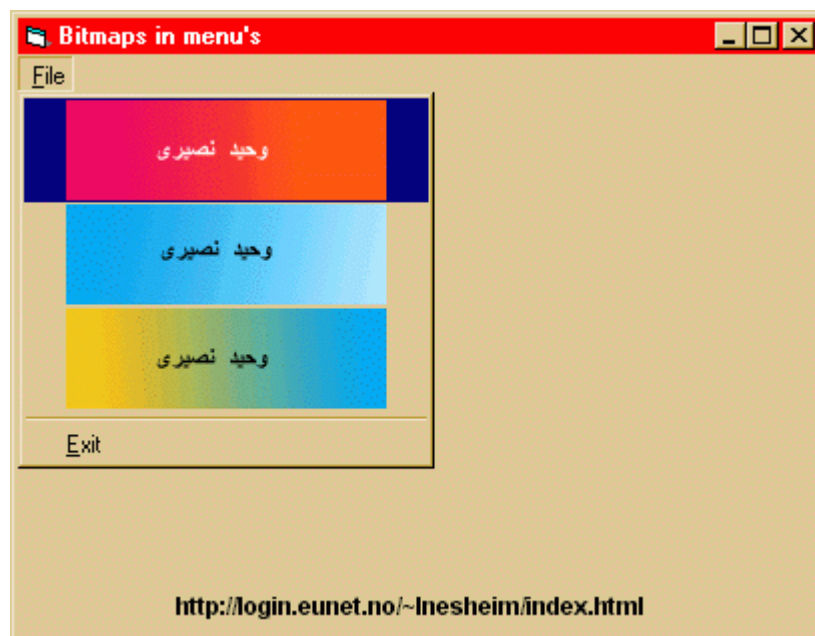
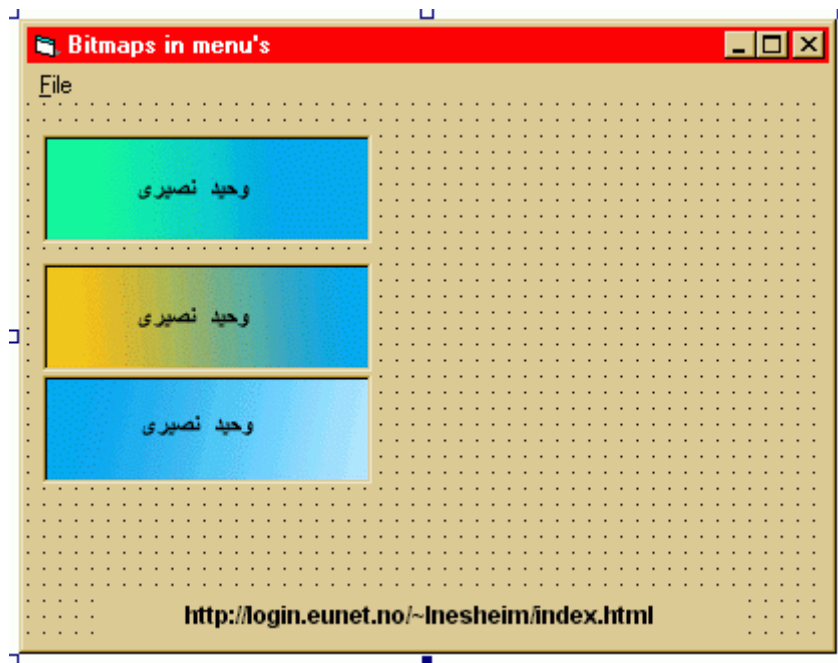


دستکاری منوها با استفاده از توابع ویندوز

- ۱- افزودن تصویر به آیتم (فقره) های منو
- ۲- افزودن آیکون (شکلک!) به آیتم های منو
- ۳- تغییر فونت منو (قابل توجه ، برای علاقمندان به فارسی کردن همه چیز!)
- ۴- حذف آیتم (فقره) های منوی سیستم
- ۵- اضافه کردن آیتم به منوی سیستم
- ۶- شبیه سازی *Menu_MouseOver*
- ۷- ایجاد منوهای شفاف (*Transparent*)
- ۸- ایجاد منو در *System Tray* ویندوز
- ۹- منوهای جهنده (*Pop Up*)

۱- افزودن تصویر به آیتم (فقره)های منو



Option Explicit

```
Private Declare Function GetMenu Lib "user32" (ByVal hWnd As Long) _
As Long
Private Declare Function GetMenuItemID Lib "user32" ( _
ByVal hMenu As Long, ByVal nPos As Long) As Long
Private Declare Function GetSubMenu Lib "user32" ( _
ByVal hMenu As Long, ByVal nPos As Long) As Long
```

```

Private Declare Function ModifyMenu Lib "user32" Alias _
"ModifyMenuA" (ByVal hMenu As Long, ByVal nPosition As Long, _
ByVal wFlags As Long, ByVal wIDNewItem As Long, _
ByVal lpString As Any) As Long

Const MF_BITMAP = 4
Const MF_CHECKED = 8

Private Sub Form_Load()

    Dim hMenu As Long, hSubMenu As Long, lngID As Long

    hMenu = GetMenu(Me.hWnd)
    hSubMenu = GetSubMenu(hMenu, 0)

    picBitmaps(0).Picture = picBitmaps(0).Image
    lngID = GetMenuItemID(hSubMenu, 0)
    Call ModifyMenu(hMenu, lngID, MF_BITMAP, lngID, _
        CLng(picBitmaps(0).Picture))

    picBitmaps(1).Picture = picBitmaps(1).Image
    lngID = GetMenuItemID(hSubMenu, 1)
    Call ModifyMenu(hMenu, lngID, MF_BITMAP, lngID, _
        CLng(picBitmaps(1).Picture))

    picBitmaps(2).Picture = picBitmaps(2).Image
    lngID = GetMenuItemID(hSubMenu, 2)
    Call ModifyMenu(hMenu, lngID, MF_BITMAP, lngID, _
        CLng(picBitmaps(2).Picture))

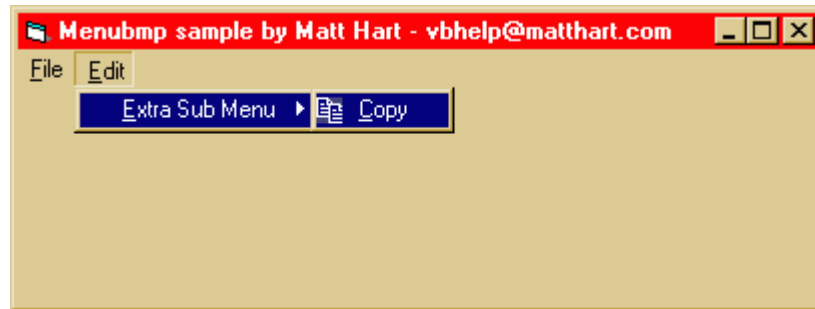
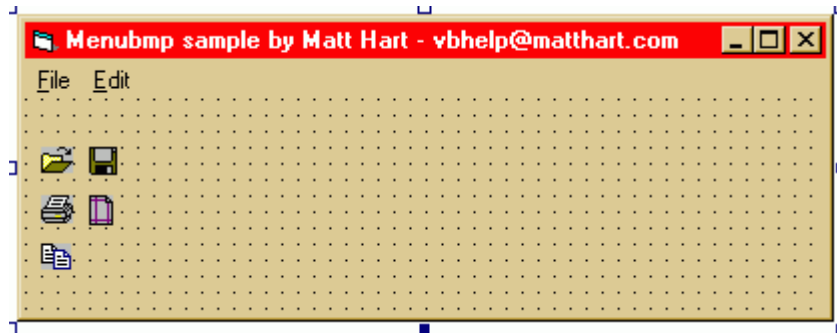
End Sub

Private Sub mnuBitmap1_Click()
    mnuBitmap1.Checked = Not mnuBitmap1.Checked
End Sub

Private Sub mnuBitmapAfsluiten_Click()
    End
End Sub

```

۲- افزودن آیکن (شکلک!) به آیتم های منو



Option Explicit

```
' This sample shows you how to add bitmaps to your
' menu items. First, you must retrieve the VB menu handle
' with the GetMenu API call. Then, you set the Unchecked and
' Checked bitmaps (I don't differentiate between the two
' with this example - usually you would put a checkmark in
' the bitmap and use that for Checked).
```

```
' Note that the Picture property of an Image control (or
' a Picture control) is the Bitmap Handle, this the Image
' controls can be used like a bitmap resource.
```

```
' I also have one sub menu so that you can see how getting
' submenu handles and item positions works.
```

```
Private Declare Function GetMenu Lib "user32" _
    (ByVal hwnd As Long) As Long
```

```
Private Declare Function GetSubMenu Lib "user32" _
    (ByVal hMenu As Long, ByVal nPos As Long) As Long
```

```

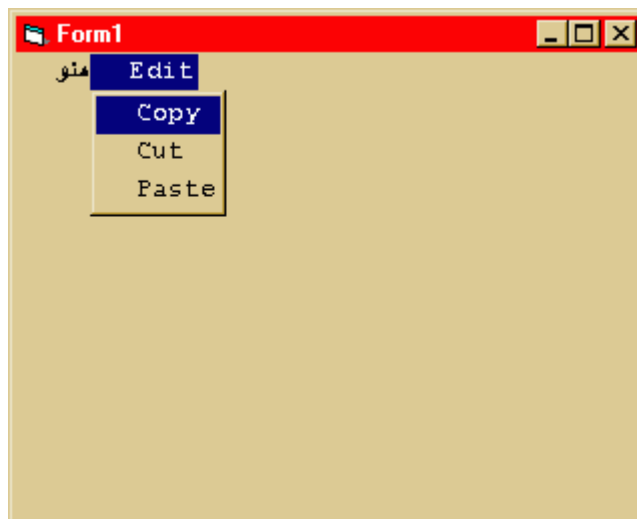
Private Declare Function SetMenuItemBitmaps Lib "user32" _
    (ByVal hMenu As Long, ByVal nPosition As Long, _
    ByVal wFlags As Long, _
    ByVal hBitmapUnchecked As Long, ByVal hBitmapChecked As Long) _
    As Long

Const MF_BYPOSITION = &H400&

Private Sub Form_Load()
    Dim mHandle As Long, lRet As Long, sHandle As Long, _
        sHandle2 As Long
    mHandle = GetMenu(hwnd)
    sHandle = GetSubMenu(mHandle, 0)
    lRet = SetMenuItemBitmaps(sHandle, 0, MF_BYPOSITION, _
        imOpen.Picture, imOpen.Picture)
    lRet = SetMenuItemBitmaps(sHandle, 1, MF_BYPOSITION, _
        imSave.Picture, imSave.Picture)
    lRet = SetMenuItemBitmaps(sHandle, 3, MF_BYPOSITION, _
        imPrint.Picture, imPrint.Picture)
    lRet = SetMenuItemBitmaps(sHandle, 4, MF_BYPOSITION, _
        imPrintSetup.Picture, imPrintSetup.Picture)
    sHandle = GetSubMenu(mHandle, 1)
    sHandle2 = GetSubMenu(sHandle, 0)
    lRet = SetMenuItemBitmaps(sHandle2, 0, MF_BYPOSITION, _
        imCopy.Picture, imCopy.Picture)
End Sub

