

Visual Basic 6.0

高级编程技巧——

多媒体通信篇

白 鹏 莫卫东 张福萍 张晓敏 编著



西安交通大学出版社

TP312

B19

Visual Basic 6.0

高级编程技巧

——多媒体通信篇

白 鹏 莫卫东 张福萍 张晓敏 编著

西安交通大学出版社

内 容 简 介

JS200/17

Visual Basic 6.0 是微软公司最新推出的 Visual Studio 6.0 可视化应用程序开发工具组件中的一员,也是当今世界上最流行的可视化编程工具。本书是“Visual Basic 6.0 高级编程技巧系列”丛书中的一本——多媒体通信篇。主要内容是全面系统介绍在 Visual Basic 6.0 环境下如何利用控件进行程序设计,除了讲解具有针对性的实用方法之外,并在其中插入了大量全新的实例。在这些实例中有许多都是作者在实际设计中采用过的。本书共分 10 章,其内容包括多媒体与网络通信的编程技巧和实例程序等。书中通过作者的编程实践经验,用大量的实例说明三类控件的编程方法,从而使读者进一步加深对 VB 控件的理解和应用技巧。

本书立意独特,体系完整,内容充实,图文并茂,使用最新的实例,讲解最实用的方法,奉献最有价值的技巧,使读者能在最短的时间内掌握使用 Visual Basic 6.0 开发出最好的应用程序的方法和技巧。

本书不仅适用于 Visual Basic 的初学者作为学习和开发 Windows 应用程序的理想参考书,更适用于所有软件开发人员作为常备手册性参考书。

图书在版编目(CIP)数据

Visual Basic6.0 高级编程技巧:多媒体通信篇/白鹏
等编. -西安:西安交通大学出版社,2000.1
ISBN 7-5605-1189-9

I.V… II.白… III.BASIC 语言-程序设计 IV.TP312

中国版本图书馆 CIP 数据核字(1999)第 72427 号

*

西安交通大学出版社出版发行

(西安市咸宁西路 28 号 邮政编码:710049 电话:(029)2668316)

西安华宇印刷厂印装

各地新华书店经销

*

开本:787 mm×1 092 mm 1/16 印张:17 字数:420 千字

1999 年 12 月第 1 版 1999 年 12 月第 1 次印刷

印数:1~5 000 定价:25.00 元

若发现本社图书有倒页、白页、少页及影响阅读的质量问题,请去当地销售
部门调换或与我社发行科联系调换。发行科电话:(029)2668357,2667874

前 言

Visual Basic 6.0 是微软公司最新推出的 Visual Studio 6.0 可视化应用程序开发工具组件中的一员，也是当今世界上最流行的可视化编程工具。新的 Visual Basic 系统不但简化了可视化编程的方法，而且使得应用程序的开发流程更加流畅，功能也更加强大。特别是 Visual Basic 6.0 在数据库和 Internet 应用程序开发两方面的改进，使程序员能够在当前的环境中轻松自如地开发出功能不凡的各种数据库与多媒体应用程序。因此，Visual Basic 6.0 已经成为最受欢迎的 Windows 应用程序可视化开发工具。

为了帮助众多的程序设计人员尽快地掌握 Visual Basic 6.0 进行程序设计的方法，有关 Visual Basic 6.0 的各类技术参考书不断问世。在这些参考书中不乏优秀之作，但更多地是介绍 Visual Basic 6.0 进行程序设计的基本方法和技巧，属于入门级的技术参考书。我们的这套“Visual Basic 6.0 高级编程技巧”丛书，试图从 Visual Basic 编程的各个角度，分别系统地介绍 Visual Basic 编程方法，更深入地分析编程中的策略与技巧，并通过大量的实例来讲解各种方法与技巧的实际应用，具有很强的实战效果。其中的许多示例都是作者在实际工作中采用过的。可见，本丛书是介绍 Visual Basic 编程方法与技巧的更高层次的技术参考书，广大的 Visual Basic 编程人员一定会在其中获得有益的帮助和启示。

