

## نکاتی در مورد استفاده از فرم ها

- ۱- On Top کردن فرم
- ۲- چگونه یک فرم را شفاف (Transparent) کنیم؟
- ۳- چگونه شکل فرم ها را از حالت مستطیلی خارج کنیم؟
- ۴- چگونه فرم برنامه را پوسته (Skin) دار کنیم؟
- ۵- چند جلوه ویژه برای نمایش فرم ها
- ۶- چگونه یک فرم را در وسط صفحه نمایش قرار دهیم؟
- ۷- چگونه یک فرم بدون عنوان (Caption) را با موشواره (Mouse) حرکت دهیم؟
- ۸- چگونه یک فرم یا آیکون مشخص از برنامه ای را مخفی کنیم؟
- ۹- چگونه مکان قرارگیری فرم برنامه ای را که با دستور shell اجرا شده تعیین کنیم؟
- ۱۰- مخفی کردن دکمه فرم برنامه در نوار وظیفه (TaskBar) ویندوز
- ۱۱- چگونه یک فرم را در وسط فرم دیگری نمایش دهیم؟
- ۱۲- چگونه عنوان فرم برنامه فعال را بدست آوریم؟
- ۱۳- چگونه می توان متغیری را به یک فرم ، بدون استفاده از متغیرهای عمومی فرستاد؟
- ۱۴- چگونه فرم دیگر برنامه را از انجام عملیاتی در فرم جاری با خبر سازیم؟
- ۱۵- چگونه مکان و اندازه یک فرم را غیرقابل تغییر نماییم؟
- ۱۶- دستکاری نوار عنوان (TITLE BAR)
- ۱۷- پویانمایی (Animation) آیکون (شکلک!) فرم

## ۱- On Top کردن فرم

فرم On Top فرمی است که همواره بالاتر از تمام فرم‌ها (روی تمام فرم‌ها) در صفحه نمایش ظاهر می‌شود.

مثال:

اگر کارت گرافیکی شما خروجی به تلویزیون ویا ویدئو داشته باشد می‌توانید فرمی On Top درست کنید و در هنگام نمایش فیلمی آنرا روی فیلم در حال نمایش بیندازید و سپس مجموع تصاویر در حال پخش را از طریق ویدئو ضبط کنید. با اینکار شما از داشتن کارت های گرافیکی گران قیمت برای انداختن یک تصویر ویا یک نوشته روی چند دقیقه از فیلم معاف خواهید شد.

Option Explicit

```
Public Declare Function SetWindowPos Lib "user32" _
    (ByVal hwnd As Long, _
    ByVal hWndInsertAfter As Long, _
    ByVal x As Long, ByVal y As Long, _
    ByVal cx As Long, ByVal cy As Long, _
    ByVal wFlags As Long) As Long
```

```
Global Const SWP_NOMOVE = 2
Global Const SWP_NOSIZE = 1
Global Const FLAGS = SWP_NOMOVE Or SWP_NOSIZE
Global Const HWND_TOPMOST = -1
Global Const HWND_NOTOPMOST = -2
```

```
Sub setTopMost(frm As Form, ByVal enabled As Boolean)
Dim res, FLAGS
FLAGS = SWP_NOMOVE Or SWP_NOSIZE
If enabled = True Then
    res = SetWindowPos(frm.hwnd, HWND_TOPMOST, _
        0, 0, 0, 0, FLAGS)
    'if res=0, there is an error
Else
    'To turn off topmost (make the form act normal again):
    res = SetWindowPos(frm.hwnd, HWND_NOTOPMOST, _
        0, 0, 0, 0, FLAGS)
End If
End Sub
```

مثال:

```
Private Sub Form_Load()
    setTopMost Me, True
End Sub
```

## ۲- چگونه یک فرم را شفاف (Transparent) کنیم؟

برای اینکه بتوانید این جلوه را ببینید چند کنترل را روی فرم قرار دهید و برنامه را اجرا کنید.

```

Private Declare Function CreateRectRgn Lib "gdi32" _
    (ByVal X1 As Long, ByVal Y1 As Long, _
    ByVal X2 As Long, ByVal Y2 As Long) As Long

Private Declare Function CombineRgn Lib "gdi32" _
    (ByVal hDestRgn As Long, ByVal hSrcRgn1 As Long, _
    ByVal hSrcRgn2 As Long, ByVal nCombineMode As Long) As Long

Private Declare Function SetWindowRgn Lib "user32" _
    (ByVal hWnd As Long, ByVal hRgn As Long, _
    ByVal bRedraw As Long) As Long

Public Sub TransparentForm(frm As Form)
    frm.ScaleMode = vbPixels
    Const RGN_DIFF = 4
    Const RGN_OR = 2

    Dim outer_rgn As Long
    Dim inner_rgn As Long
    Dim wid As Single
    Dim hgt As Single
    Dim border_width As Single
    Dim title_height As Single
    Dim ctl_left As Single
    Dim ctl_top As Single
    Dim ctl_right As Single
    Dim ctl_bottom As Single
    Dim control_rgn As Long
    Dim combined_rgn As Long
    Dim ctl As Control

    If frm.WindowState = vbMinimized Then Exit Sub

    ' Create the main form region.
    wid = frm.ScaleX(frm.Width, vbTwips, vbPixels)
    hgt = frm.ScaleY(frm.Height, vbTwips, vbPixels)
    outer_rgn = CreateRectRgn(0, 0, wid, hgt)

    border_width = (wid - frm.ScaleWidth) / 2
    title_height = hgt - border_width - frm.ScaleHeight
    inner_rgn = CreateRectRgn(border_width, title_height, _
        wid - border_width, hgt - border_width)

    ' Subtract the inner region from the outer.
    combined_rgn = CreateRectRgn(0, 0, 0, 0)
    CombineRgn combined_rgn, outer_rgn, inner_rgn, RGN_DIFF

    ' Create the control regions.
    For Each ctl In frm.Controls
        If ctl.Container Is frm Then
            ctl_left = frm.ScaleX(ctl.Left, frm.ScaleMode, vbPixels) _
                + border_width
            ctl_top = frm.ScaleX(ctl.Top, frm.ScaleMode, vbPixels) _
                + title_height
            ctl_right = frm.ScaleX(ctl.Width, frm.ScaleMode, vbPixels) _
                + ctl_left
            ctl_bottom = frm.ScaleX(ctl.Height, frm.ScaleMode, _
                vbPixels) + ctl_top
            control_rgn = CreateRectRgn(ctl_left, ctl_top, ctl_right, _

```

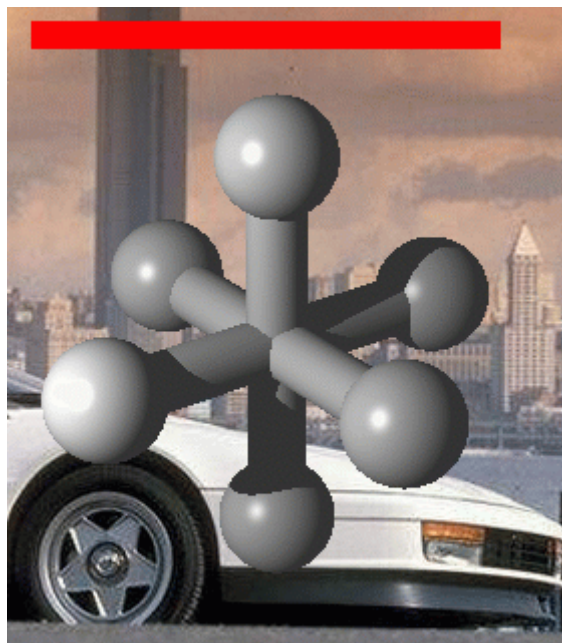
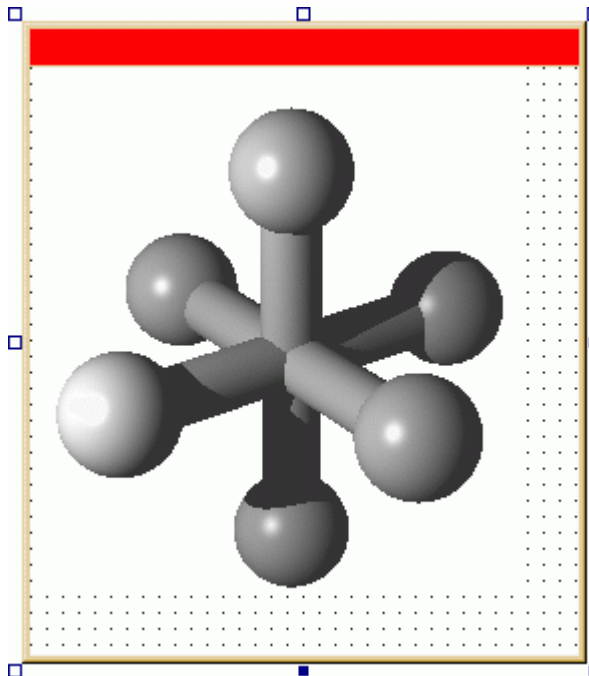
```
        ctl_bottom)
CombineRgn combined_rgn, combined_rgn, control_rgn, RGN_OR
    End If
Next ctl

'Restrict the window to the region.
SetWindowRgn frm.hWnd, combined_rgn, True
End Sub
```

