



Blog: <http://www.throglolan.blogspot.com>

Autor: Throglolan

Dedicado con todo mi amor a Tuniwiold.

.... Y que viva el guarapo.

TRUCOS EN VISUAL BASIC 6.0

Para la aplicación que personaliza folders y USBs he utilizado algunos recursos que pretendo explicar en este documento. Pueden descargar el programa y su código en mi blog.

PEDIR QUE EL USUARIO ESPECIFIQUE UNA CARPETA

peguen el siguiente código en un módulo .bas, y usen la función SystemCarpetas para que retorne la carpeta, ej:

```
dim a as string
a = SystemCarpetas
msgbox a
```

Option Explicit

Private Type BrowseInfo

```
    hwndOwner      As Long
    pidlRoot        As Long
    pszDisplayName  As Long
    lpszTitle       As Long
    ulFlags         As Long
    lpfnCallback    As Long
    lParam          As Long
    iImage          As Long
```

End Type

Private Type SHITEMID

```
    CB As Long
    abID As Byte
```

End Type

Private Type ITEMIDLIST

```
    mkid As SHITEMID
```

End Type

Private Declare Sub CoTaskMemFree Lib "ole32.dll" (ByVal hMem As Long)

Private Declare Function lstrcat Lib "kernel32" Alias "lstrcatA"

(ByVal lpString1 As String, ByVal lpString2 As String) As Long

Private Declare Function SHBrowseForFolder Lib "shell32" (lpbi As BrowseInfo) As Long

Private Declare Function SHGetPathFromIDList Lib "shell32" (ByVal pidList As Long, ByVal lpBuffer As String) As Long

Private Declare Function SHGetSpecialFolderLocation Lib "shell32.dll" (ByVal hwndOwner As Long, ByVal nFolder As Long, pidl As ITEMIDLIST) As Long

Public Function SystemCarpetas()

```
    Dim iNull As Integer, lpIDList As Long, lResult As Long
```

```
    Dim sPath As String, udtBI As BrowseInfo, idl As ITEMIDLIST
```

```

With udtBI
    .hwndOwner = frmMain.hWnd
    .lpszTitle = lstrcat("Seleccione la carpeta", "")
    .ulFlags = 1
End With
lpIDList = SHBrowseForFolder(udtBI)
If lpIDList Then
    sPath = String(260, 0)
    lResult = SHGetPathFromIDList(lpIDList, sPath)
    CoTaskMemFree lpIDList
    iNull = InStr(sPath, vbNullChar)
    If iNull Then sPath = Left(sPath, iNull - 1)
End If
SystemCarpetas = sPath
End Function

```

MANEJO DE FICHEROS DE CONFIGURACION .INI Y .INF

A continuación publico una función y una rutina que permite manejar valores en archivos de configuración.

Un archivo de configuración se divide en secciones, cada sección tiene una o varias claves a la cual se le asigna un valor, como si fuese una variable. Se ven más o menos así:

```

[.ShellClassInfo]                ---> Seccion .ShellClassInfo
ConfirmFileOp=0                  ---> clave y valor
IconFile=c:\io.ico               ---> clave y valor
IconIndex=0                      ---> clave y valor

[{BE098140-A513-11D0-A3A4-00C04FD706EC}] ---> Seccion {BE098140-A513-11D0-A3A4-00C04FD706EC}

IconArea_Image=c:\io.jpg         ---> clave y valor
IconArea_Text=0xFFffffff         ---> clave y valor

```

En las siguientes funciones los argumentos significan lo siguiente:

- lpFileName: archivo de configuracion (ej: c:\arc.ini)
- lpAppName: sección (ej: .ShellClassInfo)
- lpKeyName: clave (ej: IconFile)
- lpDefault: Si no se encuentra el valor, que valor se devolvera.
- lpString: valor a asignar.

```

Public Function IniGet(ByVal lpFileName As String, ByVal lpAppName As String, ByVal lpKeyName As String, Optional ByVal lpDefault As String = "") As String
    Dim LTmp As Long, sRetVal As String
    sRetVal = String$(255, 0)
    LTmp = GetPrivateProfileString(lpAppName, lpKeyName, lpDefault, sRetVal, Len(sRetVal), lpFileName)
    If LTmp = 0 Then
        IniGet = lpDefault
    Else
        IniGet = Left(sRetVal, LTmp)
    End If

```

```

        End If
    End Function

    Public Sub IniWrite(ByVal lpFileName As String, ByVal lpAppName As
String, ByVal lpKeyName As String, ByVal lpString As String)
        Call WritePrivateProfileString(lpAppName, lpKeyName, lpString,
lpFileName)
    End Sub

```

MODULO DE CLASE PARA PERSONALIZAR CARPETAS Y USB

Este modulo de clase permite cambiar el fondo, color de texto e icono de una carpeta e incluso de una unidad evaluando el tipo de f6lder. Permite hacer los cambios sin pensar en archivos de configuraci6n.

