evtTriggers

Soumis par Gilles LAURENT
03-06-2007
Dernière mise à jour : 03-06-2007

evtTriggers

Création de déclencheurs d'évènements permanents via WMI.
Ce script permet de configurer en ligne de commandes l'exécution d'une tâche (script, binaire, etc ...) lorsqu'un évènement / une source / un eventID particulier s'enregistre dans le journal d'évènements (eventlog) de la machine locale.

Note: Ce script reproduit le fonctionnement de l'outil EventTriggers.exe disponible sur les plateformes XP/2003. Un dysfonctionnement du binaire sur certaines plateformes XP SP2 Fr m'a conduit, lors d'une contribution sur un forum Microsoft Server, à écrire ce script pour pallier (à) ce problème ;-)

Petit extrait de la discussion :

<http://groups.google.fr/group/microsoft.public.fr.exchange/msg/1436a71f34cf4774?&hl=fr>

Historique:

Version 1.0 - Première release publique

Plateformes supportées:

Windows XP/2003

Pré requis:

Aucun

Usage:

\> cscript evtTriggers.vbs (pour obtenir la syntaxe embarquée)

Code source VBScript:

```

1 Option Explicit
2
3 ' =====
4 ' Définition des constantes
5 ' =====
6 Const NAMESPACE = "\\.\root\subscription"
7 Const PolingInterval = "10"
8
9 ' =====
10 ' Définition des variables globales
11 ' =====
12 Dim colNamedArguments
13 Dim objSWbemServices
14 Dim objFileSystem
15 Dim WQL
16 Dim EventName
17 Dim Filter
18 Dim Consumer
19 Dim Binding
20
21 ' =====
22 ' Procédure d'affichage de la syntaxe
23 ' =====
24 Sub Usage ()
25
26 WScript.Echo ""
27 WScript.Echo "EvtTriggers /paramètre:valeur [/paramètres:valeur ...]"
28 WScript.Echo ""
29 WScript.Echo "Description : "
30 WScript.Echo ""
31 WScript.Echo " Cet outil en ligne de commandes permet à un administrateur d'afficher et"
32 WScript.Echo " configurer les déclencheurs d'évènements WMI sur des ordinateurs locaux."

```

```

33 WScript.Echo ""
34 WScript.Echo "Liste des paramètres :"
35 WScript.Echo ""
36 WScript.Echo " /O:create    Crée ou met à jour un nouveau déclencheur d'évènement qui"
37 WScript.Echo "                analysera et agira sur les occurrences des évènements du"
38 WScript.Echo "                journal NT d'un critère donné."
39 WScript.Echo ""
40 WScript.Echo "                Liste des paramètres :"
41 WScript.Echo ""
42 WScript.Echo "                /TR:nom_du_déclencheur          - Obligatoire"
43 WScript.Echo "                /TK:chemin_complet_de_la_tâches    - Obligatoire"
44 WScript.Echo "                /L:journal_à_analyser (tous par défaut) - Optionnel"
45 WScript.Echo "                /EID:id_de_l'évènement            - Optionnel"
46 WScript.Echo "                /T:type_de_l'évènement (1,2,3,4 ou 5) - Optionnel"
47 WScript.Echo "                1- Erreur 2- Avertissement 3- Informations"
48 WScript.Echo "                4- Audit Succès 5- Audit Erroné"
49 WScript.Echo "                /SO:source_de_l'évènement          - Optionnel"
50 WScript.Echo ""
51 WScript.Echo " /O:query    Demande et affiche les propriétés et les paramètres du"
52 WScript.Echo "            déclencheur d'évènements."
53 WScript.Echo ""
54 WScript.Echo " /O:delete    Supprime un déclencheur d'évènement par l'ID du déclencheur."
55 WScript.Echo ""
56 WScript.Echo "            Liste des paramètres :"
57 WScript.Echo ""
58 WScript.Echo "            /TID:ID_du_déclencheur                - Obligatoire"
59 WScript.Echo ""
60 WScript.Echo ""
61
62 End Sub
63
64 ' =====
65 ' Fonction de suppression d'un évènement WMI
66 ' =====
67 Function DeleteWMIEvent ()
68 Dim TID, Instances, Instance, Index, TaskNumber
69
70 ' =====
71 ' Lecture des arguments
72 ' =====
73 TID = CInt (colNamedArguments.Item("TID"))
74
75 ' =====
76 ' Lecture des instances WMI pour les liaisons
77 ' =====
78 Set Instances = GetObject("WINMGMTS:" & NAMESPACE & ":_FilterToConsumerBinding").instances_()
79
80 ' =====
81 ' Enumération des liaisons
82 ' =====
83 For Each Instance in Instances
84
85 ' Seules les liaisons EvtTriggers sont supprimée
86 Index = InStr(Instance.Consumer, "=" & "_" )
87 If (Index <> 0) Then
88
89 ' Incrémentation du TID
90 TaskNumber = TaskNumber + 1
91
92 ' suppression de la tâche si le TID correnpond à celui spécifié
93 If (TaskNumber = TID) Then
94
95 ' Formatage du nom de la tâche
96 EventName = Mid(Instance.Consumer, Index + 3, Len (Instance.Consumer) - Index - 3)
97

