

特效视窗——

Visual Basic

开发高级界面实例

刘建东 等编著

A
人民邮电出版社
www.pptph.com.cn



特效视窗

——Visual Basic 开发高级界面实例

刘建东 等编著



人民邮电出版社

图书在版编目(CIP)数据

特效视窗: Visual Basic 开发高级界面实例/刘建东等编著.—北京:人民邮电出版社,2000.11
ISBN 7-115-08765-2

I.特… II.刘… III.BASIC 语言-程序设计 IV.TP312

中国版本图书馆 CIP 数据核字(2000)第 46331 号

特效视窗——Visual Basic 开发高级界面实例

- ◆ 编 著 刘建东 等
责任编辑 王晓明
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@pptph.com.cn
网址 <http://www.pptph.com.cn>
北京汉魂图文设计有限公司制作
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本:787×1092 1/16
印张:31.75
字数:803千字 2000年11月第1版
印数:1-4 000册 2000年11月北京第1次印刷

ISBN 7-115-08765-2/TP·1811

定价:46.00元

前 言

经过几年的发展，Microsoft 的 Visual Basic 已经成为一种真正专一化的开发语言和编程环境。全球有 100 多万 Visual Basic 开发人员。在开发 Windows 前端界面和客户/服务器应用程序时，微软的 Visual Basic 已成为许多程序开发人员的首选工具。

最新版本的 Visual Basic 6.0 不但继承了 Visual Basic 早期版本的所有优点，如图形界面、面向对象的程序开发、高性能的本地代码编译、灵活方便的可视化数据开发工具以及企业水平的客户/服务器，而且还增添了许多新功能。因此，从事开发应用程序工作的很多程序开发人员都非常愿意使用 Visual Basic 6.0 这一优秀的编程工具。毫无疑问，随着计算机应用水平的不断提高，使用 Visual Basic 6.0 从事开发工作的人还会继续增加。

本书从一个程序开发人员的角度，通过一些实用的例子，对 Visual Basic 6.0 的开发界面进行了详细的介绍，并专门编写了相应的实例。通过实例，读者将看到，用 Visual Basic 6.0 可以编写出许多高质量的 Windows 界面。作者编写本书的目的是为了使读者能够通过详实的内容和具体的实例，较快地掌握使用 Visual Basic 6.0 的编程和开发技巧，学会开发各种 Windows 高级界面。

本书主要由刘建东编写。另外参加编写和提供帮助的人员还有：刘建辉、姚东、陈红、张京平、冯峰、程卫国、王建忠、官笑萍、王野秋、王晓军、王瑞清、刘东、雯辉、张新、王进军、李瑞峰、郭卫军、曹迎芳、梁辉、刘春挥、韩志宏、秦卫忠、柴爱军、郑晓军、穆临平、张平则、李福根、郭红宇、胡燕军、王昌富、白贵林、李丽洪等。

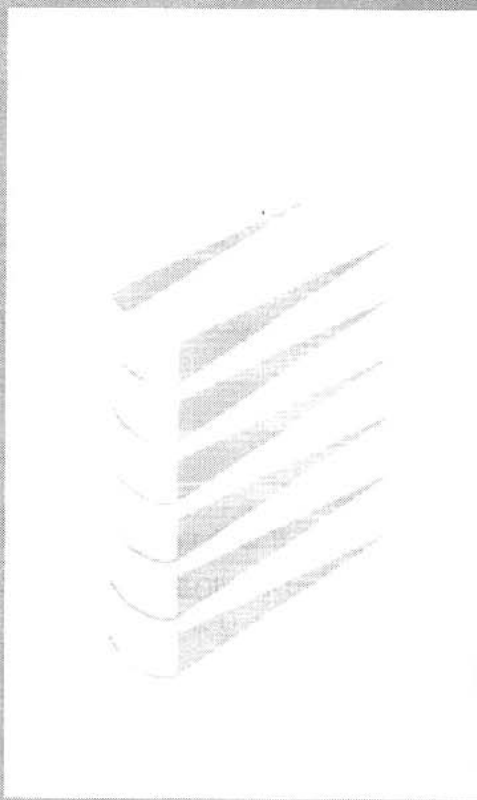
最后需要说明的是，由于计算机应用软件技术发展太快，加上作者的水平有限，书中的缺点与错误在所难免，我们诚恳地希望读者批评指正。