Crean un modulo de clase llamado cStyleFolder, a6aden el c6digo encontrado m6s abajo. Crean una instancia de la clase cStyleFolder en el form load:

```
Public carpeta As New cStyleFolder
```

No olvidemos descargar el objeto al terminar, es decir en el form_unload:

```
set carpeta = nothing
```

Despu6s debemos poner la carpeta que queremos trabajar, y empezamos a asignar propiedades:

```
carpeta.folder = App.Path
carpeta.icono = "c:\icono.ico"
```

Este es el codigo de la clase:

```

Option Explicit

Private Declare Function GetPrivateProfileSectionNames Lib "kernel32"
Alias "GetPrivateProfileSectionNamesA" (ByVal lpszReturnBuffer As
String, ByVal nSize As Long, ByVal lpFileName As String) As Long
Private Declare Function GetPrivateProfileSection Lib "kernel32" Alias
"GetPrivateProfileSectionA" (ByVal lpAppName As String, ByVal
lpReturnedString As String, ByVal nSize As Long, ByVal lpFileName As
String) As Long
Private Declare Function GetPrivateProfileString Lib "kernel32" Alias
"GetPrivateProfileStringA" (ByVal lpApplicationName As String, ByVal
lpKeyName As Any, ByVal lpDefault As String, ByVal lpReturnedString As
String, ByVal nSize As Long, ByVal lpFileName As String) As Long
Private Declare Function WritePrivateProfileString Lib "kernel32"
Alias "WritePrivateProfileStringA" (ByVal lpApplicationName As String,
ByVal lpKeyName As Any, ByVal lpString As Any, ByVal lpFileName As
String) As Long

Private Enum tTipoCarpeta
    raiz
    carpeta
End Enum

```

```

Dim vFolder As String, tipoCarpeta As tTipoCarpeta
'Dim vIcono As String, vFondo As String, vforecolor As String

Public Property Get fondo() As String
    fondo = IniGet(vFolder & "\desktop.ini", "{BE098140-A513-11D0-A3A4-00C04FD706EC}", "IconArea_Image")
End Property

Public Property let fondo(ByVal vNewValue As String)
    IniWrite vFolder & "\desktop.ini", "{BE098140-A513-11D0-A3A4-00C04FD706EC}", "IconArea_Image", vNewValue
End Property

Public Property Get icono() As String
    If tipoCarpeta = raiz Then
        icono = IniGet(vFolder & "\autorun.inf", "AUTORUN", "ICON")
    Else
        icono = IniGet(vFolder & "\desktop.ini", ".ShellClassInfo", "IconFile")
    End If
End Property

Public Property let icono(ByVal vNewValue As String)
    If tipoCarpeta = raiz Then
        IniWrite vFolder & "\autorun.inf", "AUTORUN", "ICON", vNewValue
    Else
        IniWrite vFolder & "\desktop.ini", ".ShellClassInfo", "IconFile", vNewValue
        IniWrite vFolder & "\desktop.ini", ".ShellClassInfo", "IconIndex", "0"
    End If
End Property

Public Property Get foreColor() As String
    foreColor = IniGet(vFolder & "\desktop.ini", "{BE098140-A513-11D0-A3A4-00C04FD706EC}", "IconArea_Text")
End Property

Public Property let foreColor(ByVal vNewValue As String)
    IniWrite vFolder & "\desktop.ini", "{BE098140-A513-11D0-A3A4-00C04FD706EC}", "IconArea_Text", vNewValue
End Property

Public Property Get folder() As Variant
    folder = vFolder
End Property

Public Property let folder(ByVal vNewValue As Variant)
    Dim I As Integer
    vFolder = vNewValue
    For I = 1 To Len(vFolder)
        If Mid(vFolder, I, 1) = "/" Then
            vFolder = Mid(vFolder, 1, I - 1) & "\" & Mid(vFolder, I + 1)
        End If
    Next
    If Right(vFolder, 1) = "\" Then vFolder = Mid(vFolder, 1, Len(vFolder) - 1)
    If Len(vFolder) = 2 Then

```

```

        tipoCarpeta = raiz
    Else
        tipoCarpeta = carpeta
    End If
End Property

' *****
' *****
' *****                                FUNCIONES
' *****
' *****

Public Function IniGet(ByVal lpFileName As String, ByVal lpAppName As
String, ByVal lpKeyName As String, Optional ByVal lpDefault As String
= "") As String
    Dim LTmp As Long, sRetVal As String
    sRetVal = String$(255, 0)
    LTmp = GetPrivateProfileString(lpAppName, lpKeyName, lpDefault,
sRetVal, Len(sRetVal), lpFileName)
    If LTmp = 0 Then
        IniGet = lpDefault
    Else
        IniGet = Left(sRetVal, LTmp)
    End If
End Function

Public Sub IniWrite(ByVal lpFileName As String, ByVal lpAppName As
String, ByVal lpKeyName As String, ByVal lpString As String)
    Call WritePrivateProfileString(lpAppName, lpKeyName, lpString,
lpFileName)
End Sub

```