```

۳- تغییر فونت منو (قابل توجه ، برای علاقمندان به فارسی کردن همه چیز!)





```
Option Explicit
```

```
DefLng A-Z
```

```
Const MFT_STRING = 0
```

```
Type RECT
```

```
    Left As Long
```

```
    Top As Long
```

```
    Right As Long
```

```
    Bottom As Long
```

```
End Type
```

```
Type Size
```

```
    cx As Long
```

```
    cy As Long
```

```
End Type
```

```
'MENUITEMINFO
```

```
Public 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
```

```
' MEASUREITEMSTRUCT for ownerdraw
```

```
Type MEASUREITEMSTRUCT
```

```
    CtlType As Long
```

```
    CtlID As Long
```

```
    itemID As Long
```

```
    itemWidth As Long
```

```
    itemHeight As Long
```

```
    itemData As Long
```

```
End Type
```

```

' DRAWITEMSTRUCT for ownerdraw
Type DRAWITEMSTRUCT
    CtlType As Long
    CtlID As Long
    itemID As Long
    itemAction As Long
    itemState As Long
    hwndItem As Long
    hdc As Long
    rcItem As RECT
    itemData As Long
End Type

Public Declare Function GetMenu Lib "user32" _
    (ByVal hWnd As Long) As Long

Public Declare Function GetSubMenu Lib "user32" _
    (ByVal hMenu As Long, ByVal nPos As Long) As Long

Public Declare Function GetMenuItemCount Lib "user32" _
    (ByVal hMenu As Long) As Long

Public Declare Function GetMenuItemInfo Lib "user32" _
    Alias "GetMenuItemInfoA" _
    (ByVal hMenu As Long, ByVal un As Long, _
    ByVal b As Boolean, lpmi As MENUITEMINFO) As Long

Declare Function GetMenuItemID Lib "user32" _
    (ByVal hMenu As Long, ByVal nPos As Long) As Long

Public Declare Function SetMenuItemInfo Lib "user32" _
    Alias "SetMenuItemInfoA" _
    (ByVal hMenu As Long, ByVal uItem As Long, _
    ByVal fByPosition As Long, lpmi As MENUITEMINFO) As Long

Declare Function AppendMenu Lib "user32" _
    Alias "AppendMenuA" (ByVal hMenu As Long, _
    ByVal wFlags As Long, ByVal wIDNewItem As Long, _
    ByVal lpNewItem As Any) As Long

Declare Function RemoveMenu Lib "user32" _
    (ByVal hMenu As Long, ByVal nPosition As Long, _
    ByVal wFlags As Long) As Long

Declare Function CreateFont Lib "gdi32" _
    Alias "CreateFontA" (ByVal H As Long, _
    ByVal W As Long, ByVal E As Long, ByVal O As Long, _
    ByVal W As Long, ByVal I As Long, ByVal U As Long, _
    ByVal S As Long, ByVal C As Long, ByVal OP As Long, _
    ByVal CP As Long, ByVal Q As Long, ByVal PAF As Long, _
    ByVal F As String) As Long

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

'MENUITEMINFO
Public Const MIIM_STATE = &H1
Public Const MIIM_ID = &H2
Public Const MIIM_SUBMENU = &H4
Public Const MIIM_CHECKMARKS = &H8

```

```

Public Const MIIM_TYPE = &H10
Public Const MIIM_DATA = &H20

'menustyle
Public Const MF_BYCOMMAND = &H0&
Public Const MF_BYPOSITION = &H400&

Public Const MF_STRING = &H0&
Public Const MF_BITMAP = &H4&
Public Const MF_OWNERDRAW = &H100&

'textout style
Public Const ETO_OPAQUE = 2

' Owner draw state
Public Const ODS_SELECTED = &H1
Public Const ODS_GRAYED = &H2
Public Const ODS_DISABLED = &H4
Public Const ODS_CHECKED = &H8
Public Const ODS_FOCUS = &H10

'messages:
Public Const WM_COMMAND = &H111
Public Const WM_SYSCOMMAND = &H112
Public Const WM_MENUSELECT = &H11F
Public Const WM_LBUTTONDOWN = &H202
Public Const WM_MBUTTONDOWN = &H208
Public Const WM_RBUTTONDOWN = &H205
Public Const WM_USER = &H400
Public Const WM_CREATE = &H1
Public Const WM_DESTROY = &H2
Public Const WM_DRAWITEM = &H2B
Public Const WM_MEASUREITEM = &H2C
Public Const WM_SYSCOLORCHANGE = &H15

Declare Sub MemCopy Lib "kernel32" Alias _
    "RtlMoveMemory" (dest As Any, src As Any, _
        ByVal numbytes As Long)

Public Const GWL_WNDPROC = (-4)
Public Const GWL_USERDATA = (-21)

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

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

Declare Function TextOut Lib "gdi32" Alias "TextOutA" _
    (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, _
    ByVal lpString As String, ByVal nCount As Long) As Long

Declare Function ExtTextOut Lib "gdi32" Alias _
    "ExtTextOutA" (ByVal hdc As Long, ByVal x As _
    Long, ByVal y As Long, ByVal wOptions As Long, _
    lpRect As RECT, ByVal lpString As String, _
    ByVal nCount As Long, lpDx As Long) 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 SelectObject Lib "gdi32" _
    (ByVal hdc As Long, ByVal hObject As Long) As Long

Declare Function SetBkColor Lib "gdi32" _
    (ByVal hdc As Long, ByVal crColor As Long) As Long

Declare Function SetTextColor Lib "gdi32" _
    (ByVal hdc As Long, ByVal crColor As Long) As Long

Declare Function GetSysColor Lib "user32" _
    (ByVal nIndex As Long) As Long

Declare Function GetTextExtentPoint Lib "gdi32" _
    Alias "GetTextExtentPointA" (ByVal hdc As Long, _
    ByVal lpszString As String, ByVal cbString As Long, _
    lpSize As Size) As Long

Public Const COLOR_MENU = 4
Public Const COLOR_MENUTEXT = 7
Public Const COLOR_HIGHLIGHT = 13
Public Const COLOR_HIGHLIGHTTEXT = 14
Public Const COLOR_GRAYTEXT = 17

'consts MenuItem IDs.
Public Const IDM_CHARACTER = 10
Public Const IDM_REGULAR = 11
Public Const IDM_BOLD = 12
Public Const IDM_ITALIC = 13
Public Const IDM_UNDERLINE = 14

Type myItemType
    cchItemText As Integer
    szItemText As String * 32
End Type

Public OldWindowProc
Public hMenu, hSubMenu
Public iNoOfMenuItems, MyItem() As myItemType
Public clrPrevText, clrPrevBkgnd
Public hfntPrev

Public Const ODT_MENU = 1
Public hFont As Long
Public Function NewWindowProc(ByVal hWnd As Long, _
    ByVal msg As Long, _
    ByVal wParam As Long, lParam As Long) As Long
    Dim mM As MEASUREITEMSTRUCT
    Dim dM As DRAWITEMSTRUCT
    Select Case msg
        Case WM_DRAWITEM
            MemCopy dM, lParam, Len(dM)
            If dM.CtlType = ODT_MENU Then
                OnDrawMenuItem hWnd, dM
            End If
    End Select
End Function

```

```

        Case WM_MEASUREITEM
            MemCopy mM, lParam, Len(mM)
            If mM.CtlType = ODT_MENU Then
                mM = OnMeasureItem(hWnd, mM)
                MemCopy lParam, mM, Len(mM)
            End If
        End Select
    NewWindowProc = CallWindowProc(OldWindowProc, hWnd, msg, _
                                    wParam, VarPtr(lParam))
End Function

Sub CreateMenus(hWnd As Long)
    hMenu = GetMenu(hWnd)
    hFont = CreateFont(20, 0, 0, 0, 0, 0, 0, 0, 0, 106, _
        0, 16, 0, 0, "IranSystem1")
    Dim iNoOfMenu%, iNoOfSubMenu%
    Dim iCounter1%, iCounter2%
    iNoOfMenu = GetMenuItemCount(hMenu)
    'iNoOfMenuItems

    '*****
    ReDim MyItem(1 To 7)
    'Here I choose 7 since altogether there are 7 menuitems in
    'File & Edit menu. If u want can write a function to
    'findout the No. of menu items by extending the following
    'For Loop.
    '*****
    If iNoOfMenu Then
        For iCounter1 = 0 To iNoOfMenu - 1
            CreateOwnerDrawMenus hMenu, iCounter1
            hSubMenu = GetSubMenu(hMenu, iCounter1)
            iNoOfSubMenu = GetMenuItemCount(hSubMenu)
            If iNoOfSubMenu Then
                For iCounter2 = 0 To iNoOfSubMenu - 1
                    CreateOwnerDrawMenus hSubMenu, iCounter2
                Next iCounter2
            End If
        Next iCounter1
    End If
End Sub

Sub CreateOwnerDrawMenus(hdMenu As Long, iMenuID As Integer)
    Dim minfo As MENUITEMINFO, r As Long
    iNoOfMenuItems = iNoOfMenuItems + 1
    minfo.cbSize = Len(minfo)
    minfo.fMask = MIIM_TYPE
    minfo.fType = MFT_STRING
    minfo.dwTypeData = Space$(256)
    minfo.cch = Len(minfo.dwTypeData)
    'get menuitem data
    r = GetMenuItemInfo(hdMenu, iMenuID, True, minfo)
    'and save into user array
    MyItem(iNoOfMenuItems).cchItemText = minfo.cch 'menuitem length
    MyItem(iNoOfMenuItems).szItemText = Trim(minfo.dwTypeData) 'text
    'change menu type
    minfo.fType = MF_OWNERDRAW
    minfo.fMask = MIIM_TYPE Or MIIM_DATA
    minfo.dwItemData = iNoOfMenuItems
    'into MF_OWNERDRAW
    r = SetMenuItemInfo(hdMenu, iMenuID, True, minfo)
End Sub