作者

2000 年 7 月

实例 1

制作封面



实例简介

本例详细介绍了制作封面的两种方法：多窗体制作和隐藏控件制作。在程序运行时首先弹出封面，用于装饰程序本身，或简单地演示一下程序的优越性。这样的封面在很多专业的 Windows 程序中均存在，在 Visual Basic 中可有两种简单的方法实现这一功能。

实现过程

多窗体编程方法

多窗体编程方法是通常采用的方法，即在程序中创建两个或两个以上的 Form 窗体，把其中的一个作为专门的程序封面。具体方法如下：

(1) 创建新项目 Form1，并在此窗体内安放 Command1 命令按钮，在 Command1_Click() 事件中输入结束程序代码。

```
Private Sub Command1_Click()  
    End  
End sub
```

单击菜单的“Insert”项目，加入一个新窗体 Form1。

(2) 双击 Form1 窗体空白处，在 Form1_Load() 事件中输入代码，用于隐藏主窗体 Form1，置第二窗体为最大模式，并在运行程序时首先显示第二窗体。

(3) 隐藏主窗口。

```
Private Sub Form1_Load()  
    '窗口 2 最大模式  
    Form1.Visible = 0  
    '显示窗口 2  
    Form2.WindowState = 2  
    Form2.Show  
End Sub
```

(4) 在窗体 Form2 中定义全程变量：

```
Dim str1 As String  
Dim i As Integer
```

(5) 在窗体 Form2 中放入一个时间控件 Timer1。

(6) 在窗体 Form2 中的 Form_Load() 事件中输入以下代码，用于在窗体 2 中显示图像，并设置时间控件频率为 0：

```
Private Sub Form1_Load()
```

```

Form2.Picture = LoadPicture("c:\windows\setup.bmp")
Timer1.Interval = 10
i = 0

```

```
End Sub
```

(7) 在时间控件 Timer1- Timer1 () 事件中进行程序封面的制作工作，演示简单的动画，并控制此动画运行一段时间后自动停止，关闭程序封面，返回程序主窗口。下面的代码实现在一个图片上随机显示字符串，显示 30 次字符串后自动退出：

```

Private Sub Timer1-Timer ()
    Strr1 = "程序封面演示"
    With Form2
        '透明显示字符
        .Font.Transparent = True
        '随机变化字号
        .Font.Size = Rnd * 30 + 10
        '斜体有效
        .Font.Italic = True
        '随机变化前景
        .ForeColor = RGB(Rnd * 256, Rnd * 256, Rnd * 256)
        '随机变化字体
        .FontName = Screen.Fonts (Rnd * 7 + 14)
    End With
    '取字符串高度
    hh = Form2.TextHeight (strr1)
    '取字符串宽度
    ww = Form2.TextWidth (strr1)
    '随机改变坐标
    x = Rnd * (Form2.ScaleWidth - ww)
    y = Rnd * (Form2.ScaleHeight - hh)
    Form2.Current X = x
    Form2.Current Y = y
    '显示字符
    Form2.Print strr1
    '计数
    i = i + 1
    If i > 30 Then
        '关闭封面
        Unload Me
        '调入主窗口
        Form1.Show
        Timer1.Enabled = False '关闭时间控件
    End If
End Sub

```



```
End If
End Sub
```

(8) 为了能够人为地控制程序封面的停留时间，可在窗体 Form2 鼠标单击事件中安放退出封面代码：

```
Private Sub Form-Click ()
Unload Me
Form1.Show
'关闭时间控件
Timer1.Enabled =False
End If
End Sub
```

至此工作全部完成。当运行程序时，首先弹出程序封面，封面上显示一幅 BMP 图像。然后在此图像上进行随机字符串的显示，字符串的位置、字体、字号、前景色等值都是随机变化的，显示 30 次后自动关闭封面，返回主窗口。演示过程中用鼠标单击窗口也会返回主窗口，在主窗口中按下结束按钮退出程序。具体操作时还有很多其它技巧，比如可以置第二个窗体为主窗体，这样运行时可自动先弹出程序封面，可省略隐藏窗体和调入窗体的过程。另外，第二窗体也可以在程序运行时由代码动态生成，这样编程时复杂一些，但有利于程序代码的管理。程序封面的具体制作过程，有许多动态图像控制方法可以使用，只要灵活运用 Visual Basic 的作图方法，再加上一些必要的 API 函数的配合，会制作出理想的程序封面。