这套丛书共有 4 本，分别以四个专题介绍 Visual Basic 程序设计中的更高层次的方法与技巧。它们是：

- 窗体、菜单篇
- 控件篇
- 多媒体通信篇
- ADO 数据访问篇

本书是该丛书之一，《Visual Basic 6.0 高级编程技巧——多媒体通信篇》。全书立意独特，体系完整，内容充实，图文并茂，使用最新的实例，讲解最实用的方法，奉献最有价值的技巧。使读者能在最短的时间内掌握使用 Visual Basic 6.0 编程方法与技巧，以开发出最好的应用程序。本书共有 10 章，具体的内容安排如下：

第 1 章 图像的编程技巧与实例

图像是多媒体的重要组成部分，在本章中，将讨论有关图像的编程技巧和实例，基本包括图像的各个方面，具体涉及到的有 PictureBox 控件、PictureClip 控件、Image 控件、Shape 控件、Line 控件等与图像有关的控件和与图像有关的 API 函数。

第 2 章 声音的编程技巧与实例

声音是多媒体的重要部分，在本章中详细讨论有关声音的控件和相关的 Windows 函数的使用方法、播放声音文件的方法以及声卡的检测方法。

第 3 章 文字的编程技巧与实例

文字是多媒体的重要部分，在本章中，将详细讨论有关文本框和文字在多媒体应用中的编程技巧，包括 TextBox 控件、Label 控件、RichTextBox 控件的编程技巧及有关对文字的控制技

巧。

第 4 章 可视界面的编程技巧与实例

在本章中，将详细讲述有关动态鼠标、渐变色背景、快捷工具栏、浮动快捷工具栏等有关可视化界面编程的技巧和实例。

第 5 章 通讯的编程技巧与实例

本章详细讨论两个有关通信的控件 MSComm 控件和 Winsock 控件。MSComm 控件用于串口通信，Winsock 控件用于网络间的数据通讯。

第 6 章 网络与 Internet 的编程技巧与实例

本章将全面介绍如何编写有关 Internet 方面的应用，内容包括使用 Microsoft Exchange 发送电子邮件、使用 VB 设计 Email 程序、使用 VB 设计自己的浏览器、使用 IE 控件访问 Internet、数据库数据向 HTML Web 主页输出等。

第 7 章 有关打印的编程技巧

本章通过实例，具体阐述用 VB 语言打印图文报表、图像、数据表格等问题。

第 8 章 多媒体菜单的编程技巧与实例

本章讨论几个如何使设计的菜单更漂亮、更实用的技巧，包括在菜单中加入图标、自动隐藏和显示菜单、显示菜单的历史记录等内容。

第 9 章 鼠标与键盘编程技巧

由于鼠标和键盘是计算机信息录入和控制的主要工具，故鼠标和键盘的编程和使用技巧在 VB 中占有相当重要的地位，本章通过短小精悍的实例具体讨论鼠标和键盘的编程技巧和使用技巧。

第 10 章 其他编程技巧

本章介绍一些具有较高使用价值的编程技巧。如对目录或文件夹实时监控的方法与技巧等。

作者

1999. 11

目 录

绪论	(1)
----------	-----

第 1 章 图像的编程技巧与实例

1.1 图像的旋转	(3)
1.1.1 逐点计算的旋转方法	(3)
1.1.2 使用 API 函数的旋转方法	(6)
1.2 图像变换效果	(9)
1.2.1 “百叶窗”效果	(9)
1.2.2 “马赛克”效果	(11)
1.2.3 “窗帘”效果	(15)
1.3 动画的实现	(18)
1.4 播放动态图像	(20)
1.4.1 用 Animation 控件播放动态图像	(20)
1.4.2 用 Multimedia MCI 控件播放动态图像	(21)
1.4.3 用 API 函数 mciExecute 播放动态图像	(23)
1.5 不规则多边形与平滑曲线的绘制	(24)
1.5.1 不规则多边形的绘制	(24)
1.5.2 平滑曲线的绘制	(26)
1.6 图像的漫游、放大和缩小	(30)
1.6.1 图像的漫游	(30)
1.6.2 图像的放大和缩小	(32)
1.7 图像的复制和粘贴	(38)
1.8 保存图像到数据库的实例	(39)
1.9 综合实例--地理信息系统 (GIS) 中图像热点信息查询	(42)

