Windows Scripting Host
Windows Management Instrumentation
Active Directory Services Interface

I. Le langage Vbscript

A. *Constantes et variables*

1. Tester les variables

IsArray()
IsDate()
IsEmpty()
IsError()
IsNull()
IsNumeric()
IsObject()

B. *Les fonctions*

1. Chaînes

LCase(), UCase()
Left(), Right(),Mid(),InStr(), Len()

2. Dates / Heures

Date(), Time(), DateAdd(),DateDiff(),Day(), Month(), Year(),Hour(), Minute(), Second(),WeekdayName(), MonthName()

C. *Instructions d'interaction*

1. MsgBox

a) Syntaxe

MsgBox("Message", Boutons, "Titre")

b) Boutons

vbOKOnly	0
VbOKCancel	1
VbAbortRetryIgnore	2
VbYesNoCancel	3
VbYesNo	4
VbRetryCancel	5
vbCritical	16
VbQuestion	32
VbExclamation	48
VbInformation	64

2. InputBox

D. *Les structures de contrôle*

1. Les instructions conditionnelles

a) If

```
If Condition Then
   ...
End If
If Condition Then
   ...
Else
   ...
End If
If Condition1 Then
```

```
    ...
ElseIf Condition2 Then
    ...
End If
```

b) Select

```
Select Case Expression
Case Valeur1
    ...
Case Valeur2
    ...
Case Else
    ...
End Select
```

2. Les boucles

a) For

```
For Compteur = Début To Fin Step Pas
    ...
Next Compteur
```

b) While

```
While condition
    ...
Wend
```

c) Do… Loop

```
Do While|Until Condition
    ...
Loop
Do
    ...
Loop While|Until Condition
```

E. La gestion d'erreur

```
On Error Resume Next
If Err.Number<>O Then
    ...
End If
```

F. Utilisation d'éditeurs

AderSoft VbsEdit 2.0
Sapien Primal Script 3.1.438 (The Best)

II. Les objets Wsh

A. *Présentation*

WSH Windows Scripting Host
Technologie fondée sur ActiveX
Langage interprété

B. *Composants*

L'utilisation de ce langage nécessite le chargement de plusieurs composants :

Vbscript.dll	vscript
Jscript.dll	jscript
wshcon.dll	Windows Script Controler
wshext.dll	Shell Extension for Windows Script Host
wshfr.dll	Ressources internationales de Windows Script Host
wshom.ocx	Windows Script Host Runtime Library

Ces technologies sont très "dangereuses". Elles supposent d'être employées dans le cadre d'un réseau parfaitement sécurisé et de l'utilisation d'un navigateur autre que Internet Explorer qui permet l'utilisation de ces technologies dans le cadre d'Internet.
Pour activer ces technologies:

```
@echo off
cls
rem
rem L'objet de ce batch est d'inscrire les Dll nécessaires à l'exécution des
scripts Wsh
rem
set WIN=%systemroot%\system32
rem
rem Microsoft (r) Windows Script Controller
rem
%win%\regsvr32.exe -s %WIN%\wshcon.dll
rem
rem Microsoft (r) Shell Extension for Windows Script Host
rem
%win%\regsvr32.exe -s %WIN%\wshext.dll
rem
rem Ressources internationales de Microsoft (r) Windows Script Host
rem
%win%\regsvr32.exe -s %WIN%\wshfr.dll
rem
rem Windows Script Host Runtime Library
rem
%win%\regsvr32.exe -s %WIN%\wshom.ocx
rem
rem Microsoft (r) VBScript
rem
%win%\regsvr32.exe -s %WIN%\vbscript.dll
rem
rem Microsoft (r) JScript
rem
%win%\regsvr32.exe -s %WIN%\jscript.dll
rem
rem Microsoft (r) VBScript - Ressources internationales
rem
%win%\regsvr32.exe -s %WIN%\vbsfr.dll
SET WIN=
```

Pour désactiver ces technologies, utilisez regsvr32.exe –u –s.