مثال :

```
Private Sub Form_Resize()
    TransparentForm Me
End Sub
```

نوعی دیگر :



Option Explicit

Private Type BITMAP

    bmType As Long  
    bmWidth As Long  
    bmHeight As Long  
    bmWidthBytes As Long  
    bmPlanes As Integer  
    bmBitsPixel As Integer  
    bmBits As Long

End Type

Private Declare Function GetBitmapBits Lib "gdi32" ( \_  
    ByVal hBitmap As Long, ByVal dwCount As Long, lpBits As Any) As Long

Private Declare Function GetObject Lib "gdi32" Alias "GetObjectA" \_  
    (ByVal hObject As Long, ByVal nCount As Long, lpObject As Any) \_  
    As Long

Private Declare Function CreateRectRgn Lib "gdi32" ( \_  
    ByVal x1 As Long, \_  
    ByVal y1 As Long, ByVal x2 As Long, ByVal y2 As Long) As Long

Private Declare Function CombineRgn Lib "gdi32" ( \_  
    ByVal hDestRgn As Long, ByVal hSrcRgn1 As Long, \_

    ByVal hSrcRgn2 As Long, ByVal nCombineMode As Long) As Long

Private Declare Function SetWindowRgn Lib "user32" ( \_  
    ByVal hWnd As Long, ByVal hRgn As Long, ByVal bRedraw As Long) \_  
    As Long

Private Declare Function DeleteObject Lib "gdi32" ( \_  
    ByVal hObject As Long) As Long

' Restrict the form to its non-white pixels.

Private Sub FitToPicture()

Const RGN\_OR = 2

Dim border\_width As Single

Dim title\_height As Single

Dim bm As BITMAP

Dim bytes() As Byte

Dim R As Integer

Dim C As Integer

Dim start\_c As Integer

Dim stop\_c As Integer

Dim x0 As Long

Dim y0 As Long

Dim combined\_rgn As Long

Dim new\_rgn As Long

Dim offset As Integer

Dim colourDepth As Integer

ScaleMode = vbPixels

picShape.ScaleMode = vbPixels

picShape.AutoRedraw = True

picShape.Picture = picShape.Image

' Find the form's corner.

border\_width = (ScaleX(Width, vbTwips, vbPixels) \_  
    - ScaleWidth) / 2

title\_height = ScaleX(Height, vbTwips, vbPixels) \_  
    - border\_width - ScaleHeight

```

' Find the picture's corner.
x0 = picShape.Left + border_width
y0 = picShape.Top + title_height

' Get the bitmap information.
GetObject picShape.Image, Len(bm), bm
Select Case bm.bmBitsPixel
Case 24:
    colourDepth = 3
Case 32:
    colourDepth = 4
Case Else
MsgBox "Sorry, this program only works for 24-bit or 32-bit color." _
    & vbCrLf & "For information on color modes, "& _
    "go to www.vb-helper.com/highclr.htm"
Exit Sub
End Select

' Allocate space for the bitmap data.
ReDim bytes(0 To bm.bmWidthBytes - 1, 0 To bm.bmHeight - 1)

' Get the bitmap data.
GetBitmapBits picShape.Image, _
    bm.bmHeight * bm.bmWidthBytes, bytes(0, 0)

' Create the form's regions.
For R = 0 To bm.bmHeight - 2
    ' Create a region for this row.
    C = 0
    Do While C < bm.bmWidth
        start_c = 0
        stop_c = 0

        ' Find the next non-white column.

        ' calculate the initial offset
        offset = C * colourDepth

        Do While C < bm.bmWidth
            If bytes(offset, R) <> 255 Or _
                bytes(offset + 1, R) <> 255 Or _
                bytes(offset + 2, R) <> 255 Then Exit Do

            C = C + 1
            offset = offset + colourDepth
        Loop

        start_c = C

        ' Find the next white column.
        ' Note the offset will be set correctly
        Do While C < bm.bmWidth
            If bytes(offset, R) = 255 And _
                bytes(offset + 1, R) = 255 And _
                bytes(offset + 2, R) = 255 _
                Then Exit Do
            C = C + 1
            offset = offset + colourDepth
        Loop
        stop_c = C
    
```

```

' Make a region from start_c to stop_c.
If start_c < bm.bmWidth Then
    If stop_c >= bm.bmWidth Then stop_c = bm.bmWidth - 1

    ' Create the region.
    new_rgn = CreateRectRgn( _
        start_c + x0, R + y0, _
        stop_c + x0, R + y0 + 1)

    ' Add it to what we have so far.
    If combined_rgn = 0 Then
        combined_rgn = new_rgn
    Else
        CombineRgn combined_rgn, _
            combined_rgn, new_rgn, RGN_OR
        DeleteObject new_rgn
    End If
End If
Loop
Next R

' Restrict the form to the region.
SetWindowRgn hWnd, combined_rgn, True
DeleteObject combined_rgn
End Sub
Private Sub picShape_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    ' Center the form.
    Move (Screen.Width - Width) / 2, (Screen.Height - Height) / 2

    FitToPicture
End Sub

```

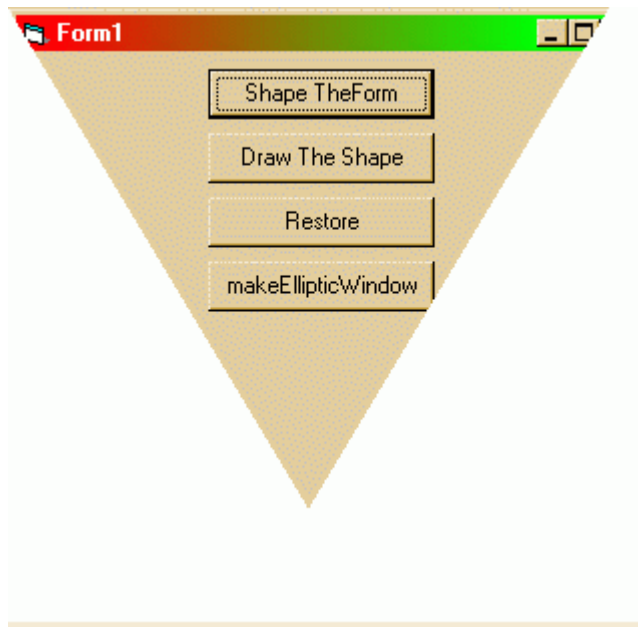
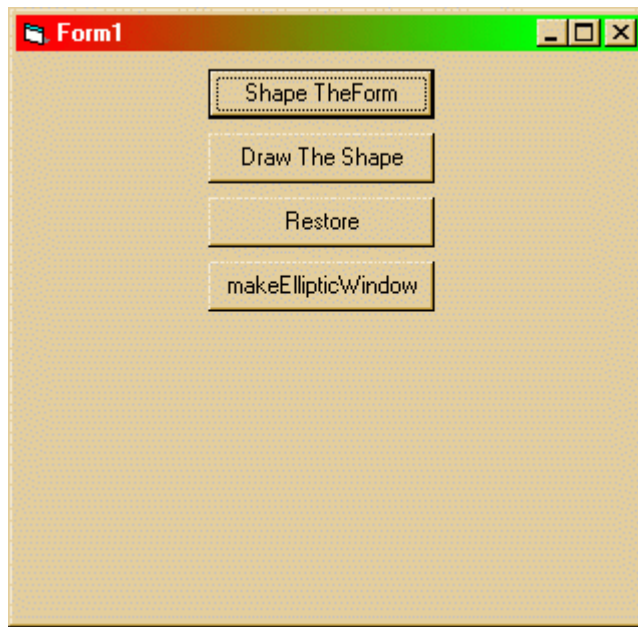
۳- چگونه شکل فرم ها را از حالت مستطیلی خارج کنیم؟

