



# Tray Cool!



by Carl Franklin

## Q HAVE IT YOUR WAY

Some apps, such as PowerToys (FSR Monitor and QuickCD) and a mail notification program, run on the tray of Windows 95 in the area where the volume indicator and time are located. How can I create a program with the icon loaded into this area? Is it an API call?

—Matthew A. Griffith, received by CompuServe

**A** Steve Cramp of Dolphin Systems ([www.dolphinsys.com](http://www.dolphinsys.com)) found the answer to this question on the Microsoft Developer Network CD. The Windows 95 Shell library contains a function to quickly and easily add, modify, or delete an icon from the tray. You also can give a tray icon a tool tips-style message to display when the mouse moves over the icon. The code involves the use of a user-defined type called NOTIFYICONDATA and the function Shell\_NotifyIconA in SHELL32.DLL (see Listing 1).

You will probably want to download this code from either the *VBPI* Forum on CompuServe (GO VBPJFO, Magazine Library, QA0396.ZIP) or Carl & Gary's VB Home Page (<http://www.apexsc.com/vb/ftp/misc/q&a0396.zip>) because this sample app has 16 icons that, when animated, look like a spinning globe. The app cycles through each icon, updating both the form icon and the tray. When the app unloads the form, it removes the icon from the tray.

## Q CANCELING THE UNLOAD EVENT

How do I exit a Form\_Unload procedure? I'm trying to make my app look as professional as possible. I have a routine set up that asks the user if he wants to save his work before exiting. I query the user "Save file before exiting?" with three options: Yes, No, or Cancel. If the user decides instead to exit by double-clicking on the top-left form control box, Cancel will not allow me to exit the unload procedure via an Exit Sub statement. Is there an easy solution to this?

*Carl Franklin is a software developer, and the co-owner of Carl & Gary's Visual Basic Home Page (<http://www.apexsc.com/vb>). Contact Carl by e-mail at [carlf@apexsc.com](mailto:carlf@apexsc.com) or through Visual Basic Programmer's Journal. CompuServe users, address your mail to internet:carlf@apexsc.com.*

*This is your forum for addressing the intricacies of the Visual Basic language. Send your questions, clever tips, and techniques. Visual Basic Programmer's Journal will pay \$25 for any submission, tip, or question we print. If your submission includes code, please send a disk along with your hard copy. Mail submissions to Q&A Columnists, c/o Fawcette Technical Publications, 209 Hamilton Avenue, Palo Alto, CA, USA, 94301-2500. CompuServe: 74774.305.*

Second question: is there a way to put text into a MaskedEdit Text control? There is no Text or Caption property for this control, so if I have a default value for this entry box I have to display it outside in a label.

—Zsolt Halmos, Los Gatos, California

### VB4

```

ICON.BAS
'-- Used by Shell_NotifyIconA
Type NOTIFYICONDATA
  cbSize          As Long
  '-- Handle of the window that receives
  ' notification messages
  hWnd           As Long
  '-- App-defined identifier of the
  ' taskbar icon
  uID            As Long
  '-- Flags
  uFlags         As Long
  '-- App-defined message identifier
  uCallbackMessage As Long
  '-- Handle to an icon
  hIcon         As Long
  '-- Tool text display message
  szTip         As String * 64
End Type
Global Const NIM_ADD = 0
Global Const NIM_MODIFY = 1
Global Const NIM_DELETE = 2
Global Const NIF_MESSAGE = 1
Global Const NIF_ICON = 2
Global Const NIF_TIP = 4
Declare Function Shell_NotifyIconA Lib _
  "shell32" (ByVal dwMessage As Long, _
  lpData As NOTIFYICONDATA) As Integer
ICON.FRM
Option Explicit
Dim nPos As Integer
Private Sub Form_Load()
  Dim nd As NOTIFYICONDATA
  Dim nRet As Integer
  '-- Fill the nd structure.
  '-- Size of the structure

```

CONTINUED ON PAGE 138.

**LISTING 1** **Drop Icons in the Tray.** The 32-bit Shell\_NotifyIconA routine lets you add, change, or delete an icon from the Windows 95 Toolbar Tray. To set up, create a form with a Timer control. Add one picture control (Picture2) and 16 other picture controls as an array (Picture1(0) through Picture1(15)). For an icon that is not animated, remove the Timer1\_Timer procedure and the picture controls.



