

第一章 快速浏览

1.1 绪论

Microsoft Visual Basic for Windows 不同于一般传统的程序语言。当你知道如何使用它所提供的工具时,你所设计出来的程序会有 Windows 风格及专业水准,如下拉式菜单、对话框、图标、工具箱和鼠标控制等。这些在时间和精力方面都不是在 DOS 操作系统下所能相比的,这就是为什么许多程序语言开发厂商都纷纷将 DOS 版本改成 Windows 版本。如果你是经验丰富的 DOS 程序员,则刚刚开始使用 Windows 操作系统写程序时,也许会觉得有些繁琐,而且觉得需要学习的地方很多,这就是你必须付出代价的地方。

Visual Basic 是承袭 Microsoft QuickBasic 的编译程序,因此当你在设计程序时会觉得比其他程序语言来得简单方便、快速。但是电脑硬件和软件是两个互相促进的工业,有好的硬件而没有软件来操作是不行的,同样,有好的软件但没有可相容的硬件则也是废物一件,两者相互依靠但同时也互相竞争。如 1986 年时 AI(人工智能)理论真是轰动,可是不到几年就不见其形影,其原因就是它提出理论太多而真正所能提出的东西又太少。而在近几年,软件界又提出面向对象程序设计(OOP)的概念,但这一次不像 AI 一样,它并没有提出太多理论,取而代之的是给最终用户提供能够直接触觉到的、简易且完善的开发环境,并提供许多辅助性的程序设计工具以提高用户的效率。如果使用不属于 OOP 的程序语言,如 C,则在设计 Windows 风格及专业式的程序时会变得非常困难,因此才有 C++ 的推出。于 1989 年 Pascal 也推出了它的 OOP Module-2 编译器,可见 OOP 不但是一种潮流,更是时势所趋。因此 Microsoft Visual Basic 也融入这种风格,但什么是 OOP 呢?我们可以举一个简单的例子来说明。例如阁下所带的电子表,就是 OOP 中的对象(Object),而此表是由 LCD 显示器、小型键盘、控制钮、表壳以及里面的电子计算机、备忘录和闹钟等元件组成,以上的概念则可引用到 Windows 应用程序上(当然包括 Visual Basic),如每一个对话框中的按钮就是一个对象(Object)。而它显示的特性及功能就如同电子表一样,它主要的功能是显示时间、日期,但如果我们按表下边的按钮则可显示电话号码。当你按下此按钮时,电子信号会传到内部的逻辑电路,而得到主菜单(Main Menu),让你选择要看的电话号码、备忘录或行程表。这种事件驱动程序设计的概念完全不同于传统的一行一行程序设计的方法。

我们再选一个例子来说明,如公司和公司的职员。公司是一个对象(Object),而每一个职员有其头衔及负责的工作项目。公司每天通过外面的事件需要职员处理:电话、信件、开会等,而每一个发生事件都有专人负责。例如每人的信件是由公司的收发处人员来分送,而不是由会计或接待人员来处理。就如同公司一样,Visual Basic 通过鼠标或由键盘所得到的信息而去执行适当的过程(Procedure),如果找不到的话则放弃(Abandoned)。以下是一个典型的伪说明事件驱动程序。

```
Initialize—application
Exit_flag = false
Do
    GetNextEvent(NewEvent)
    Select Case NewEvent
        case Event 1
            handle_event 1
        case Event 2
            handle_event 2
        case Event 3
            handle_event 3
        case end_application
            handle_exit
            Exit_flag = true
        case else
            ignore_event
    End select
Loop until Exit_flag = true
```

当设计 Visual Basic 程序时,必须为每一个对象(Object)定义一些事件(Events),而每一个对象(Object)都通过你所写的过程(Procedure)来完成。就如同以上的例子一样,其基本语法如下:

```
Sub Objectname_eventname()
```

以下将开始对 Visual Basic 中可用的工具及资源(Resource)从浅至深地介绍给读者,分成三部分来说明:

1. 简介及如何使用 Visual Basic。
2. 介绍一个 Visual Basic 项目,也就是数据库程序,可让读者了解使用 Visual Basic 的特性。
3. 详细介绍。