☛ 用隐藏控件方法隐藏窗体

实际上通过一个窗体也可以编制出类似的封面效果。基本原理就是在程序运行时首先通过 Visible 属性隐藏全部控件，然后在主窗体内进行程序封面设计和演示，延迟一定时间再恢复所有控件的显示，即达到了独立程序封面的制作过程。由于窗体内的控件很多，逐一隐藏或显示非常繁琐，因此可采用容器控件简化操作过程，即把控件全部放到 Frame1 等控件上，这样只要执行 Frame1.Visible=0，即可使其上面的所有控件隐藏起来。

由于程序只涉及一个窗体，程序设计的大部分工作只是编码过程，因此不再分步介绍。在窗体 Form1 中安放一个时间控件 Timer1 和一个容器控件 Frame1，再在 Frame1 上安放一个命令按钮 Command1，具体程序设计时可把所有主程序工作控件放在 Frame1 上；然后把下面的代码填入相应的事件之中，即可完成全部程序设计工作。运行此程序后首先在全屏幕窗体内显示图像，并从左上角开始逐渐放大字符串“程序封面演示”，放大过程中颜色发生平滑变化，形成美丽的拖尾效果，连续显示 5 次后自动关闭封面，返回主程序窗口，同样，在显示过程中用鼠标单击窗口也可立即关闭程序封面，返回主程序窗口，单击主窗体中的命令按钮退出程序。

定义全程变量：

```
Dim i, j As Integer
'退出程序按钮
Private Sub Command1-Click ()
End
End Sub
```


'单击窗体关闭封面

Private Sub Form-Click ()

'恢复容器控件

Frame1.Visible =1

'关闭时间控件

timer1.Enabled =False

'恢复缺省窗口模式

Form1.WindowState =0

End Sub

'准备工作

Private Sub Form -Load ()

'窗口最大化

Form1.WindowState =2

'调整容器控件尺寸

Frame1.Width =Form1.Width

Frame1.Height =Form1.Height

'调整容器控件位置

Frame1.Top =0

Frame1.Left =0

'隐藏容器控件

Frame1.Visible =0

'置时间频率

timer1.Interval =10

'装入图像

Form1.Picture =LoadPicture ("c:\windows\setup.bmp")

'置坐标刻度

Form1.ScaleMode=3

'"黑体" '置字体名称

Form1.FontName =

End Sub

'封面制作 (动画演示)

Private Sub timer1-Timer ()

Strr ="程序封面演示"

'透明显示

With Form1

.Font.Transparent =True

.Font.Size = i + 1 '放大字号

```

.Font.Italic =True '斜体有效
.ForeColor=RGB(i*3,56,256-i*3) '平滑变化颜色
End With
string =str
Form1.Current X=i '连续改变坐标
Form1.Current Y=i
Form1.Print stringg '显示字符串
i = i +1
'放大到 72 号字
If i >72 Then
    '循环计数
    j =j +1
    i =0
    Form1.Current X=72
    Form1.Current Y=72
    '重新显示一次字符
    Form1.ForeColor =RGB(255,56,0)
    Form1.Print stringg
End If
'动画计数
If j =5 Then
    '显示容器控件
    Frame1.Visible =1
    '关闭时间控件
    timer1.Enabled =False
    '恢复窗口模式
    Form1.WindowState=0
End If
End Sub

```

实例注释和详解

多窗口编程存在着一定的复杂性，需要对多个窗体的 Frm 文件进行管理，编码时还需要频繁地在多个窗口间进行转换。隐藏控件的方法比较简单，而且将控件全部放到 Frame1 等控件上，来一起隐藏窗体上的控件；在程序运行时首先通过 Visible 属性隐藏全部控件，然后在主窗体内进行程序封面设计和演示，延迟一定时间再恢复所有控件的显示，即可完成独立程序封面的制作过程。两种方法各有利弊，在窗体少的情况下建议用第一种方法，在窗体控件少的情况下建议用第二种方法。