۴- چگونه فرم برنامه را پوسته (skin) دار کنیم؟

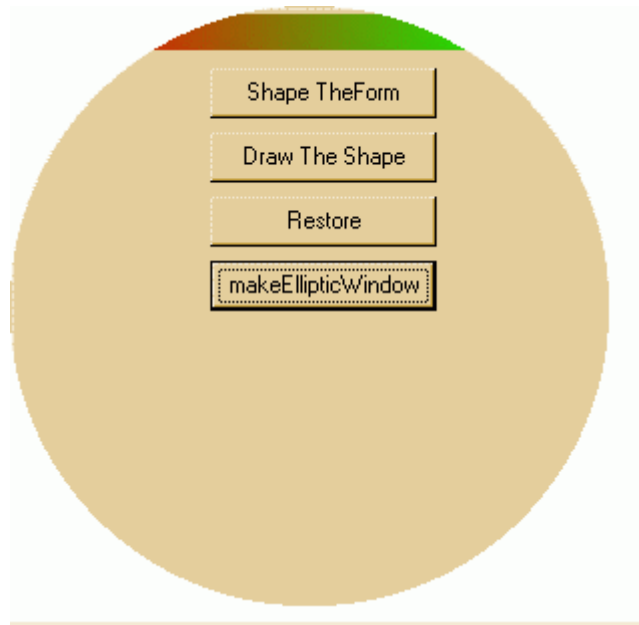
تابع `SetWindowRgn` ایجاد فرم های غیرمستطیلی را ممکن می سازد. باید توجه داشت که نقاط اول و آخر چند ضلعی مورد نظر برای تغییر شکل فرم یکی می باشند (چند ضلعی باید به این صورت بسته شود). توابع زیر را برای کنترل های روی فرم نیز می توان بکار برد. `Points(0).X` و `Points(0).Y` در مثال زیر مختصات نقطه اول چند ضلعی می باشند. اگر نقاطی را به نقاط برنامه زیر افزودید به روز درآوردن عبارت زیر را فراموش نکنید:

```
Dim Points(0 To 3) As POINTAPI
```

مثال زیر از `shapePoints(0)` بعنوان آرگومان ورودی تابع استفاده می کند. همانطور که می دانید برای فرستادن آرایه ها بعنوان آرگومان به توابع زبان C باید آدرس اولین عضو آن آرایه را مورد استفاده قرار داد.







Option Explicit

```
Declare Function CreatePolygonRgn Lib "gdi32" _
    (lpPoint As POINTAPI, ByVal nCount As Long, _
    ByVal nPolyFillMode As Long) As Long
```

```
Declare Function SetWindowRgn Lib "user32" _
    (ByVal hWnd As Long, ByVal hRgn As Long, _
    ByVal bRedraw As Boolean) As Long
```

```
Declare Function Polyline Lib "gdi32" _
    (ByVal hdc As Long, lpPoint As POINTAPI, _
    ByVal nCount As Long) As Long
```

```
Public Type POINTAPI
    X As Long
    Y As Long
End Type
```

```
Declare Function CreateEllipticRgn Lib "gdi32" _
    (ByVal X1 As Long, ByVal Y1 As Long, _
    ByVal X2 As Long, ByVal Y2 As Long) As Long
```

```
Dim Result As Long
```

```
Public Sub ShapeTheControlOrForm(shapePoints() _
    As POINTAPI, ctrlToShape As Object)
    Dim n, hRgn
    n = UBound(shapePoints()) + 1
    hRgn = CreatePolygonRgn(shapePoints(0), n, 1)
    Result = SetWindowRgn(ctrlToShape.hWnd, hRgn, True)
End Sub
```

```
Sub drawTheShape(shapePoints() As POINTAPI, frm As Form)
    Dim n
    n = UBound(shapePoints()) + 1
    Call Polyline(frm.hdc, shapePoints(0), n)
End Sub
```

```

Public Sub restoreTheShape(obj As Object)
    Result = SetWindowRgn(obj.hWnd, 0, True)
End Sub

Public Sub makeEllipticWindow(hWnd As Long, X1 As Long, _
    Y1 As Long, X2 As Long, Y2 As Long)
    Dim lRetVal As Long, lRgn As Long
    'Create an elliptic region
    lRgn = CreateEllipticRgn(X1, Y1, X2, Y2)
    'Make the window look like the region we created
    lRetVal = SetWindowRgn(hWnd, lRgn, True)
End Sub

```

مثال :

یک پروژۀ جدید باز کنید و چهار دکمه مطابق شکل های فوق روی آن قرار دهید و سپس کد زیر را وارد کنید:

```

Private Points(0 To 3) As POINTAPI

Private Sub Command1_Click()
    Call ShapeTheControlOrForm(Points, Me)
End Sub

Private Sub Command2_Click()
    Call drawTheShape(Points, Me)
End Sub

Private Sub Command3_Click()
    restoreTheShape Me
End Sub

Private Sub Command4_Click()
    makeEllipticWindow Me.hWnd, 0, 0, 300, 300
End Sub

Private Sub Form_Load()
    Points(0).X = 150
    Points(0).Y = 250
    Points(1).X = 0
    Points(1).Y = 0
    Points(2).X = 300
    Points(2).Y = 0
    Points(3).X = 150
    Points(3).Y = 250
End Sub

```

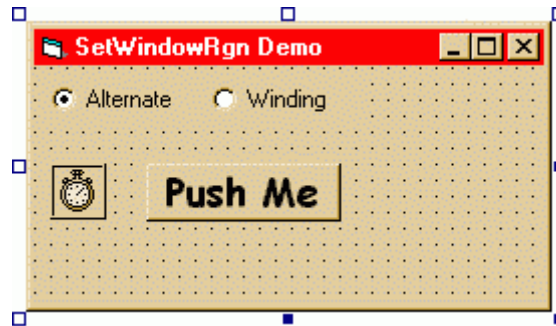
مثال :

```

'Polygon region - This draws a points
Points(0).X = 231:    Points(0).Y = 12
Points(1).X = 220:    Points(1).Y = 57
Points(2).X = 259:    Points(2).Y = 31
Points(3).X = 209:    Points(3).Y = 31
Points(4).X = 245:    Points(4).Y = 57
Points(5).X = 231:    Points(5).Y = 12

```

نوعی دیگر:



Option Explicit

```
Private Declare Function CreatePolygonRgn Lib "gdi32" ( _
    lpPoint As POINTAPI, ByVal nCount As Long, _
    ByVal nPolyFillMode As Long) As Long
Private Declare Function SetWindowRgn Lib "user32" ( _
    ByVal hWnd As Long, ByVal hRgn As Long, _
    ByVal bRedraw As Boolean) As Long
Private Declare Function CreateSolidBrush Lib "gdi32" ( _
    ByVal crColor As Long) As Long
Private Declare Function DeleteObject Lib "gdi32" ( _
    ByVal hObject As Long) As Long
Private Declare Function FillRgn Lib "gdi32" (ByVal hDC As Long, _
    ByVal hRgn As Long, ByVal hBrush As Long) As Long
Private Declare Function GetSystemMetrics Lib "user32" ( _
    ByVal nIndex As Long) As Long
Private Declare Function Polyline Lib "gdi32" (ByVal hDC As Long, _
    lpPoint As POINTAPI, ByVal nCount As Long) As Long

Private Declare Function SendMessage Lib "user32" Alias _
    "SendMessageA" (ByVal hWnd As Long, ByVal wParam As Long, _
    ByVal lParam As Long, lParam As Any) As Long
Private Declare Function ReleaseCapture Lib "user32" () As Long
Private Declare Function GetCursorPos Lib "user32" ( _
    lpPoint As POINTAPI) As Long

Private Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type

Private Type POINTAPI
    X As Long
    Y As Long
End Type

Private scnPts() As POINTAPI
Private rgnPts() As POINTAPI

Private Const SM_CYCAPTION = 4
Private Const SM_CXFRAME = 32
Private Const SM_CYFRAME = 33
```

```

' PolyFill() Modes
Private Const ALTERNATE = 1
Private Const WINDING = 2

' Used to support captionless drag
Private Const WM_NCLBUTTONDOWN = &HA1
Private Const HTCAPTION = 2
' Undocumented message constant.
Private Const WM_GETSYSTEMMENU = &H313

Private m_FillMode As Long
Private Const nPts& = 36

Private Sub Command1_Click()
    Dim hRgn As Long
    Static UsingPoly As Boolean
    '
    ' Flag variable tracks current state.
    '
    UsingPoly = Not UsingPoly
    If UsingPoly Then
        '
        ' Create a region, then turn on
        ' clipping to that region.
        '
        hRgn = CreatePolygonRgn(rgnPts(0), nPts, m_FillMode)
        Call SetWindowRgn(Me.hWnd, hRgn, True)
    Else
        '
        ' Turn off clipping.
        '
        Call SetWindowRgn(Me.hWnd, 0&, True)
    End If

    Timer1.Enabled = UsingPoly
End Sub

Private Sub Form_Load()
    m_FillMode = ALTERNATE
    With Me
        .ScaleMode = vbPixels
        .Width = Screen.Width \ 2
        .Height = .Width
        .Move (Screen.Width - .Width) \ 2, _
            (Screen.Height - .Height) \ 2
        .Icon = Nothing
    End With
End Sub

Private Sub Form_MouseDown(Button As Integer, Shift As Integer, _
    X As Single, Y As Single)
    '
    ' Allow captionless drag if form is clipped to region
    '
    If Button = vbLeftButton Then
        If Timer1.Enabled Then
            Call ReleaseCapture
            Call SendMessage(Me.hWnd, WM_NCLBUTTONDOWN, HTCAPTION, ByVal 0&)
        End If
    End If
End Sub
End Sub