```

```

Function OnMeasureItem(hWnd As Long, lpmis As MEASUREITEMSTRUCT) _
    As MEASUREITEMSTRUCT
    On Error GoTo E2
    Dim xM As MEASUREITEMSTRUCT, hfntOld As Long
    Dim S As Size, hdc As Long

    'find DC
    hdc = GetDC(hWnd)

    hfntOld = SelectObject(hdc, hFont)

    GetTextExtentPoint hdc, MyItem(lpmis.itemData).szItemText, _
        MyItem(lpmis.itemData).cchItemText, S

    'set menu item rect
    xM.itemWidth = S.cx + 10
    xM.itemHeight = S.cy

    SelectObject hdc, hfntOld
    ReleaseDC hWnd, hdc

    LSet OnMeasureItem = xM
    Exit Function
E2:
    Form1.Caption = lpmis.itemData
    Exit Function
End Function

Sub OnDrawMenuItem(hWnd As Long, lpdis As DRAWITEMSTRUCT)
    On Error GoTo E1
    Dim x, y

    'set the menuitem colors
    If (lpdis.itemState And ODS_SELECTED) Then 'if selected
        clrPrevText = SetTextColor(lpdis.hdc, _
            GetSysColor(COLOR_HIGHLIGHTTEXT))
        clrPrevBkgnd = SetBkColor(lpdis.hdc, _
            GetSysColor(COLOR_HIGHLIGHT))
    Else
        clrPrevText = SetTextColor(lpdis.hdc, _
            GetSysColor(COLOR_MENUTEXT))
        clrPrevBkgnd = SetBkColor(lpdis.hdc, GetSysColor(COLOR_MENU))
    End If

    'leave space for checkmark
    'may use GetMenuCheckMarkDimensions
    x = lpdis.rcItem.Left + 20
    y = lpdis.rcItem.Top

    hfntPrev = SelectObject(lpdis.hdc, hFont)

    ExtTextOut lpdis.hdc, x, y, ETO_OPAQUE, _
        lpdis.rcItem, Trim(" "), 1&, 0&

    TextOut lpdis.hdc, x, y, MyItem(lpdis.itemData).szItemText, _
        MyItem(lpdis.itemData).cchItemText
    'Form1.Caption = lpdis.itemData
    'may put some bitblt function here also.

    SelectObject lpdis.hdc, hfntPrev
    SetTextColor lpdis.hdc, clrPrevText

```

```

        SetBkColor lpdis.hdc, clrPrevBkgnd
    Exit Sub
E1:
    Form1.Caption = lpdis.itemData
    Exit Sub
End Sub
Sub OnDestroy()
    Dim r As Long
    'do some clean works
    Dim minfo As MENUITEMINFO, id As Integer
    Dim iNoOfMenu%, iNoOfSubMenu%
    Dim iCounter1%, iCounter2%
    iNoOfMenu = GetMenuItemCount(hMenu)
    'iMenuItemBound
    If iNoOfMenu Then
        For iCounter1 = 0 To iNoOfMenu - 1
            minfo.fMask = MIIM_DATA
            r = GetMenuItemInfo(hMenu, iCounter1, True, minfo)
            DeleteObject minfo.dwItemData
            r = SetMenuItemInfo(hMenu, iCounter1, True, minfo)
            hSubMenu = GetSubMenu(hMenu, iCounter1)
            iNoOfSubMenu = GetMenuItemCount(hSubMenu)
            If iNoOfSubMenu Then
                For iCounter2 = 0 To iNoOfSubMenu - 1
                    minfo.fMask = MIIM_DATA
                    r = GetMenuItemInfo(hSubMenu, iCounter2, _
                        True, minfo)
                    DeleteObject minfo.dwItemData
                    r = SetMenuItemInfo(hSubMenu, _
                        iCounter2, True, minfo)
                Next iCounter2
            End If
        Next iCounter1
    End If
    DeleteObject hFont
    Erase MyItem
End Sub

```

: مثال

```

Option Explicit

Private Sub close_Click()
    MsgBox "close"
End Sub

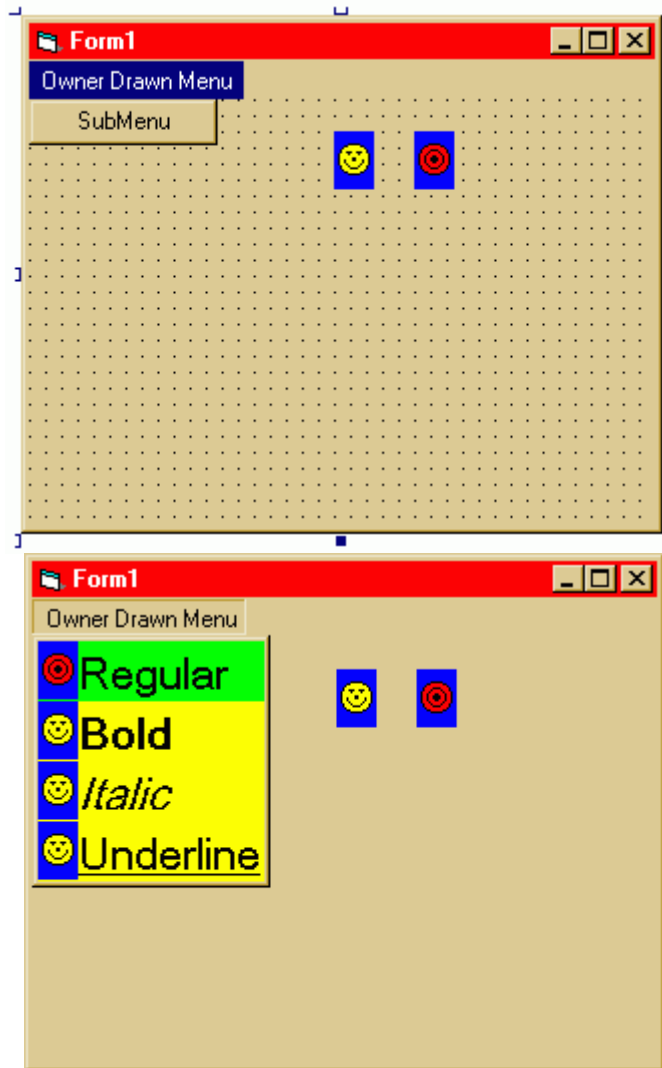
Private Sub Form_Load()
    Call CreateMenus(Me.hWnd)
    'set Callback
    OldWindowProc = SetWindowLong(Me.hWnd, _
        GWL_WNDPROC, AddressOf NewWindowProc)
End Sub

Private Sub Form_Unload(Cancel As Integer)
    'do some clean work
    Call OnDestroy
End Sub

Private Sub mnuClose_Click()
    Unload Me
End Sub

```

نوعی دیگر:



```
Option Explicit
```

```
DefLng A-Z
```

```
Const MFT_STRING = 0
```

```
Type RECT
```

```
    Left As Long
```

```
    Top As Long
```

```
    Right As Long
```

```
    Bottom As Long
```

```
End Type
```

```
Type Size
```

```
    cx As Long
```

```
    cy As Long
```

```
End Type
```

```
'MENUITEMINFO
```

```
Public 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

' MEASUREITEMSTRUCT for ownerdraw
Type MEASUREITEMSTRUCT
    CtlType As Long
    CtlID As Long
    itemID As Long
    itemWidth As Long
    itemHeight As Long
    itemData As Long
End Type

' DRAWITEMSTRUCT for ownerdraw
Type DRAWITEMSTRUCT
    CtlType As Long
    CtlID As Long
    itemID As Long
    itemAction As Long
    itemState As Long
    hwndItem As Long
    hdc As Long
    rcItem As RECT
    itemData As Long
End Type

Public Declare Function GetMenu Lib "user32" _
    (ByVal hwnd As Long) As Long

Public Declare Function GetSubMenu Lib "user32" _
    (ByVal hMenu As Long, ByVal nPos As Long) As Long

Public Declare Function GetMenuItemCount Lib "user32" _
    (ByVal hMenu As Long) As Long

Public Declare Function GetMenuItemInfo Lib "user32" _
    Alias "GetMenuItemInfoA" _
    (ByVal hMenu As Long, ByVal un As Long, _
    ByVal b As Boolean, lpmi As MENUITEMINFO) As Long

Declare Function GetMenuItemID Lib "user32" _
    (ByVal hMenu As Long, ByVal nPos As Long) As Long

Public Declare Function SetMenuItemInfo Lib "user32" _
    Alias "SetMenuItemInfoA" _
    (ByVal hMenu As Long, ByVal uItem As Long, _
    ByVal fByPosition As Long, lpmi As MENUITEMINFO) As Long

Declare Function AppendMenu Lib "user32" _
    Alias "AppendMenuA" (ByVal hMenu As Long, _
    ByVal wFlags As Long, ByVal wIDNewItem As Long, _

