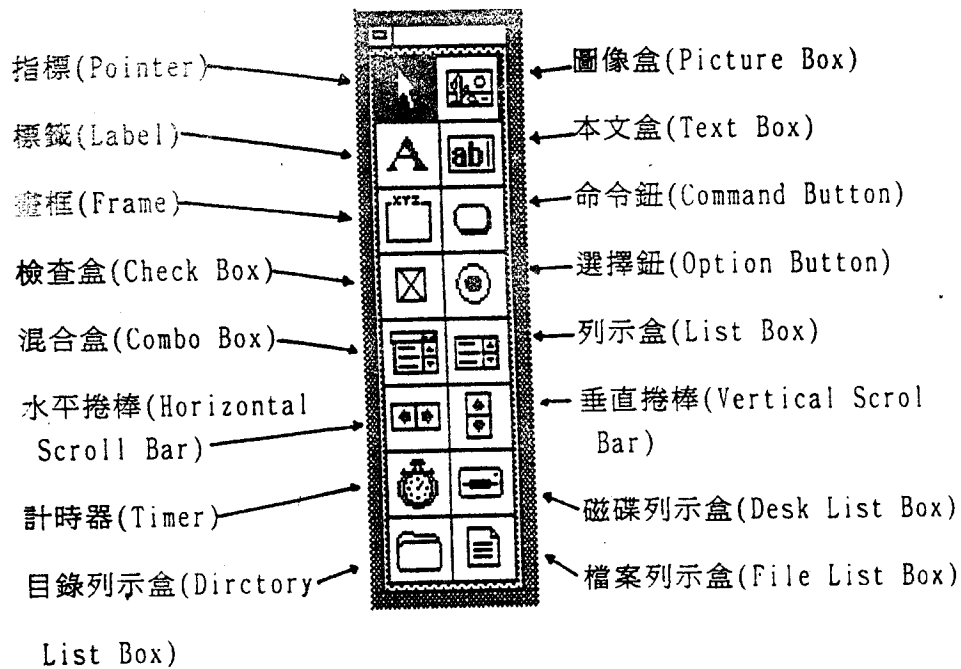


图 1-13 工具框

4. 计划窗口(Project)

如图 1-14 所示，计划窗口是用来显示出 1 个应用程序（1 个完整的计划）内所有的文件。它可能包括：

A. 表格文件(Form File):

1 个表格可能由 1 个或多个事件函数(Event Procedure)或通用函数(General Procedure)组成。1 个计划至少包括 1 个表格文件。

B. 模块文件(Module File):

所有通用函数和声明(Declaration)部份的集合称为模块。它能够被不同表格内的程序码调用。通常较大或较复杂的应用程式才使用模块文件。

C. 全局模块文件(Global Module File):

全局模块（最上层）存放着能被各程序取用的变量、常量以及自定类型数据等信息。一个计划只能有一个整体模块文件。

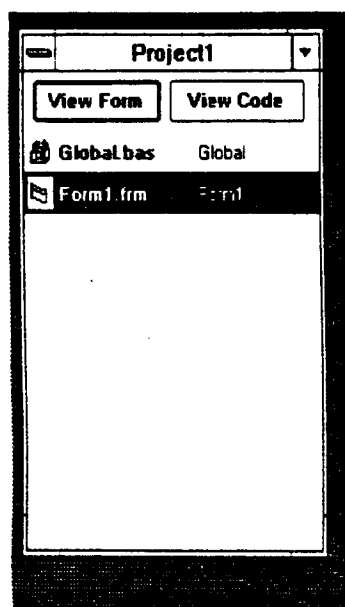


图 1-14 计划窗口

计划窗口除了显示出1个计划内的所有文件外，同时也可以利用它上面的“View Form”和“View Code”两命令按钮来调出各文件的表格或程序码。

1-2 控制图

所谓控制图(control)就是放在表格上的图式对象(Graphical object)。它的主要功能是用来接受输出的信息，作为用户与计算机间的界面。

1. 控制图的种类

控制图大部份都可以由左边的工具框中找到并画出，除了菜单(Menu)必须由菜单条中 WINDOW 项下的“Menu Design Window”来产生。下面列出所有控制图的项目及简略说明：