至此,假设读者的电脑中已经有 Visual Basic 软件了。在 Windows 中用鼠标双击 Microsoft Visual Basic 3.0 图标,几秒之后您的屏幕上出现了 Visual Basic 开发环境,如图 1-0 所示。

我想当第一次看到此屏幕时,也许你会目不暇接的感觉,而且不知如何开始。但当你将本章读完后,定会有一个详细的概念,而且更加喜爱用 Visual Basic 去设计你的程序。

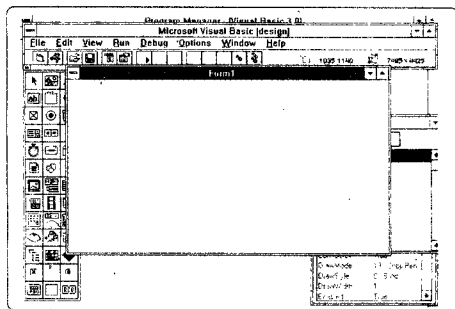


图 1-0

1.2 简介及如何使用 Visual Basic

我们现在开始设计一个 Visual Basic 程序。如图 1-1 所示,此窗口的程序中有三个命令按钮:Display Hello, Clear 及 Exit 和一个空的正文框。

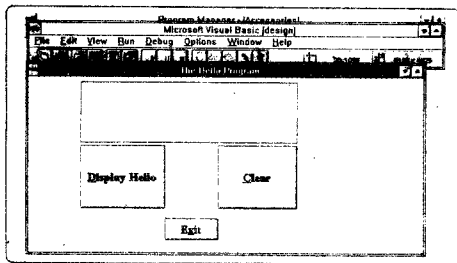


图 1-1

当鼠标移到 Display Hello 上时按一下,则图 1-2 的正文框中会有 Hello World 出现。

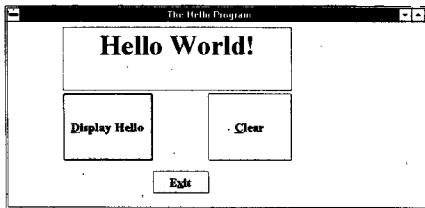


图 1-2

当你将鼠标移到 Clear 时按一下,则 Hello World 会消失于正文框中。如要结束程序,则将鼠标移到 Exit,按一下,则程序结束。

我们了解程序功能之后,必须着手去完成这个程序。首先,我们必须做的是打开一个新的项目。

1.2.1 打开新项目

你可以从图 1-0 的 File 菜单中选 Select New 项目。下列图形会在你的屏幕出现,如图 1-3 所示。

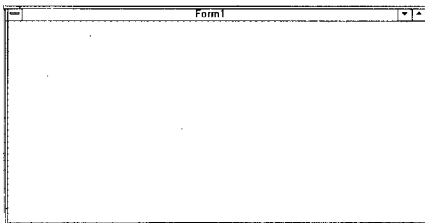


图 1-3

这是一个空白的,其顶端上有 Form1(这是 Visual Basic 所设的名称,我们可以改成有意义、好记的名字)。这有点像图 1-1,而里面都是空白的。

1.2.2 存储新项目

我们假设在这空白的窗体中不做任何修改,于是必须把它存储起来。当我们存储这一个项目之后,有二个文件(File)会被存储。

1. Make 文件:这一个文件的扩展文件名是 .MAK,它用来存储 Visual Basic 建立项目时的一些有用的数据。

2. Form 文件:这一个文件的扩展文件名是 .FRM,它用来存储窗体的数据。

我们又回到 File 菜单中选 Select Save As 项目,于是 Visual Basic 会问你是否要存储起来,如图 1-4 所示。

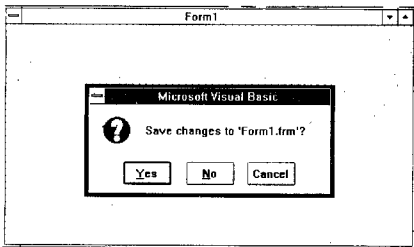


图 1-4

当你选择 Yes, Visual Basic 会通过对话框的方式将你要存储的数据再次显示一遍。

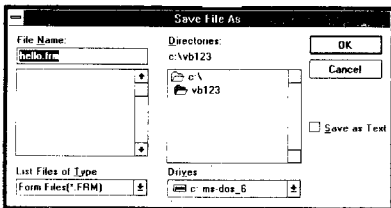


图 1-5

这时你就可将文件名称改成一个好记、有意义的名字。如图 1-5 中改成 Hello. frm, 同时存储至 C:\VB123 目录下(VB123 目录可在 DOS 下建立)。