```

```

    ByVal lpNewItem As Any) As Long

Declare Function RemoveMenu Lib "user32" _
    (ByVal hMenu As Long, ByVal nPosition As Long, _
    ByVal wFlags As Long) As Long

Declare Function CreateFont Lib "gdi32" _
    Alias "CreateFontA" (ByVal H As Long, _
    ByVal W As Long, ByVal E As Long, ByVal O As Long, _
    ByVal S As Long, ByVal I As Long, ByVal U As Long, _
    ByVal S As Long, ByVal C As Long, ByVal OP As Long, _
    ByVal CP As Long, ByVal Q As Long, ByVal PAF As Long, _
    ByVal F As String) As Long

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

'MENUITEMINFO
Public Const MIIM_STATE = &H1
Public Const MIIM_ID = &H2
Public Const MIIM_SUBMENU = &H4
Public Const MIIM_CHECKMARKS = &H8
Public Const MIIM_TYPE = &H10
Public Const MIIM_DATA = &H20

'menustyle
Public Const MF_BYCOMMAND = &H0&
Public Const MF_BYPOSITION = &H400&

Public Const MF_STRING = &H0&
Public Const MF_BITMAP = &H4&
Public Const MF_OWNERDRAW = &H100&

'textout style
Public Const ETO_OPAQUE = 2

' Owner draw state
Public Const ODS_SELECTED = &H1
Public Const ODS_GRAYED = &H2
Public Const ODS_DISABLED = &H4
Public Const ODS_CHECKED = &H8
Public Const ODS_FOCUS = &H10

'messages:
Public Const WM_COMMAND = &H111
Public Const WM_SYSCOMMAND = &H112
Public Const WM_MENUSELECT = &H11F
Public Const WM_LBUTTONDOWN = &H202
Public Const WM_MBUTTONDOWN = &H208
Public Const WM_RBUTTONDOWN = &H205
Public Const WM_USER = &H400
Public Const WM_CREATE = &H1
Public Const WM_DESTROY = &H2
Public Const WM_DRAWITEM = &H2B
Public Const WM_MEASUREITEM = &H2C
Public Const WM_SYSCOLORCHANGE = &H15

Declare Sub MemCopy Lib "kernel32" Alias _
    "RtlMoveMemory" (dest As Any, src As Any, _
    ByVal numbytes As Long)

```

```

Public Const GWL_WNDPROC = (-4)
Public Const GWL_USERDATA = (-21)

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

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

Declare Function TextOut Lib "gdi32" Alias "TextOutA" _
    (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, _
    ByVal lpString As String, ByVal nCount As Long) As Long

Declare Function ExtTextOut Lib "gdi32" Alias _
    "ExtTextOutA" (ByVal hdc As Long, ByVal x As _
    Long, ByVal y As Long, ByVal wOptions As Long, _
    lpRect As RECT, ByVal lpString As String, _
    ByVal nCount As Long, lpDx As Long) 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 SelectObject Lib "gdi32" _
    (ByVal hdc As Long, ByVal hObject As Long) As Long

Declare Function SetBkColor Lib "gdi32" _
    (ByVal hdc As Long, ByVal crColor As Long) As Long

Declare Function SetTextColor Lib "gdi32" _
    (ByVal hdc As Long, ByVal crColor As Long) As Long

Declare Function GetSysColor Lib "user32" _
    (ByVal nIndex As Long) As Long

Declare Function GetTextExtentPoint Lib "gdi32" _
    Alias "GetTextExtentPointA" (ByVal hdc As Long, _
    ByVal lpszString As String, ByVal cbString As Long, _
    lpSize As Size) As Long

Public Const COLOR_MENU = 4
Public Const COLOR_MENUTEXT = 7
Public Const COLOR_HIGHLIGHT = 13
Public Const COLOR_HIGHLIGHTTEXT = 14
Public Const COLOR_GRAYTEXT = 17

'consts MenuItem IDs.
Public Const IDM_CHARACTER = 10
Public Const IDM_REGULAR = 11
Public Const IDM_BOLD = 12
Public Const IDM_ITALIC = 13
Public Const IDM_UNDERLINE = 14

Type myItemType
    hFont As Long

```



```

        cchItemText As Integer
        szItemText As String * 32
    End Type

    Public OldWindowProc
    Public hMenu, hSubMenu
    Public mnuItemCount, MyItem() As myItemType
    Public clrPrevText, clrPrevBkgnd
    Public hfntPrev

    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 SRCCOPY = &HCC0020 ' (DWORD) dest = source

    Public Function NewWindowProc(ByVal hwnd As Long, _
        ByVal msg As Long, ByVal wParam As Long, _
        lParam As Long) As Long

        Dim mM As MEASUREITEMSTRUCT
        Dim dM As DRAWITEMSTRUCT

        Select Case msg

            Case WM_DRAWITEM

                MemCopy dM, lParam, Len(dM)
                OnDrawMenuItem hwnd, dM

            Case WM_MEASUREITEM

                MemCopy mM, lParam, Len(mM)
                mM = OnMeasureItem(hwnd, mM)
                MemCopy lParam, mM, Len(mM)

            Case WM_COMMAND

                'Put your Menu Command here.

            Case WM_SYSCOLORCHANGE

                'Put your code here.

            Case Else

        End Select

        NewWindowProc = CallWindowProc(OldWindowProc, _
            hwnd, msg, wParam, VarPtr(lParam))

    End Function

    Sub CreateMenus(hwnd As Long)

        'get Menus
        hMenu = GetMenu(hwnd)
        hSubMenu = GetSubMenu(hMenu, 0)

        'remove original menu item

```

```

RemoveMenu hSubMenu, 0, MF_BYPOSITION

'creates string menus
AppendMenu hSubMenu, MF_STRING, IDM_REGULAR, "Regular"
AppendMenu hSubMenu, MF_STRING, IDM_BOLD, "Bold"
AppendMenu hSubMenu, MF_STRING, IDM_ITALIC, "Italic"
AppendMenu hSubMenu, MF_STRING, IDM_UNDERLINE, "Underline"

'call to make OwnerDrawMenus
CreateOwnerDrawMenus

End Sub
Sub CreateOwnerDrawMenus()

    Dim minfo As MENUITEMINFO, id As Integer

'get the menuitem handle
hSubMenu = GetSubMenu(GetMenu(Form1.hwnd), 0)
mnuItemCount = GetMenuItemCount(hSubMenu)

'ReDim usertype array for menuitems
ReDim MyItem(0 To mnuItemCount - 1) As myItemType
Dim r As Long

'loop to fill array
For id = 0 To mnuItemCount - 1
    minfo.cbSize = Len(minfo)
    minfo.fMask = MIIM_TYPE
    minfo.fType = MFT_STRING
    minfo.dwTypeData = Space$(256)
    minfo.cch = Len(minfo.dwTypeData)

'get menuitem data
r = GetMenuItemInfo(hSubMenu, id, True, minfo)

'and save into user array
MyItem(id).cchItemText = minfo.cch 'menuitem length
MyItem(id).szItemText = Trim(minfo.dwTypeData) 'text
MyItem(id).hFont = CreateMenuItemFont(id) 'font

'change menu type
minfo.fType = MF_OWNERDRAW
minfo.fMask = MIIM_TYPE Or MIIM_DATA
minfo.dwItemData = id

'into MF_OWNERDRAW
r = SetMenuItemInfo(hSubMenu, id, True, minfo)

Next id

End Sub

Function OnMeasureItem(hwnd As Long, lpmis As MEASUREITEMSTRUCT) _
    As MEASUREITEMSTRUCT

    Dim xM As MEASUREITEMSTRUCT, hfntOld As Long
    Dim S As Size, hdc As Long

'find DC

```

```

hdc = GetDC(hwnd)

hfntOld = SelectObject(hdc, MyItem(lpmis.itemData).hFont)

GetTextExtentPoint hdc, MyItem(lpmis.itemData).szItemText, _
    MyItem(lpmis.itemData).cchItemText, S

'set menu item rect
xM.itemWidth = S.cx + 10
xM.itemHeight = S.cy

SelectObject hdc, hfntOld
ReleaseDC hwnd, hdc

LSet OnMeasureItem = xM

End Function

Sub OnDrawMenuItem(hwnd As Long, lpdis As DRAWITEMSTRUCT)

Dim x, y

'set the menuitem colors
If (lpdis.itemState And ODS_SELECTED) Then 'if selected
'   clrPrevText = SetTextColor(lpdis.hdc, _
'       GetSysColor(COLOR_HIGHLIGHTTEXT))
'   clrPrevBkgnd = SetBkColor(lpdis.hdc, _
'       GetSysColor(COLOR_HIGHLIGHT))
  clrPrevBkgnd = SetBkColor(lpdis.hdc, vbRed)
  clrPrevBkgnd = SetBkColor(lpdis.hdc, vbGreen)
Else
'   clrPrevText = SetTextColor(lpdis.hdc, _
'       GetSysColor(COLOR_MENUTEXT))
'   clrPrevBkgnd = SetBkColor(lpdis.hdc, _
'       GetSysColor(COLOR_MENU))
  clrPrevBkgnd = SetBkColor(lpdis.hdc, vbBlue)
  clrPrevBkgnd = SetBkColor(lpdis.hdc, vbYellow)
End If

'leave space for checkmark
'may use GetMenuCheckMarkDimensions
x = lpdis.rcItem.Left + 20
y = lpdis.rcItem.Top

hfntPrev = SelectObject(lpdis.hdc, MyItem(lpdis.itemData).hFont)

ExtTextOut lpdis.hdc, x, y, ETO_OPAQUE, _
    lpdis.rcItem, Trim(" "), 1&, 0&

TextOut lpdis.hdc, x, y, MyItem(lpdis.itemData).szItemText, _
    MyItem(lpdis.itemData).cchItemText

' Draw a picture.
If (lpdis.itemState And ODS_SELECTED) Then 'if selected
  BitBlt lpdis.hdc, 0, y, Form1.Picture2.ScaleWidth, _
    Form1.Picture2.ScaleHeight, Form1.Picture2.hdc, _
    0, 0, SRCCOPY
Else
  BitBlt lpdis.hdc, 0, y, Form1.Picture1.ScaleWidth, _
    Form1.Picture1.ScaleHeight, Form1.Picture1.hdc, 0, 0, SRCCOPY
End If