A. 图像框(Picture Box):

图像框不但可以在上面画图，键入文本，还可以从VB外面装入，拷贝位图像(Bitmap)或肖像(Icon)。

B. 文本框(Text Box):

文本框可以接受文本的输入及输出。

C. 标签(Label):

标签可显示出（输出）文本的数据，但无法作为输入的介面。

D.画框(Frame):

画框可以用来集合性质相同的控制图。

E.命令按钮(Command Button):

当用户选择此按钮时,即可执行一个反应或动作。

F.选择按钮(Option Button):

对于一些项目较少的选择可用此控制图让用户选择。例如WINDOWS内应用程序, Paintbrush 中打印选择(⊙)。

G.检查框(Check Box):

检查框可以让用户选择 True / False或 Yes / No的控制图。例如要跳出WINDOWS 时的“☐ Safe Chage”, 敲一下☐内出现, 再敲一下以变成☒。

H.混合框(Combo Box):

混合框是列示框和文本框的混合, 它可经由键入方式或选择方式来输入数据。

I.列示框(List Box):

列示框可以显示一系列的项目供用户挑选。就如属性条上的属性列表一样。

J.垂直滚动条(Vertical Scroll Bar):

水平滚动条可在一系列区域或表格中显示数据或位置的工具。

K.水平滚动条(Horizontal Scroll Bar):

水平滚动条可在一系列区域或表格中显示数据或位置的工具。

L.计时器(Timer):

计时器可以作为执行程序计时之用。

M.磁盘列式框(Disk List Box):

磁盘列示框可以查找或切换有效磁盘机, 就如WINDOWS上文件管理员 (File Manager)一样。它可与文件列示框及目录列示框同步作用。

N.文件列示框(File List Box):

文件列示框可以操控文件的访问。它可与磁盘列示框及目录列示框同步作用。

O.目录列式框(Directory List Box):

目录列示框可以查找或切换有效磁盘机上的目录及路径。它可与磁盘列示框及文件列示框同步作用。

2. 控制图的画法

在表格上建立控制图有二种方法: 第一种是将鼠标器指针在工具框上适当的控制图上敲一下, 鼠标器指针在表格上已经变成了“+”符号, 而且工具框上的该控制图转变成黑色。此时将指针移到表格上适当位置拖曳(Drag,所谓拖曳就是按着左钮不放移动鼠标器)出一个合适大小的控制图。如图 1-15 所示 (以文本框为例)。

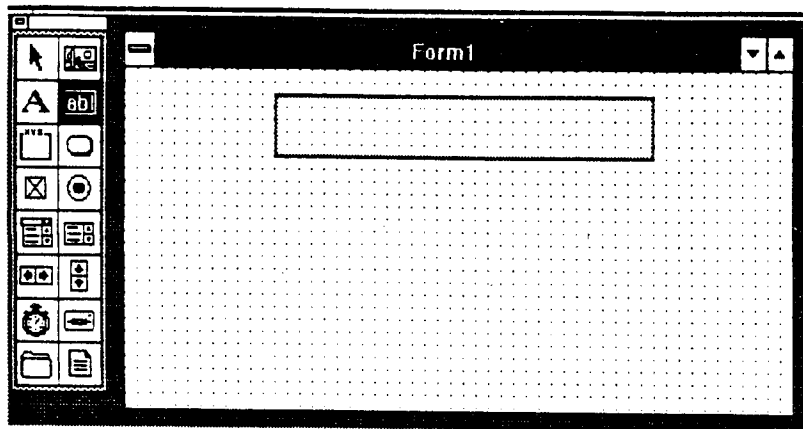


图 1-15 建立文本框控制图

按着放开左钮，就形成了一个适当的文本框控制图，这时在此控制图的四周会附有 8 个小黑色方块，这表示此控制图当前活动控制图(Active Control)，如图 1-16 所示。

