WSH Shell : Afficher les tâches

Soumis par Gilles LAURENT
14-10-2007
Dernière mise à jour : 10-04-2008

WSH Shell : Afficher le titre des fenêtres des tâches en cours d'exécution

Une question d'un usager sur le forum Scripting (Fr) était de savoir comment récupérer par script le titre des fenêtres des tâches en cours d'exécution. Vous pouvez retrouver ce fil ici :

Plusieurs solutions ont été proposées :
- KO: Tasklist /v (seules les premières instances sont affichées)
- OK: AutoIt (via la fonction WinList)
- OK: Python (Via l'import win32gui et l'API Win32 EnumWindows et le support du callback !)
- KO: Composant COM AutoItX (la méthode WinList n'est malheureusement pas implémentée !)
- KO: Composant COM WScript.Application (seules les fenêtres IE et Explorer sont retournées)
- KO: PowerShell (seules les premières instances sont affichées et Explorer est ignoré !)

J'ai donc tenté ma chance en VBScript avec la console WSH Shell. J'ai finalement abouti à la solution proposée ci-dessous qui se compose :
- De la classe VBScript _wshTaskView.inc
Ce module est à déposer dans le sous-dossier Include de la console WSH Shell

- Du composant COM externe dyncall.dll (wrapper Dynawrap)
Ce composant a besoin d'être inscrit dans le registre via la commande regsvr32
Toutefois si la console WSH Shell est démarrée sous l'autorité de l'administrateur local alors il suffit de déposer le composant dans le sous-dossier Com de la console WSH Shell. Dans ce cas l'enregistrement du composant est automatique ;-)

Note : Dans le cadre de ce projet, le composant COM est utilisé pour invoquer les API Win32 suivantes :
- GetTopWindow
- GetWindow
- GetWindowText
- GetWindowThreadProcessId
- IsWindowVisible
- lstrcat
- RtlMoveMemory
- MultiByteToWideCharMicrosoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. Tous droits réservés.

```
 _ _ _  ___  _ _  ___  _       _  _
| | | |/ __>| | |/ __>| |_  ___ | || |
| | | |\__ \|   | \__ \| . |/ ._>| || |
|__/_/ <___/|_|_| <___/|_|_|\___.|_||_|

Windows Script Host (WSH) Shell v1.0.0.5 starting ...

Registering components ...
  Registering dyncall.dll ...

Loading external modules ...
  Loading _wshAdsi.inc ...
  Loading _wshIni.inc ...
  Loading _wshTaskView.inc ...
  Loading _wshWmi.inc ...

Welcome ...
It's 14/10/2007 16:39:44 and WSH Shell is up !

Ready.

WSH D:\Test> ' le composant COM externe dyncall a été enregistré de manière
WSH D:\Test> ' automatique lors du démarrage de la console (cela nécessite que
```

```
WSH D:\Test> ' la console soit démarrée avec les droits appropriés)
WSH D:\Test> ' notre nouveau module est bien présent dans la liste des modules
WSH D:\Test> ' automatiquement chargés. Il ne reste plus qu'à créer une
WSH D:\Test> ' instance pour en bénéficier
WSH D:\Test> Set oTV=New wshTaskView
WSH D:\Test> ' détermination des membres (méthodes et propriétés)
WSH D:\Test> gm(oTV)

Category  Name
--------  ----
Function  GetAllRunningTasks ()
Property  TaskCount
Property  Version

WSH D:\Test> ' combien de tâches sont actuellement en cours d'exécution ?
WSH D:\Test> echo oTV.TaskCount
15
WSH D:\Test> ' affichage formaté des tâches en cours (id, nom et titre)
WSH D:\Test> ' comme le fait si bien le gestionnaire des tâches Windows
WSH D:\Test> ft oTV.GetAllRunningTasks(),"Name","","*"

Id    Name           WindowTitle
--    ----           -----------
3916  cmd.exe        C:\WINDOWS\system32\cmd.exe
2068  cmd.exe        Cmd Running as Administrator
808   explorer.exe   D:\Test
808   explorer.exe   Poste de travail
3396  IEXPLORE.EXE   glsft.free.fr - WSH Shell : Les modules externes - Micr...
3396  IEXPLORE.EXE   Google - Microsoft Internet Explorer
3240  IEXPLORE.EXE   Aide et Support en ligne Microsoft - Microsoft Internet...
2696  msimn.exe      microsoft.public.fr.scripting - Outlook Express
672   notepad.exe    test.cmd - Bloc-notes
1812  notepad.exe    History.log - Bloc-notes
2332  notepad.exe    _wshTaskView.inc - Bloc-notes
3860  powershell.exe Windows PowerShell
148   taskmgr.exe    Gestionnaire des tâches de Windows
3496  vmware.exe     Windows 2000 Professional - VMware Workstation
3884  WSH.exe        Windows Script Host (WSH) Shell

WSH D:\Test> ' NOTE: on remarque que trois instances d'Internet Explorer sont
WSH D:\Test> ' affichées ici avec deux Id dictincts (3396) et (3240)
WSH D:\Test> ' - le premier processus (3396) gère deux instances de fenêtre
WSH D:\Test> ' - le second processus (3240) gère une seule instance de fenêtre
WSH D:\Test> ' NOTE: les instances de l'explorateur de fichiers sont également
WSH D:\Test> ' affichées (808)
WSH D:\Test> ' Enjoy !
WSH D:\Test>
```