C. *Mode d'exécution*

1. Les pages Wsf

Ce fichier peut être tapé dans le bloc note. Pensez à le sauvegarder avec l'extension .wsf.
Pour l'exécuter, tapez wscript //job:1 wsf.wsf
<package>

```
    <comment>
        Mon commentaire
    </comment>
    <job id="1">
        <object id="O_SHELL" progid="Wscript.shell" />
        <reference object="Wscript.Shell"/>
        <resource id="LOGICIEL">Votre logiciel</resource>
        <resource id="CHOIX"></resource>
        <script language="vbscript">
            CHOIX=InputBox(GetResource("LOGICIEL"),"Votre choix")
            O_SHELL.Run CHOIX
        </script>
    </job>
</package>
```

2. Les pages Html

```
<html>
    <body>
    <script language="vbscript">
        Option Explicit
        Dim NETWORK,USER,COMPUTER
        Set NETWORK=CreateObject("WScript.NETWORK")
        With NETWORK
            USER=.UserName : COMPUTER=.ComputerName
        End With
        document.Write USER & "," & COMPUTER
        Set USER=Nothing
        Set COMPUTER=Nothing
        Set NETWORK=Nothing
    </script>
    </body>
</html>
```

D. Utilisation de Jscript

```
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body >
    <script language="javascript">
        for(prop in navigator)
        {
            document.write('navigator.'+prop+':'+navigator[prop]+'<br>');
        }
        if(navigator.appName!='Microsoft Internet Explorer')
        {
            location.href='http://www.alltheweb.com';
        }
    </script>
    <script language="jscript">
        var e=new Error("0","Pas d'erreur");
        try
        {
            var OBJ_WSH_NETWORK=new ActiveXObject("Wscript.Network");
        }
        catch(e)
        {
```

```
            document.write('Erreur : ' + e.number + ' ' +e.description + '<br>')
        }
        if(e.number==0)
        {
            with(OBJ_WSH_NETWORK)
            {
                document.write("En jscript : " + ComputerName + "  " + UserName + "<br>");
            }
        }
        else
        {
            //location.href='http://www.alltheweb.com';
        }
    </script>
    <script language="vbscript">
        On Error Resume Next
        Set OBJ_WSH_NETWORK=CreateObject("Wscript.Network")
        If Err.number=0 then
            With OBJ_WSH_NETWORK
                document.write "En vbscript : " & .ComputerName & "  " & .UserName & "<br>"
            End With
        Else
            With Err
                document.write "Erreur : " & .number & "  " & .description & "<br>"
            End With
        End If
        location.href="http://www.alltheweb.com"
    </script>
</body>

</html>
```

E. Utilisation d'un fichier hta

```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
        <HTA:APPLICATION
            application="Vive Ie"
            border="dialog"
            borderStyle="normal"
            caption="yes"
            icon=""
            maximizeButton="yes"
            minimizeButton="yes"
            showInTaskbar="no"
            sysmenu="yes"
            windowState="normal"
            innerBorder="yes"
            navigable="yes"
            scroll="auto"
            selection="no"
            scrollFlat="yes"
        />
    </head>
    <body>
        <script language="vbscript">
            Document.Write "<b>Attention : l'utilisation de Windows peut nuire !!!</b><br><br>"
            'Set WSH=CreateObject("WScript.Shell")
            'With WSH
```

```
        '    .RegWrite "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\RegisteredOrganization","Sherlok Holmes","REG_SZ"
        '    .RegWrite "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\RegisteredOwner","Sherlok Holmes","REG_SZ"
        'End With
        Set FSO=CreateObject("Scripting.FileSystemObject")
        Set F=FSO.OpenTextFile("c:\boot.ini",1)
        Do
            document.write F.ReadLine & "<br>"
        Loop Until F.AtEndOfStream
        Set F=Nothing
        Set FSO=Nothing
        'Set WSH=Nothing
    </script>
  </body>
</html>
```