```

```

Private Sub Form_MouseUp(Button As Integer, Shift As Integer, _
    X As Single, Y As Single)
    Dim pt As POINTAPI

    ' This is relative to the screen, so we can't
    ' use the coordinates passed in the event
    Call GetCursorPos(pt)

    If Button = vbRightButton Then
        If Timer1.Enabled Then
            Call SendMessage(Me.hWnd, WM_GETSYSTEMMENU, 0, _
                ByVal MakeLong(pt.Y, pt.X))
        End If
    End If
End Sub

Private Sub Form_Paint()
    Dim hBrush As Long
    Dim hRgn As Long
    '
    ' Create region and a brush to fill it with.
    '
    hBrush = CreateSolidBrush(vbRed)
    hRgn = CreatePolygonRgn(scnPts(0), nPts, m_FillMode)
    Call FillRgn(Me.hDC, hRgn, hBrush)
    '
    ' Clean up GDI objects.
    '
    Call DeleteObject(hRgn)
    Call DeleteObject(hBrush)
    '
    ' Draw outline around polygon.
    '
    Call Polyline(Me.hDC, scnPts(0), nPts + 1)
End Sub

Private Sub Form_Resize()
    With Me
        Command1.Move (.ScaleWidth - Command1.Width) \ 2, _
            (.ScaleHeight - Command1.Height) \ 2
        If .Visible Then
            CalcRgnPoints
            .Refresh
        End If
    End With
End Sub

Private Static Sub CalcRgnPoints()
    ReDim scnPts(0 To nPts) As POINTAPI
    ReDim rgnPts(0 To nPts) As POINTAPI
    Dim offset As Long
    Dim angle As Long
    Dim theta As Double
    Dim radius1 As Long
    Dim radius2 As Long
    Dim x1 As Long
    Dim y1 As Long
    Dim xOff As Long
    Dim yOff As Long
    Dim n As Long

```

```

'
' Some useful constants.
'
Const Pi# = 3.14159265358979
Const DegToRad# = Pi / 180
'
' Calc radius based on form size.
'
x1 = Me.ScaleWidth \ 2
y1 = Me.ScaleHeight \ 2
If x1 > y1 Then
    radius1 = y1 * 0.85
Else
    radius1 = x1 * 0.85
End If
radius2 = radius1 * 0.5
'
' Offsets to move origin to upper
' left of window.
'
xOff = GetSystemMetrics(SM_CXFRAME)
yOff = GetSystemMetrics(SM_CYFRAME) + _
        GetSystemMetrics(SM_CYCAPTION)
'
' Step through a circle, 10 degrees each
' loop, finding points for polygon.
'
n = 0
For angle = 0 To 360 Step 10
    theta = (angle - offset) * DegToRad
    '
    ' First region is for drawing.
    ' One long, one short, one long...
    '
    If n Mod 2 Then
        scnPts(n).X = x1 + (radius1 * (Sin(theta)))
        scnPts(n).Y = y1 + (radius1 * (Cos(theta)))
    Else
        scnPts(n).X = x1 + (radius2 * (Sin(theta)))
        scnPts(n).Y = y1 + (radius2 * (Cos(theta)))
    End If
    '
    ' Second region is for clipping.
    ' Add offsets.
    '
    rgnPts(n).X = scnPts(n).X + xOff
    rgnPts(n).Y = scnPts(n).Y + yOff
    n = n + 1
Next angle

offset = (offset + 2) Mod 360
End Sub

Private Sub Option1_Click(Index As Integer)
    m_FillMode = Index + 1
End Sub

Private Static Sub Timer1_Timer()
    Dim nRet As Long
    Dim hRgn As Long

```

```

CalcRgnPoints
hRgn = CreatePolygonRgn(rgnPts(0), nPts, m_FillMode)
nRet = SetWindowRgn(Me.hWnd, hRgn, True)
End Sub

Public Function MakeLong(ByVal WordHi As Variant, _
    ByVal WordLo As Integer) As Long
    '
    ' High word is coerced to a variant on call to allow
    ' it to overflow limits of multiplication which shifts
    ' it left.
    '
    MakeLong = (WordHi * &H10000) + (WordLo And &HFFFF&)
End Function

```

## ۵- چند جلوه ویژه برای نمایش فرم ها

منفجر شدن فرم !:

```

'Declarations

#If Win16 Then
    Type RECT
        Left As Integer
        Top As Integer
        Right As Integer
        Bottom As Integer
    End Type
#Else
    Type RECT
        Left As Long
        Top As Long
        Right As Long
        Bottom As Long
    End Type
#End If

'User and GDI Functions for Explode/Implode to work

#If Win16 Then
    Declare Sub GetWindowRect Lib "User" (ByVal hwnd As Integer, _
        lpRect As RECT)
    Declare Function GetDC Lib "User" (ByVal hwnd As Integer) As _
        Integer
    Declare Function ReleaseDC Lib "User" (ByVal hwnd As Integer, _
        ByVal hdc As Integer) As Integer
    Declare Sub SetBkColor Lib "GDI" (ByVal hdc As Integer, ByVal _
        crColor As Long)
    Declare Sub Rectangle Lib "GDI" (ByVal hdc As Integer, ByVal X1 _
    As Integer, ByVal Y1 As Integer, ByVal X2 As Integer, ByVal Y2 As _
    Integer)
    Declare Function CreateSolidBrush Lib "GDI" (ByVal crColor As _
    Long) As Integer
    Declare Function SelectObject Lib "GDI" (ByVal hdc As Integer, _
        ByVal hObject As Integer) As Integer
    Declare Sub DeleteObject Lib "GDI" (ByVal hObject As Integer)
#Else
    Declare Function GetWindowRect Lib "user32" (ByVal hwnd As Long, _

```

```

        lpRect As RECT) As Long
Declare Function GetDC Lib "user32" (ByVal hwnd As Long) As Long
Declare Function ReleaseDC Lib "user32" (ByVal hwnd As Long, _
        ByVal hdc As Long) As Long
Declare Function SetBkColor Lib "gdi32" (ByVal hdc As Long, _
        ByVal crColor As Long) As Long
Declare Function Rectangle Lib "gdi32" (ByVal hdc As Long, _
        ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, _
        ByVal Y2 As Long) As Long
Declare Function CreateSolidBrush Lib "gdi32" (ByVal crColor _
        As Long) As Long
Declare Function SelectObject Lib "user32" (ByVal hdc As Long, _
        ByVal hObject As Long) As Long
Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long _
        ) As Long

#End If

'*****
'*Description:
'*The higher the "Movement", the slower the window explosion.
'*****

Sub ExplodeForm(f As Form, Movement As Integer)
    Dim myRect As RECT
    Dim formWidth%, formHeight%, i%, X%, Y%, Cx%, Cy%
    Dim TheScreen As Long
    Dim Brush As Long

    GetWindowRect f.hwnd, myRect
    formWidth = (myRect.Right - myRect.Left)
    formHeight = myRect.Bottom - myRect.Top
    TheScreen = GetDC(0)
    Brush = CreateSolidBrush(f.BackColor)

    For i = 1 To Movement
        Cx = formWidth * (i / Movement)
        Cy = formHeight * (i / Movement)
        X = myRect.Left + (formWidth - Cx) / 2
        Y = myRect.Top + (formHeight - Cy) / 2
        Rectangle TheScreen, X, Y, X + Cx, Y + Cy
    Next i

    X = ReleaseDC(0, TheScreen)
    DeleteObject (Brush)

End Sub

Public Sub ImplodeForm(f As Form, Direction As Integer, _
        Movement As Integer, ModalState As Integer)
'*****
'*Description:
'*The larger the "Movement" value, the slower the "Implosion"
'*****

    Dim myRect As RECT
    Dim formWidth%, formHeight%, i%, X%, Y%, Cx%, Cy%
    Dim TheScreen As Long
    Dim Brush As Long

    GetWindowRect f.hwnd, myRect