第 2 章 声音的编程技巧与实例

2.1 有关声音的基本概念	(56)
2.2 Multimedia MCI 控件	(57)
2.3 MCI 接口处理声音方法	(62)
2.4 用 Multimedia MCI 控件播放 *.WAV 和 *.MID 格式音乐	(65)
2.5 用 Multimedia MCI 控件 CD 音乐光盘播放器	(67)
2.6 检测声卡的方法	(69)
2.7 用 XmPlayer 自动播放 VCD	(70)

第3章 文字的编程技巧与实例

3.1 文字的旋转——用任意角度显示文字技术实现的数字钟实例	(72)
3.2 用 API 函数实现英文取词的实例	(79)
3.3 用 API 函数计算 TextBox 中行数的实例	(81)
3.4 如何进行数字与汉字大写的转换	(82)
3.5 用 Label 控件创建访问键	(87)
3.6 用 TextBox 控件显示文件	(88)
3.7 用 TextBox 控件检测文件的改变	(89)
3.8 限定汉字和西文字符输入个数的实例	(90)
3.9 用 RichTextBox 控件实现的综合字符编辑器	(92)

第4章 可视界面的编程技巧与实例

4.1 动态鼠标的实现	(97)
4.2 在 PictureBox 控件中实现渐变颜色	(100)
4.3 如何创建浮动的快捷菜单栏	(101)
4.4 创建动态凹陷的快捷菜单栏	(104)
4.5 如何建立动态的过渡窗体	(109)
4.6 建立自己的应用软件屏幕保护程序	(111)
4.7 变暗的屏幕	(114)
4.8 利用 TreeView 控件建立可视层次和结构的数据库	(117)
4.9 利用 ImageList 控件进行图像的叠加	(126)

第5章 通信的编程技巧与实例

5.1 MSComm 控件功能和使用说明	(130)
5.2 用 MSComm 控件进行数据通信的实例	(131)
5.3 用 MSComm 控件拨电话的实例	(134)
5.4 用 Winsock 控件实现点对点数据通信的实例	(135)
5.5 用 API 函数实现串行通信的查询方法	(137)

第6章 网络与 Internet 的编程技巧与实例

6.1 使用 Microsoft Exchange 发送电子邮件	(143)
6.2 使用 VB 设计 E-mail 程序	(144)
6.3 使用 VB 设计自己的浏览器	(148)
6.4 使用 IE 控件访问 Internet	(151)
6.5 数据库数据向 HTML Web 主页输出实例	(153)
6.6 建立可判断计算机在网络中名称的登录窗体	(156)
6.7 MUD 客户端服务程序	(158)

6.8 断开与 Internet 的连接实例	(162)
6.9 获取系统当前是否处于 Internet 连接状态的实例	(164)

第 7 章 有关打印的编程技巧与实例

7.1 利用 Excel 预览、打印报表	(166)
7.2 利用 Word 打印文本文件	(170)
7.3 可打印超宽 (A3 纸) 表格程序	(173)
7.4 图文混排预览及打印程序	(179)
7.5 图像任意区域的打印程序	(208)

第 8 章 多媒体菜单的编程技巧与实例

8.1 在菜单中加入图标	(211)
8.2 能自动隐藏和显示的菜单	(213)
8.3 使用 API 函数的自动隐藏和显示菜单	(219)
8.4 能显示和保存历史记录菜单	(222)
8.5 如何改变屏幕的分辨率	(227)

第 9 章 鼠标与键盘编程技巧

9.1 显示 ListBox 控件中鼠标所在选项的位置	(230)
9.2 鼠标的“离开”事件	(231)
9.3 用 API 函数获取鼠标位置的实例	(232)
9.4 键码值检测实例	(233)
9.5 鼠标移动里程的计算实例	(234)
9.6 捕捉窗体鼠标的实例	(235)

第 10 章 其他编程技巧与实例

10.1 能计算任意四则运算的计算器	(238)
10.2 对目录或文件夹实时监控的实例	(252)
10.3 格式化磁盘的实例	(255)
10.4 获取 CPU 使用率的实例	(257)
10.5 帮助文件的制作与实例	(259)

绪 论

1. VB 的发展历史与特点

1991 年 7 月, Microsoft 公司推出了 VisualBasic 语言 (简称 VB)。VB 是可视化的计算机编程语言。它鲜明的特点是:

- . 容易学习。
- . 容易使用, 面向对象。
- . 数据处理能力强。
- . 语法简单, 程序容易修改。

这些原因使 VB 成为当前计算机流行的编程语言之一。VB 经过几年的发展从 16 位计算机的编程已发展为 32 位计算机的编程, 其版本号已升级为 6.0, VB 在可视化界面和数据库方面与其他语言相比有绝对的优势。

2. VB 程序的组成

一个标准的 VB 程序由工程、窗体和代码三部分组成。

工程 (Project): 完成一个 Windows 下的任务称工程。它由若干个功能窗体组成。

窗体 (Form): 窗体是由多个图像式的对象组成。它的主要功能是用来接受和输出信息, 作为用户与计算机间的界面。

代码 (Code): 代码是程序的核心, 整个程序的运转由代码来协调。

3. 控件的基本概念与简介

在 VB 中, 控件有方法、属性和事件三个特性。这三个特性充分体现了 VB 的面向对象的特点。控件的集合构成了窗体 (Form), 窗体的集合构成工程 (Project), 代码 (Code) 有机地将控件窗体和工程组合在一起。在某种程度上, 编程可以看成是“组合的艺术”, 通过有限的程序语言和各类控件, 经过程序语言“组合的艺术”形成应用软件, 完成强大的功能。本书所讨论的编程技巧就是“组合艺术”的具体表现。

4. 本书中控件中英文名称及表示方法

在本书中采用两种方法表示控件:

(1) VB 系统的默认形式。例如 CommandButton 控件的 VB 默认表示形式为 Command1, Command2 等。

(2) 采用前缀形式表示控件。即每一个控件用在其英文名称中选取的三个

字母来表示。例如 CommandButton 控件的前缀形式表示为 Cmd，有一个 CommandButton 控件，完成退出系统的功能，采用前缀形式表示为 CmdExit。本书中常见的控件中英文名称和前缀列表如下：

控件英文名称	控件中文名称	前缀
Animation	动画按钮	ani
CheckBox	复选框	chk
ComboBox	组合框	cbo
CommandButton	命令按钮	cmd
CommonDialog	公用对话框	dlg
Communications	通信	com
DbcomboBox	数据约束组合框	dbc
DbGrid	数据约束表格	dbg
DbListBox	数据约束列表框	dbl
DirectoryListBox	目录列表框	dir
Drive list box	驱动器列表框	drv
File list box	文件列表框	fil
Form	窗体	frm
Frame	框架	fra
HorizontalScrollBar	水平滚动条	hsb
Image	图像	img
ImageList	图像列表	ils
Label	标签	lbl
List box	列表框	lst
ListView	列表视图	lsv
OLE	OLE 客户	ole
PictureBox	图片框	pic
Picture clip	图片粘贴板	clp
ProgressBar	进度栏	prg
Shape	形状	shp
Slider	滑动条	sld
StatusBar	状态栏	stb
TextBox	文本框	txt
TreeView	树形视图	tre
Toolbar	工具栏	tlb
TabStrip	条纹标签钮	tbs
Timer	定时器	tmr
UpDown		upd
VerticalScrollBar	垂直滚动条	vsb

在本书中，为了说明问题的方便，在示例的代码中采用上述的两种控件表示方法。在可能的条件下，建议尽量采用前缀表示方法来表示控件。

第1章

图像的编程技巧与实例

图像是多媒体的重要组成部分，在本章中，将讨论有关图像的编程技巧和实例，基本包括图像各个方面，具体涉及到的有 PictureBox 控件、PictureClip 控件、Image 控件、Shape 控件、Line 控件等与图像有关的控件和与图像有关的 API 函数。

1.1 图像的旋转

图像的旋转是图像的基本处理方法。图像的旋转有两种方法：逐点计算的方法和使用 API 函数的方法。

1.1.1 逐点计算的旋转方法

1. 逐点计算方法的功能与思路

图像是由多个点组成的，如果将每一个点都按一定的角度进行旋转，整幅图像也就进行了旋转，因此，图像的旋转变为点的旋转。

2. 原理与步骤

逐点计算方法的原理与步骤为：

- (1) 取将要变换图像框中的一点，包括颜色信息，称为源点；
- (2) 在另外图像框相同的坐标系中，根据要旋转的角度计算该点的新坐标，称为目标点；
- (3) 将源点的颜色赋值给计算出来的目标点坐标；
- (4) 反复进行上述步骤，直到将所有的点都计算完毕。

