

frmTrayExample - 1

Dim runthis As Boolean

```
Private Sub ExitCommand_Click()  
    frmTrayExample.WindowState = vbMinimized  
End Sub
```

```
Private Sub Form_Load()  
    runthis = False  
End Sub
```

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, x As Single, Y As Single)  
    If RespondToTray(x) <> 0 Then  
        runthis = True  
        Call ShowFormAgain(Me)  
        frmTrayExample.Show  
        frmTrayExample.WindowState = vbNormal  
        frmTrayExample.Refresh  
    End If  
End Sub
```

```
Private Sub Form_Resize()  
    If Not runthis Then  
        If frmTrayExample.WindowState = vbMinimized Then  
            Call AddToTray(Me.Icon, Me.Caption, Me)  
        End If  
    End If  
    runthis = False  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)  
    Call RemoveFromTray  
End Sub
```

TrayModule - 1

```
'TrayModule Basic Module
'Coded by Michael Barnathan
'E-mail at webmaster@programmerfaq.com
'Declare a user-defined variable to pass to the Shell_NotifyIcon
'function.
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

'Declare the constants for the API function. These constants can be
'found in the header file Shellapi.h.

'The following constants are the messages sent to the
'Shell_NotifyIcon function to add, modify, or delete an icon from the
'taskbar status area.
Global Const NIM_ADD = &H0
Global Const NIM_MODIFY = &H1
Global Const NIM_DELETE = &H2

'The following constant is the message sent when a mouse event occurs
'within the rectangular boundaries of the icon in the taskbar status
'area.
Global Const WM_MOUSEMOVE = &H200

'The following constants are the flags that indicate the valid
'members of the NOTIFYICONDATA data type.
Global Const NIF_MESSAGE = &H1
Global Const NIF_ICON = &H2
Global Const NIF_TIP = &H4

'The following constants are used to determine the mouse input on the
'the icon in the taskbar status area.

'Left-click constants.
Global Const WM_LBUTTONDOWNBLCLK = &H203 'Double-click
Global Const WM_LBUTTONDOWN = &H201 'Button down
Global Const WM_LBUTTONUP = &H202 'Button up

'Right-click constants.
Global Const WM_RBUTTONDOWNBLCLK = &H206 'Double-click
Global Const WM_RBUTTONDOWN = &H204 'Button down
Global Const WM_RBUTTONUP = &H205 'Button up

'Declare the API function call.
Declare Function Shell_NotifyIcon Lib "shell32" _
    Alias "Shell_NotifyIconA" (ByVal dwMessage As Long, pnid As NOTIFYICONDATA) As Boolean

'Dimension a variable as the user-defined data type.
Global nid As NOTIFYICONDATA

Sub AddToTray(TrayIcon, TrayText As String, trayform As Form)
    'Set the individual values of the NOTIFYICONDATA data type.
    nid.cbSize = Len(nid)
    nid.hwnd = trayform.hwnd
    nid.uId = vbNull
```

```

nid.uFlags = NIF_ICON Or NIF_TIP Or NIF_MESSAGE
nid.uCallbackMessage = WM_MOUSEMOVE
nid.hIcon = TrayIcon 'You can replace form1.icon with loadpicture=("icon's file name")
nid.szTip = TrayText & vbNullChar

```

```

'Call the Shell_NotifyIcon function to add the icon to the taskbar
'status area.

```

```

Shell_NotifyIcon NIM_ADD, nid
trayform.Hide

```

```
End Sub
```

```
Sub ModifyTray(TrayIcon, TrayText As String, trayform As Form)
'Set the individual values of the NOTIFYICONDATA data type.

```

```

nid.cbSize = Len(nid)
nid.hwnd = trayform.hwnd
nid.uId = vbNull
nid.uFlags = NIF_ICON Or NIF_TIP Or NIF_MESSAGE
nid.uCallbackMessage = WM_MOUSEMOVE
nid.hIcon = TrayIcon 'You can replace form1.icon with loadpicture=("icon's file name")
nid.szTip = TrayText & vbNullChar

```

```

'Call the Shell_NotifyIcon function to modify the icon to the taskbar
'status area.

```

```

Shell_NotifyIcon NIM_MODIFY, nid

```

```
End Sub
```

```
Sub RemoveFromTray()
```

```

Shell_NotifyIcon NIM_DELETE, nid

```

```
End Sub
```

```
Function RespondToTray(x As Single)
```

```

'Call this sub from the mousemove event on a form
'Event occurs when the mouse pointer is within the rectangular
'boundaries of the icon in the taskbar status area.

```

```
RespondToTray = 0
```

```
Dim msg As Long
```

```
Dim sFilter As String
```

```
If frmTrayExample.ScaleMode <> 3 Then
```

```

    msg = x / Screen.TwipsPerPixelX

```

```
Else
```

```

    msg = x

```

```
End If
```

```
Select Case msg
```

```

    Case WM_LBUTTONDOWN

```

```

    Case WM_LBUTTONUP

```

```

    Case WM_LBUTTONDOWNBLCLK 'Left button double-clicked

```

```

    RespondToTray = 1

```

```

    Case WM_RBUTTONDOWN 'Right button pressed

```

```

    RespondToTray = 2

```

```

    Case WM_RBUTTONUP

```

```

    Case WM_RBUTTONDOWNBLCLK

```

```
End Select
```

```
End Function
```

```
Sub ShowFormAgain(trayform As Form)
```

```

Call RemoveFromTray

```

```

Load trayform

```

```
End Sub
```