

Soumis par Gilles LAURENT  
15-04-2008  
Dernière mise à jour : 16-04-2008

## WSH Shell : Enumérer les options régionales et linguistiques

Dans le cadre d'un projet initié par un client, j'ai été sollicité, entre autre, pour concevoir un outil en ligne de commande permettant d'une part de déterminer les différentes locales installées sur des systèmes d'exploitation Windows 2000, XP et Vista et d'autre part de sauvegarder quelques options régionales et linguistiques dans un fichier de type XML . Sans outil natif à disposition (à ma connaissance tout du moins !), j'ai naturellement lancé mon compilateur MSVC6.0 (eh eh oui ! toujours ma superbe plateforme virtuelle de dev !) et écrit rapidement un bout de code C utilisant les API Win32 EnumSystemLocales et GetLocaleInfo. Par curiosité, sur l'heure du déjeuner avec un sandwich Tomates-Mozzarella dans la main, j'ai jeté un oeil du coté des classes du Framework .Net sur mon portable XP SP2 sur lequel Windows PowerShell est installé. Il s'avère que la classe .Net CultureInfo possède tous les membres (méthodes et propriétés) nécessaires au besoin. Ci-dessous un exemple one-liner<sup>1</sup> en PowerShell avec la méthode statique GetCultures :  
PS D:\Test> [Globalization.CultureInfo]::GetCultures("InstalledWin32Cultures") |  
sort DisplayName | select -first 5

| LCID | Name       | DisplayName                  |
|------|------------|------------------------------|
| 1078 | af-ZA      | Afrikaans (South Africa)     |
| 1052 | sq-AL      | Albanian (Albania)           |
| 2092 | az-AZ-Cyrl | Azeri (Cyrillic, Azerbaijan) |
| 1068 | az-AZ-Latn | Azeri (Latin, Azerbaijan)    |
| 1069 | eu-ES      | Basque (Basque)              |

one-liner<sup>1</sup> : Désigne un traitement basé sur un code d'une ligne unique. C'est fréquent sous Perl et maintenant également sous PowerShell :-)

Note : Et oui ! Toute la puissance du Framework .Net en ligne de commandes ou scripting grâce à Windows PowerShell ! Pour faire connaissance avec ce langage dans les meilleures conditions possibles, je ne peux que vous recommander le livre "Windows PowerShell : Guide de référence pour l'administration système" aux éditions ENI écrit par Arnaud PETITJEAN, Microsoft MVP et Robin LEMESLE. Un Must-have pour les francophones souhaitant s'initier et/ou approfondir leurs connaissances de ce nouveau langage de scripting et ligne de commandes devenu maintenant incontournable dans l'environnement Microsoft ! Vous pouvez retrouver ces deux individus sur le site communautaire PowerShell-Scripting.com.

Bon ! Nous en étions aux locales ;-) J'ai donc ensuite essayé de réaliser ces mêmes opérations en VBScript en m'appuyant sur la console WSH Shell. J'ai finalement abouti à la solution ci-dessous qui se compose :

- De la classe VBScript\_wshCultures.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 :

- IsValidLocale

- GetLocaleInfoMicrosoft (R) Windows Script Host Version 5.6

Copyright (C) Microsoft Corporation 1996-2001. Tous droits réservés.

[illegible]

Windows Script Host (WSH) Shell v1.0.0.8 starting ...

## Registering components ...

Registering dyncall.dll ...

Loading external modules ...

Loading \_wshAdsi.inc ...

Loading \_wshCalendar.inc ...

Loading \_wshCodeSyntaxing.inc ...

Loading \_wshCultures.inc ...

Loading \_wshIni.inc ...

Loading \_wshWmi.inc ...

Loading profiles ...

Loading D:\Users\Dev\Copyright\WSH\Release\WSHShell\_Profile.inc ...

Loading D:\Documents and Settings\Gilles\Mes documents\WSH Shell\WSHShell\_P...

Welcome ...

It's 15/04/2008 20:26:44 and WSH Shell is up !

Ready.

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 oCult=New wshCultures

WSH D:\Test>

WSH D:\Test> ' détermination des membres (méthodes et propriétés)

WSH D:\Test> gm(oCult)

Category Name

-----

Function EnumSystemLocales ()

Function GetLocaleInfo (LCID, LCType)

Property InstalledCultureCount

Property Version

WSH D:\Test> ' combien de locales sont installées sur ce système ?

