曹国钧 编著

Visual BASIC 3.0—4.0 for Windows 实用技巧

上海科学普及出版社

Visual BASIC 3.0－4.0 for Windows 实用技巧

曹国钧　编著

(沪)新登字第 305 号

责任编辑：胡名正　刘瑞莲

Visual BASIC 3.0—4.0 for
Windows 实用技巧
曹国钧　编著
上海科学普及出版社出版
（上海曹杨路 500 号　邮政编码　200063）

新华书店上海发行所发行　常熟高专印刷厂印刷
开本 787×1092 1/16　印张　23　字数　553 000
1996 年 9 月第 1 版　1996 年 9 月第 1 次印刷

ISBN 7-5427-1147-4/TP・277　定价：29.00 元

内 容 提 要

VB 3.0—4.0 是 Windows 环境下的最好的程序设计工具，自从 1.0 版发表以来，Visual BASIC 几乎每年被美国的 Byte 杂志评为优秀奖。

由于 Visual BASIC 具有 C++和 Windows 的特征，因此在使用 VB 时应充分发挥这一优势。本书着重从这个角度帮助读者更灵活地使用 VB，针对 Windows 用户十分关心的问题，介绍 VB3.0—4.0 的应用技巧，例如：VB 在图形图像处理、QueryDef、数据库、定制光标、Windows 资源、多媒体等方面的应用，HELP 的制作与调用以及 VBX 的自定义编制与应用等。在附录中还给出详尽而实用的 VB 使用资料，指导读者用好、用活 VB。

本书适用于 VB 用户，Windows 用户，程序员，培训班师生，大专院校有关专业师生。

目　录

试读结束：需要全本请在线购买：　www.ertongbook.com

第一章　VB 的图形图像处理技巧

随着 Windows 图形用户界面的流行，在 Windows 中处理图形图像就成为重要的内容。在本章中，我们给出若干图形图像的技巧。包括如下内容：

1. VB 下的图形卷轴与 HotSpot。
2. 利用 VB 方便地实现剪贴板图像快速存盘。
3. 利用 VB 中的 3D 控件制作界面。
4. 利用 VB 特别功能构造动态界面。
5. 用 VB 在 Windows 中实现图像的旋转。
6. Windows 图标动画效果的实现方法。

1.1　VB 的图形卷轴与 HotSpot

在使用 VB 进行图形程序设计中，通常尽量把图片制作成符合屏幕或显示区域范围的大小。但是，有时要求考虑整体效果必须将一大张图片放在一个较小的区域中显示，或者因图形太大而屏幕无法完整地显示，此时就需要采用其他方法，例如卷轴，以便让用户观察其他部分。

另外，在 VB 程序中，使用系统图作为软件的主菜单，在屏幕上看不到一个按钮，您可以用鼠标器单击系统图中的任何一个部件，相应的部件功能就实现了，这就是我们称之为“HotSpot”的技巧。

下面我们讨论一下卷轴与 HotSpot 的实现方法及使用技巧。

1.1.1　图形卷轴的视线

首先我们在 VB 中建立一个 txForm1 的窗体(Form)，利用 VB 的 ToolBox(工具箱) 在 txForm1 中建立一个图形框(PictureBox) txPic1 以及两个滚动条 HScroll1 和 VScroll1，如图 1-1 所示：