同样道理,你也必须存储你的项目,Visual Basic 设置的文件名(File name)是 Project1. mak, 但如同存储窗体一样,你可以定义一个容易记的名字。现在我们定义为 Hello. mak, 并存储在 C:\VB123 目录下,如图 1-6 所示。

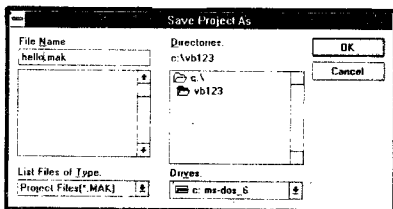


图 1-6

现在我们已经存储了两个文件:Hello. mak 和 Hello. frm。

1.2.3 检查项目窗口

现在我们的项目叫 Hello. mak。它只有一个窗体文件,叫做 Hello. frm。事实上,当你在设计程序时,可能会产生一个以上的窗体文件。如你想要看有哪些窗体文件在此项目中,则可到项目窗口中去看,其步骤如下。

1. 至窗口菜单中选项目,则项目窗口会出现如图 1-7 所示的画面。



图 1-7

2. 如果你是用专业版的话,则项目窗口会如图 1-8 所示的画面。

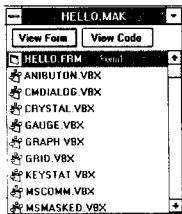


图 1-8

由以上两个图,可以看出项目文件名 Hello.mak 以及窗体名称 Hello.frm 都是由上节中我们存储时给的名称。同时项目窗口也出现一些有 .VBX 扩展文件名的文件,我们称之为定制控件,而其所属的 .VBX 文件是用于定义(Define)一个工具框控件面不是属于 Visual Basic 环境之中。在 Visual Basic 3.0 标准版中有网格控制件和对象链接与嵌入控件,其特性定义在 Grid.vbx 和 OLECLLine.vbx 中。在安装(Set up)Visual Basic 时,这些文件会被复制到 \windows\system 之下。如果你要用这些定制控件的 .VBX 文件,则必须包括于你的项目窗口中,否则你可以把它们删除掉。

1.2.4 窗体的属性(Property)与标题

当初我们由 Visual Basic 所显示出来的一个空白窗体(图 1-3),其标题定义为 Form1。当然我们要将它改成一个好记的名称,如叫 The Hello Program 也可。以下步骤就是教读者如何去修改。

我们仍然从 Window 菜单中先选 Project,则图 1-7 出现的高亮度显示就是你要的窗口,如例题为 Hello.frm,然后用鼠标选 ViewForm,此时在窗口中移动鼠标选 Properties,则 Visual Basic 会显示图 1-9。

到 Properties 窗口中高亮度显示 Caption 这一项,然后在设置框(Setting Box)中按一下,则可以修改标题属性(Caption Property)了。只要将窗体 1 改成 The Hello program,则窗体的标题将变成 The Hello Program,如图 1-10 所示:

在继续讨论之前,首先要了解什么是属性(Property)。以上我们所修改的标题(Caption),只是窗体中的一项属性(Property)。也就是说窗体有许多的属性(Property),而在先前也提到过 Visual Basic 程序每一个物体都叫对象(Object)。例如对象(Object)有窗体命令按钮、滚动条、图片和其他。

而对象(Object)的属性(Property)则是定义对象(Object)的外观和功能,例如窗体是一个对象(Object),而刚才所修改的标题属性就是定义窗体显示时的标题(The Hello