```

```

98 ' Suppression de l'instance
99 Instance.delete_()
100 End If
101 End If
102 Next
103
104 ' =====
105 ' Lecture des instances WMI pour les filtres
106 ' =====
107 Set Instances = GetObject("WINMGMTS:" & NAMESPACE & ".__EventFilter").instances_()
108
109 ' =====
110 ' Enumération des filtres
111 ' =====
112 For Each Instance in Instances
113
114 If (Instance.Name = EventName) Then
115
116 ' Suppression de l'instance
117 Instance.delete_()
118 End If
119 Next
120
121 ' =====
122 ' Lecture des instances WMI pour les Consumers
123 ' =====
124 Set Instances = GetObject("WINMGMTS:" & NAMESPACE & ":CommandLineEventConsumer").instances_()
125
126 ' =====
127 ' Enumération des Consumers
128 ' =====
129 For Each Instance in Instances
130
131 If (Instance.Name = "_" & EventName) Then
132
133 ' Suppression de l'instance
134 Instance.delete_()
135 End If
136 Next
137
138 if (EventName <> "") Then WScript.Echo "Tâche [" & EventName & "] *SUPPRIMEE*"
139
140 ' =====
141 ' Traitement terminé
142 ' =====
143 WScript.Echo ""
144 WScript.Echo "--"
145 WScript.Echo "Opération achevée avec succès."
146
147 End Function
148
149 ' =====
150 ' Fonction d'interrogation des évènements WMI
151 ' =====
152 Function QueryWMIEvents ()
153 Dim Bind, Binds, Consumers, Index, TaskNumber
154
155 ' =====
156 ' Connexion à l'espace de nom WMI
157 ' =====
158 Set Binds = GetObject("WINMGMTS:" & NAMESPACE & ".__FilterToConsumerBinding").instances_()
159
160 ' =====
161 ' Enumération des liaisons filter / consumer
162 ' =====

```

```

163 For Each Bind in Binds
164
165 ' Seules les liaisons EvtTriggers sont affichées
166 Index = InStr(Bind.Consumer, "=" & "_")
167 If (Index <> 0) Then
168
169 ' Incrémentation du TID
170 TaskNumber = TaskNumber + 1
171
172 ' Formatage du nom de la tâche
173 EventName = Mid(Bind.Consumer, Index + 3, Len (Bind.Consumer) - Index - 3)
174
175 ' =====
176 ' Affichage des informations sur la tâche
177 ' =====
178 WScript.Echo "-----"
179 WScript.Echo "TID   : " & TaskNumber
180 WScript.Echo "Name   : " & EventName
181
182 ' Lecture de la ligne de commandes associée à la tâche
183 Set Consumers = GetObject("WINMGMTS:" & NAMESPACE & ":CommandLineEventConsumer").instances_()
184
185 ' =====
186 ' Enumération des Consumers
187 ' =====
188 For Each Consumer in Consumers
189
190 ' affichage de la ligne de commande associée à la tâche
191 If (Consumer.Name = "_" & EventName) Then
192     WScript.Echo "CmdLine : " & Consumer.CommandLineTemplate
193 End If
194 Next
195 End If
196 Next
197
198 ' =====
199 ' Traitement terminé
200 ' =====
201 WScript.Echo ""
202 WScript.Echo "--"
203 WScript.Echo "Opération achevée avec succès."
204
205 End Function
206
207 ' =====
208 ' Fonction de création d'un évènement WMI
209 ' =====
210 Function CreateWMIEvent ()
211 Dim TaskName
212 Dim Logfile
213 Dim EventType
214 Dim SourceName
215 Dim EventCode
216
217 ' =====
218 ' Lecture des arguments
219 ' =====
220 EventName = colNamedArguments.Item("TR")
221 TaskName = colNamedArguments.Item("TK")
222 LogFile = colNamedArguments.Item("L")
223 EventType = colNamedArguments.Item("T")
224 SourceName = colNamedArguments.Item("SO")
225 EventCode = colNamedArguments.Item("EID")
226
227 ' =====