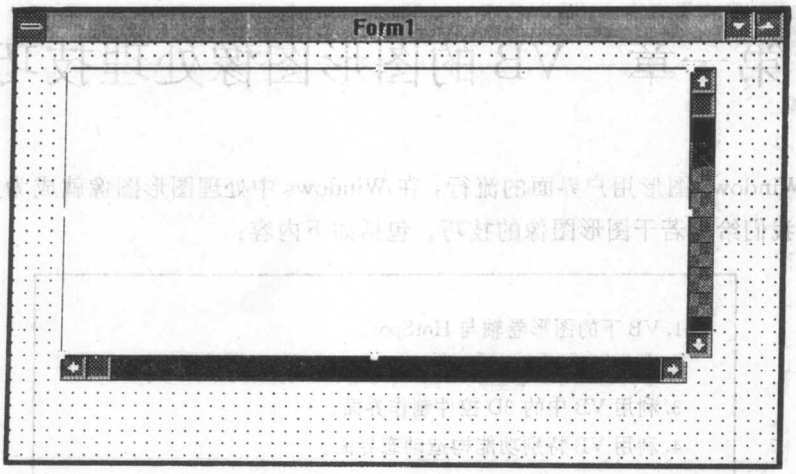

图 1-1

下面的 VB 代码可实现图形卷轴功能：

```
sub Form _ load()
' 以下属性也可在 VB 的属性窗口中给出
txPic1. autosize=True
' 这里选用的 BMP 图形为 Windows 95 的安装图形
txPic1. picture=loadPicture("d:\pwin\安装. bmp")
txForm1. HScroll1. Min=0
' 减去一个屏幕的宽度
txForm1. HScroll1. Max=txForm1. txPic1. Width-640
txForm1. VScroll1. Min=0
' 减去一个屏幕的高度
txForm1. VScroll1. Max=txForm1. txPic1. Height-480
txForm1. VScroll1. SmallChange=20                    ' 步长
txForm1. HScroll1. SmallChange=20                    ' 步长
end sub
' 卷轴程序
sub VScroll1 _ change()
txPic1. Top=-VScroll1. Value                         ' 垂直卷轴
end sub
sub HScroll1 _ change()
txPic1. left=-HScroll1. Value                        ' 水平卷轴
end sub
```

在 VB 中执行上面的程序，我们可以得到下面的屏幕，参见图 1-2：

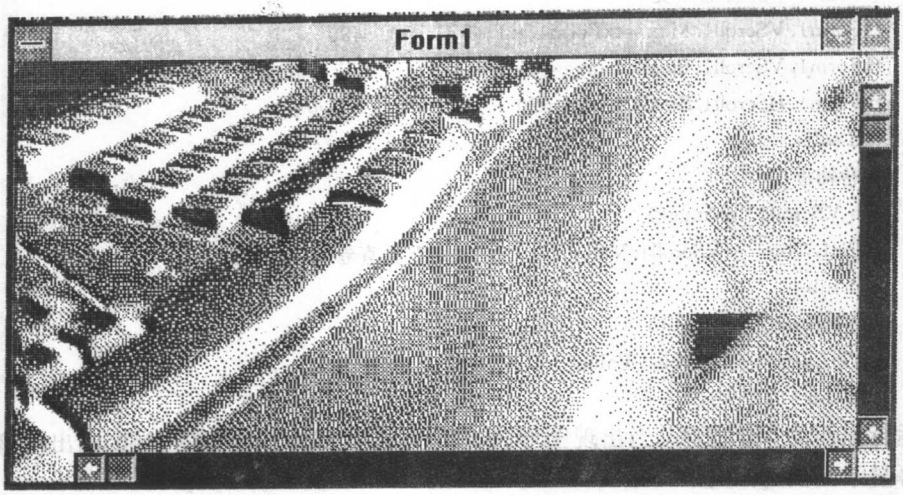

图 1-2

您可以用鼠标器按住左键移动左右滚动条或上下滚动条，从而看到“安装. BMP”图形的全部。

在上面的 VB 程序中，我们将图形框的左边界 txPic1. left 和上界 txPic1. top 设置为负值，目的在于当卷轴往右移动时，图形往左移动，当卷轴向左移动时，图形往右移动，这样就可以看到隐藏的部分了。

另外，为了配合 HotSpot 的实现，可在卷轴程序中增加一个整屏显示功能，即双击图形时就进入了 HotSpot 的整屏显示。

建立另一个窗体 txForm2，在此窗体中建立一个图像(Image)框 txImage1 整屏显示程序。

```
sub txPic1 _ DblClick()
txForm2. txImage1. Width = 640
txForm2. txImage1. Height = 480
txForm2. txImage1. Stretch = True
end sub
```