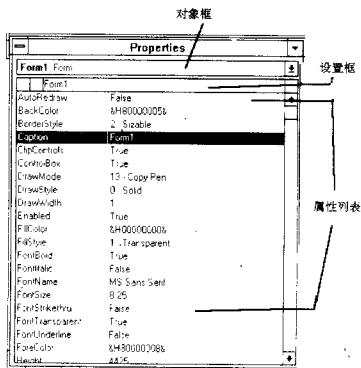


图 1-9

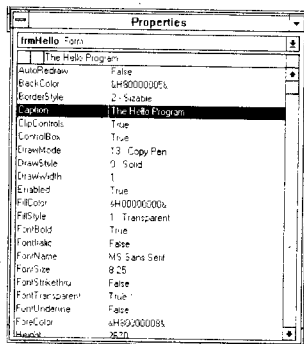


图 1-10

Program)。如果你再看图 1 11 的话,像 Backcolor 就如同字面意义一样,它是让你修改窗体

的背景颜色。

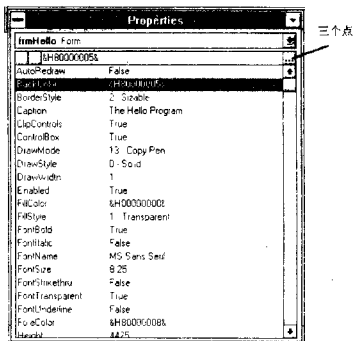


图 1-11

同样的，将鼠标移到 Backcolor，然后按一下，如果在设置框(Setting Box)有三个点，在那里用鼠标按一下，则图 1-12 的调色板会出现。

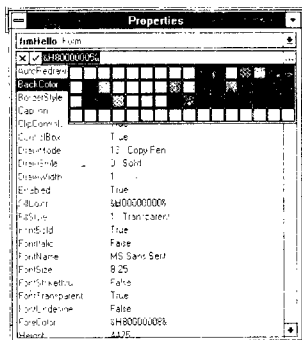


图 1-12

此时,你可象艺术家一样自己选颜色,直到满意为止。

1.2.5 修改窗体的属性名称

在 Visual Basic 中每一个对象(Object)都有名字,而它的名字是在名称属性中定义的。在 Hello Program 中产生一个新的对象(Object)时,Visual Basic 会自动在名称属性的窗体中赋予名称 Form1,而我们必须将它改成我们所要的名字。同样的,我们到 Window 菜单中选 Properties Form,然后用鼠标选名称属性(Property),然后再移动鼠标到设置框(Setting Box),此时 Visual Basic 就可让你去修改名字,例如可以改为 FrmHello,所以现在 FrmHello 就是 Hello Program 的正式名称了。

至此我们还没有完成整个工作,我们的窗体必须像图 1-1 一样才行。但我们必须将已完成的工作存储起来,所以必须回到 File 菜单中选 Save Protect Form,存储起来。

1.2.6 在 frmHello 窗体中增加 Exit 按钮

在图 1-1 中我们三个控制按钮,分别为:Display Hello, Clear 和 Exit,要将此控制按钮放在产生的窗体中(frmHello),必须用工具箱窗口(图 1-0 左边的部分),这是包含许多控制对象的工具箱,而你的责任就是挑选要用的工具(Tool)放入窗体中。在图 1-13 中是此工具箱窗口。