WSH D:\Test> echo oCult.InstalledCultureCount

117

WSH D:\Test> ' affichage formaté des locales installées sur ce système

WSH D:\Test> ' c'est ici un Windows XP Pro SP2 up-to-date :-)

WSH D:\Test> ft oCult.EnumSystemLocales (), "Language", "", "x"

LCID ISO Language

---- ---

1078 af-ZA Afrikaans

1052 sq-AL Albanais

1031 de-DE Allemand (Allemagne)

3079 de-AT Allemand (Autriche)

5127 de-LI Allemand (Liechtenstein)

4103 de-LU Allemand (Luxembourg)

2055 de-CH Allemand (Suisse)

7177 en-ZA Anglais (Afrique du Sud)

3081 en-AU Anglais (Australie)

10249 en-BZ Anglais (Belize)

4105 en-CA Anglais (Canada)

9225 en-CB Anglais (Caraïbes)

1033 en-US Anglais (États-Unis)

11273 en-TT Anglais (Île de la Trinité)

6153 en-IE Anglais (Irlande)

8201 en-JM Anglais (Jamaïque)

5129 en-NZ Anglais (Nouvelle-Zélande)

13321 en-PH Anglais (Philippines)

2057 en-GB Anglais (Royaume-Uni)

12297 en-ZW Anglais (Zimbabwe)

2092 az-AZ-Cyrl Azéri (cyrillique)

1068 az-AZ-Latn Azéri (latin)

|       |            |                                            |
|-------|------------|--------------------------------------------|
| 1069  | eu-ES      | Basque                                     |
| 1059  | be-BY      | Biélorusse                                 |
| 5146  | bs-BA      | Bosniaque                                  |
| 1026  | bg-BG      | Bulgare                                    |
| 1027  | ca-ES      | Catalan                                    |
| 1050  | hr-HR      | Croate                                     |
| 4122  | hr-BA      | Croate (Bosnie-Herzégovine)                |
| 1030  | da-DK      | Danois                                     |
| 3082  | es-ES      | Espagnol (alphabet international)          |
| 11274 | es-AR      | Espagnol (Argentine)                       |
| 16394 | es-BO      | Espagnol (Bolivie)                         |
| 13322 | es-CL      | Espagnol (Chili)                           |
| 9226  | es-CO      | Espagnol (Colombie)                        |
| 5130  | es-CR      | Espagnol (Costa Rica)                      |
| 12298 | es-EC      | Espagnol (Équateur)                        |
| 4106  | es-GT      | Espagnol (Guatemala)                       |
| 18442 | es-HN      | Espagnol (Honduras)                        |
| 17418 | es-SV      | Espagnol (Le Salvador)                     |
| 2058  | es-MX      | Espagnol (Mexique)                         |
| 19466 | es-NI      | Espagnol (Nicaragua)                       |
| 6154  | es-PA      | Espagnol (Panama)                          |
| 15370 | es-PY      | Espagnol (Paraguay)                        |
| 10250 | es-PE      | Espagnol (Pérou)                           |
| 20490 | es-PR      | Espagnol (Porto Rico)                      |
| 7178  | es-DO      | Espagnol (République dominicaine)          |
| 1034  | es-ES      | Espagnol (traditionnel)                    |
| 14346 | es-UY      | Espagnol (Uruguay)                         |
| 8202  | es-VE      | Espagnol (Venezuela)                       |
| 1061  | et-EE      | Estonien                                   |
| 1080  | fo-FO      | Féroïen                                    |
| 1035  | fi-FI      | Finnois                                    |
| 2060  | fr-BE      | Français (Belgique)                        |
| 3084  | fr-CA      | Français (Canada)                          |
| 1036  | fr-FR      | Français (France)                          |
| 5132  | fr-LU      | Français (Luxembourg)                      |
| 6156  | fr-MC      | Français (Monaco)                          |
| 4108  | fr-CH      | Français (Suisse)                          |
| 1110  | gl-ES      | Galicien                                   |
| 1106  | cy-GB      | Gallois                                    |
| 1032  | el-GR      | Grec                                       |
| 1038  | hu-HU      | Hongrois                                   |
| 1057  | id-ID      | Indonésien                                 |
| 1039  | is-IS      | Islandais                                  |
| 1040  | it-IT      | Italien (Italie)                           |
| 2064  | it-CH      | Italien (Suisse)                           |
| 1087  | kk-KZ      | Kazakh                                     |
| 1088  | ky-KG      | Kyrgyz (cyrillique)                        |
| 1062  | lv-LV      | Letton                                     |
| 1063  | lt-LT      | Lituanien                                  |
| 1071  | mk-MK      | Macédonien Ex Rép. yougoslave de Macédoine |
| 2110  | ms-BN      | Malais (Brunei Darussalam)                 |
| 1086  | ms-MY      | Malais (Malaisie)                          |
| 1082  | mt-MT      | Maltais                                    |
| 1153  | mi-NZ      | Maori                                      |
| 1104  | mn-MN      | Mongol (cyrillique)                        |
| 2067  | nl-BE      | Néerlandais (Belgique)                     |
| 1043  | nl-NL      | Néerlandais (Pays-Bas)                     |
| 1044  | nb-NO      | Norvégien (Bokmål)                         |
| 2068  | nn-NO      | Norvégien (Nynorsk)                        |
| 2115  | uz-UZ-Cyrl | Ouzbek (cyrillique)                        |
| 1091  | uz-UZ-Latn | Ouzbek (latin)                             |
| 1045  | pl-PL      | Polonais                                   |
| 1046  | pt-BR      | Portugais (Brésil)                         |
| 2070  | pt-PT      | Portugais (Portugal)                       |
| 1131  | qu-BO      | Quechua (Bolivie)                          |

```

2155 qu-EC    Quechua (Équateur)
3179 qu-PE    Quechua (Pérou)
1048 ro-RO    Roumain
1049 ru-RU    Russe
4155 se-NO    Same de Lule (Norvège)
5179 se-SE    Same de Lule (Suède)
8251 se-FI    Same de Skolt (Finlande)
9275 se-FI    Same d'Inari (Finlande)
3131 se-FI    Same du nord (Finlande)
1083 se-NO    Same du nord (Norvège)
2107 se-SE    Same du nord (Suède)
6203 se-NO    Same du sud (Norvège)
7227 se-SE    Same du sud (Suède)
3098 sr-SP-Cyrl Serbe (cyrillique)
7194 sr-BA-Cyrl Serbe (cyrillique, Bosnie-Herzégovine)
2074 sr-SP-Latn Serbe (latin)
6170 sr-BA-Latn Serbe (latin, Bosnie-Herzégovine)
1051 sk-SK    Slovaque
1060 sl-SI    Slovène
1132 ns-ZA    Sotho du nord
1053 sv-SE    Suédois
2077 sv-FI    Suédois (Finlande)
1089 sw-KE    Swahili
1092 tt-RU    Tatar
1029 cs-CZ    Tchèque
1074 tn-ZA    Tswana
1055 tr-TR    Turc
1058 uk-UA    Ukrainien
1076 xh-ZA    Xhosa
1077 zu-ZA    Zoulou