另外，还需要将前面的卷轴 VB 程序修改如下：

```
sub Form _ load()
' 以下属性也可在 VB 的属性窗口中给出
txPic1. autosize = True
' 这里选用的 BMP 图形为 Windows 95 的安装图形
txPic1. picture = loadPicture("d:\pwin\安装. bmp")
' 全屏显示安装. BMP 图形
txForm2. txImage1. Picture = loadPicture("D:\pwin\安装. bmp")
txForm1. HScroll1. Min = 0
' 减去一个屏幕的宽度
```

```
txForm1. HScroll1. Max=txForm1. txPic1. Width-640
txForm1. VScroll1. Min=0
' 减去一个屏幕的高度
txForm1. VScroll1. Max=txForm1. txPic1. Height-480
txForm1. VScroll1. SmallChange=20                    ' 步长
txForm1. HScroll1. SmallChange=20                    ' 步长
end sub
' 卷轴程序
sub VScroll1 _ change()
txPic1. Top=-VScroll1. Value                         ' 垂直卷轴
end sub
sub HScroll1 _ change()
txPic1. left=-HScroll1. Value                        ' 水平卷轴
end sub
```

运行上面的 VB 程序，当出现图形时，在该图形上双击鼠标器的左键，则出现该图形的全屏幕显示画面，如下图 1-3：

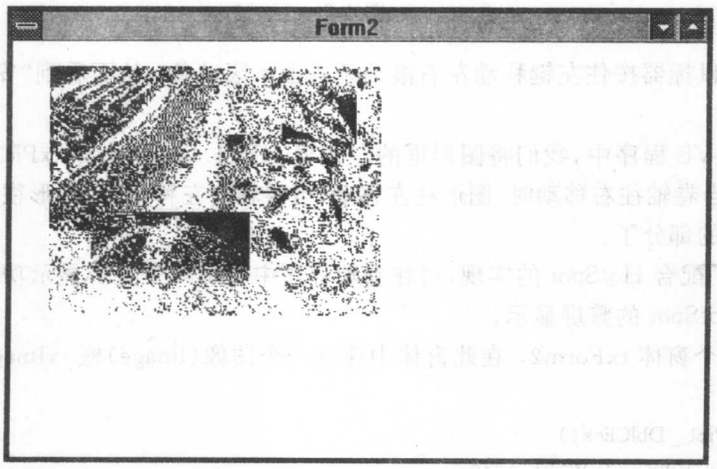

图 1-3

1.1.2 HotSpot 的实现

首先将窗体 txForm2 的图像框 txImgage1 的参数设置如下：

```
txImage1. Caption="Image"
txImage1. Stretch=True
txImage1. Height=480
txImage1. Width=640
```

下面的 VB 代码可实现 HotSpot 的效果。其中 XX、YY 设置为 Global 变量，它主要用于记录鼠标器当前在图像中的位置。

```
sub Image1 _ DblClick()
```


试读结束：需要全本请在线购买： www.ertongbook.com

```
' 将 HotSpot 点定位到屏幕的中心位置上
txForm1.txPic1.Left=320-(xx/15.1875/640)*txForm1.txPic1.Width
txForm1.txPic1.Top=240-(yy/15.25/480)*txForm1.txPic1.Height
if -txForm1.txPic1.Left>txForm1.txPic1.Width-640 then
    txForm1.txPic1.Left=-txForm1.txPic1.Width+640
end if
if -txForm1.txPic1.top>txForm1.txPic1.Height-480 then
    txForm1.txPic1.top=-txForm1.txPic1.Height+480
end if
if txForm1.txPic1.Left> 0 then
    txForm1.txPic1.Left=0
end if
if txForm1.txPic1.Top> 0 then
    txForm1.txPic1.Top=0
end if
if -txForm1.txPic1.Left>txForm1.HScroll1.Max then
    txForm1.txPic1.HScroll1.Value=txForm1.HScroll1.Max
else
    txForm1.HScroll1.value=-txForm1.txPic1.Left
end if
if -txForm1.txPic1.Top>txForm1.VScroll1.Max then
    txForm1.VScroll1.Value=txForm1.VScroll1.Max
else
    txForm1.VScroll1.Value=txForm1.VScroll1.Top
end if
Hide
end sub
Sub Image1_MouseMove(Button as Integer,Shift as integer,X as Single,Y as Single)
XX=X
YY=Y
end sub
```