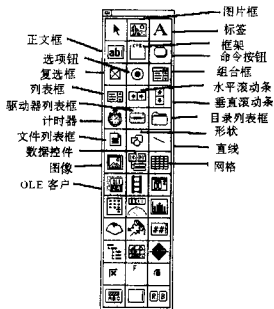


图 1-13

因版本的不同,所以位置或许有些不同,但图形仍是一样的。

指针 : 有一个箭头图形。

图片框 : 在箭头的右侧从位图,图标或一些特殊文件中来显示影像图形。

- 标签框 : (有一个 A 图形记号) 用于识别对话框(Dialog Box)中的元素(Element)。
- 正文框 : (有 ab 图形记号) 是一个有输入功能的框, 一般是由键盘敲入数据, 通过此框而输入电脑。
- 框架 : (有 xy 图形记号) 将一个窗体中所有的功能集合到一起。
- 命令按钮 : (小的长方形而有侧影的记号), 控制当用鼠标按下时而去执行你要的功能。
- 复选框 : (有一个 x 记号) 是属于一个打开/关闭的按钮。
- 选项框 : (圆形而中心有一圆点) 又叫收音机按钮, 其功能和复选框一样, 控制是打开/关闭, 但它是一个单一动作而不能一次将一群对象打开/关闭, 其所以叫收音机按钮, 是如同收音机一样一次只能选一台来听。
- 组合框 : 它有正文框和列表框的特性, 即可从键盘上输入数据, 或从列表中选数据。
- 列表框 : 是控制从列表中去选所要的项目。
- 水平滚动条 : (有二个水平箭头图形) 是在设计滚动条程序时用的, 而其出现图的形状是水平面。
- 垂直滚动条 : (有二个垂直箭头图形) 其功能和水平滚动条相同, 其图形是垂直的。
- 计时器 : (有一个表图形) 用来显示时间。
- 驱动器列表框 : (有硬盘图形) 让用户能看到系统中有效的硬盘驱动器。
- 目录列表框 : (有一个卷宗的图形) 显示硬盘中的目录。
- 文件 : (有一记事纸图形) 在这里可以从目录中选择需要的文件。
- 形状 : (三种不同图形重叠在一起) 在此可以选你要的形状如圆形、正方形、长方形、椭圆形等。
- 直线 : (有一横线的图形) 此功能可以将一个窗体分成不同的区域。
- 图像 : (有一个框的图形) 如图形控件, 但能提供较佳功能及内存。
- 数据控件 : 让使用 Visual Basic 的用户能够使用已有的数据库。
- 通用对话 : 操作一些基本的功能如打开、存储、打印文件等。
- 网格 : 有二维矩阵功能, 很像 Microsoft Excel。
- OLE : 可让你使用对象链接和嵌入的功能, 当使用不同的应用程序都能分享数据。

以下步骤就是如何将 Exit 按钮放在窗体里。

我们在窗口中选工具箱(Tool Box), 则图 1-14 会出现, 然后按两次命令按钮(Command Button), 则 Visual Basic 会将命令按钮(Command Button)放在窗体的中央, 如图 1-14 所示。而同时 Visual Basic 会设置属性(Property)的值, 如它设主属性(Capital Property)为 Command1, 名称属性(Name Property)为 Command1, 因此下而我们必须将它改成我们要的名字。

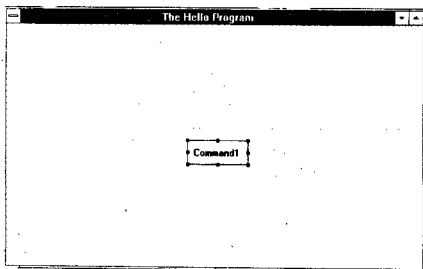
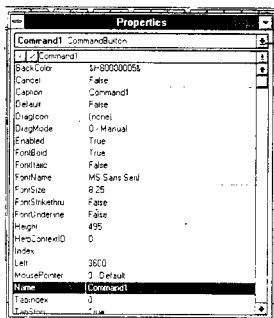


图 1-14

1.2.7 修改 Exit 的属性

要将名称属性(Name Property)改成 cmdExit,我们需在 Window 菜单中选 Properties,但必须注意在 Properties 的对象箱(Object Box)中,是 Command1 的命令按钮(Command Button)。如果不是的话,则必须将鼠标移到 ↓ 处去找所要的对象(Object),如图 1-15 所示,则我们即可将名称属性(Name Property)中的 Command1 改成 cmdExit。



单击此处,切换
至另一对象

图 1-15

要将主属性(Capital Property)的 cmdExit 从 Command1 改成 E&x.it。为什么 &. 号放至 x 字母之前呢?如你仔细注意一下图 1-1 时会发现 X 字母之下的底线,其意思是在电脑键盘上按[Alt]+[x]和按 Exit 按钮是一样效果,因此才有这个字母 &. 出现。

1.2.8 改变 Exit 按钮的位置

在图 1-1 中,Exit 按钮的正确位置应该接近下半部中央的部分,因此我们下一步骤就是改变它的位置,而其方法和使用普通窗口相同,即将鼠标移到 Exit 按钮上,手按着鼠标左边按钮不要放,一直到将 Exit Button 移至下半部中央部分才放开鼠标左边的按钮,此时就是我们所要的位置了。当然面积大小也可用普通窗口的做法将它放大。