```



```

formWidth = (myRect.Right - myRect.Left)
formHeight = myRect.Bottom - myRect.Top
TheScreen = GetDC(0)
Brush = CreateSolidBrush(f.BackColor)

    For i = Movement To 1 Step -1
    Cx = formWidth * (i / Movement)
    Cy = formHeight * (i / Movement)
    X = myRect.Left + (formWidth - Cx) / 2
    Y = myRect.Top + (formHeight - Cy) / 2
    Rectangle TheScreen, X, Y, X + Cx, Y + Cy
    Next i

X = ReleaseDC(0, TheScreen)
DeleteObject (Brush)

End Sub

```

مثال: یک پروژه جدید باز کنید و دکمه‌ای روی آن قرار دهید و...

```

Private Sub Command1_Click()
    Call ImplodeForm(Me, 2, 500, 1)
    End
    Set Form1 = Nothing
End Sub

Private Sub Form_Load()
    Call ExplodeForm(Me, 500)
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, _
                               UnloadMode As Integer)
    Call ImplodeForm(Me, 2, 500, 1)
End Sub

```

نوعی دیگر:

```

Public Sub Form_Animation_Opener( _
    your_form As Form, _
    ByVal loop_step As Integer, _
    ByVal how_open As Integer _
)

Dim f_h, f_w, i, j

'your_form.ScaleMode = 3
f_w = your_form.Width
f_h = your_form.Height

Select Case how_open

    Case Is = 1
        For i = 0 To f_h Step loop_step
            your_form.Height = i
            your_form.Show
        Next i

    Case Is = 2
        For i = 0 To f_w Step loop_step
            your_form.Width = i

```

```

        your_form.Show
    Next i

Case Is = 3

    your_form.Height = 0
    For i = 0 To f_w Step loop_step
        your_form.Width = i
        your_form.Show
        'DoEvents
    Next i
    For i = 0 To f_h Step loop_step
        your_form.Height = i
        your_form.Show
        'DoEvents
    Next i

Case Is = 4

    For i = 0 To f_w Step loop_step
        your_form.Height = j
        your_form.Width = i
        your_form.Show
        If j < f_h Then j = j + loop_step
    Next i

End Select

End Sub

Public Sub form_Animation_Closer( _
    your_form As Form, _
    loop_step As Integer)
On Error GoTo Exit_sub

Dim f_h, f_w, i, j

f_w = your_form.Width
f_h = your_form.Height

For i = f_h To 0 Step loop_step
    your_form.Height = i
    your_form.Show
    DoEvents
Next i

For i = f_w To 0 Step loop_step
    your_form.Width = i
    your_form.Show
    DoEvents
Next i

Exit_sub:

End Sub

Private Sub Form_Load()
    Form_Animation_Opener Me, 2, 3
End Sub

```

مثال :

```

Private Sub Form_Unload(Cancel As Integer)
    form_Animation_Closer Form1, -2
    Me.WindowState = 1
End Sub

```

### ۶- چگونه یک فرم را در وسط صفحه نمایش قرار دهیم؟

```

Public Sub Center_This_Form(f As Form)
    f.Move (Screen.Width - f.Width) \ 2, _
           (Screen.Height - f.Height) \ 2
End Sub

```

مثال :

```

Private Sub Form_Load()
    Center_This_Form Me
End Sub

```

### ۷- چگونه یک فرم بدون عنوان (Caption) را با موشواره (Mouse) حرکت دهیم؟

```

Sub moveform(ff As Form, xx, yy, bt)
    Static oldx, oldy, mf
    Dim moveleft, movetop
    moveleft = ff.Left + xx - oldx
    movetop = ff.Top + yy - oldy
    If bt = vbLeftButton Then
        If mf = 0 Then
            ff.Move moveleft, movetop
            ff.Refresh
            mf = 1
        Else
            mf = 0
        End If
    End If
    oldx = xx
    oldy = yy
End Sub

```

مثال :

```

Private Sub Form_MouseMove(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    moveform Me, X, Y, Button
End Sub

```

راه دوم :

```

Public Const WM_SYSCOMMAND = &H112
Public Const SC_MOVE = &HF010&
Public Declare Function ReleaseCapture Lib "user32" () _
    As Long

Public Declare Function SendMessage Lib "user32" Alias _
    "SendMessageA" (ByVal hwnd As Long, _
    ByVal wParam As Long, _
    ByVal lParam As Long, _

```

lParam As Any) As Long

مثال :

```
Private Sub Form_MouseMove(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    If Button = 1 Then ' Left button
        ReleaseCapture
        ret = SendMessage(Me.hwnd, WM_SYSCOMMAND, _
            SC_MOVE, 0)
    End If
End Sub
```

۸- چگونه یک فرم یا آیکون مشخص از برنامه ای را مخفی کنیم ؟

```
Public Const SW_ERASE = &H4
Public Const SW_HIDE = 0
Public Const SW_INVALIDATE = &H2
Public Const SW_MAX = 10
Public Const SW_MAXIMIZE = 3
Public Const SW_MINIMIZE = 6
Public Const SW_NORMAL = 1
Public Const SW_OTHERUNZOOM = 4
Public Const SW_OTHERZOOM = 2
Public Const SW_PARENTCLOSING = 1
Public Const SW_PARENTOPENING = 3
Public Const SW_RESTORE = 9
Public Const SW_SCROLLCHILDREN = &H1
Public Const SW_SHOW = 5
Public Const SW_SHOWDEFAULT = 10
Public Const SW_SHOWMAXIMIZED = 3
Public Const SW_SHOWMINIMIZED = 2
Public Const SW_SHOWMINNOACTIVE = 7
Public Const SW_SHOWNA = 8
Public Const SW_SHOWNOACTIVATE = 4
Public Const SW_SHOWNORMAL = 1
Public Const SWP_FRAMECHANGED = &H20
Public Const SWP_DRAWFRAME = SWP_FRAMECHANGED
Public Const SWP_HIDEWINDOW = &H80
Public Const SWP_NOACTIVATE = &H10
Public Const SWP_NOCOPYBITS = &H100
Public Const SWP_NOMOVE = &H2
Public Const SWP_NOOWNERZORDER = &H200
Public Const SWP_NOREDRAW = &H8
Public Const SWP_NOREPOSITION = SWP_NOOWNERZORDER
Public Const SWP_NOSIZE = &H1
Public Const SWP_NOZORDER = &H4
Public Const SWP_SHOWWINDOW = &H40

Public Declare Function FindWindow Lib "user32" Alias _
    "FindWindowA" (ByVal lpClassName As String, _
        ByVal lpWindowName As String) As Long

Public Declare Function ShowWindow Lib "user32" _
    (ByVal hwnd As Long, ByVal nCmdShow As Long) As Long
```

مثال :

```

Sub Form_Load()
    Dim Handle As Long
    Handle = FindWindow(0, "Document - WordPad")
    Call ShowWindow(Handle, SW_HIDE)
End Sub

```

۹- چگونه مکان قرارگیری فرم برنامه ای را که با دستور shell اجرا شده تعیین کنیم؟

```

Public Declare Function SetWindowPos Lib "user32" _
    (ByVal hwnd As Long, ByVal hWndInsertAfter As Long, _
    ByVal x As Long, ByVal y As Long, ByVal cx As Long, _
    ByVal cy As Long, ByVal wFlags As Long) As Long

```

مثال :

```

Sub Form_Load()
    Dim r
    Dim myhWnd

    r = Shell("Notepad c:\autoexec.bat", 1)
    DoEvents ' Let the app load completely
    'Get the executable window handle based on window's title
    myhWnd = FindWindow(0, "Notepad - AUTOEXEC.BAT")
    ' Move the window to the desired location
    ' Params 3 and 4 are top and left,
    ' 5 and 6 are window height and width
    SetWindowPos myhWnd, -1, 10, 10, 200, 300, _
        SWP_SHOWWINDOW
End Sub

```

۱۰- مخفی کردن دکمه فرم برنامه در نوار وظیفه (TaskBar) ویندوز

```

Public Declare Function SetWindowLong Lib "user32" Alias _
    "SetWindowLongA" (ByVal hwnd As Long, _
    ByVal nIndex As Long, ByVal dwNewLong As Long) As Long

Public Const GWL_EXSTYLE = (-20)
Public Const WS_EX_TOOLWINDOW = &H80&

Public Sub setShowInTaskbar( _
    Visible As Boolean, hwnd As Long)
    Dim L As Long

    L = ShowWindow(hwnd, SW_HIDE)
    DoEvents
    L = SetWindowLong(hwnd, GWL_EXSTYLE, _
        IIf(Visible, -WS_EX_TOOLWINDOW, WS_EX_TOOLWINDOW))
    DoEvents
    L = ShowWindow(hwnd, SW_SHOW)
End Sub

```

مثال :

```

Private Sub Command1_Click()
    setShowInTaskbar False, Me.hwnd
    MsgBox "click here!"

