

Notifylcon

Soumis par Gilles LAURENT

03-12-2008

Dernière mise à jour : 04-12-2008

Afficher un icône dans la zone de notification

Une question de Sylvain (Alias Scarabé) reçue par voie de messagerie il y a quelques jours était de connaître la méthode à utiliser pour afficher un icône dans la zone de notification en VBScript avec le composant COM DynaWrap. Vous trouverez donc ci-dessous, à titre d'exemple, le script Notifylcon.vbs permettant de réaliser cette tâche.

Note: Dans le cadre de ce projet, le composant COM DynaWrap est utilisé pour invoquer les API Win32 suivantes :

- 1- Librairie dynamique User32.dll
 - GetDesktopWindow
 - LoadImage
 - DestroyIcon
- 2- Librairie dynamique Shell32.dll
 - Shell_NotifyIcon
- 3- Librairie dynamique Kernel32.dll
 - GetCurrentProcessId

Historique:

Version 1.1 - 20081204

- possibilité d'exécuter le script de manière simultanée
- un icône est affiché pour chaque instance avec le Pid comme uID
- le Pid est affiché dans le titre de la fenêtre et également dans le Tip de l'icône

Version 1.0 - 20081203

- Première Release Publique

Plateformes supportées:

Windows 2000 et versions supérieures

Listing 1 : Notifylcon.vbs

- Option Explicit
- ' déclaration des constantes
- Const DW_BSTR = 0
- Const DW_LONG = 4
- Const IMAGE_ICON = 1
- Const LR_LOADFROMFILE = 16
- Const NIF_ICON = 2
- Const NIF_TIP = 4
- Const NIM_ADD = 0
- Const NIM_DELETE = 2
- ' déclaration des variables
- Dim oDyn
- Dim hWnd, hIcon
- Dim dwPid, dwNotifyIconDataAddr
- Dim sTip, sNotifyIconData
- ' instanciation du Wrapper DynaWrap (version étendue)
- Set oDyn=CreateObject("DynamicWrapper")
- ' définition des API Win32 utilisées pour ce projet
- oDyn.Register "Shell32.dll", "Shell_NotifyIcon", "i=I", "r=b"
- oDyn.Register "User32.dll", "GetDesktopWindow", "r=h"
- oDyn.Register "User32.dll", "LoadImage", "i=hsIII", "r=h"
- oDyn.Register "User32.dll", "DestroyIcon", "i=h", "r=b"
- oDyn.Register "Kernel32.dll", "GetCurrentProcessId", "r=I"
- ' détermination du Pid du processus courant
- ' le Pid sera l'identificateur de l'icone de la zone de notification

- ' cette solution permet à priori l'exécution d'instances simultanées
- dwPid=oDyn.GetCurrentProcessId()
- ' détermination du handle du bureau
- ' cette fenêtre recevra tous les messages de notification
- hWnd=oDyn.GetDesktopWindow()
- ' chargement de l'icone à partir du fichier NotifyIcon.ico
- ' le fichier icone doit se situer dans le répertoire courant
- hIcon=oDyn.LoadImage(VBNull, "NotifyIcon.ico", IMAGE_ICON, 0, 0, LR_LOADFROMFILE)
- ' définition du texte de l'icone (Tip)
- ' ce texte s'affichera lors du survol de l'icone
- ' le Pid du processus courant sera également affiché
- sTip="Notify Icon From VBScript ! Pid:" & CStr(dwPid)
- ' allocation de la structure NOTIFYICONDATA
- ' cette structure a une taille de 88 bytes soit 44 en unicode
- sNotifyIconData=String(44,0)
- ' renseignement des valeurs de la structure
- ' les valeurs sont injectées dans le buffer à l'offset spécifié
- dwNotifyIconDataAddr=oDyn.GetBSTRAddr(sNotifyIconData)
- oDyn.SetMemInBSTRAddr dwNotifyIconDataAddr, 0, DW_LONG, 88 ' .cbSize
- oDyn.SetMemInBSTRAddr dwNotifyIconDataAddr, 4, DW_LONG, hWnd ' .hWnd
- oDyn.SetMemInBSTRAddr dwNotifyIconDataAddr, 8, DW_LONG, dwPid ' .uID
- oDyn.SetMemInBSTRAddr dwNotifyIconDataAddr, 12, DW_LONG, NIF_ICON Or NIF_TIP ' .uFlags
- oDyn.SetMemInBSTRAddr dwNotifyIconDataAddr, 20, DW_LONG, hIcon ' .hIcon
- oDyn.SetMemInBSTRAddr dwNotifyIconDataAddr, 24, DW_BSTR, sTip ' .szTip
- ' affichage de l'icone dans la zone de notification
- oDyn.Shell_NotifyIcon NIM_ADD, dwNotifyIconDataAddr
- ' simulation du traitement
- ' on se contente ici d'afficher une boite de dialogue
- ' le Pid du processus courant est affiché dans le titre
- MsgBox "Processing, please wait !", 0, sTip
- ' traitement terminé
- ' suppression de l'icone de la zone de notification
- oDyn.Shell_NotifyIcon NIM_DELETE, dwNotifyIconDataAddr
- ' libération des ressources utilisées
- oDyn.DestroyIcon(hIcon)

Téléchargement:

Guide PDF: -

Archive: NotifyIcon.zip