运行上面的 VB 程序，在 Image 图像上连击两下就可在 Picture 图片框中得到连击点恰好位于屏幕的中心位置，实现了 HotSpot 效果。

1.2 利用 VB 方便地实现剪贴板图像快速存盘

Windows 的用户经常需要将剪贴板中的图像保存在图形文件中，传统的方法是将剪贴板中的图像粘贴到画笔(PaintBrush)中，但这种方法的一个不便之处就是必须事先估计好图像的尺寸，有时需要反复多次才能获得满意的结果。另一方面，由于画笔软件的限制，采用这种方法对于全屏幕图像往往会丢掉最下面的一部分。

为了解决这个问题，我们利用 VB 代码可把剪贴板中的图像直接存储到 BMP 文件中，还可以存储剪贴板上的图片(.WMF 图元文件)，非常方便。

1. 我们首先建立一个窗体(Form)，将该窗体的 Name 属性设置为 frmClipSave，不必画任何控制。

2. 在窗体中设计五个菜单控制项。

这五个菜单控制项的正文分别为：

- Option——选项(主菜单项)

下面四个为 Option 的子菜单项。

- ReRead from ClipBoard——从剪贴板中重读图像
- Save——保存图像
- About——程序信息
- Exit——退出程序

这些菜单项可用 VB 的“Windows”菜单中的“Menu design”命令直接设计。

下图 1-4 和 1-5 就是“Option”，“ReRead from ClipBoard”菜单项的设计画面。其他类似。

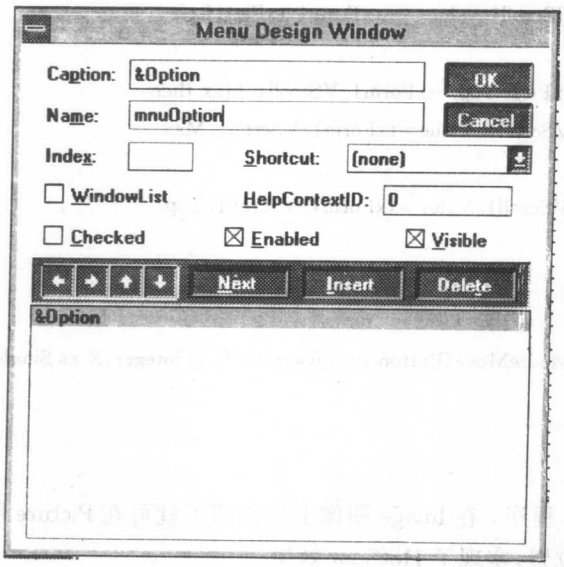

图 1-4

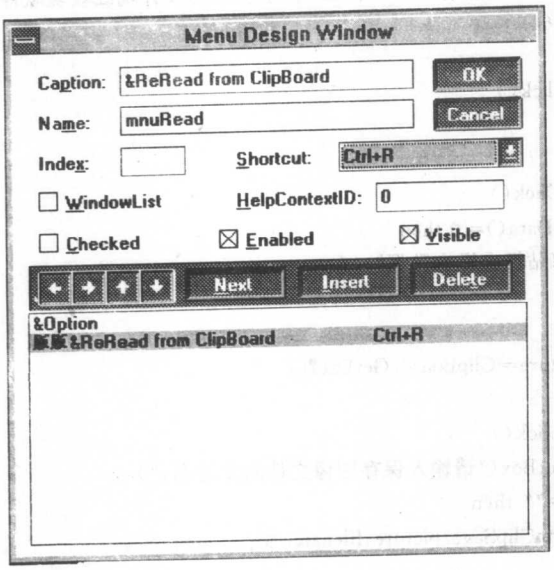

图 1-5

下面的图 1-6 就是菜单设计完成后的画面：

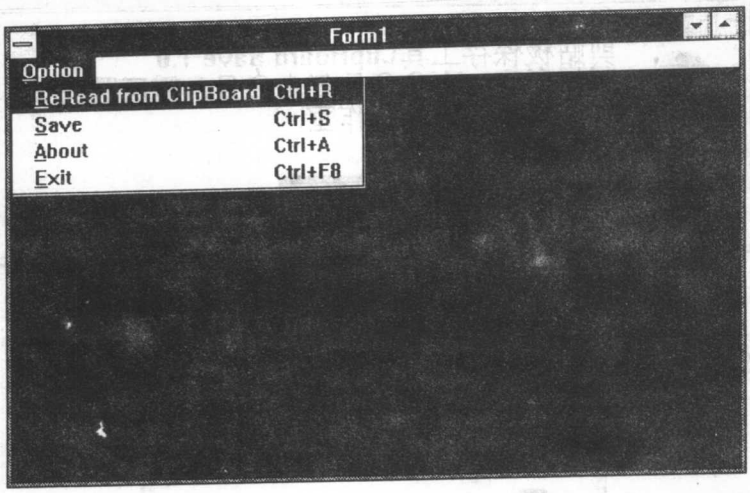

图 1-6

然后，输入下面的 VB 代码就能实现剪贴板中图像快速存盘。

```
sub Form _ load()
mnuAbout _ Click
mnuRead _ Click
End Sub
Sub MnuAbout _ Click()
```

```
msg $ = "剪贴板保存工具 ClipBoard Save 1.0"+chr $ (13)+chr $ (10)
msg $ =msg $ +"版权所有,1995 年 12 月,重庆医药设计院红玫瑰软件工作室"
MsgBox msg $ ,64
End Sub
sub mnuExit _ Click()
  end
End Sub
sub mnuRead _ Click()
if ClipBoard. GetData()=0 then
  msgBox "在剪贴板中没有数据",16
  end
end if
frmClipSave. Picture=ClipBoard. GetData()
End Sub
Sub mnuSave _ Click()
filename $ =InputBox("请输入保存图像文件的文件名:")
if filename $ <>"" then
   savePicture frmClipSave. picture,filename $
end if
End Sub
```