实例2

下雪的界面特技



实例简介

对于下雪的景象大家可能都不陌生，本例中我们将介绍用一幅背景图片作下雪的背景，然后用随机函数 Rnd 产生雪花来不断地下移形成一个下雪的景象的特效界面。

实现过程

创建 Form 窗体

如图 2-1 所示，窗体的名称为 Form1，窗体包括一个按钮 Command1，其属性名称为 Command1,Caption 为开始下雪。其实制作这样一个下雪的景象并不复杂，它的原理是首先在底色为黑色的屏幕上随机画出许多白点（雪花），然后使这些雪花不断地向下移动（重画），反复循环，就成功地模拟了下雪的景象。还可以修改该程序中的一些数据来调整雪花的密度和雪花落下的快慢。

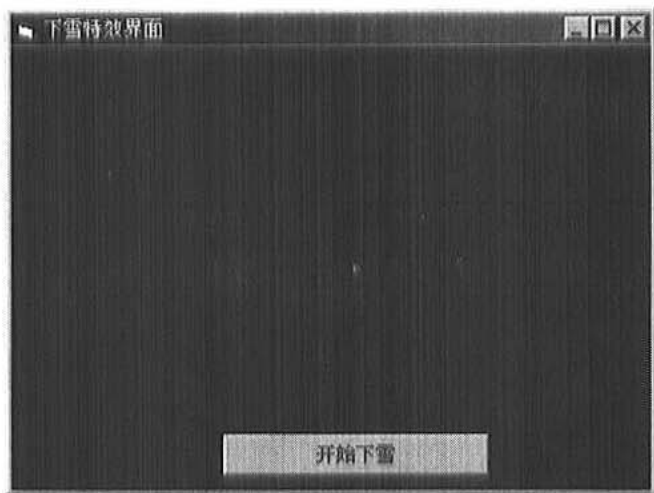


图 2-1 Form1 窗体设计

Form1 窗体的全部代码如下：

```
Dim Snow(1000, 2), Amounty As Integer
Private Sub Command1_Click()
    Form1.Show
    '转让控制权，以便让操作系统处理其它的事件
    DoEvents
```

```

'对随机数生成器做初始化的动作
Randomize
'屏幕上随机画出许多白点（雪花）
Amounty = 325
For J = 1 To Amounty
Snow(J, 0) = Int(Rnd * Form1.Width)
Snow(J, 1) = Int(Rnd * Form1.Height)
Snow(J, 2) = 10 + (Rnd * 20)
Next J
Do While Not (DoEvents = 0)
For LS = 1 To 10
For I = 1 To Amounty
OldX = Snow(I, 0): OldY = Snow(I, 1)
Snow(I, 1) = Snow(I, 1) + Snow(I, 2)
'雪花不断地向下移动（重画）
If Snow(I, 1) > Form1.Height Then
Snow(I, 1) = 0: Snow(I, 2) = 5 + (Rnd * 30)
Snow(I, 0) = Int(Rnd * Form1.Width)
OldX = 0: OldY = 0
End If
Coloury = 8 * (Snow(I, 2) - 10): Coloury = 60 + Coloury
PSet (OldX, OldY), QBColor(0)
PSet (Snow(I, 0), Snow(I, 1)), RGB(Coloury, Coloury, Coloury)
Next I
Next LS
Loop
End
End Sub

```

按以下代码编写在窗体中单击鼠标的事件:

```

Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
unload me
End Sub

```

运行上面的程序，就可以看到漫天的雪花在缓缓落下，地面上还会有积雪。单击鼠标可结束程序，如图 2-2 所示。

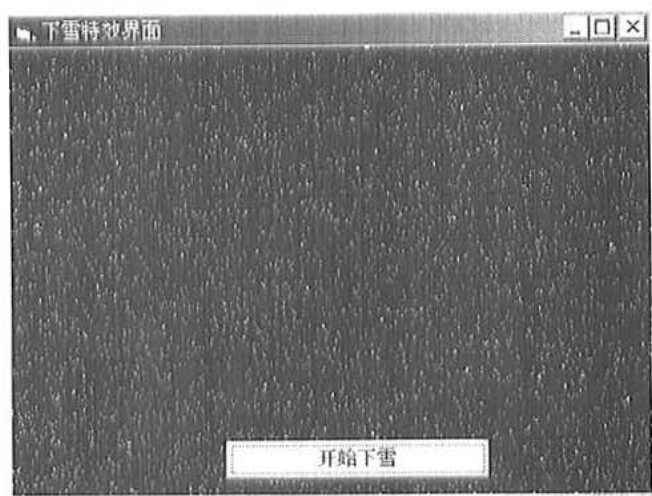


图 2-2 窗体 Form1 运行结果

实例注释和详解

Randomize 语句

Randomize 用 number 将 Rnd 函数的随机数生成器初始化，该随机数生成器给 number 一个新的种子值。如果省略 number，则用系统计时器返回的值作为新的种子值。如果没有使用 Randomize，则（无参数的）Rnd 函数使用第一次调用 Rnd 函数的种子值。