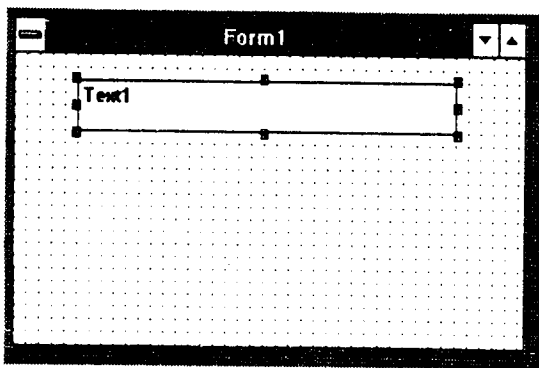


图 1-16 完成的活动控制图

控制图周围有 8 个小方块，除了表示当前它是活动控制图外，还表示此控制图可以改变大小。用鼠标器拖曳四角落的小黑色方块则可以使控制图朝该角落方向变大或缩小，若拖曳上下左右四个小黑色方块则控制图可朝该方向拉长或缩短，如图 1-17 是拖曳上面小黑色方块控制朝上面方向拉长。

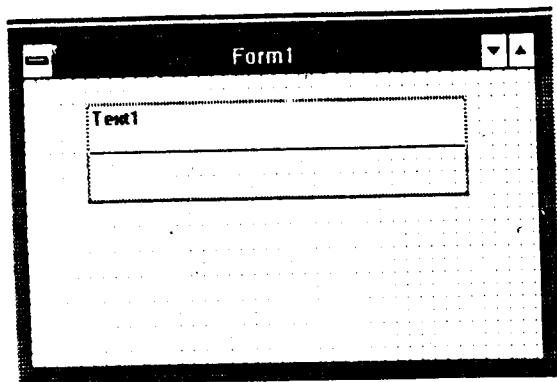


图 1-17 拖曳中（向上）的控制图

完成控制图后，如果想改变控制图的位置，只要将鼠标器指针放入控制图，然后以拖曳方式就可拖着该控制图到处跑了。

第二种建立控制图的方法是在所选定的工具框上适当控制图中连续敲二下，则在表格下中央处就会出现一个控制图了。如图 1-18 所示。

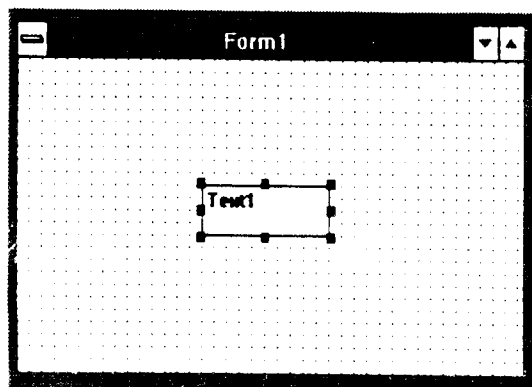


图 1-18 出现正中央的文本框

复制数目较多可以用“拷贝”(copy)进行。首先将要复制的控制图成为活动控制图（周围出现黑色小方块，如果不是活动控制图，只要在该控制图上敲一下即可），接着在菜单条(Menu Bar)编辑(Edit)上的“copy”敲一下，VB 会将此控制图拷贝到 WINDOWS 的 Clipblard 内，然后再在此命令群上的“Pase”上敲一下，画面会现现一对话框，问“ You alerady have acontrol names “Text1”. Do you want to create a control!?”，我们的回

答是“No”，因为我们不是要复制数组控制图（以后讨论）。最后会在左上角出现一个与活动控制图完全一样的控制图了。如图 1-19 所示。

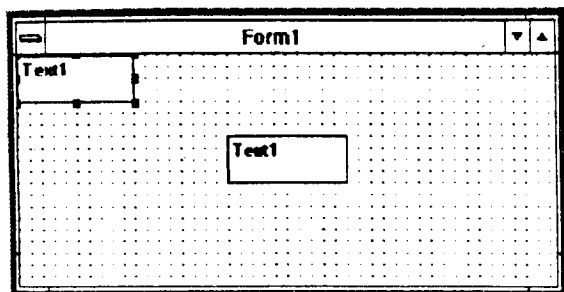


图 1-19 复制控制图

清除控制图是一件很简单的事，只要将该控制图变成活动控制图，然后在键盘上按 DEL 键，该控制图就立该消失。

除了用直接拖曳可以改变控制图或表格的大小外，从属性条上也能记变表格或控制图的大小和位置。敲一下属性表右边的向下箭头一下，属性列表中有 4 种属性是有关表格或控制图的大小及位置：“Width”、“Height”、“Top”和“Left”。

当敲一下该属性名称后中央的设置会显示当前活动对象的属性大小值。例如图 1-20 中当前活动对象是表格，在属性表中找出“Width”属性敲一下，中央设置区处立该显示出“6090”以示该表格宽度。