```

```

        SelectObject lpdis.hdc, hfntPrev
        SetTextColor lpdis.hdc, clrPrevText
        SetBkColor lpdis.hdc, clrPrevBkgnd

End Sub
Function CreateMenuItemFont(uID As Integer) As Long
Dim Weight As Long
Dim use_italic As Long
Dim use_underline As Long
Dim use_strikethrough As Long

    Select Case uID + 11

        Case IDM_BOLD

            Weight = 700

        Case IDM_ITALIC

            use_italic = True

        Case IDM_UNDERLINE

            use_underline = True

    End Select

CreateMenuItemFont = CreateFont(30, 0, _
    0, 0, Weight, _
    use_italic, use_underline, _
    use_strikethrough, 136, 0, _
    16, 0, 0, "Times New Roman")

End Function

Sub OnDestroy()
Dim r As Long

    'do some clean works
    Dim minfo As MENUITEMINFO, id As Integer

    hSubMenu = GetSubMenu(GetMenu(Form1.hwnd), 0)
    mnuItemCount = GetMenuItemCount(hSubMenu)

    For id = 0 To mnuItemCount - 1
        minfo.fMask = MIIM_DATA

        r = GetMenuItemInfo(hSubMenu, id, True, minfo)

        DeleteObject minfo.dwItemData

        r = SetMenuItemInfo(hSubMenu, id, True, minfo)
    Next

    Erase MyItem

End Sub

Option Explicit

```

: مثال

```

Private Sub Form_Load()

    Call CreateMenus(Me.hwnd)

    'set Callback
    OldWindowProc = SetWindowLong(Me.hwnd, _
        GWL_WNDPROC, AddressOf NewWindowProc)

End Sub

Private Sub Form_Unload(Cancel As Integer)

    'do some clean work
    Call OnDestroy

End Sub

```

۴- حذف آیتم (فقره)های منوی سیستم

```

Public Const MF_BYPOSITION = &H400&
Public Declare Function RemoveMenu Lib "user32" _
    (ByVal hMenu As Long, _
    ByVal nPosition As Long, _
    ByVal wFlags As Long) As Long
Public Declare Function GetSystemMenu Lib "user32" _
    (ByVal hWnd As Long, _
    ByVal bRevert As Long) As Long
Public Sub RemoveCloseButton(hWnd As Long)
    Dim hSysMenu As Long
    hSysMenu = GetSystemMenu(hWnd, 0)
    RemoveMenu hSysMenu, 6, MF_BYPOSITION
    RemoveMenu hSysMenu, 5, MF_BYPOSITION
End Sub

```

مثال:

```

Private Sub Form_Load()
    RemoveCloseButton Me.hWND
End Sub

```

۵- اضافه کردن آیتم به منوی سیستم



Option Explicit

```

Declare Function GetMenu Lib "user32" (ByVal hwnd As Long) As Long
Declare Function GetSubMenu Lib "user32" (ByVal hMenu As Long, _
    ByVal nPos As Long) As Long
Declare Function ModifyMenu Lib "user32" Alias "ModifyMenuA" ( _
    ByVal hMenu As Long, ByVal nPosition As Long, _
    ByVal wFlags As Long, ByVal wIDNewItem As Long, _
    ByVal lpString As String) As Long
Declare Function AppendMenu Lib "user32" Alias "AppendMenuA" ( _
    ByVal hMenu As Long, ByVal wFlags As Long, _
    ByVal wIDNewItem As Long, ByVal lpNewItem As String) As Long
Declare Function CreatePopupMenu Lib "user32" () As Long

```

```

Public Const MF_BYPOSITION = &H400&
Public Const MF_POPUP = &H10&
Public Const MF_ENABLED = &H0&
Public Const MF_STRING = &H0&

```

```
Public OldWindowProc As Long
```

```

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

```

```

Public Const GWL_WNDPROC = (-4)
Public Const WM_USER = &H400

```

```
Public MsgNames As New Collection
```

```

Public Const WM_MENUSELECT = &H11F
Public Const WM_COMMAND = &H111

```

```

Declare Function GetSystemMenu Lib "user32" (ByVal hwnd As Long, _
    ByVal bRevert As Long) As Long
Public Const WM_SYSCOMMAND = &H112
Public Const MF_SEPARATOR = &H800&
Public Const IDM_ABOUT = 1999

```

```

' *****
' Pass along all messages except the one that
' makes the context menu appear.
' *****

```

```
Public Function NewWindowProc(ByVal hwnd As Long, _
```

```

ByVal msg As Long, _
ByVal wParam As Long, ByVal lParam As Long) As Long
Const MIN_MENU = 1001
Const MAX_MENU = 1010

If msg = WM_COMMAND Then
    MsgBox "Command ID" & Str$(wParam)
    If wParam >= MIN_MENU And _
        wParam <= MAX_MENU _
        Then Exit Function
End If

If msg = WM_SYSCOMMAND And _
    wParam = IDM_ABOUT _
Then
    MsgBox "Show About dialog here"
    Exit Function
End If

NewWindowProc = CallWindowProc( _
    OldWindowProc, hwnd, msg, wParam, _
    lParam)
End Function

Public Function GetMainMenuStructure(frm As Form) As Integer
'MenuItem
Dim main_handle As Long
Dim submenu As MenuItem
Dim i As Integer
Dim num As Integer

' Get the main menu item.
main_handle = GetMenu(frm.hwnd)

i = 0
Do
    submenu = New MenuItem
    submenu.GetMenuStructure frm.hwnd
    i = i + 1
Loop

' GetMenuItems
'
' Dim hFormMenu As Long
'Dim hSubMenu2 As Long
'Dim hSubMenu2List As Long
'Dim iCounter As Integer
'Dim hSystemMenu As Long
'
' ' Get the handle to the Menu on the
' ' current form
' hFormMenu = GetMenu(Me.hWnd)

End Function

```

MenuItem.cls

```
Option Explicit
```

```

Private Handle As Long
Private Caption As Long

Private Items() As MenuItem
Public Sub GetMenuStructure(New_Handle As Long)
Dim buf As String * 256
Dim length As Integer

    Handle = New_Handle

    length = GetMenuString(Handle, 1, buf, Len(buf), MF_BYPOSITION)
    m_Caption = Left$(buf, length)

End Sub

```

: مثال

Option Explicit

```

Private Sub CreateMenus()
Dim hFormMenu As Long
Dim hSubMenu2 As Long
Dim hSubMenu2List As Long
Dim iCounter As Integer
Dim hSystemMenu As Long

    ' Get the handle to the Menu on the
    ' current form
    hFormMenu = GetMenu(Me.hwnd)

    ' Create a new popup menu to be used to
    ' place the required menu items in
    hSubMenu2List = CreatePopupMenu()

    ' Modify the second menu option on the Form
    ' Menu (denoted by the position 1 -
    ' menu positions start with 0 for the first
    ' item, so 1 specifies the second item).
    ' Specify that the Submenu option will
    ' activate the new popup menu just created
    ' (given the new popup menu handle). Note:
    ' The last string parameter in the function
    ' call specifies the new menu option caption
    ' so if it is to stay the same, make this
    ' string the same as the original caption
    ModifyMenu hFormMenu, 0, MF_BYPOSITION + MF_POPUP, _
        hSubMenu2List, "New Menu Item"

    ' Get the handle of the second menu
    ' option on the Form Menu (denoted by the
    ' position 1)
    hSubMenu2 = GetSubMenu(hFormMenu, 0)

    ' Add new items to the second menu option
    ' list ("Document List")
    For iCounter = 1 To 10
        Call AppendMenu(hSubMenu2, MF_STRING + MF_ENABLED, _
            1000 + iCounter, "Document " & CStr(iCounter))
    Next

```



```

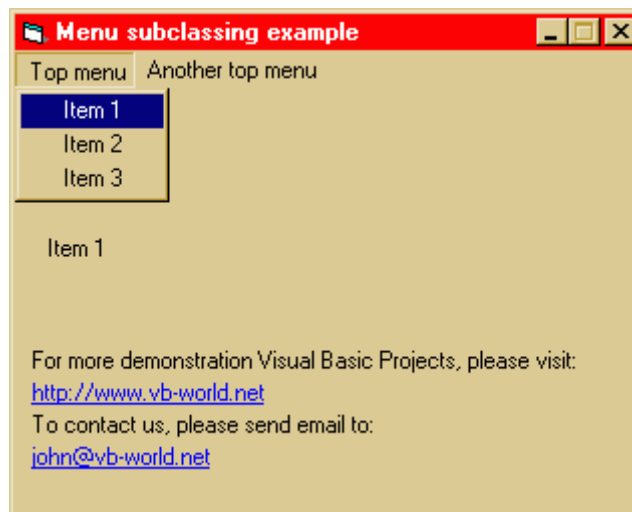
' Add "About" to the system menu.
hSystemMenu = GetSystemMenu(hwnd, False)
AppendMenu hSystemMenu, MF_SEPARATOR, 0, ""
AppendMenu hSystemMenu, MF_STRING + MF_ENABLED, _
            IDM_ABOUT, "About..."
End Sub
Private Sub Form_Load()
    CreateMenus

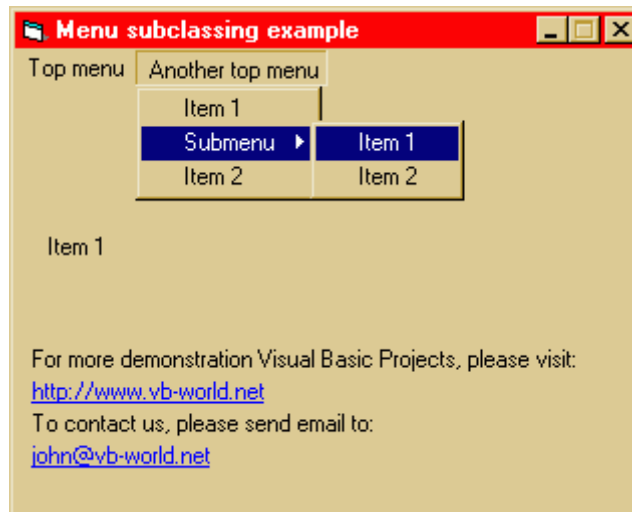
    ' Replace the default WindowProc.
    OldWindowProc = SetWindowLong( _
        hwnd, GWL_WNDPROC, _
        AddressOf NewWindowProc)
End Sub

Private Sub mnuFileExit_Click()
    Unload Me
End Sub