F. Shell

1. Méthodes

BrowseForFolder	Recherche d'un dossier
CascadeWindows	Afficher les fenêtres en cascade
ControlPanelItem	Composants du panneau de configuration
EjectPC	Ejecte le PC de son support
Explore	Ouvrir le dossier
FileRun	Boîte de dialogue Exécuter
FindComputer	Recherche d'ordinateurs
FindFiles	Recherche de fichiers
Help	Affiche l'aide
MinimizeAll	Tout réduire
NameSpace	Ouvrir un dossier spécial
Open	Ouvrir un odssier
RefreshMenu	Actualiser le menu
SetTime	Propriétés Date et heure
ShutdownWindows	Quitter Windows
SubHelp	Afficher l'aide
Suspend	Fonction suspendre
TileHorizontally	Réorganiser horizontalement
TileVertically	Réorganiser verticalement
TrayProperties	Propriétés de la barre de tâches
UndoMinimizeAll	Annulation de la réduction de toutes les fenêtres
Windows	Ferme toutes les fenêtres ouvertes

2. Pour fonctionner…

Vous devez disposer de la version **Version 4.71** ou supérieure de Shell32.dll

3. Exemple

```
Set ObjAppli=CreateObject("Shell.Application")
objAppli.SetTime
```

G. Objet Wscript

1. Propriétés et méthodes

Propriété	Description
Application	Retourne l'interface IDispatch pour Wscript
Arguments	Collection des paramètres
ConnectObject	Connecte les évènements d'un objet à des fonctions avec un

Propriété	Description
	préfixe donné.
CreateObject	Crée un objet et met en place la gestion des évènements.
DisconnectObject	Déconnecte un objet précédemment connecté de WSH.
Echo	Affiche des paramètres dans une fenêtre ou au prompt d'une commande dans la fenêtre DOS.
FullName	Chemin complet de l'exécutable utilisé
GetObject	Retourne un objet Automation à partir d'un fichier.
Name	Nom de wscript (propriété par défaut)
Path	Nom du répertoire où se trouvent Wscript.exe ou Cscript.exe
Quit	Stoppe l'exécution avec un code d'erreur particulier.
Read (avec *strStream* représentant une propriété stdIn)	Lie un nombre spécifié de caractères dans un flux d'entrée et renvoie la chaîne résultante.
ReadAll (avec *strStream* représentant une propriété stdIn)	Lie un flux d'entrée et renvoie la chaîne résultante.
ReadLine (avec *strStream* représentant une propriété stdIn)	Lie une ligne d'un flux d'entrée et renvoie la chaîne résultante.
ScriptFullName	Chemin complet du script qui est exécuté par WSH
ScriptName	Nom du fichier de script qui est exécuté par WSH
Skip (avec *strStream* représentant une propriété stdIn)	Ignore un certain nombre de caractères lors de la lecture d'un flux de données entrant (« input stream »).
SkipLine (avec *strStream* représentant une propriété stdIn)	Ignore la ligne suivante lors de la lecture d'un flux de données entrant (« input stream »).
Sleep	Place le process dans un état inactif pour un nombre spécifié de millisecondes et continue ensuite l'exécution.
StdErr	Expose le flux de sortie en écriture seule concernant l'erreur (terme US : write-only error output stream) pour le script courant.
StdIn	Expose le flux d'entrée en lecture seule (terme US : read-only input stream) pour le script courant.
StdOut	Expose le flux de sortie en écriture seule (terme US : a write-only output stream) pour le script courant.
strStream.AtEndOfLine (avec strStream représentant une propriété stdIn)	Renvoie True si le pointeur d'entrée précède juste l'indicateur de fin de ligne dans un flux d'entrée
strStream.AtendOfStream (avec strStream représentant une propriété stdIn)	Renvoie True si le pointeur d'entrée est à la fin du flux d'entrée
strStream.Close (avec *strStream* représentant une propriété stdIn, stdOut, stdErr)	Ferme un flux ouvert
strStream.Column (avec strStream représentant une propriété stdIn)	Renvoie le numéro de colonne de la position du caractère courant dans le flux d'entrée
Version	Version de WSH
Write (avec *strStream* représentant une propriété stdOut ou stdErr)	Ecrit une chaîne spécifiée dans un flux de données sortant (« output stream »).
WriteBlankLines (avec *strStream* représentant une propriété stdOut ou stdErr)	Ecrit un certain nombre de nouvelles lignes de caractères dans un flux de données sortant (« output stream »).
WriteLine (avec *strStream* représentant une propriété stdOut ou stdErr)	Ecrit une ligne spécifié et un caractère newline dans un flux de données sortant (« output stream »).