```

```

WSH D:\Test> ' il est également possible d'obtenir des informations pour une
WSH D:\Test> ' langue données. Par exemple détermination du nom de la monnaie
WSH D:\Test> ' au Royaume-Uni (2057)
WSH D:\Test> echo oCult.GetLocaleInfo (2057, &H1007)
UK Pound Sterling
WSH D:\Test>
WSH D:\Test> ' Note: La valeur &H1007 correspond à la constante du fichier
WSH D:\Test> ' include WinNls.h LOCALE_SENGCURRNAME qui spécifie le nom de la
WSH D:\Test> ' monnaie en Anglais
WSH D:\Test>
WSH D:\Test> ' Enjoy !
WSH D:\Test>

```

Note : Les constantes supportées par la méthode GetLocaleInfo sont disponibles sur MSDN

Listing 1 : \_wshCultures.inc

```

_ '
_ '  Windows Script Host (WSH) Shell
_ '  (c) 2008 Gilles LAURENT
_ '  wshCultures Class v1.0.0.1
_ '

_ Option Explicit
_ Class WSHCultures
_ ' =====
_ ' == PRIVATE PROPERTIES
_ ' =====
_ Private oFs, oRe, oDyn
_ ' =====
_ ' == PUBLIC PROPERTIES
_ ' =====
_ Public Property Get InstalledCultureCount ()
_ InstalledCultureCount = UBound (Me.EnumSystemLocales ())