```

```
    setShowInTaskbar True, Me.hwnd
End Sub
```

## ۱۱- چگونه یک فرم را در وسط فرم دیگری نمایش دهیم؟

```
Option Explicit

Public Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type

Public Declare Function GetWindowRect Lib "user32" _
    (ByVal hwnd As Long, lpRect As RECT) As Long

Sub DialogCenterParent( _
    ByVal hwndParent As Integer, frmDialog As Form)
    Dim iLeft As Integer
    Dim iTop As Integer
    Dim iMidX As Integer
    Dim iMidY As Integer
    Dim rcParent As RECT

    ' Find the ideal center point.
    If hwndParent = 0 Then
        'No parent, so center over the enter screen
        iMidX = Screen.Width / 2
        iMidY = Screen.Height / 2
    Else

        'Center over the form's parent.

        Call GetWindowRect(hwndParent, rcParent)
        ' Next 2 lines as one single line.
        iMidX = ((rcParent.Left * Screen.TwipsPerPixelX) _
            + (rcParent.Right * Screen.TwipsPerPixelY)) / 2
        ' Next 2 lines as one single line.
        iMidY = ((rcParent.Top * Screen.TwipsPerPixelY) _
            + (rcParent.Bottom * Screen.TwipsPerPixelY)) / 2
    End If

    ' Find the form's upper left based on that

    iLeft = iMidX - (frmDialog.Width / 2)
    iTop = iMidY - (frmDialog.Height / 2)

    ' If the form is outside the screen, move it inside

    If iLeft < 0 Then
        iLeft = 0
        ElseIf (iLeft + frmDialog.Width) > Screen.Width Then
            iLeft = Screen.Width - frmDialog.Width
        End If

    If iTop < 0 Then
        iTop = 0
        ElseIf (iTop + frmDialog.Height) > Screen.Height Then
            iTop = Screen.Height - frmDialog.Height
        End If
End Sub
```

```
' Move the form to it's new position
frmDialog.Move iLeft, iTop
```

```
End Sub
```

مثال :

```
Private Sub Command1_Click()
    Form2.Show
    DialogCenterParent Me.hwnd, Form2
End Sub
```

## ۱۲- چگونه عنوان فرم برنامه فعال را بدست آوریم؟

```
Option Explicit
```

```
Private Declare Function GetWindowTextLength Lib "user32" _
    Alias "GetWindowTextLengthA" (ByVal hWnd As Long) As Long
```

```
Private Declare Function GetWindowText Lib "user32" Alias _
    "GetWindowTextA" (ByVal hWnd As Long, _
        ByVal lpString As String, _
        ByVal cch As Long) As Long
```

```
Private Declare Function GetForegroundWindow Lib _
    "user32" () As Long
```

```
Private Declare Function GetParent Lib "user32" _
    (ByVal hWnd As Long) As Long
```

```
' Returns the handle of the active window.
' if GetParent = true then the parent window is
' returned.
```

```
Public Function GetActiveWindow( _
    ByVal ReturnParent As Boolean) As Long
```

```
    Dim i As Long
    Dim j As Long
```

```
    i = GetForegroundWindow
```

```
    If ReturnParent Then
```

```
        Do While i <> 0
```

```
            j = i
```

```
            i = GetParent(i)
```

```
        Loop
```

```
        i = j
```

```
    End If
```

```
    GetActiveWindow = i
```

```
End Function
```

```
Public Function GetWindowTitle( _
    ByVal hWnd As Long) As String
```

```
    Dim L As Long
```

```
    Dim s As String
```

```
    L = GetWindowTextLength(hWnd)
```

```
    s = Space(L + 1)
```

```
    GetWindowText hWnd, s, L + 1
```

```
    GetWindowTitle = Left$(s, L)
```

```
End Function
```

مثال :

```
Private Sub Command1_Click()  
    MsgBox GetWindowText(GetActiveWindow(False))  
End Sub
```

### ۱۳- چگونه می توان متغیری را به یک فرم ، بدون استفاده از متغیرهای عمومی فرستاد؟

روش استاندارد برای انجام این کار این است که یک جعبه متن مخفی را روی فرم قرار دهید و سپس در فرمی دیگر با دستور `Form.myTextbox.Text="Value"` مقداری را به آن فرم دلخواه بفرستید. سپس با استفاده از روال رخداد `Change` آن کنترل می توانید کار مورد نظر خود را در آن فرم انجام دهید. خاصیت `Tag` کنترل را نیز امتحان کنید.

### ۱۴- چگونه فرم دیگر برنامه را از انجام عملیاتی در فرم جاری با خبر سازیم ؟

فرض کنید برنامه شما دو فرم دارد که در یکی از آنها کاربر دو عدد را وارد می سازد و سپس روی دکمه محاسبه کلیک کرده و عملیاتی روی این داده ها در فرم دوم صورت می گیرد و فرم دوم لازم دارد که مجموعه محاسبات خود را لحظه ای متوقف کند. یک روش انجام آن به صورت زیر است :

```
Private Sub Command1_Click()  
    Dim Myform as frmEntry  
    Set Myform = New frmEntry  
    With Myform  
        .Show  
        Do  
            DoEvents  
        Loop Until Myform.Ready  
        'Do some calculations based on the entry  
        txtResults = .txtNum1 * .txtNum2  
    End With  
    Unload frm  
    Set frm = Nothing  
End Sub
```

متغیر `Ready` متغیری است عمومی که فرم دوم را از پایان یافتن ورود اطلاعات آگاه می سازد. وجود تابع `DoEvents` ضروری است اما منابع زیادی از سیستم را مصرف می کند. راه بهتر ایجاد روال رخداد می باشد :

```
Public Event NumbersSubmitted()  
Public NumOne As Long  
Public NumTwo As Long  
  
Private Sub cmdSubmit_Click()  
    NumOne = CLng(txtNum1)
```



```

NumTwo = CLng(txtNum2)
Unload Me
RaiseEvent NumbersSubmitted
End Sub

```

سپس در فرم اصلی روال زیر را برای پاسخ دادن به رخداد تحریک شده می توان ایجاد کرد.

```

Private Sub frmNumEntry_NumbersSubmitted()
With frmNumEntry
    txtResults = .NumOne * .NumTwo
    Set frmNumEntry = Nothing
End With
End Sub

```

frmNumEntry متغیری است عمومی در سطح فرم :

```

Dim WithEvents frmNumEntry As frmNumberEntry

```

۱۵- چگونه مکان و اندازه یک فرم را غیرقابل تغییر نماییم ؟

اندازه غیرقابل تغییر :



```

Option Explicit
Public OldWindowProc As Long ' Original window proc

' Function to retrieve the address of the current Message-Handling
' routine
Declare Function GetWindowLong Lib "user32" Alias "GetWindowLongA" _
    (ByVal hwnd As Long, ByVal nIndex As Long) As Long
' Function to define the address of the Message-Handling routine
Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" _
    (ByVal hwnd As Long, ByVal nIndex As Long, ByVal dwNewLong As Long) _
    As Long
' Function to copy an object/variable/structure passed by reference
' onto a variable of your own
Declare Sub CopyMemory Lib "kernel32" Alias "RtlMoveMemory" (pDest _
    As Any, pSource As Any, ByVal ByteLen As Long)
' Function to execute a function residing at a specific memory
' address
Declare Function CallWindowProc Lib "user32" Alias _
    "CallWindowProcA" _
    (ByVal lpPrevWndFunc As Long, ByVal hwnd As Long, _
    ByVal Msg As Long, _
    ByVal wParam As Long, ByVal lParam As Long) As Long