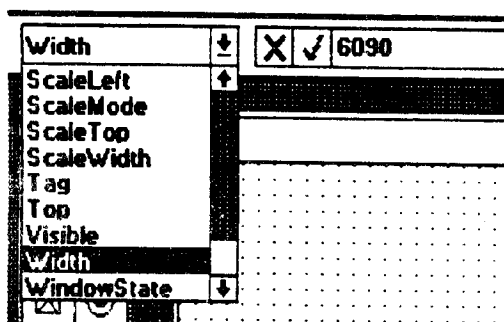


图 1-20 属性列表

如图 1-21 所示，从属性列表中选择“Height”，则中央设置区立该显示出活动对象的高度(500)。同时你可以注意属性条最右边方框上的高度数字也为 500。如果你希望将高度改为 1000，只要键入“1000”，则下高度属性将由 500 变成 1000，换句话说，此活动对象高度会增长。当然在设置键入完数据时，不要忘了在它右边的“v”敲一下以示确认。

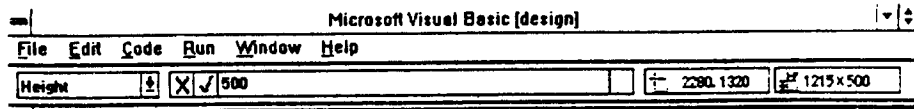


图 1-21 "Height"属性

1-3 对象(Object)

所谓对象就是表格与控制图的统称。

1. 对象属性(Properties)

所谓属性就是定义该对象的特性，如标题(Caption)，控制名称(CtlName)，颜色(Color)，字体大小(FontSize)，窗口状态(Window State)，可见否(Visible)...等。不同的对象 VB 会赋予不同的属性。

定义对象属性的方法有二种：第一种是在设计状态下，用属性条中属性选择直接设置。第二种是用程序码设置。其设置的文法：

格式

对象名称，属性名称 = 新高设置属性值
(Object Name)(Property) (New Setting)

范例

```
Display.Text = "早安"
Display.Text = "Good Morning!"
```

这段程序表示在表格上有一个文本框控制图，它的控制名称(CtlName)为"Display"，我们希望程序执行时将"Good Morning"或"早安"这两字符串赋予 Display 文本框控制图的"Text"属性。

范例

```
Apple.Visible = 0(表示 False, -1 表示 True)
```

这个语句表示在表格上有一个文本框控制名称(CtlName)为"Apple"，我们希望程序执行时，将 0 值赋予 Apple 这控制图的"Visible"属性。当某控制图的 Visible 属性赋予 0 时，程序执行时，它会消失不见。若赋予-1 则会出现，通常缺省值都是-1。

2. 对象事件(Event)

所谓事件(Event)是由 VB 预先设置好能被对象认识的动作，如 Click (敲一下鼠标器)，Dblick (敲二下鼠标器)，Load (装入)，MousMove (移动鼠标器)，GotFocus

(获得焦点), LostFocus (焦点离开) ...等。不同的对象能认识的事件当然不一定相同。当事件由用户触动(如 Click)或由系统引发(如 Load)时, 对象就会对该事件作出反应(Respond)。例如写个程序让用户只要敲一下鼠标器左键(Click)即可将某些信息印在屏幕上。

所谓事件函数(Event Procedure)就是附在一个对象上的程序码。通常一个对象会认识一个以上的事件, 当然它就能使用一个以上的程序码(事件函数)来对用户或系统的触动作出反应。虽然一个对象可以包含许多事件函数, 但是不一定每一个事件函数都都会用上, 这是由设计者视程序复杂的程度而定。事件函数的格式如下:

格式

```
Sub 对象名称_事件名称()  
:  
: (加入所要反应动作的程序码)  
:  
End Sub
```

对象名称指的是该对象控制名称(CtlName), 而事件名称必须是 VB 所定义好给予该对象的所有可能认识的事件。这些“事件”都不必让我们操心, 因为当你建立好一个表格或控制图后 VB 会自动配好什么对象配那些事件, 存放在程序码窗口中供加入程序码。如图 1-22 所示。