```

۶- شبیه سازی *Menu_MouseOver*





Option Explicit

```
Public Const URL = "http://www.vb-world.net"
Public Const email = "john@vb-world.net"
```

```
Public Declare Function ShellExecute Lib "shell32.dll" Alias
"ShellExecuteA" (ByVal hwnd As Long, ByVal lpOperation As String,
ByVal lpFile As String, ByVal lpParameters As String, ByVal
lpDirectory As String, ByVal nShowCmd As Long) As Long
Public Const SW_SHOWNORMAL = 1
```

```
Public Const MF_BYCOMMAND = &H0&
Public Const MF_BYPOSITION = &H400&
Public Const MF_POPUP = &H10&
Declare Function GetMenuString Lib "user32" Alias "GetMenuStringA"
(ByVal hMenu As Long, ByVal wIDItem As Long, ByVal lpString As
String, ByVal nMaxCount As Long, ByVal wFlag As Long) As Long
```

```
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 GetWindowLong Lib "user32" Alias
"GetWindowLongA" (ByVal hwnd As Long, ByVal nIndex As Long) 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
Declare Function GetSystemMenu Lib "user32" (ByVal hwnd As Long,
ByVal bRevert As Long) As Long
```

```
Private oldwndproc As Long
Private subclassedhwnd As Long
```

```
Public Const WM_MENUSELECT = &H11F
Public Const WM_NCDESTROY = &H82
Public Const GWL_WNDPROC = (-4)
```

```
Public Sub HookWindow(SubClassForm As Form)
```

```
' if something is already subclassed, don't subclass anything else
If oldwndproc <> 0 Then Exit Sub
```

```

subclassedhWnd = SubClassForm.hwnd

'Get the handle for the old window procedure so it can be replaced
and used later
oldwndproc = GetWindowLong(SubClassForm.hwnd, GWL_WNDPROC)

'Install custom window procedure for this window
SetWindowLong SubClassForm.hwnd, GWL_WNDPROC, AddressOf WndProc

End Sub

Private Function WndProc(ByVal hwnd As Long, ByVal Msg As Long, ByVal
wParam As Long, ByVal lParam As Long) As Long
'Does control want this message?
If Msg = WM_MENUSELECT Then

    ' This occurs when the menu is being closed
    If lParam = 0 Then Exit Function

    Dim MenuItemStr As String * 128
    Dim MenuHandle As Integer

    ' Get the low word from wParam: this contains the command ID or
    position of the menu entry
    MenuHandle = GetLowWord(wParam)

    'If the highlighted menu is the top of a popup menu, pass menu item
    by position
    If (GetHighWord(wParam) And MF_POPUP) = MF_POPUP Then

        'Get the caption of the menu item
        If GetMenuString(lParam, MenuHandle, MenuItemStr, 127,
MF_BYPOSITION) = 0 Then Exit Function

    Else    ' Otherwise pass it by command ID

        'Get the caption of the menu item
        If GetMenuString(lParam, MenuHandle, MenuItemStr, 127,
MF_BYCOMMAND) = 0 Then Exit Function

    End If

    ' Add status bar message here!
    frmMenu.lblSelItem = Trim$(MenuItemStr)

Else

    'Otherwise, just call default window handler
    WndProc = CallWindowProc(oldwndproc, hwnd, Msg, wParam, lParam)

End If

'Unhook this window if it is being destroyed
If Msg = WM_NCDESTROY Then
    UnHookWindow
End If
End Function

Public Sub UnHookWindow()
' If there is nothing subclassed, there is nothing to unsubclass!

```

```

If oldwndproc = 0 Then Exit Sub

'Return to default window handler
SetWindowLong subclassedhWnd, GWL_WNDPROC, oldwndproc
oldwndproc = 0
End Sub

Public Function GetLowWord(Word As Long)
GetLowWord = CInt("&H" & Right$(Hex$(Word), 4))
End Function

Public Function GetHighWord(Word As Long)
GetHighWord = CInt("&H" & Left$(Hex$(Word), 4))
End Function

Public Sub gotoweb()
Dim Success As Long

Success = ShellExecute(0&, vbNullString, URL, vbNullString, "C:\",
SW_SHOWNORMAL)

End Sub

Public Sub sendemail()
Dim Success As Long

Success = ShellExecute(0&, vbNullString, "mailto:" & email,
vbNullString, "C:\", SW_SHOWNORMAL)

End Sub

```

مثال :

```

Option Explicit

Private Sub Form_Load()
HookWindow Me
lblemail = email
lblurl = URL
End Sub

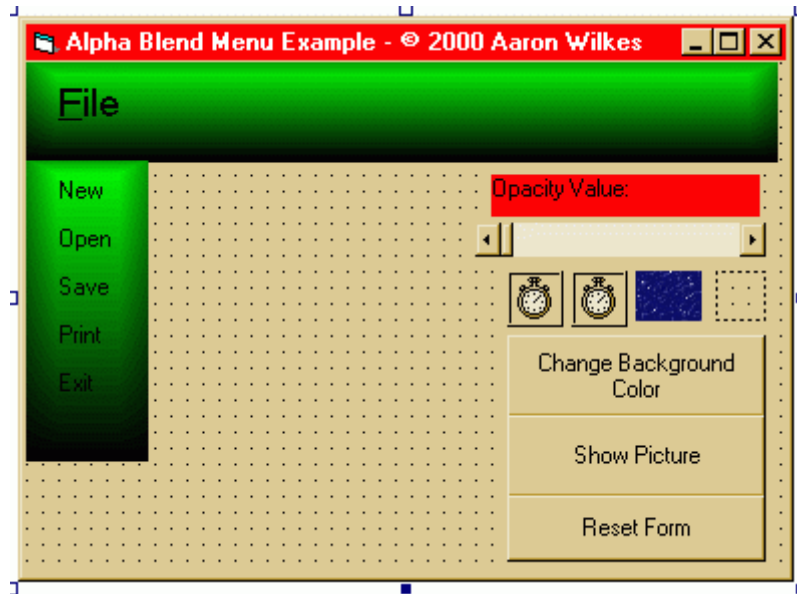
Private Sub lblemail_Click()
sendemail
End Sub

Private Sub lblurl_Click()
gotoweb
End Sub

Private Sub Form_Unload(Cancel As Integer)
UnHookWindow
End Sub

```

۷- ایجاد منوهای شفاف (Transparent)



Option Explicit

```
Private Type BLENDFUNCTION
    BlendOp As Byte
    BlendFlags As Byte
    SourceConstantAlpha As Byte
    AlphaFormat As Byte
End Type
```

```
Const AC_SRC_OVER = &H0
Dim I As Long, A As Long
Dim Opacity As Integer
Dim BF As BLENDFUNCTION, lBF As Long
```

```
Private Declare Function AlphaBlend Lib "msimg32.dll" _
    (ByVal hdc As Long, ByVal lInt As Long, ByVal lInt As Long, _
    ByVal lInt As Long, ByVal lInt As Long, ByVal hdc As Long, _
    ByVal lInt As Long, ByVal lInt As Long, ByVal lInt As Long, _
    ByVal lInt As Long, ByVal BLENDFUNCT As Long) As Long
```

```
Private Declare Sub RtlMoveMemory Lib "kernel32.dll" _
    (Destination As Any, Source As Any, ByVal Length As Long)
```

```
Dim MnuFileOpened As Boolean
```

```
Private Sub Command1_Click()
    If MnuFileOpened Then
        Timer2.Enabled = True
        MnuFileOpened = False
    End If
    Form1.BackColor = vbRed
    Picture1.Cls
    With BF
        .BlendOp = AC_SRC_OVER
        .BlendFlags = 0
        .SourceConstantAlpha = Opacity
        .AlphaFormat = 0
    End With
End Sub
```

```

End With

RtlMoveMemory lBF, BF, 4

AlphaBlend Picture1.hdc, 0, 0, Picture1.ScaleWidth, _
           Picture1.ScaleHeight, Form1.hdc, 0, 0, _
           Form1.ScaleWidth, Form1.ScaleHeight, lBF

Picture1.Refresh
End Sub

Private Sub Command2_Click()
If MnuFileOpened Then
Timer2.Enabled = True
MnuFileOpened = False
End If
Form1.Picture = Image1.Picture
Picture1.Cls
With BF
.BlendOp = AC_SRC_OVER
.BlendFlags = 0
.SourceConstantAlpha = Opacity
.AlphaFormat = 0
End With
RtlMoveMemory lBF, BF, 4
AlphaBlend Picture1.hdc, 0, 0, Picture1.ScaleWidth, _
           Picture1.ScaleHeight, Form1.hdc, 0, 0, _
           Form1.ScaleWidth, Form1.ScaleHeight, lBF

Picture1.Refresh
End Sub

Private Sub Command3_Click()
If MnuFileOpened Then
Timer2.Enabled = True
MnuFileOpened = False
End If
Form1.BackColor = &H8000000F
Form1.Picture = Image2.Picture
Picture1.Cls
With BF
.BlendOp = AC_SRC_OVER
.BlendFlags = 0
.SourceConstantAlpha = Opacity
.AlphaFormat = 0
End With
RtlMoveMemory lBF, BF, 4
AlphaBlend Picture1.hdc, 0, 0, Picture1.ScaleWidth, _
           Picture1.ScaleHeight, Form1.hdc, 0, 0, _
           Form1.ScaleWidth, Form1.ScaleHeight, lBF

Picture1.Refresh
End Sub

Private Sub Form_Click()
If MnuFileOpened Then
Timer2.Enabled = True
MnuFileOpened = False
End If
End Sub