执行上面的 VB 程序，首先出现版权信息，参见下面的图 1-7：

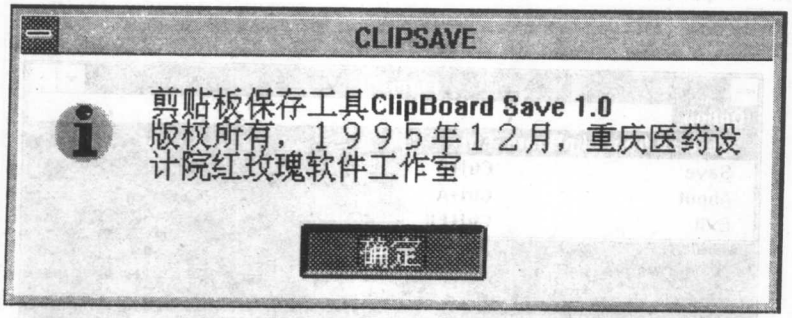

图 1-7

在图中按“确定”按钮，若剪贴板中没有数据，则出现图 1-8 所示的屏幕：

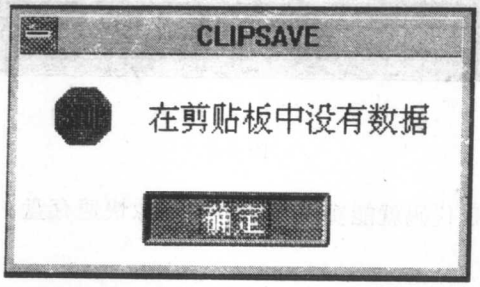

图 1-8

图 1-8 提示“剪贴板中没有数据”。按“确定”按钮，则退出该程序。否则，就出现图 1-9 所

示的屏幕，在屏幕上显示剪贴板中的内容。

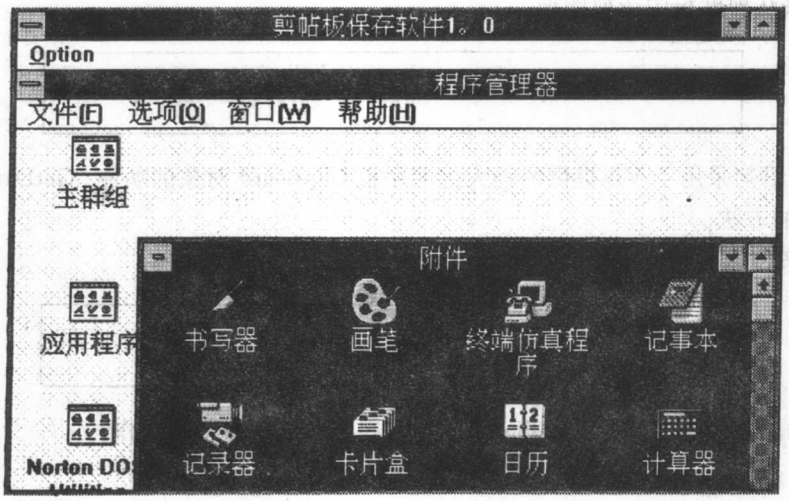

图 1-9

若您的剪贴板中的内容发生了变化，则可选取“Option”菜单中的“ReRead from ClipBoard”命令，在剪贴板中重新读入，参见图 1-6；

若您需要将剪贴板中的内容保存在图像文件中，可选取“Option”菜单中的“Save”命令，此时，将出现下图 1-10 所示的提示屏幕。

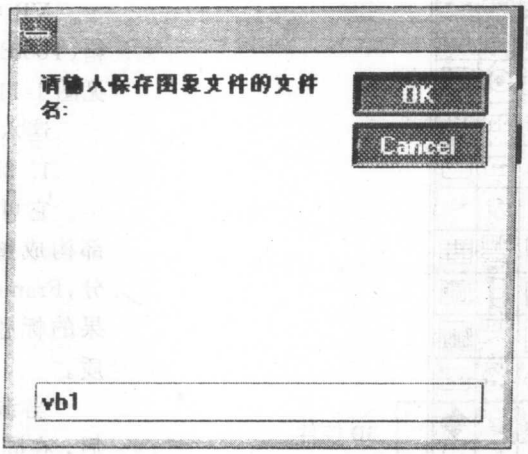

图 1-10

图 1-10 要求您输入一个图像文件名，例如 VB1，按“OK”按钮，则将剪贴板的内容保存在 VB1.BMP 文件中。

要退出此程序，只要选取“Option”菜单中的“Exit”命令即可。

另外，我们在“Option”菜单中各个命令设置了热键，例如“ReRead from ClipBoard”的

热键为 Ctrl+R、"Save"的热键为 Ctrl+S 等。