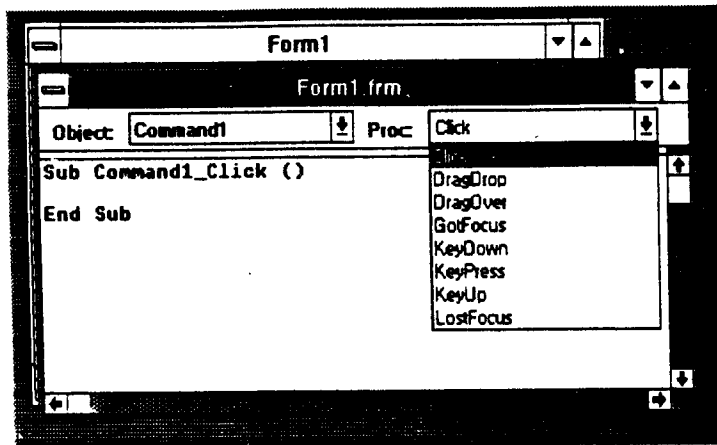


图 1-22 事件函数(1)

图 1-22 上的 Command1 是表格上一个命令按钮控制图的缺省控制图的缺省控制名称 (CtlName)。“Sub Command1_Click()”事件函数是建立好控制的缺省控制后自动缺省的事件函数。如果想改变或增加事件函数，只要在右边列示框上的事件名称上敲一下，此事件就会改变或自动附到“Command1”对象上。

有三个方法可以进入具有事件函数式码窗口，如图 1-22 所示。第一种方法是直接在已建立的控制图上敲二下，第二种利用菜单条“Code”填内的“View Code”命令，第三种方法是选择计划窗口上方右边的“View Code”命令按钮。

3. 对象方法(Method)

VB 有些关键字(Keywords)可以用来显示对象、画图或打印等功能，这些关键字就称为“方法”(Method)，其格式为：

格式

对象名称.方法名称

范例

Picture1.Print“Good Morning”

Print“早安!”

这段程序可以在 Picture1 控制图内印出“Good Morning”，前面没有加对象名称，则在表格上印出“您好!”

范例

Picture.Newpage

这段程序可令 Picture1 控制图移到鼠标器所在的位置。

1-4 实例设计

本节要以一个实例来说明 VB 应用程序的步骤。只要耐心的跟着实例走，而且上机实习，你会慢慢发现 VB 确是“太帅了”。一般应用程序设计的步骤有三：

- (1) 建立用户界面的对象(Object)
- (2) 为这些对象设置属性
- (3) 加程序码到对象的函数中。

本例是在表格上建立一个文本框控制图，一个命令按钮控制图，用户只要用鼠标器在命令按钮上敲一下，则文本框控制图马上会印出“VB 欢迎您!”

1. 控制图的建立

本例共要建立二个控制图：一个文本控制图，另一个命令按钮控制图。首先在工具框上的文本框(Text Box)上连敲二下，在表格 1(Form1)的中央立即出现一个“Text1”文本控制图。使用拖曳(Drag)方法将此文本控制图移到上方，如图 1-22 所示。

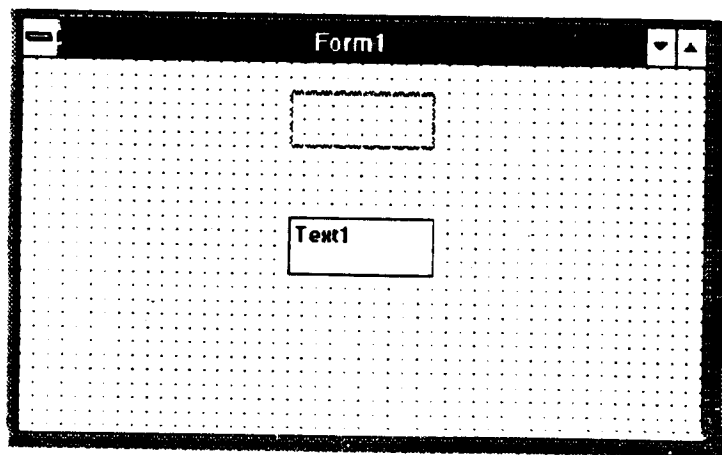


图 1-22 文本控制图拖曳到上方(2)

利用控制图周围的八个黑色小方块拖曳成适当大小的控制图，如图 1-23 所示。