```

```

Private Sub Form_Load()
HScroll11.Value = 128
For I = 0 To MnuLab.Count - 1
MnuLab(I).Tag = MnuLab(I).Top
MnuLab(I).Visible = False
Next I
Picture1.Visible = True
Form1.AutoRedraw = True
Picture1.AutoRedraw = True
Form1.ScaleMode = vbPixels
Picture1.ScaleMode = vbPixels
With BF
.BlendOp = AC_SRC_OVER
.BlendFlags = 0
.SourceConstantAlpha = Opacity
.AlphaFormat = 0
End With
RtlMoveMemory lBF, BF, 4
AlphaBlend Picture1.hdc, 0, 0, Picture1.ScaleWidth, _
, Picture1.ScaleHeight, Form1.hdc, 0, 0, Form1.ScaleWidth, _
Form1.ScaleHeight, lBF

End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, _
X As Single, Y As Single)
Label1.ForeColor = RGB(0, 0, 0)
For A = 0 To MnuLab.Count - 1
MnuLab(A).ForeColor = RGB(0, 0, 0)
Next A
End Sub

Private Sub Form_Unload(Cancel As Integer)
End
End Sub

Private Sub HScroll11_Change()
Label2.Caption = "Opacity Value" & Str$(HScroll11.Value)
Opacity = HScroll11.Value
Picture1.Cls
With BF
.BlendOp = AC_SRC_OVER
.BlendFlags = 0
.SourceConstantAlpha = Opacity
.AlphaFormat = 0
End With
RtlMoveMemory lBF, BF, 4
AlphaBlend Picture1.hdc, 0, 0, Picture1.ScaleWidth, _
Picture1.ScaleHeight, Form1.hdc, 0, 0, Form1.ScaleWidth, _
Form1.ScaleHeight, lBF

Picture1.Refresh
End Sub

Private Sub HScroll11_Scroll()
Label2.Caption = "Opacity Value" & Str$(HScroll11.Value)
Opacity = HScroll11.Value
Picture1.Cls
With BF
.BlendOp = AC_SRC_OVER
.BlendFlags = 0

```

```

.SourceConstantAlpha = Opacity
.AlphaFormat = 0
End With
RtlMoveMemory lBF, BF, 4

AlphaBlend Picture1.hdc, 0, 0, Picture1.ScaleWidth, _
Picture1.ScaleHeight, Form1.hdc, 0, 0, Form1.ScaleWidth, _
Form1.ScaleHeight, lBF

Picture1.Refresh
End Sub

Private Sub Labell_Click()
If Not MnuFileOpened Then
Picture2.Visible = True
Form1.AutoRedraw = True
Picture2.AutoRedraw = True
Form1.ScaleMode = vbPixels
Picture2.ScaleMode = vbPixels
With BF
.BlendOp = AC_SRC_OVER
.BlendFlags = 0
.SourceConstantAlpha = 255
.AlphaFormat = 0
End With
RtlMoveMemory lBF, BF, 4
AlphaBlend Picture2.hdc, 0, 0, Picture2.ScaleWidth, _
Picture2.ScaleHeight, Form1.hdc, 0, 0, _
Form1.ScaleWidth, Form1.ScaleHeight, lBF

Timer1.Enabled = True
MnuFileOpened = True
End If
End Sub

Private Sub Labell_MouseDown(Button As Integer, Shift As Integer, _
X As Single, Y As Single)
Labell.ForeColor = RGB(0, 255, 0)
If Labell.Top < 9 Then
Labell.Top = Labell.Top + 8
End If
End Sub

Private Sub Labell_MouseMove(Button As Integer, Shift As Integer, _
X As Single, Y As Single)
Labell.ForeColor = RGB(0, 255, 0)
For A = 0 To MnuLab.Count - 1
MnuLab(A).ForeColor = RGB(0, 0, 0)
Next A
End Sub

Private Sub Labell_MouseUp(Button As Integer, Shift As Integer, _
X As Single, Y As Single)
Labell.ForeColor = RGB(0, 255, 0)
If Labell.Top > 14 Then
Labell.Top = Labell.Top - 8
End If
End Sub

Private Sub MnuLab_Click(Index As Integer)
Dim Ret As Long

```



```

Select Case Index
Case 0
MsgBox "New", vbOKOnly + vbInformation, "You Chose..."
Case 1
MsgBox "Open", vbOKOnly + vbInformation, "You Chose..."
Case 2
MsgBox "Save", vbOKOnly + vbInformation, "You Chose..."
Case 3
MsgBox "Print", vbOKOnly + vbInformation, "You Chose..."
Case 4
MsgBox "Quit", vbOKOnly + vbInformation, "You Chose..."
Ret = MsgBox("Do you want to quit?", vbYesNo + vbInformation,
"Quit?!?")
If Ret = vbYes Then
Unload Me
End If
End Select
End Sub

Private Sub MnuLab_MouseDown(Index As Integer, Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
If MnuLab(Index).Top < MnuLab(Index).Tag + 1 Then
MnuLab(Index).Top = MnuLab(Index).Top + 4
End If
End Sub

Private Sub MnuLab_MouseMove(Index As Integer, Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
For A = 0 To MnuLab.Count - 1
MnuLab(A).ForeColor = RGB(0, 0, 0)
Next A
MnuLab(Index).ForeColor = RGB(0, 255, 0)
End Sub

Private Sub MnuLab_MouseUp(Index As Integer, Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
If MnuLab(Index).Top > MnuLab(Index).Tag + 1 Then
MnuLab(Index).Top = MnuLab(Index).Top - 4
End If
End Sub

Private Sub Picture1_Click()
If MnuFileOpened Then
Timer2.Enabled = True
MnuFileOpened = False
End If
End Sub

Private Sub Picture1_MouseMove(Button As Integer, Shift As Integer, _
    X As Single, Y As Single)
Label1.ForeColor = RGB(0, 0, 0)
For A = 0 To MnuLab.Count - 1
MnuLab(A).ForeColor = RGB(0, 0, 0)
Next A
End Sub

Private Sub Picture2_MouseMove(Button As Integer, Shift As Integer, _
    X As Single, Y As Single)
For A = 0 To MnuLab.Count - 1
MnuLab(A).ForeColor = RGB(0, 0, 0)
Next A

```

```

End Sub

Private Sub Timer1_Timer()
For I = 255 To Opacity Step -30
If I > Opacity - 1 Then
For A = 0 To MnuLab.Count - 1
MnuLab(A).Visible = True
Next A
End If
Timer1.Enabled = False
Picture2.Cls
With BF
.BlendOp = AC_SRC_OVER
.BlendFlags = 0
.SourceConstantAlpha = I
.AlphaFormat = 0
End With
RtlMoveMemory lBF, BF, 4
AlphaBlend Picture2.hdc, 0, 0, Picture2.ScaleWidth,
Picture2.ScaleHeight, Form1.hdc, 0, 0, Form1.ScaleWidth,
Form1.ScaleHeight, lBF
Picture2.Refresh
Next I
End Sub

Private Sub Timer2_Timer()
For I = Opacity To 255 Step 30
If I > 254 - 30 Then
Picture2.Visible = False
Timer2.Enabled = False
For A = 0 To MnuLab.Count - 1
MnuLab(A).Visible = False
Next A
End If
Picture2.Cls
With BF
.BlendOp = AC_SRC_OVER
.BlendFlags = 0
.SourceConstantAlpha = I
.AlphaFormat = 0
End With
RtlMoveMemory lBF, BF, 4
AlphaBlend Picture2.hdc, 0, 0, Picture2.ScaleWidth, _
Picture2.ScaleHeight, Form1.hdc, 0, 0, _
Form1.ScaleWidth, Form1.ScaleHeight, lBF

Picture2.Refresh
Next I
End Sub

```

۸- ایجاد منو در System Tray ویندوز

```

Public Const WM_LBUTTONDOWN = &H201
' // Tray notification definitions
Global IconIndex As Integer
Global FlashOn As Boolean

Type NOTIFYICONDATA

```

```

        cbSize As Long
        hwnd As Long
        uID As Long
        uFlags As Long
        uCallbackMessage As Long
        hIcon As Long
        szTip As String * 64
End Type

Public Const NIM_ADD = &H0
Public Const NIM_MODIFY = &H1
Public Const NIM_DELETE = &H2

Public Const NIF_MESSAGE = &H1
Public Const NIF_ICON = &H2
Public Const NIF_TIP = &H4

Declare Function Shell_NotifyIcon Lib "shell32.dll" Alias
"Shell_NotifyIconA" (ByVal dwMessage As Long, lpData As
NOTIFYICONDATA) As Long

Type SHFILEINFO
        hIcon As Long           ' out: icon
        iIcon As Long          ' out: icon index
        dwAttributes As Long    ' out: SFGAO_ flags
        szDisplayName As String * 255 'MAX_PATH
' out: display name (or path)
        szTypeName As String * 80      ' out: type name
End Type

Public Const SHGFI_ICON = &H100           ' get icon
Public Const SHGFI_DISPLAYNAME = &H200   ' get display name
Public Const SHGFI_TYPENAME = &H400      ' get type name
Public Const SHGFI_ATTRIBUTES = &H800    ' get attributes
Public Const SHGFI_ICONLOCATION = &H1000  ' get icon location
Public Const SHGFI_EXETYPE = &H2000      ' return exe type
Public Const SHGFI_SYSICONINDEX = &H4000 ' get system icon index
Public Const SHGFI_LINKOVERLAY = &H8000 ' put a link overlay on icon
Public Const SHGFI_SELECTED = &H10000    ' show icon in selected state
Public Const SHGFI_LARGEICON = &H0       ' get large icon
Public Const SHGFI_SMALLICON = &H1       ' get small icon
Public Const SHGFI_OPENICON = &H2        ' get open icon
Public Const SHGFI_SHELLICONSIZE = &H4   ' get shell size icon
Public Const SHGFI_PIDL = &H8            ' pszPath is a pidl
Public Const SHGFI_USEFILEATTRIBUTES = &H10
' use passed dwFileAttribute

Declare Function SHGetFileInfo Lib "shell32.dll" Alias " _
SHGetFileInfoA" (ByVal pszPath As String, _
ByVal dwFileAttributes As Long, psfi As SHFILEINFO, _
ByVal cbFileInfo As Long, ByVal uFlags As Long) As Long

Declare Function SHGetNewLinkInfo Lib "shell32.dll" Alias _
"SHGetNewLinkInfoA" (ByVal pszLinkto As String, _
ByVal pszDir As String, ByVal pszName As String, _
pfMustCopy As Long, ByVal uFlags As Long) As Long

Public Const SHGNLI_PIDL = &H1           ' pszLinkTo is a pidl
Public Const SHGNLI_PREFIXNAME = &H2    ' Make name "Shortcut to xxx"

' // End SHGetFileInfo