3. 具体实现算法

在图像框中有一源点，其坐标为 (X, Y) ，图像要旋转的角度为 K ，计算公式为：

$$X0 = X \cdot \cos(-K) + Y \cdot \sin(-K)$$

$$Y0 = Y \cdot \cos(-K) - X \cdot \sin(-K)$$

其中坐标 $(X0, Y0)$ 为旋转后的目标点坐标。

4. 运行和设计时屏幕

图 1.1 所示为采用上述方法编写的实例运行屏幕。在屏幕中有两个图像控件 PictureBox, 名称分别为 PicSource 和 PicTarget, PicSource 为源图像, PicTarget 为经过旋转后的目标图像, 在文本框 TextBox 控件 TxtValue 中输入所要旋转的角度, 再选择“确定”即可。

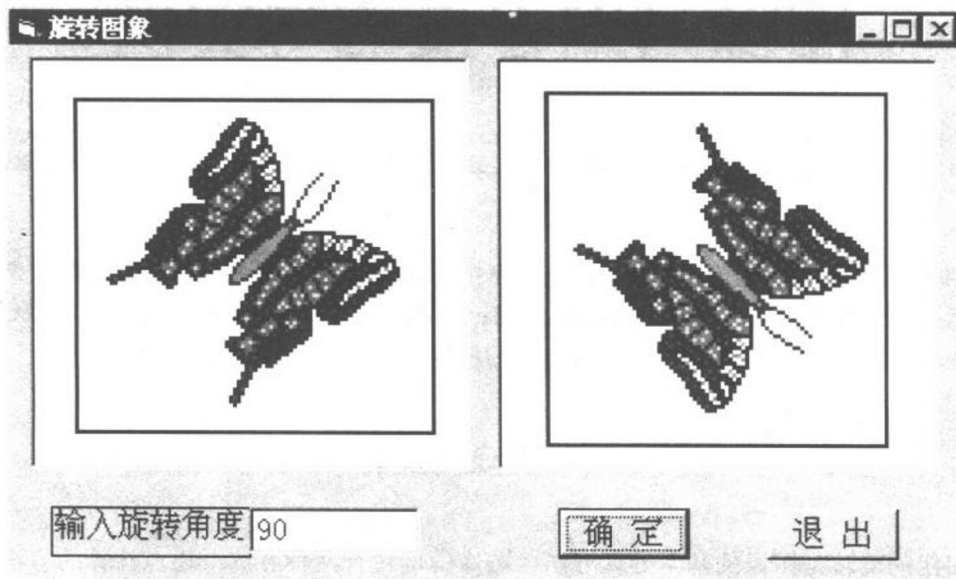


图 1.1 逐点计算的旋转方法实例运行屏幕

窗体设计时的屏幕, 如图 1.2 所示。

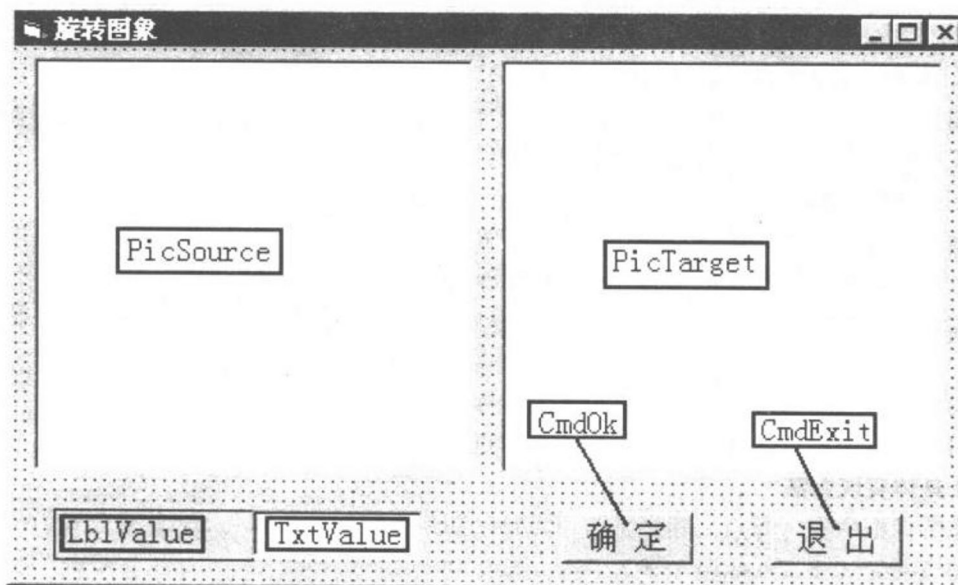


图 1.2 逐点计算的旋转方法设计时的窗体屏幕

5. 完整程序代码清单

实现上述功能的完整程序代码如下:

```
Option Explicit
Const Pi = 3.141592653589
Private Sub CmdOk_Click()
    Dim X As Double
    Dim Y As Double
    Dim X0 As Double
    Dim Y0 As Double
    Dim Xc As Double
    Dim Yc As Double
    Dim Xn As Double
    Dim Yn As Double
    Dim K As Single
    PicTarget.Cls
    K = Val(TxtValue.Text) * Pi / 180
    For X = 0 To PicSource.ScaleWidth - 1
        Xc = X - PicSource.ScaleWidth / 2
        For Y = 0 To PicSource.ScaleHeight - 1
            Yc = Y - PicSource.ScaleHeight / 2
            X0 = Xc * Cos(-K) + Yc * Sin(-K)
            Y0 = Yc * Cos(-K) - Xc * Sin(-K)
            Xn = X0 + PicTarget.ScaleWidth / 2
            Yn = Y0 + PicTarget.ScaleHeight / 2
            PicTarget.PSet (Xn, Yn), PicSource.Point(X, Y)
        Next Y
    Next X
End Sub

Private Sub Form_Load()
    PicSource.Picture = LoadPicture("d:\vbsouce\pic1.bmp")
    TxtValue.Text = "0"
    LblValue.Caption = "输入旋转角度"
    CmdOk.Caption = "确定"
    CmdExit.Caption = "退出"
    PicSource.ScaleMode = vbPixels
    PicTarget.ScaleMode = vbPixels
End Sub

Private Sub CmdExit_Click()
```

```
End  
End Sub
```

1.1.2 使用 API 函数的旋转方法

1. 用 API 函数方法的思路

此种方法与 1.1.1 节逐点计算的思路基本相似，在将点按一定的角度进行旋转时，用 API 函数 GetPixel 函数获得某一点的颜色信息，经过计算后用 API 函数 SetPixel 重新设置一点，保持与原来点信息的一致。