1.2.9 修改 Exit 的字体属性

我们知道由 Visual Basic 产生的字体如图 1-14 所示,和我们所要的不一样(如图 1-1 所示),因此我们也必须将它改成我们所需要的样子。在 cmdExit 按钮的 Font Name Property 中改成 Time New Roman,并在 Font Size Property 中改成 13.5,才会使大小相等。而 Visual Basic 其中一个最大的好处就是当你一改变其设置值时,你可以马上看到改正以后的样子。因此你可以改到满意为止。

1.2.10 增加其他按钮

到目前还剩下两个命令按钮需要加入窗体中:Display Hello 按钮和 Clear 按钮。

所以我们再次光临正文框窗口中的命令按钮,将 Display Hello 按钮放在左边,Clear 按钮放在右边,接下来修改 Caption 属性,将左边的改成 &.Display Hello 而右边的改成 &.Clear。最后修改 Font 属性,将 cmdHello 按钮中的 Font Name 改成 Times New Roman,Font Size 属性改成 13.5,同样方法修改 cmdClear。

1.2.11 增加正文框

最后我们要加入 frmHello,即图 1-2 有 Hello World! 长方形的长方形字幕,此时我们仍必须借用正文框窗口中的辅助工具,将鼠标指针移至 Text Box 的图标中,然后按两次,同时移动方块和调整大小后移至如图 1-2 的位置。

1.2.12 修改正文框的属性

首先我们将 Visual Basic 所产生的 Text1 的 Name 属性改成 Text display,然后改其字体名称属性(Font Name Property)。这可把正文框中的 MS Sans Serif 改成你要的字型如 Dom Casual 等,第三,改其字体大小(Font Size),由 825 改成 30,第四,改其间距(Alignment Property),由 0-left Justify 改为 2-center,如此其字体才会显示在长方形中央,但有一点要注意,Visual Basic 不会将字体放至长方形中央,除非 MultiLine 属性中的正文框的值改为 True,因此 MultiLine 属性不要忘了改。

到此,Visual Basic 程序的部分终于完成。将刚才所修改的存储起来,以下将我们所做的

归纳起来,如表 1-1 所示。

表 1-1 Hello 程序属性表

对象	属性	设置值
窗体	Name	frmHello
	Caption	The Hello Program
命令按钮	Name	cmdExit
	Caption	E&xit
	FontName	Times New Roman
	FontSize	13.5
命令按钮	Name	cmdHello
	Caption	&Ddisplay Hello
	FontName	Times New Roman
	FontSize	13.5
命令按钮	Name	cmdClear
	Caption	&Cclear
	FontName	Times New Roman
	FontSize	13.5
正文框	Name	txtDisplay
	Text	(空白)
	FontName	Dom Casual
	Font Size	30
	Alignment	2—Center
	MultiLine	True

1.2.13 把程序放在对象中

因为我们已将对象(Object)放入窗体且设置好了它的属性,所以 Visual Basic 程序这一部分已完成。现在,要做的就是将代码(Code)放入对象中。Visual Basic 是属于事件驱动的程序语言,也就是说代码(Code)接到事情的反应时才执行。例如你按下 Exit 按钮,则代码(Code)因得到退出的反应,而执行这一个功能。如按 Display Hello 按钮的话则执行 Display Hello 的功能,而我们的工作就是写一些合适的代码(Code)放至适当的项目和事件里以使整个程序完成。

1.2.14 把程序放在 Exit 按钮中

以下是说明如何将程序放入 Exit 按钮中。首先,用鼠标在 Exit 按钮中按两下(Double-Click),于是 Visual Basic 会有代码(Code)窗出现,如图 1-16 所示。

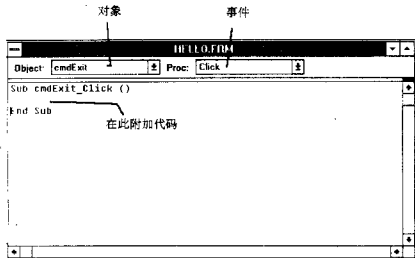


图 1-16

由图 1-16 中可看出 Visual Basic 使你很容易识别目前是什么代码在代码窗中,左上方的组合框和右上方的组合框分别显示对象和事件的名字,而且 Visual Basic 已经有:

```
Sub cmdExit_Click()
```

```
End Sub
```