## Q & A

**A** The QueryUnload event provides an opportunity to ask the user for information, such as when closing files and saving data. You can set the Cancel variable to True to avoid the form being unloaded. QueryUnload occurs before the Unload event, so the code in Unload never even executes. The UnloadMode parameter tells you what action is attempting to unload your application, such as the system menu's close command or a normal exit.

Here's how to do this:

```
Sub Form_QueryUnload (UnloadMode As Integer, _
    Cancel As Integer)

    Select Case UnloadMode

        Case vbFormControlMenu          ' 0
            '-- The user has chosen the Close
            ' command from the Control-menu box
            ' on the form.

        Case vbFormCode                 ' 1
            '-- The Unload method has been invoked
            ' from code.

        Case vbAppWindows               ' 2
```

```
            '-- The current Windows-environment
            ' session is ending.
            Cancel = True

        Case vbAppTaskManager          ' 3
            '-- The Microsoft Windows Task Manager
            ' is closing the application.
            Cancel = True

        Case vbFormMDIForm             ' 4
            '-- An MDI child form is closing
            ' because the MDI form is closing.
            Cancel = True

    End Select
End Sub
```

The Masked Edit control has a Text property, but it isn't visible in the property window. It works like the standard text control works, except that according to the manual, when you set it programmatically, the string "must match the characters in the input mask exactly, including literal characters and underscores." In other words, if you use the default social security number mask, "###-##-####," you would have to set the Text to something like "123-12-3213" or some other number that uses the mask format. 

### CONTINUED FROM PAGE 136.

```
nd.cbSize = Len(nd)
'-- The form's hWnd
nd.hWnd = Form1.hWnd
'-- Specify Null for the ID
nd.uID = vbNull
'-- No callback procedure
nd.uCallbackMessage = vbNull
'-- Specify the Icon
nd.hIcon = Form1.Icon
'-- Set the flags to tell Shell that we are
' specifying the CallbackMessage, Icon,
' and Tip
nd.uFlags = NIF_MESSAGE Or NIF_ICON Or NIF_TIP
'-- Add the Icon
nRet = Shell_NotifyIconA(NIM_ADD, nd)
'-- Adjust the size of the form
Width = 3100
Height = 1500
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Dim nd As NOTIFYICONDATA
    Dim nRet As Integer
    '-- Size of the structure
    nd.cbSize = Len(nd)
    '-- The form's hWnd
    nd.hWnd = Form1.hWnd
    '-- Specify Null for the ID
    nd.uID = vbNull
    '-- No callback procedure
    nd.uCallbackMessage = vbNull
    '-- Specify the Icon
    nd.hIcon = Form1.Icon
    '-- Specify the Tip
    nd.szTip = "Buzz around the World, dude." _
        & Chr$(0)
    '-- Set the flags to tell Shell that we are
    ' specifying the CallbackMessage, Icon,
    ' and Tip
    nd.uFlags = NIF_MESSAGE Or NIF_ICON Or _
        NIF_TIP
    '-- Update the Icon
    nRet = Shell_NotifyIconA(NIM_MODIFY, nd)
End Sub
```

```
        NIF_TIP
        '-- Delete the Icon
        nRet = Shell_NotifyIconA(NIM_DELETE, nd)
    End Sub
```

```
Private Sub Timer1_Timer()
    Dim nd As NOTIFYICONDATA
    Dim nRet As Integer
    '-- Cycle the picture and icon
    Form1.Icon = Picture1(nPos).picture
    Picture2.picture = Picture1(nPos).picture
    nPos = nPos + 1
    If nPos = 16 Then
        nPos = 0
    End If
    '-- Size of the structure
    nd.cbSize = Len(nd)
    '-- The form's hWnd
    nd.hWnd = Form1.hWnd
    '-- Specify Null for the ID
    nd.uID = vbNull
    '-- No callback procedure
    nd.uCallbackMessage = vbNull
    '-- Specify the Icon
    nd.hIcon = Form1.Icon
    '-- Specify the Tip
    nd.szTip = "Buzz around the World, dude." _
        & Chr$(0)
    '-- Set the flags to tell Shell that we are
    ' specifying the CallbackMessage, Icon,
    ' and Tip
    nd.uFlags = NIF_MESSAGE Or NIF_ICON Or _
        NIF_TIP
    '-- Update the Icon
    nRet = Shell_NotifyIconA(NIM_MODIFY, nd)
End Sub
```