2. 原理与步骤

与 1.1.1 节原理与方法基本类似。

3. 具体实现算法

假设坐标原点为图像的中心，在图像框中有一源点，其坐标为 (P2x, P2y)，图像要旋转的角度为 Theta，计算公式为：

$$R = \text{sqr}(1\& * P2x * P2x + 1\& * P2y * P2y)$$

$$P1x = R * \cos(A + \text{Theta})$$

$$P1y = R * \sin(A + \text{Theta})$$

其中坐标 (P1x, P1y) 为旋转计算后的目标点坐标，R 为半径。

4. 运行和设计时屏幕

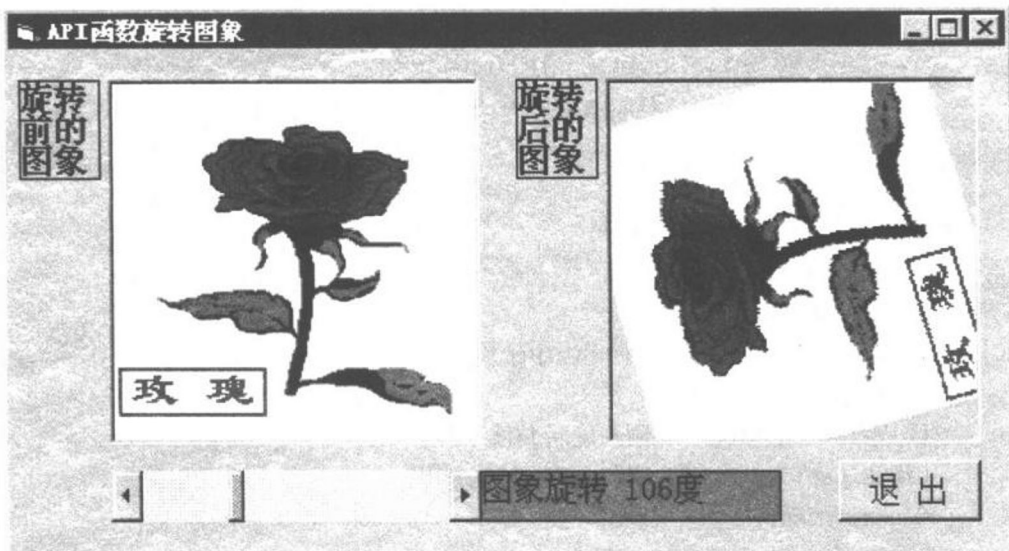


图 1.3 用 API 函数旋转图像的运行屏幕

图 1.3 所示为采用上述方法编写的实例运行屏幕。在屏幕中有两个两个图像控件 PictureBox, 名称分别为 PicSource 和 PicTarget, PicSource 为源图像, PicTarget 为经

过旋转后的目标图像，当改变水平滚动条控件 HsbDegree 的值时，即改变了对左边 PictureBox 控件 Picture1 中图像的旋转角度，角度的值在 Label 控件 LblDegree 中显示，旋转后的图像在右边的 PictureBox 控件 PicTarget 中显示出来，图中所示图像逆时针旋转了 106° 。