```

```

' This is the message constant
Public Const WM_GETMINMAXINFO = &H24

' This is a structure referenced by the MINMAXINFO structure
Type POINTAPI
    x As Long
    y As Long
End Type

' This is the structure that is passed by reference (ie an address)
' to your message handler
' The key items in this structure are ptMinTrackSize and
' ptMaxTrackSize
Type MINMAXINFO
    ptReserved As POINTAPI
    ptMaxSize As POINTAPI
    ptMaxPosition As POINTAPI
    ptMinTrackSize As POINTAPI
    ptMaxTrackSize As POINTAPI
End Type
Public Function SubClass1_WndMessage(ByVal hwnd As Long, _
    ByVal Msg As Long, ByVal wp As Long, ByVal lp As Long) As Long

    ' Watch for the pertinent message to come in
    If Msg = WM_GETMINMAXINFO Then

        Dim MinMax As MINMAXINFO

        ' This is necessary because the structure was passed by its
        ' address and there
        ' is currently no intrinsic way to use an address in Visual
        ' Basic
        CopyMemory MinMax, ByVal lp, Len(MinMax)

        ' This is where you set the values of the MinX,MinY,MaxX, and
        ' MaxY
        ' The values placed in the structure must be in pixels. The
        ' values
        ' normally used in Visual Basic are in twips. The conversion
        ' is as follows:
        ' pixels = twips\twipsperpixel
        MinMax.ptMinTrackSize.x = 3975 \ Screen.TwipsPerPixelX
        MinMax.ptMinTrackSize.y = 1740 \ Screen.TwipsPerPixelY
        MinMax.ptMaxTrackSize.x = Screen.Width \ _
            Screen.TwipsPerPixelX \ 2
        MinMax.ptMaxTrackSize.y = 3480 \ Screen.TwipsPerPixelY

        ' Here we copy the datastructure back up to the address
        ' passed in the parameters
        ' because Windows will look there for the information.
        CopyMemory ByVal lp, MinMax, Len(MinMax)

        ' This message tells Windows that the message was handled
        ' successfully
        SubClass1_WndMessage = 1
        Exit Function

    End If

    ' Here, we forward all irrelevant messages on to the default
    ' message handler.

```

```

SubClass1_WndMessage = CallWindowProc(OldWindowProc, hwnd, Msg, _
    wp, lp)

End Function

```

مثال :

```

Option Explicit

' This constant is used to refer to the Message Handling function in
' a given window
Private Const GWL_WNDPROC = (-4)
Private Sub Form_Load()

    ' First, we need to store the address of the existing Message
    ' Handler
    OldWindowProc = GetWindowLong(Me.hWnd, GWL_WNDPROC)

    ' Now we can tell windows to forward all messages to our own
    ' Message Handler
    Call SetWindowLong(Me.hWnd, GWL_WNDPROC, _
        AddressOf SubClass1_WndMessage)

End Sub

Private Sub Form_Unload(Cancel As Integer)

    ' We must return control of the messages back to windows before
    ' the program exits
    Call SetWindowLong(Me.hWnd, GWL_WNDPROC, OldWindowProc)

End Sub

```

مكان غير قابل تغيير :

```

Option Explicit
' Used to get menu information.
Private Type MENUITEMINFO
    cbSize As Long
    fMask As Long
    fType As Long
    fState As Long
    wID As Long
    hSubMenu As Long
    hbmpChecked As Long
    hbmpUnchecked As Long
    dwItemData As Long
    dwTypeData As String
    cch As Long
End Type

' Menu information constants.
Private Const MIIM_STATE As Long = &H1
Private Const MIIM_ID As Long = &H2
Private Const MIIM_SUBMENU As Long = &H4
Private Const MIIM_CHECKMARKS As Long = &H8
Private Const MIIM_TYPE As Long = &H10
Private Const MIIM_DATA As Long = &H20

' System menu command values commonly used by VB.
Private Const SC_SIZE = &HF000&

```

```

Private Const SC_MOVE = &HF010&
Private Const SC_MINIMIZE = &HF020&
Private Const SC_MAXIMIZE = &HF030&
Private Const SC_CLOSE = &HF060&
Private Const SC_RESTORE = &HF120&

' Enumerated system menu items.
Public Enum SysMenuItems
    smRestore = SC_RESTORE
    smMove = SC_MOVE
    smSize = SC_SIZE
    smMinimize = SC_MINIMIZE
    smMaximize = SC_MAXIMIZE
    smClose = SC_CLOSE
End Enum

' Used to select which menu to remove.
Private Const MF_BYCOMMAND = &H0&
Private Const MF_BYPOSITION = &H400

' Toggles enabled state of menu item.
Private Const MF_ENABLED = &H0&
Private Const MF_GRAYED = &H1&
Private Const MF_DISABLED = &H2&

Private Declare Function GetSystemMenu Lib "user32" (ByVal hWnd As _
    Long, ByVal revert As Long) As Long
Private Declare Function GetMenuItemCount Lib "user32" (ByVal hMenu _
    As Long) As Long
Private Declare Function GetMenuItemID Lib "user32" (ByVal hMenu As _
    Long, ByVal nPos As Long) As Long
Private Declare Function GetMenuItemInfo Lib "user32" Alias _
    "GetMenuItemInfoA" (ByVal hMenu As Long, ByVal un As Long, _
    ByVal b As Long, lpMenuItemInfo As MENUITEMINFO) As Long
Private Declare Function SetMenuItemInfo Lib "user32" Alias _
    "SetMenuItemInfoA" (ByVal hMenu As Long, ByVal un As Long, _
    ByVal bool As Long, lpMenuItemInfo As MENUITEMINFO) As Long

Private Function GetMenuItemPosition(frm As Form, _
    ByVal MenuItem As SysMenuItems) As Long
    Dim hMenu As Long
    Dim ID As Long
    Dim i As Long
    Const HighBit As Long = &H8000&

    ' Default to returning -1 in case of
    ' failure, since menu is 0-based.
    GetMenuItemPosition = -1

    ' Retrieve handle to system menu.
    hMenu = GetSystemMenu(frm.hWnd, False)

    ' Loop through system menu, scanning
    ' for requested standard menu item.
    For i = 0 To GetMenuItemCount(hMenu) - 1
        ID = GetMenuItemID(hMenu, i)
        If ID = MenuItem Then
            ' Return position of normal
            ' enabled menu item.

```

```

        GetMenuItemPosition = i
    Exit For
ElseIf ID = (MenuItem And Not HighBit) Then
    ' This item is disabled.
    ' Return position and alter
    ' MenuItem with new ID.
    MenuItem = ID
    GetMenuItemPosition = i
    Exit For
End If
Next i
End Function

Public Sub EnableMenuItem(frm As Form, _
    ByVal MenuItem As SysMenuItems, _
    Optional ByVal Enabled As Boolean = True)
    ' This routine is automatically called whenever the
    ' MinButton, MaxButton, or Movable properties are
    ' set.
    Dim hMenu As Long
    Dim nPosition As Long
    Dim uFlags As Long
    Dim mii As MENUITEMINFO
    Const HighBit As Long = &H8000&

    ' Retrieve handle to system menu.
    hMenu = GetSystemMenu(frm.hWnd, False)

    ' Translate ID to position.
    nPosition = GetMenuItemPosition(frm, MenuItem)
    If nPosition >= 0 Then

        ' Initialize structure.
        mii.cbSize = Len(mii)
        mii.fMask = MIIM_STATE Or MIIM_ID Or MIIM_DATA Or MIIM_TYPE
        mii.dwTypeData = String$(80, 0)
        mii.cch = Len(mii.dwTypeData)
        Call GetMenuItemInfo(hMenu, nPosition, MF_BYPOSITION, mii)

        ' Set appropriate state.
        If Enabled Then
            mii.fState = MF_ENABLED
        Else
            mii.fState = MF_GRAYED
        End If

        ' New ID uses highbit to signify that
        ' the menu item is enabled.
        If Enabled Then
            mii.wID = MenuItem
        Else
            mii.wID = MenuItem And Not HighBit
        End If

        ' Modify the menu!
        mii.fMask = MIIM_STATE Or MIIM_ID
        Call SetMenuItemInfo(hMenu, nPosition, MF_BYPOSITION, mii)
    End If
End Sub

```

```

Private Function GetMenuItemState(ByVal hMenu As Long, ByVal
nPosition As Long) As Long
    Dim mii As MENUITEMINFO

    ' Initialize structure.
    mii.cbSize = Len(mii)
    mii.fMask = MIIM_STATE
    Call GetMenuItemInfo(hMenu, nPosition, MF_BYPOSITION, mii)

    ' Return current state.
    GetMenuItemState = mii.fState
End Function

Public Sub setMoveable(frm As Form, ByVal Value As Boolean)
    ' Toggle SC_MOVE menu appropriately.
    Call EnableMenuItem(frm, smMove, Value)
End Sub

Public Function isMoveable(frm As Form) As Boolean
    ' Return whether SC_MOVE menu is enabled.
    isMoveable = Not CBool(GetMenuItemState( _
        GetSystemMenu(frm.hWnd, False), _
        GetMenuItemPosition(frm, smMove)))
End Function

```

مثال :