2. Exemple : Arguments, Echo, Quit

```
Set OBJ_ARG = Wscript.Arguments
For Each ARG in OBJ_ARG
    AFFICHE=AFFICHE & Chr(13) & ARG
Next
With Wscript
    .Echo AFFICHE
```

```
    .Quit
End With
```

H. Objet Wscript.Shell

1. Méthodes et propiétésl

<table>
<tr><th>Méthode</th><th>Description</th></tr>
<tr><td>AppActivate</td><td>Active une fenêtre d'application
object.AppActivate StrTitreFenêtre</td></tr>
<tr><td>LogEvent</td><td>Enregistre un évènement dans l'observateur d'évènements de NT ou dans un fichier wsh.log sur Windows 9x
object.LogEvent(intType, strMessage [,strTarget])
intType Valeur entière décrivant le type d'événement
strMessage Message à consigner
strTarget Ordinateur distant
<table>
<tr><th>intType</th><th>Value</th></tr>
<tr><td>0</td><td>SUCCESS</td></tr>
<tr><td>1</td><td>ERROR</td></tr>
<tr><td>2</td><td>WARNING</td></tr>
<tr><td>4</td><td>INFORMATION</td></tr>
<tr><td>8</td><td>AUDIT_SUCCESS</td></tr>
<tr><td>16</td><td>AUDIT_FAILURE</td></tr>
</table></td></tr>
<tr><td>SendKeys</td><td>Envoie une séquence de touches à la fenêtre active.
object.SendKeys (strSéquenceTouches)
A chaque fois que l'état vidéo (l'affichage) est modifié, employez l'instruction Wscript.Sleep intTempsMs pour synchroniser le déroulement du script avec les événements vidéo.
<table>
<tr><th>Touche</th><th>Séquence</th></tr>
<tr><td>BACKSPACE</td><td>{BACKSPACE}, {BS}, or {BKSP}</td></tr>
<tr><td>BREAK</td><td>{BREAK}</td></tr>
<tr><td>CAPS LOCK</td><td>{CAPSLOCK}</td></tr>
<tr><td>DEL or DELETE</td><td>{DELETE} or {DEL}</td></tr>
<tr><td>DOWN ARROW</td><td>{DOWN}</td></tr>
<tr><td>END</td><td>{END}</td></tr>
<tr><td>ENTER</td><td>{ENTER} or ~</td></tr>
<tr><td>ESC</td><td>{ESC}</td></tr>
<tr><td>HELP</td><td>{HELP}</td></tr>
<tr><td>HOME</td><td>{HOME}</td></tr>
<tr><td>INS or INSERT</td><td>{INSERT} or {INS}</td></tr>
<tr><td>LEFT ARROW</td><td>{LEFT}</td></tr>
<tr><td>NUM LOCK</td><td>{NUMLOCK}</td></tr>
<tr><td>PAGE DOWN</td><td>{PGDN}</td></tr>
<tr><td>PAGE UP</td><td>{PGUP}</td></tr>
<tr><td>PRINT SCREEN</td><td>{PRTSC}</td></tr>
<tr><td>RIGHT ARROW</td><td>{RIGHT}</td></tr>
<tr><td>SCROLL LOCK</td><td>{SCROLLLOCK}</td></tr>
<tr><td>TAB</td><td>{TAB}</td></tr>
<tr><td>UP ARROW</td><td>{UP}</td></tr>
<tr><td>F1</td><td>{F1}</td></tr>
<tr><td>F2</td><td>{F2}</td></tr>
<tr><td>F3</td><td>{F3}</td></tr>
<tr><td>F4</td><td>{F4}</td></tr>
<tr><td>F5</td><td>{F5}</td></tr>
<tr><td>F6</td><td>{F6}</td></tr>
<tr><td>F7</td><td>{F7}</td></tr>
</table></td></tr>
</table>