窗体设计时的屏幕，如图 1.4 所示。

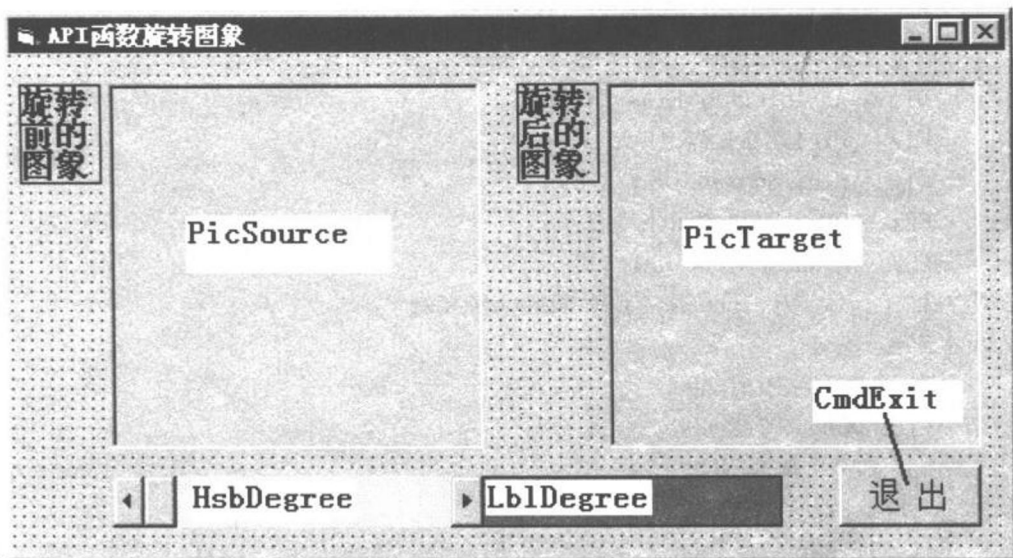


图 1.4 用 API 函数旋转图像实例设计时的屏幕

E、完整程序代码清单

实现上述功能的完整程序代码如下：

```
Const SRCCOPY = &HCC0020
Const Pi = 3.14159265359
Private Declare Function SetPixel Lib "GDI32" (ByVal hDC As Integer, ByVal
X As Integer, ByVal Y As Integer, ByVal crColor As Long) As Long
Private Declare Function GetPixel Lib "GDI32" (ByVal hDC As Integer, ByVal
X As Integer, ByVal Y As Integer) As Long
Sub Form_Load()
    PicSource.Picture = LoadPicture("d:\vbsouce\Pic2.bmp")
    PicSource.ScaleMode = 3
    PicTarget.ScaleMode = 3
    HsbDegree.Max = 360
    HsbDegree.Min = 0
    HsbDegree.LargeChange = 5
    HsbDegree.SmallChange = 1
    LblDegree.Caption = "图像旋转 0 度"
End Sub
```

```

Sub BmpRotate(Pic1 As PictureBox, Pic2 As PictureBox, ByVal Theta)
    Dim C1x As Integer, C1y As Integer
    Dim C2x As Integer, C2y As Integer
    Dim A As Single
    Dim P1x As Integer, P1y As Integer
    Dim P2x As Integer, P2y As Integer
    Dim N As Integer, R As Integer
    C1x = Pic1.ScaleWidth \ 2
    C1y = Pic1.ScaleHeight \ 2
    C2x = Pic2.ScaleWidth \ 2
    C2y = Pic2.ScaleHeight \ 2
    If C2x < C2y Then N = C2y Else N = C2x
    N = N - 1
    Pic1hDC% = Pic1.hDC
    Pic2hDC% = Pic2.hDC
    For P2x = 0 To N
        For P2y = 0 To N
            If P2x = 0 Then A = Pi / 2 Else A = Atn(P2y / P2x)
            R = Sqr(1& * P2x * P2x + 1& * P2y * P2y)
            P1x = R * Cos(A + Theta)
            P1y = R * Sin(A + Theta)
            C0& = GetPixel(Pic1hDC%, C1x + P1x, C1y + P1y)