```

```

228 ' Connexion à l'espace de nom WMI
229 ' =====
230 Set objSWbemServices = GetObject("winmgmts:{impersonationLevel=impersonate,(security)}" & NAMESPACE)
231
232 ' =====
233 ' Chargement de la classe pour l'accès au Filesystem
234 ' =====
235 Set objFileSystem = CreateObject("Scripting.FileSystemObject")
236
237 ' =====
238 ' Préparation de la requête WQL
239 ' =====
240 WQL = "SELECT * FROM __InstanceCreationEvent WITHIN " & PolingInterval & " " & "WHERE TargetInstance ISA
'Win32_NTLogEvent"
241
242 ' Adaptation de la requête WQL en fonction des arguments
243 If Logfile <> "" Then WQL = WQL & " AND TargetInstance.Logfile = " & Logfile & ""
244 If EventType <> "" Then WQL = WQL & " AND TargetInstance.EventType = " & EventType
245 If SourceName <> "" Then WQL = WQL & " AND TargetInstance.SourceName = " & SourceName & ""
246 If EventCode <> "" Then WQL = WQL & " AND TargetInstance.EventCode = " & EventCode
247
248 ' =====
249 ' Instanciation du filtre
250 ' =====
251 Set Filter = objSWbemServices.Get("__EventFilter").SpawnInstance_()
252 With Filter
253 .Name = EventName
254 .EventNamespace = "root\cimv2"
255 .QueryLanguage = "WQL"
256 .Query = WQL
257 End With
258
259 ' création de l'instance
260 Filter.Put_()
261
262 ' =====
263 ' Instanciation du consumer
264 ' =====
265 Set Consumer = objSWbemServices.Get("CommandLineEventConsumer").SpawnInstance_()
266 With Consumer
267 .Name = "_" & EventName
268 .CommandLineTemplate = TaskName
269 .RunInteractively = False
270 .WorkingDirectory = objFileSystem.GetParentFolderName(TaskName)
271 End With
272
273 ' création de l'instance
274 Consumer.Put_()
275
276 ' =====
277 ' Création de la liaison filtre / consumer
278 ' =====
279 Set Binding = objSWbemServices.Get("__FilterToConsumerBinding").SpawnInstance_()
280 With Binding
281 .Filter = Filter.Path_.RelPath
282 .Consumer = Consumer.Path_.RelPath
283 End With
284
285 ' Création de la liaison
286 Binding.Put_()
287
288 if (TaskName <> "") Then WScript.Echo "Tâche [" & EventName & "] *CREEE*"
289
290 ' =====
291 ' Traitement terminé

```

```
292 ' =====
293 WScript.Echo ""
294 WScript.Echo "--"
295 WScript.Echo "Opération achevée avec succès."
296
297 End Function
298
299 ' =====
300 ' POINT D'ENTREE : Lecture des arguments
301 ' =====
302 Set colNamedArguments = WScript.Arguments.Named
303
304 ' =====
305 ' Lecture de l'ordre (create / delete / query)
306 ' =====
307 Select Case LCase (colNamedArguments.Item("O"))
308
309 ' =====
310 ' Création d'un évènement WMI
311 ' =====
312 Case "create"
313 CreateWMIEvent ()
314
315 Case "delete"
316 DeleteWMIEvent ()
317
318 Case "query"
319 QueryWMIEvents ()
320
321 Case Else Usage ()
322
323 End Select
```

Téléchargement:

Guide PDF: -

Archive: EvtTriggers.zip