```

```

- End Property
- Public Property Get Version ()
-     oRe.Pattern="v\d\\.d.*)"
-     Version = oRe.Execute (oFs.OpenTextFile (shell.Path & "\\Include_wshCultures.inc", 1, False).ReadAll ())(0).SubMatches
- 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 "kernel32.dll", "IsValidLocale", "f=s", "r=t", "i=ll"
-     oDyn.Register "kernel32.dll", "GetLocaleInfo", "f=s", "r=l", "i=llrl"
-     ' 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
- ' =====
- ' == PUBLIC MEMBERS
- ' =====
- Public Function GetLocaleInfo (LCID, LCType)
-     ' déclaration des variables
-     Dim strLocaleData
-     Dim nLen
-     Dim LCID_INSTALLED: LCID_INSTALLED = 1
-     ' vérification de la validité du LCID
-     If oDyn.IsValidLocale (LCID, LCID_INSTALLED) Then
-         ' le LCID est valide
-         ' lecture de la donnée associée à la valeur
-         ' détermination de la taille du buffer à allouer
-         nLen=oDyn.GetLocaleInfo (LCID, LCType, strLocaleData, 0)
-         If nLen > 0 Then
-             ' allocation du buffer et lecture de la donnée
-             strLocaleData = String (nLen, VBNullChar)
-             nLen=oDyn.GetLocaleInfo (LCID, LCType, strLocaleData, nLen)
-         Else
-             ' la donnée n'est pas disponible
-             strLocaleData = ""
-         End If
-     End If
-     ' retour de la donnée au format texte
-     GetLocaleInfo = strLocaleData
- End Function
- Public Function EnumSystemLocales ()
-     ' déclaration des variables
-     Dim strLanguage, strEngLanguage, strISOCountry, strISOLanguage
-     Dim strLocaleName
-     Dim LCID
-     Dim arrSL(): Redim arrSL (0)
-     Dim LCID_INSTALLED: LCID_INSTALLED = 1
-     Dim LOCALE_SLANGUAGE: LOCALE_SLANGUAGE = 2
-     Dim LOCALE_SENGLANGUAGE : LOCALE_SENGLANGUAGE = &H1001
-     Dim LOCALE_SISO3166CTRYNAME: LOCALE_SISO3166CTRYNAME = &H5A
-     Dim LOCALE_SISO639LANGNAME: LOCALE_SISO639LANGNAME = &H59
-     Dim LOCALE_SNAME: LOCALE_SNAME = &H5C

```

```

- ' définition du header
- arrSL (0) = "LCID" & shell.strTableFieldSep & "ISO" & shell.strTableFieldSep & "Language"
- ' énumération des langues installées
- For LCID=0 To 65535
-     ' vérification de la validité du LCID
-     If oDyn.IsValidLocale (LCID, LCID_INSTALLED) Then
-         ' le LCID est valide
-         ' lecture du nom de la langue (locale)
-         strLanguage = Me.GetLocaleInfo (LCID, LOCALE_SLANGUAGE)
-         ' lecture du nom du langage (English)
-         strEngLanguage = Me.GetLocaleInfo (LCID, LOCALE_SENGLANGUAGE)
-
-         ' lecture du code pays
-         strISOCountry = Me.GetLocaleInfo (LCID, LOCALE_SISO3166CTRYNAME)
-         ' lecture du code langue
-         strISOLanguage = Me.GetLocaleInfo (LCID, LOCALE_SISO639LANGNAME)
-
-         ' détermination du tag de langue (Vista et supérieur)
-         strLocaleName = Me.GetLocaleInfo (LCID, LOCALE_SNAME)
-         ' vérification de la présence du tag de langue
-         If strLocaleName = "" Then
-             ' la constante LOCALE_SNAME n'est pas définie sur le système
-             ' tentative de construction manuelle du tag
-             strLocaleName = strISOLanguage & "-" & strISOCountry
-             ' alphabet cyrillique ou latin
-             If InStr (strEngLanguage, "Cyrillic") Then strLocaleName = strLocaleName & "-Cyril"
-             If InStr (strEngLanguage, "Latin") Then strLocaleName = strLocaleName & "-Latn"
-         End If
-         ' sauvegarde des informations
-         Redim Preserve arrSL (UBound (arrSL) + 1)
-         arrSL (UBound (arrSL)) = LCID & shell.strTableFieldSep & strLocaleName & shell.strTableFieldSep & strLanguage
-     End If
- Next
- ' fin de traitement
- EnumSystemLocales = arrSL
- End Function
- End Class

```