            C1& = GetPixel(Pic1hDC%, C1x - P1x, C1y - P1y)
            C2& = GetPixel(Pic1hDC%, C1x + P1y, C1y - P1x)
            C3& = GetPixel(Pic1hDC%, C1x - P1y, C1y + P1x)
            If C0&<>-1 Then Xret&=SetPixel(Pic2hDC%, C2x+P2x, C2y+P2y, C0&)
            If C1&<>-1 Then Xret&=SetPixel(Pic2hDC%, C2x-P2x, C2y-P2y, C1&)
            If C2&<>-1 Then Xret&=SetPixel(Pic2hDC%, C2x+P2y, C2y-P2x, C2&)
            If C3&<>-1 Then Xret&=SetPixel(Pic2hDC%, C2x-P2y, C2y+P2x, C3&)
        Next
        t% = DoEvents()
    Next
End Sub

Private Sub HsbDegree_Change()
    PicTarget.Cls
    Call BmpRotate(PicSource, PicTarget, 2 * 3.14 * HsbDegree.Value / 360)
    LblDegree.Caption = "图像旋转" & Str(HsbDegree.Value) & "度"

```

```
End Sub

Private Sub CmdExit_Click()
    End
End Sub
```

1.2 图像变换效果

图像的变换效果是指当图像在加载和卸载时进行的一些图形“特技”效果，如模拟百叶窗翻转过程的“百叶窗”效果等。

1.2.1 “百叶窗”效果

1. 功能描述

“百叶窗”效果主要是模拟百叶窗翻转过程，从而实现图像的效果。

2. 实现思路

制作时先在窗体中绘制两个图像框，将图像载入一个图像框，同时将另一个图像框设为空白，然后调用 BitBlt 函数将第一个图像框中的图形一部分一部分地复制到第二个图像框中，这样就可以实现“百叶窗”的图形特效。

3. 实现方法与步骤

(1) 新建一窗体，加入两个图像框 PicSource 和 PicDest，两个命令按钮 cmdStart 和 cmdExit；

(2) 在载入窗体同时，在 PicSource 图像框中载入图片；

(3) 在窗体中加入命令按钮 CmdStart，在 CmdStart_Click 事件中调用 BitBlt 函数和 Sleep 函数，Sleep 函数用于延时。

4. 代码说明

```
H% = PicSource.ScaleHeight '图像的高度
W% = PicSource.ScaleWidth  '图像的宽度
ChangWidth = 14            '设置百叶窗叶片高度
For i = 0 To (ChangWidth - 1)
    For j = i To H% Step ChangWidth
        s% = BitBlt(PicDest.hDC, 0, j, W%, 1, PicSource.hDC, 0, j, COPY_PUT)
        '调用 API 函数 BitBlt 实现图像变换效果
        Sleep (100) '延时
```