```

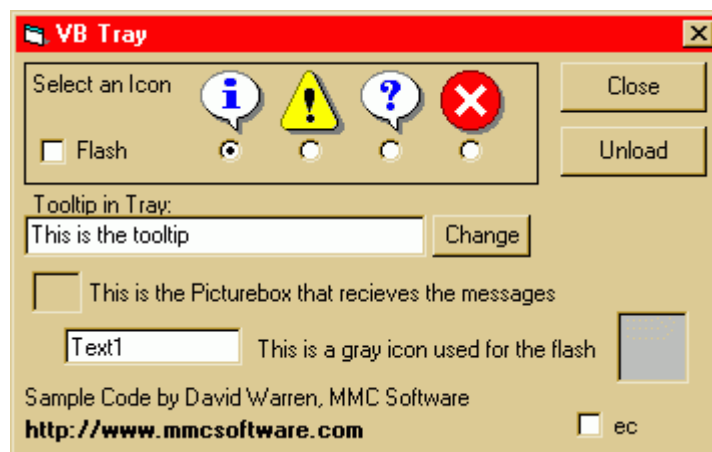
```

'Get the menu handle
Declare Function GetMenu Lib "user32" (ByVal hwnd As Long) As Long
Declare Function GetSubMenu Lib "user32" (ByVal hMenu As Long, _
    ByVal nPos As Long) As Long
'make it pop up
Declare Function TrackPopupMenu Lib "user32" ( _
    ByVal hMenu As Long, ByVal wFlags As Long, _
    ByVal x As Long, ByVal y As Long, _
    ByVal nReserved As Long, ByVal hwnd As Long, lpRect As Any) As Long
' Flags for TrackPopupMenu
Public Const TPM_LEFTBUTTON = &H0&
Public Const TPM_RIGHTBUTTON = &H2&
Public Const TPM_LEFTALIGN = &H0&
Public Const TPM_CENTERALIGN = &H4&
Public Const TPM_RIGHTALIGN = &H8&

Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type
Declare Function FindWindow Lib "user32" Alias "FindWindowA" ( _
    ByVal lpClassName As Any, ByVal lpWindowName As Any) As Long
Declare Function GetWindowRect Lib "user32" (ByVal hwnd As Long, _
    lpRect As RECT) As Long

'Tray messages
Public Const TRAY_MSG_MOUSEMOVE = 7680
Public Const TRAY_MSG_LEFTBTN_DOWN = 7695
Public Const TRAY_MSG_LEFTBTN_UP = 7710
Public Const TRAY_MSG_LEFTBTN_DBLCLICK = 7725
Public Const TRAY_MSG_RIGHTBTN_DOWN = 7740
Public Const TRAY_MSG_RIGHTBTN_UP = 7755
Public Const TRAY_MSG_RIGHTBTN_DBLCLICK = 7770

```



```

Dim IconIndex As Integer, IntDown As Integer
Sub MyPopupmenu(x As Integer, y As Integer)
    hMenu = GetMenu(Form2.hwnd)
    hSubMenu = GetSubMenu(hMenu, 0)
    hTray = FindWindow("Shell_TrayWnd", 0&)
    If hTray > 0 Then

```

```

        Dim TrayRect As RECT
        Result = GetWindowRect(hTray, TrayRect)
        lResult = TrackPopupMenu(hSubMenu, TPM_RIGHTALIGN,
TrayRect.Right, TrayRect.Bottom, 0&, Form2.hwnd, ByVal 0&)
    End If
End Sub

```

```

Private Sub cmdChangeTip_Click()

```

```

    Dim iData As NOTIFYICONDATA
    iData.cbSize = Len(iData)
    iData.hwnd = Picture1.hwnd
    iData.uID = 9999
    iData.uFlags = NIF_TIP
    iData.szTip = (txtTip.Text) & Chr$(0)

    Result = Shell_NotifyIcon(NIM_MODIFY, iData)

```

```

End Sub

```

```

Private Sub cmdClose_Click()

```

```

    Me.Hide

```

```

End Sub

```

```

Private Sub cmdUnload_Click()

```

```

    Unload Me

```

```

End Sub

```

```

Private Sub Form_Load()

```

```

'   VB4 Tray
'   Code sample for Visual Basic 4
'   Copyright © 1996 by David Warren
'

```

```

    Dim iData As NOTIFYICONDATA

    iData.cbSize = Len(iData)
    iData.hwnd = Picture1.hwnd
    iData.uID = 9999
    iData.uFlags = NIF_MESSAGE + NIF_ICON + NIF_TIP
    iData.uCallbackMessage = WM_LBUTTONDOWN
    iData.hIcon = Image1(0).Picture
    iData.szTip = Text2.Text & Chr$(0)

```

```

    Result = Shell_NotifyIcon(NIM_ADD, iData)
    Me.Hide

```

```

End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)

```

```

    Dim iData As NOTIFYICONDATA

```

```

iData.cbSize = Len(iData)
iData.hwnd = Picture1.hwnd
iData.uID = 9999

Result = Shell_NotifyIcon(NIM_DELETE, iData)

End

End Sub

Private Sub Image1_Click(Index As Integer)

    Option1(Index).Value = True

End Sub

Private Sub Option1_Click(Index As Integer)
    Dim iData As NOTIFYICONDATA
    IconIndex = Index
    iData.cbSize = Len(iData)
    iData.hwnd = Picture1.hwnd
    iData.uID = 9999
    iData.uFlags = NIF_ICON
    iData.uCallbackMessage = WM_LBUTTONDOWN
    iData.hIcon = Image1(IconIndex).Picture

    Result = Shell_NotifyIcon(NIM_MODIFY, iData)

End Sub

Private Sub Picture1_MouseDown(Button As Integer, Shift As Integer, x
As Single, y As Single)

    Select Case x
        Case TRAY_MSG_MOUSEMOVE

        Case TRAY_MSG_LEFTBTN_DOWN

        Case TRAY_MSG_LEFTBTN_UP
            MyPopupMenu (x), (y)
        Case TRAY_MSG_LEFTBTN_DBLCLICK
            Me.Show
            Me.SetFocus
        Case TRAY_MSG_RIGHTBTN_DOWN

        Case TRAY_MSG_RIGHTBTN_UP
            MyPopupMenu (x), (y)
        Case TRAY_MSG_RIGHTBTN_DBLCLICK

    End Select

End Sub

Private Sub Timer1_Timer()

```

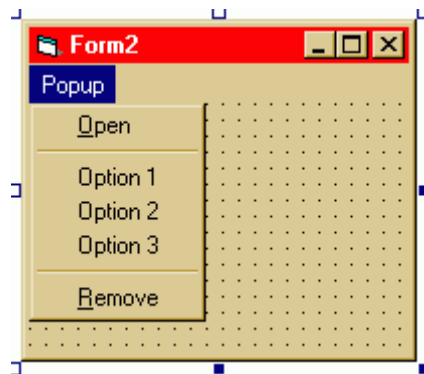
```

Dim iData As NOTIFYICONDATA
If Check1.Value = 1 Then
iData.cbSize = Len(iData)
iData.hwnd = Picture1.hwnd
iData.uID = 9999
If FlashOn Then
FlashOn = False
iData.uFlags = NIF_ICON
iData.hIcon = Image2.Picture
Result = Shell_NotifyIcon(NIM_MODIFY, iData)

Else
iData.uFlags = NIF_ICON
iData.hIcon = Image1(IconIndex).Picture
Result = Shell_NotifyIcon(NIM_MODIFY, iData)
FlashOn = True
End If
Else
If FlashOn = False Then

iData.cbSize = Len(iData)
iData.hwnd = Picture1.hwnd
iData.uID = 9999
iData.uFlags = NIF_ICON
iData.hIcon = Image1(IconIndex).Picture
Result = Shell_NotifyIcon(NIM_MODIFY, iData)
FlashOn = True
End If
End If
End Sub

```



```

Sub GeneralMessage()

MsgBox "Menu functionality is contained in Form2"
End Sub

Private Sub mOpen_Click()
Form1.Show
End Sub

Private Sub mOpt1_Click()
GeneralMessage
End Sub

Private Sub mOpt2_Click()
GeneralMessage

```

```

End Sub

Private Sub mOpt3_Click()
    GeneralMessage
End Sub

Private Sub mRemove_Click()
    Unload Form1
End Sub

```

۹- منوهای جهنده (Pop Up)



```

' Textmenu sample by Matt Hart - vbhelp@matthart.com
' http://matthart.com

Option Explicit

Public Const GWL_WNDPROC = (-4)
Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" _
    (ByVal hwnd As Long, ByVal nIndex As Long, ByVal dwNewLong _
    As Long) As Long
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
Declare Sub CopyMemory Lib "kernel32" Alias "RtlMoveMemory" _
    (hpvDest As Any, hpvSource As Any, ByVal cbCopy As Long)

Public Const WM_MOUSEACTIVATE = &H21
Public Const WM_RBUTTONDOWN = &H204

Public origWndProc As Long

Public Sub SetHook(hwnd, bSet As Boolean)
    If bSet Then
        origWndProc = SetWindowLong(hwnd, GWL_WNDPROC, _
            AddressOf AppWndProc)
    ElseIf origWndProc Then
        Dim lRet As Long
        lRet = SetWindowLong(hwnd, GWL_WNDPROC, origWndProc)
    End If
End Sub

```



```

Public Function AppWndProc(ByVal hwnd As Long, _
    ByVal Msg As Long, ByVal wParam As Long, _
    ByVal lParam As Long) As Long
    Select Case Msg
        Case WM_MOUSEACTIVATE
            Dim C As Integer
            Call CopyMemory(C, ByVal VarPtr(lParam) + 2, 2)
            If C = WM_RBUTTONDOWN Then
                Form1.PopupMenu Form1.mnuBP
                SendKeys "{ESC}"
            End If
        End Select
    AppWndProc = CallWindowProc(origWndProc, hwnd, Msg, _
        wParam, lParam)
End Function

```

مثال :

```

Option Explicit

'
' This sample shows how to subclass a text box and replace its
' default right-click edit menu with one of your own.

Private Sub Form_Load()
    Call SetHook(Text1.hwnd, True)
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, _
    UnloadMode As Integer)
    Call SetHook(Text1.hwnd, False)
End Sub

Private Sub mnuPrint_Click()
    MsgBox "Print!"
End Sub

```

مراجع و منابع :

<http://vbaccelerator.com/>
<http://www.vb-world.net/>
<http://www.vb-helper.com/HowTo/>
<http://matthart.com/>
http://www.domaindlx.com/e_morcillo/
<http://www.acky.net/>
<http://business.vsnl.com/rationalbrains/index.html>