我们从上面的 VB 代码中可以看出，该程序的核心仅仅两条语句：

· 用来从剪贴板中读取图像：

```
fmtClipSave. Picture=ClipBoard. GetData()
```

上面该语句采用了 VB 提供的强大的剪贴板 ClipBoard 对象的功能。GetData() 为 ClipBoard 对象的方法。

· 用来存储剪贴板中的图像：

```
SavePicture frmClipSave. Picture,Filename $
```

上面的语句利用 VB 的 SavePicture 功能，避免了繁琐的图像操作。

1.3 利用 VB 中的 3D 控件制作界面

在 VB 中，利用六个 3D 控件可容易地制作出具有艺术化三维外观的窗口界面，而且不需要特别编写程序代码。

1.3.1 VB 的 3D 控件

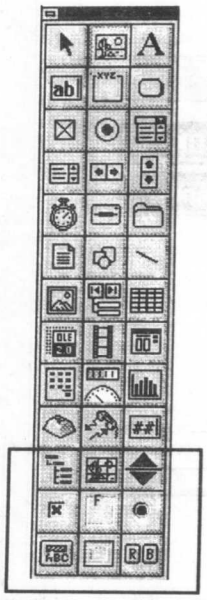

图 1-11

VB 中的 3D 控件位于工具箱(ToolBox)最后六个图标上，参见图 1-11 所示：

这六个 3D 控件介绍如下：

1. 3D 框架(SSFrame3D)

它可以把别的控件放到其内部构成整个界面的一个组成部分，Frame 可以选择具有三维效果的标题，并有 Alignment 的性质。

当调整好框架的三维外观后，在同一表格中创建新的框架时，其属性如同前一个框架。

下图 1-12 就是 3D 框架的屏幕显示效果：

试读结束：需要全本请在线购买： www.ertongbook.com