Méthode	Description
	F8 {F8} F9 {F9} F10 {F10} F11 {F11} F12 {F12} F13 {F13} F14 {F14} F15 {F15} F16 {F16} SHIFT + CTRL ^ ALT %
Run	Lance une application *object.Run(strCommande, [intWindowStyle], [bWaitOnReturn])* strCommande Commande incluant le chemin si celui-ci n'est pas indiqué dans la variable PATH intWindowStyle Valeur entière précisant le mode d'ouverture bWaitOnReturn Si la valeur est à True, le script attend que le programme soit terminé pour continuer à s'exécuter. Par défaut, la valeur est à False. **intWindowStyle Description** 0 Cache la fenêtre et en active une autre 1 Active la fenêtre dans sa taille intermédiaire 2 Active la fenêtre et la passe en icône. 3 active la fenêtre en plein écran 4 Rappelle la fenêtre dans la dernière dimension utilisée 5 Activates the window and displays it in its current size and position. 6 Minimizes the specified window and activates the next top-level window in the Z order. 7 Displays the window as a minimized window. The active window remains active. 8 Displays the window in its current state. The active window remains active. 9 Activates and displays the window. If the window is minimized or maximized, the system restores it to its original size and position. An application should specify this flag when restoring a minimized window. 10 Sets the show-state based on the state of the program that started the application.
CreateShortCut	Création d'un raccourci vers un fichier ou une Url *object.CreateShortcut(strPathname)* strPathnamechemin d'accès complet au fichier lnk ou url
ExpandEnvironmentStrings	*object.ExpandEnvironmentStrings(strString)* strString Variable d'environnement
Popup	Affiche le texte à l'écran et gère le choix de l'utilisateur intButton = object.Popup(strText,[nSecondsToWait],[strTitle],[nType]) strText Texte du message à afficher nSecondsToWait Nombre de secondes durant lesquelles la fenêtre sera affichée strTitle Titre de la fenêtre nType Type de la boîte IntButton Valeur renvoyée par le choix de l'utilisateur **Valeur nType Description**

<table>
<tr><th>Méthode</th><th>Description</th></tr>
<tr><td></td><td>
0 Show OK button.
1 Show OK and Cancel buttons.
2 Show Abort, Retry, and Ignore buttons.
3 Show Yes, No, and Cancel buttons.
4 Show Yes and No buttons.
5 Show Retry and Cancel buttons.
16 Show "Stop Mark" icon.
32 Show "Question Mark" icon.
48 Show "Exclamation Mark" icon.
64 Show "Information Mark" icon.

Value intButton — Description
1 — OK button
2 — Cancel button
3 — Abort button
4 — Retry button
5 — Ignore button
6 — Yes button
7 — No button
</td></tr>
<tr><td>Exec</td><td>Exécute un script
object.Exec(strCommand)
strCommand Nom du script à exécuter</td></tr>
<tr><td>RegRead</td><td>object.RegRead(strName)
strName Entrée de la base de regsitres à lire

Root key Name — Abbreviation
HKEY_CURRENT_USER — HKCU
HKEY_LOCAL_MACHINE — HKLM
HKEY_CLASSES_ROOT — HKCR
HKEY_USERS — HKEY_USERS
HKEY_CURRENT_CONFIG — HKEY_CURRENT_CONFIG

Type — Description — In the Form of
REG_SZ — A string — A string
REG_DWORD — A number — An integer
REG_BINARY — A binary value — A VBArray of integers
REG_EXPAND_SZ — An expandable string (e.g., "%windir%\\calc.exe") — A string</td></tr>
<tr><td>RegDelete</td><td>Détruit l'entrée spécifiée
object.RegDelete(strName)</td></tr>
<tr><td>RegWrite</td><td>object.RegWrite(strName, anyValue [,strType])

anyValue — strType
string — REG_SZ
string — REG_EXPAND_SZ
integer — REG_DWORD
string — REG_BINARY

Dans la chaîne à écrire, vous pouvez remplacer le nom littéral de la branche par l'alias indiqué ci-dessous