注意：

若想得到重复的随机数序列，在使用具有数值参数的 Randomize 之前，直接调用具有负参数值的 Rnd。使用具有同样 number 值的 Randomize 是不会得到重复的随机数序列的。

本示例用 Randomize 语句初始化随机数生成器。由于忽略了数值参数，因此 Randomize 用 Timer 函数的返回值作为新的随机数种子值。

```
Dim MyValue  
'对随机数生成器做初始化的动作  
Randomize  
'生成 1 到 100 之间的随机数值  
MyValue = Int((100 * Rnd) + 1)
```

LoadPicture 函数

将图形载入到窗体的 Picture 属性、PictureBox 控件或 Image 控件。

语法：

```
LoadPicture([filename], [size], [colordepth],[x,y])
```

LoadPicture 函数语法有下列部分：

Filename——可选的。字符串表达式指定一个文件名。可以包括文件夹和驱动器。如果未指定文件名，LoadPicture 清除图像或 PictureBox 控件。

size——可选变体。如果 filename 是光标或图标文件，指定想要的图像大小。

colordepth——可选变体。如果 filename 是一个光标或图标文件，指定想要的颜色深度。

x——可选变体。如果使用 y，则必须使用。如果 filename 是一个光标或图标文件，指定想要的宽度。在包含多个独立图像的文件中，如果那样大小的图像不能得到时，则使用可能的最好匹配。只有当 colordepth 设为 vbLPCustom 时，才使用 x 和 y 值。

y——可选变体。如果使用 x，则必须使用。如果 filename 是一个光标或图标文件，指定想要的高度。在包含多个独立图像的文件中，如果那样大小的图像不能得到时，则使用可能的最好匹配。

size 的设置值见表 2-1。

表 2-1 size 的设置值

常 量	值	描 述
VbLPSmall	0	系统小图标
VbLPLarge	1	系统大图标，由视频驱动程序决定
VbLPSmallShell	2	外壳小图标大小，由“控制面板”中的 Display Properties 对话框中 Appearance 选项卡上的 Caption Buttons 的大小设置决定
VbLPLargeShell	3	外壳大图标大小，由“控制面板”中的 Display Properties 对话框中 Appearance 选项卡上的图标大小设置决定
VbLPCustom	4	自定义大小，由 x 和 y 参数提供值

colordepth 的设置值见表 2-2。

表 2-2 colordepth 的设置值

常 量	值	描 述
VbLPDefault	0	如果使用指定文件，则为最佳可用匹配
VbLPMonochrome	1	2 色
VbLPVGAColor	2	16 色
VbLPColor	3	256 色

说明：

Visual Basic 可以识别的图形格式有：位图(.bmp)文件、图标(.ico)文件、光标(.cur)文件、行程编码(.rle)文件、元(.wmf)文件、增强的元文件(.emf)、GIF(.gif)文件以及 JPEG(.jpg)文件。赋值不带参数的 LoadPicture 将清除窗体、图片框及图像控件中的图形。

为了加载在 PictureBox 控件和 Image 控件中显示的图形或加载作为窗体背景的图形，必须将 LoadPicture 的返回值赋给要显示该图片的对象的 Picture 属性。

例如：

```
Set Picture = LoadPicture("PARTY.BMP")
```



```
Set Picture1.Picture = LoadPicture("PARTY.BMP")
```

如果要将图标赋予窗体，则要把 LoadPicture 函数的返回值赋给 Form 对象的 Icon 属性：

```
Set Form1.Icon = LoadPicture("MYICON.ICO")
```

图标也可以被赋予除 Timer 控件和 Menu 控件外的其它控件的 DragIcon 属性。

例如：

```
Set Command1.DragIcon = LoadPicture("MYICON.ICO")
```

使用 LoadPicture 可将图形文件载入到系统剪贴板，如下所示：

```
Clipboard.SetData LoadPicture("PARTY.BMP")
```

实例3

分割窗体