Listing 1 : _wshTaskView.inc

```
- '
- '    Windows Script Host (WSH) Shell
- '    (c) 2007 Gilles LAURENT
- '    wshTaskView Class v1.0.0.1
- '
- Option Explicit
- Class WSHTaskView
- ' ========================
- ' == PRIVATE PROPERTIES
- ' ========================
-     Private oFs, oRe, oDyn
- ' ========================
- ' == PUBLIC PROPERTIES
- ' ========================
```

```
-    Public Property Get TaskCount ()
-        TaskCount = UBound (Me.GetAllRunningTasks ())
-    End Property
-    Public Property Get Version ()
-        oRe.Pattern="v(\d\.\d.*)"
-        Version = oRe.Execute (oFs.OpenTextFile ("Include\_wshTaskView.inc", 1, False).ReadAll ())(0).SubMatches (0)
-    End Property
- ' ========================
- ' == PRIVATE MEMBERS
- ' ========================
-    Private Sub Class_Initialize ()
-        ' initialisation des objets
-        Set oDyn = CreateObject ("DynamicWrapper")
-        Set oFs = CreateObject ("Scripting.FileSystemObject")
-        Set oRe = New RegExp
-        ' déclaration des API Win32 (prototypes)
-        oDyn.Register "User32.dll", "GetTopWindow", "f=s", "r=h", "i=h"
-        oDyn.Register "User32.dll", "GetWindow", "f=s", "r=h", "i=hu"
-        oDyn.Register "User32.dll", "GetWindowText", "f=s", "r=l", "i=hll"
-        oDyn.Register "User32.dll", "GetWindowThreadProcessId", "f=s", "r=l", "i=hl"
-        oDyn.Register "User32.dll", "IsWindowVisible", "f=s", "r=t", "i=h"
-        oDyn.Register "kernel32.dll", "lstrcat", "f=s", "r=l", "i=ws"
-        oDyn.Register "kernel32.dll", "RtlMoveMemory", "f=s", "r=l", "i=lll"
-        oDyn.Register "kernel32.dll", "MultiByteToWideChar", "f=s", "r=l", "i=llllll"
-        ' définition des propriétés de l'expression régulière
-        oRe.IgnoreCase = True
-        oRe.Multiline = False
-        oRe.Global = True
-    End Sub
-    Private Sub Class_Terminate ()
-        ' libération des objets
-        Set oFs = Nothing
-        Set oRe = Nothing
-        Set oDyn = Nothing
-    End Sub
-    Private Function LPSTR (str)
-        Dim lpSrc, lpDest
-        lpSrc = oDyn.lstrcat (str, "")
-        lpDest = oDyn.lstrcat (LPSTR, "")
-        LPSTR = CLng (0)
-        oDyn.RtlMoveMemory lpDest+8, lpSrc+8, 4
-    End Function
-    Private Function ReadMemory (lpSrc, nOffset, nSize)
-        Dim lpDest
-        lpDest = oDyn.lstrcat (ReadMemory, "")
-        ReadMemory = CLng (0)
-        oDyn.RtlMoveMemory lpDest+8, lpSrc+nOffset, nSize
-    End Function
- ' ========================
- ' == PUBLIC MEMBERS
- ' ========================
-    Public Function GetAllRunningTasks ()
-        ' déclaration des variables
-        Dim oProc
-        Dim hWnd
-        Dim nLen
-        Dim strTitleBuf, strPid, strTitle
-        Dim arrART(): Redim arrART (0)
-        Dim CP_ACP: CP_ACP = 0
-        ' initialisation des buffers (allocation mémoire)
-        strTitleBuf = String (256, VBNullChar)
-        strPid = String (4, VBNullChar)
-        ' définition du header
-        arrART (0) = "Id" & shell.strTableFieldSep & "Name" & shell.strTableFieldSep & "WindowTitle"
-        ' recherche du handle de la fenêtre racine (top level)
```

```
-       hWnd = oDyn.GetTopWindow (0)
-       ' énumération des fenêtres visibles
-       do
-          If hWnd <> 0 And oDyn.IsWindowVisible (hWnd) Then
-             nLen = oDyn.GetWindowText (hWnd, LPSTR (strTitleBuf), Len (strTitleBuf))
-             If nLen > 0 Then
-                strTitle = String (nLen + 1, VBNullChar)
-                oDyn.MultiByteToWideChar CP_ACP, 0, LPSTR (strTitleBuf), -1, LPSTR (strTitle), Len (strTitle)
-                strTitle = Replace (strTitle, Chr (0), "")
-                If strTitle <> "Program Manager" Then
-                   oDyn.GetWindowThreadProcessId hWnd, LPSTR (strPid)
-                   Set oProc = GetObject ("winmgmts:/root/cimv2:Win32_Process.Handle='" & ReadMemory (LPSTR (strPid), 0, 4
-                   Redim Preserve arrART (UBound (arrART) + 1)
-                   arrART (UBound (arrART)) = oProc.ProcessId & shell.strTableFieldSep & oProc.Name & shell.strTableFieldSep
-                End If
-             End If
-          End If
-          hWnd = oDyn.GetWindow (hWnd, 2)
-       Loop Until hWnd = 0
-       GetAllRunningTasks = arrART
-    End Function
- End Class
```