Branche — Alias
HKEY_CURRENT_USER — HKCU
HKEY_LOCAL_MACHINE — HKLM
HKEY_CLASSES_ROOT — HKCR
HKEY_USERS — HKEY_USERS</td></tr>
</table>

Méthode	Description
	HKEY_CURRENT_CONFIG HKEY_CURRENT_CONFIG
CurrentDirectory	Retourne le répertoire courant
SpecialFolders	Collection des répertoires constituant l'environnement de l'utilisateur : object.SpecialFolders(objWshSpecialFolders) ObjSpecialFolders AllUsersDesktop AllUsersStartMenu AllUsersPrograms AllUsersStartup Desktop Favorites Fonts MyDocuments NetHood PrintHood Programs Recent SendTo StartMenu Startup Templates
Environment	object.Environment ([strType]) **strType** **Os** System Windows NT/2000 Process Windows 95/98/Me

Variable	Description	System	User	NT 2000)	98 ME
NUMBER_OF_PROCESSORS	Nombre de processeurs	X	-	X	-
PROCESSOR_ARCHITECTURE	Type de processeur	X	-	X	-
PROCESSOR_IDENTIFIER	Id du processeur	X	-	X	-
PROCESSOR_LEVEL	Niveau du processeur	X	-	X	-
PROCESSOR_REVISION	Version du processeur.	X	-	X	-
OS	Os	X	-	X	-
COMSPEC	Interpréteur de commandes	X	-	X	X
HOMEDRIVE	Lecteur logique de la partition d'amorçage	-	-	X	-
HOMEPATH	Répertoire de travail	-	-	X	-
PATH	Chemin d'accès aux exécutables	X	X	X	X
PATHEXT	Extensions pour les fichiers exécutables	X	-	X	-
PROMPT	Invite	-	-	X	X
SYSTEMDRIVE	Lecteur du répertoire système	-	-	X	-
SYSTEMROOT	Equivalent à WINDIR.	-	-	X	-
WINDIR	Répertoire du système.	X	-	X	X
TEMP	Répertoire des fichiers temporaires,	-	X	X	X
TMP	Autre variable des fichiers temporaires	-	X	X	X

2. Exemples

a) 1er exemple : activate, sleep, sendkeys

```
set WshShell = WScript.CreateObject("WScript.Shell")
WshShell.Run "calc"
WScript.Sleep 100
WshShell.AppActivate "Calculator"
WScript.Sleep 100
```

```
WshShell.SendKeys "1{+}"
WScript.Sleep 500
WshShell.SendKeys "2"
WScript.Sleep 500
WshShell.SendKeys "~"
WScript.Sleep 500
WshShell.SendKeys "*3"
WScript.Sleep 500
WshShell.SendKeys "~"
WScript.Sleep 2500
```

b) 2ème exemple : CreateShortcut

```
  set WshShell = WScript.CreateObject("WScript.Shell")
Set WSHSHELL=Wscript.CreateObject("WScript.Shell")
CHEMIN_BUREAU = WSHSHELL.SpecialFolders("Desktop")
Set RACCOURCI_CALC = WSHSHELL.CreateShortcut(CHEMIN_BUREAU  & "\Calculatrice.lnk")
With RACCOURCI_CALC
    .TargetPath = WScript.ScriptFullName
    .WindowStyle = 1
    .Hotkey = "CTRL+SHIFT+C"
    .IconLocation = "calc.exe, 0"
    .Description = "Calculatrice"
    .WorkingDirectory = CHEMIN_BUREAU
    .Save
End With
Set LIEN_GOOGLE = WshShell.CreateShortcut(CHEMIN_BUREAU & "\Google.url")
With LIEN_GOOGLE
    .TargetPath = "http://www.google.fr"
    .Save
End With
```

3. 3ème exemple : ExpandEnvironmentStrings

```
set WshShell = WScript.CreateObject("WScript.Shell")
WScript.Echo "WinDir is " & WshShell.ExpandEnvironmentStrings("%WinDir%")
```