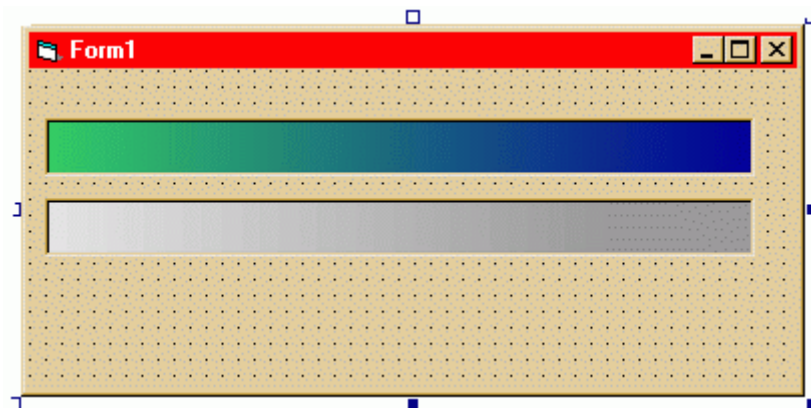
```

Private Sub Command1_Click()
    setMoveable Me, False
End Sub

Private Sub Command2_Click()
    MsgBox isMoveable(Me)
End Sub

```

۱۶- دستکاری نوار عنوان (TITLE BAR)



Option Explicit

```

Public OldWindowProc As Long

Private Declare Function CallWindowProc Lib "user32" Alias
"CallWindowProcA" (ByVal lpPrevWndFunc As Long, ByVal hwnd As Long,
ByVal msg As Long, ByVal wParam As Long, ByVal lParam As Long) As
Long
Private Declare Function DefWindowProc Lib "user32" Alias
"DefWindowProcA" (ByVal hwnd As Long, ByVal wMsg As Long, ByVal
wParam As Long, ByVal lParam As Long) As Long
' *****
' Display message names.
' *****
Public Function NewWindowProc(ByVal hwnd As Long, ByVal msg As Long,
ByVal wParam As Long, ByVal lParam As Long) As Long
Const WM_NCPAINT = &H85
Const WM_ACTIVATE = &H6
Const WM_NCACTIVATE = &H86
Const WM_MDIACTIVATE = &H222
Const WM_SETTEXT = &HC
Const WM_SYSCOMMAND = &H112
Const SC_CLOSE = &HF060&

' Assume we will return True.
NewWindowProc = True

' Process messages.
Select Case msg
    Case WM_NCPAINT
        DefWindowProc hwnd, msg, wParam, lParam
        Form1.PaintActive

    Case WM_NCACTIVATE
        If wParam Then
            ' The form is active.
            DefWindowProc hwnd, msg, wParam, lParam
            Form1.PaintActive
        Else
            ' The form is inactive.
            DefWindowProc hwnd, msg, wParam, lParam
            Form1.PaintInactive
        End If

    Case WM_SETTEXT
        DefWindowProc hwnd, msg, wParam, lParam
        Form1.PaintActive

    Case WM_SYSCOMMAND
        DefWindowProc hwnd, msg, wParam, lParam
        If wParam <> SC_CLOSE Then
            Form1.PaintActive
        End If

    Case Else
        ' Invoke the original WindowProc.
        NewWindowProc = CallWindowProc( _
            OldWindowProc, hwnd, msg, wParam, _
            lParam)
End Select
End Function

```

: مثال

Option Explicit

```
Private Declare Function SetWindowLong Lib "user32" Alias _
    "SetWindowLongA" (ByVal hwnd As Long, ByVal nIndex As Long, _
        ByVal dwNewLong As Long) As Long
Private Declare Function GetWindowDC Lib "user32" ( _
    ByVal hwnd As Long) As Long
Private Declare Function ReleaseDC Lib "user32" ( _
    ByVal hwnd As Long,
    ByVal hdc As Long) As Long
Private Declare Function GetSystemMetrics Lib "user32" ( _
    ByVal nIndex As Long) As Long
Private Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, _
    ByVal x As Long, ByVal y As Long, ByVal nWidth As Long, _
    ByVal nHeight As Long, ByVal hSrcDC As Long, _
    ByVal xSrc As Long, ByVal ySrc As Long, ByVal dwRop As Long) As Long

Private Const GWL_WNDPROC = (-4)
Private Const SM_CXFRAME = 32
Private Const SM_CYFRAME = 33
Private Const SM_CXSIZE = 30
Private Const SM_CYCAPTION = 4
Private Const SRCCOPY = &HCC0020

' Install the new WindowProc.
Private Sub Form_Load()
    ' This reduces flicker.
    Caption = ""

    ' Draw the caption on the PictureBoxes.
    picActive.AutoRedraw = True
    picActive.ScaleMode = vbPixels
    picActive.CurrentX = 2
    picActive.CurrentY = 2
    picActive.Font.Bold = True
    picActive.Print "GradMenu"

    picInactive.AutoRedraw = True
    picInactive.ScaleMode = vbPixels
    picInactive.CurrentX = 2
    picInactive.CurrentY = 2
    picInactive.Font.Bold = True
    picInactive.Print "GradMenu"

    OldWindowProc = SetWindowLong( _
        hwnd, GWL_WNDPROC, _
        AddressOf NewWindowProc)
End Sub

' Paint the title bar for an active form.
Public Sub PaintActive()
    Dim window_dc As Long
    Dim border_width As Long
    Dim border_height As Long
    Dim title_button_width As Long
    Dim wid As Long
    Dim hgt As Long
```



```

window_dc = GetWindowDC(hwnd)
border_width = GetSystemMetrics(SM_CXFRAME)
border_height = GetSystemMetrics(SM_CYFRAME)
title_button_width = GetSystemMetrics(SM_CXSIZE)

' Get the width of the area to draw. This is our
' width in pixels, minus 2 times the border width,
' minus room for the three buttons on the right.
wid = ScaleX(Width, vbTwips, vbPixels) - _
    2 * border_width - _
    3 * title_button_width

' Get the height of the area to draw. This is
' the height of a normal caption minus 1 pixel.
hgt = GetSystemMetrics(SM_CYCAPTION) - 1

' Paint the title bar.
BitBlt window_dc, border_width, border_height, _
    wid, hgt, picActive.hdc, 0, 0, SRCCOPY

' Release the window's DC.
ReleaseDC hwnd, window_dc
End Sub

' Paint the title bar for an inactive form.
Public Sub PaintInactive()
Dim window_dc As Long
Dim border_width As Long
Dim border_height As Long
Dim title_button_width As Long
Dim wid As Long
Dim hgt As Long

window_dc = GetWindowDC(hwnd)
border_width = GetSystemMetrics(SM_CXFRAME)
border_height = GetSystemMetrics(SM_CYFRAME)
title_button_width = GetSystemMetrics(SM_CXSIZE)

' Get the width of the area to draw. This is our
' width in pixels, minus 2 times the border width,
' minus room for the three buttons on the right.
wid = ScaleX(Width, vbTwips, vbPixels) - _
    2 * border_width - _
    3 * title_button_width

' Get the height of the area to draw. This is
' the height of a normal caption minus 1 pixel.
hgt = GetSystemMetrics(SM_CYCAPTION) - 1

' Paint the title bar.
BitBlt window_dc, border_width, border_height, _
    wid, hgt, picInactive.hdc, 0, 0, SRCCOPY

' Release the window's DC.
ReleaseDC hwnd, window_dc
End Sub

```

یک کنترل Timer روی فرم قرار دهید، نام آنرا tmrIcon گذاشته و Enabled آنرا در برگه خواص False کنید. Interval آنرا نیز برابر ۱۰۰۰ قرار دهید.

دو کنترل Picture Box با یک نام pictIcon روی فرم قرار دهید. چون نام آنها یکی است یک آرایه کنترلی از آنها تشکیل خواهد شد. خاصیت مرئی بودن هر دو را False کنید. در قسمت تصویر هر کدام یک آیکون قرار دهید. و سپس کد زیر را به فرم اضافه کنید :

```
Private Sub Form_Resize()  
If Me.WindowState = 1 Then  
    tmrIcon.Enabled = True  
Else  
    tmrIcon.Enabled = False  
End If  
End Sub  
  
Private Sub tmrIcon_Timer()  
If Me.Icon <> pictIcon(0).Picture Then  
    Me.Icon = pictIcon(0).Picture  
Else  
    Me.Icon = pictIcon(1).Picture  
End If  
End Sub
```

پس از اجرای برنامه، فرم را Minimize کنید.

منابع و مراجع :

<http://developer.ecorp.net>  
<http://www.vb-helper.com>  
<http://www.cadvision.com>  
<http://www.vbworld.com>  
<http://www.Abstractvb.com>  
<http://www.mvps.org/vb>