这二行于代码窗中,而你要写的代码就放在此二行之中,cmdExit_click()就称为过程。但为什么 Visual Basic 会指定这一个名称呢?因 Visual Basic 知道你是要放代码于 cmdExit_中,而你用鼠标按下去调用此事件,而要写的程序是属于 Click 事件,故用 Click(),所以整个过程就是 cmdExit_Click()。而什么样的程序你必须写过程入 CmdExit_click()中呢?因主要功能是要结束整个程序,因此程序是 End,所以整个过程就变为:

```
Sub cmdExit_click()
```

```
End
```

```
End Sub
```

我们先将此过程存储起来,到 File 菜单中选 Save 项目即可。如果你要试试去执行程序,则可从 Run 菜单选 Start 即可执行。而 Visual Basic 则将图 1-1 的窗口出现在你的电脑屏幕中,你可以用鼠标按 Display Hello 按钮,但没有动作发生,因为我们还没有放任何程序至此按钮里。同样的 Clear 按钮,也是一样,但如按 Exit 按钮的话,则在过程 cmdExit_click()中的程序会执行,因是放 End 命令,则程序结束了整个 Hello 程序。

1.2.15 把程序放在 Display Hello 按钮中

首先我们选 frmHello 窗体,然后按两次鼠标(Double click),Visual Basic 则出现 cmdHello_click()的过程,而有

```
sub cmdHello_click()
```

```
End Sub
```

两行出现,而你必须在此两行中加入 `txtDisplay.Text="Hello World!"`,所以它基本的命令为:

```
ObjectName.property=The new value of property
```

而整个过程就变成

```
Sub cmdhello_click()
```

```
txtDisplay.Text="Hello World!"
```

```
End Sub
```

最后将它存储起来。

1.2.16 把程序放在 Clear 按钮中

同样按两次 Clear 按钮,Visual Basic 则产生代码窗,其中有 `cmdClear_click()` 的过程。因过程的功能是将屏幕清掉,所以我们必须加入

```
txtDisplay.Text=""
```

而整个过程变成 `Sub cmdClear_click()`

```
Sub cmdClear_click()
```

```
txtDisplay.Text=""
```

```
End Sub
```

同样将它存储起来。

1.2.17 执行 Hello 程序

现在 Hello 程序终于有完整的 Visual Basic 程序和代码两部分。如我们要执行则在 Run 菜单中选 Start,用鼠标按 Display Hello 和 Clear 按钮,则会显示和清除 Hello World! 这一串文字,当然你也可以用 `[Alt]+[D]` 和 `[Alt]+[C]` 来显示和清除 Hello World!,将会得到同样效果,如要结束程序则按 Exit 按钮。

1.2.18 产生可执行文件

我们执行 Hello 程序都是从 Run 菜单中选 Start,但这不是一个简单方便的执行方法,而且如果用户没有 Visual Basic 软件的话,就根本没办法去执行程序,因此我们必须产生一个 .EXE 文件,也就是 Hello.exe。

首先在 File 菜单中选 Make Exec,因此就会有图 1-17 中 Make Exec File 的对话框出现。

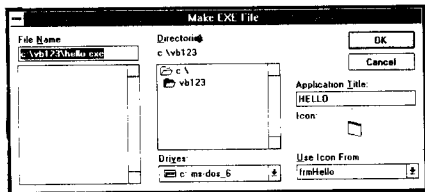


图 1-17

此时我们从中间的框中去选 EXE 文件所存在的目录(Directory), 选定(HightLight)后, 则左边的方框会出现, 此子目录中所定义的文件名(Filename)选定(HightLight)后, 则在上边的 File Name 显示你所选的一切, 然后按 OK, 则 Visual Basic 立刻转换成 EXE 文件。此例题变成 Hello. EXE 后, 我们就可以用 Windows 的功能去执行了。