4. 4ème exemple :RegRead, RegWrite

```
Set WSHELL=Wscript.CreateObject("Wscript.Shell")
On Error Resume Next
TEXTE_CLE_CALC="HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Calculatrice"
CLE_CALC=WSHELL.RegRead(TEXTE_CLE_CALC)
If CLE_CALC ="" Then
    WSHELL.RegWrite TEXTE_CLE_CALC,"c:\windows\calc.exe","REG_SZ"
    Wscript.Echo "Entrée ajoutée !"
Else
    Wscript.Echo "La clé existe déjà !"

End If
```

5. Exemple 5 : Run

```
'Ouvre le fichier boot.ini
Set OBJ_WSH_SHELL=CreateObject("Wscript.Shell")
OBJ_WSH_SHELL.Run("notepad.exe c:\boot.ini")
' OBJ_WSH_SHELL.Run("c:\boot.ini")
```

6. Exemple 6 : Popup

```
'Utilisation de Popup qui remplace MsgBox de Vb
Set OBJ_WSH_SHELL=CreateObject("Wscript.Shell")
V_REPONSE=OBJ_WSH_SHELL.Popup("Votre texte ici",120,"Votre texte là",0)
```

```
With Wscript
    Select Case V_REPONSE
        Case 6:
            .echo "Vous avez répondu oui"
        Case 7:
            .echo "Vous avez répondu non"
    End Select
    .quit
End With
```

I. Objet Wscript.Environment

Cette manipulation se fait à partir de l'objet objShell défini précédemment. La propriété Environment de cet objet permet de récupérer les variables d'environnement sous forme d'une collection. La méthode Environment peut prendre en argument une des valeurs suivantes : « USER », « SYSTEM » et « PROCESS ». Ces valeurs seront dépendantes soit de l'utilisateur, soit du système, soit du process en cours.
Si aucune valeur n'est fournie, la valeur utilisée sera « SYSTEM » sur WindowsNT et « PROCESS » sur Windows95. Sur Windows95, « PROCESS » est d'ailleurs la seule valeur supportée.
Dans la documentation, la variable « WshEnvironnement » est souvent utilisée pour désigner cette collection, mais on peut appeler cette collection différemment. Il ne s'agit pas d'un objet intrinsèque.

1. Propriétés et méthodes

Item
Length
Count
Remove

2. Exemples

a) Exemple 1

```
Set WSHSHELL = WScript.CreateObject("WScript.Shell")
Set WSHENV = WshShell.Environment("PROCESS")
Wscript.Echo WSHENV.Length
Wscript.Echo WSHENV.Count
For Each VAR_ENV in WSHENV
    AFFICHE=AFFICHE & Chr(13) & VAR_ENV
Next
Wscript.Echo AFFICHE
```

b) Exemple 2

```
Set shell = WScript.CreateObject( "WScript.Shell" )
WScript.Echo shell.Environment("PROCESS").Item("WINDIR")
WScript.Echo shell.Environment("PROCESS").Item("PATH")
WScript.Echo shell.Environment("PROCESS").Item("PROMPT")
WScript.Echo shell.ExpandEnvironmentStrings("PROCESS").("%WINDIR%")
shell.Environment.Remove( "PATH" )
Wscript.Echo shell.ExpandEnvironmentStrings( _
"The Windows directory is: %WINDIR%" )
```

J. objet Wscript.Network

1. Propriétés et méthodes

a) AddWindowsPrinterConnection

(1) Intérêt

Ajoute une imprimante

(2) Syntaxe

Windows NT/2000 object.AddWindowsPrinterConnection(strPrinterPath)
Windows 9x/Me object.AddWindowsPrinterConnection(strPrinterPath,strDriverName[,strPort])

(3) Arguments

strPrinterPath	Chemin UNC désignant l'imprimante
strDriverName	Nom du pilote (inutile sous Windows NT/2000)
strPort	Port utilisé (inutile sous Windows NT/2000)

(4) Exemples

```
Set WshNetwork = WScript.CreateObject("WScript.Network")
PrinterPath = "\\printserv\DefaultPrinter"
WshNetwork.AddWindowsPrinterConnection PrinterPath
Set WshNetwork = WScript.CreateObject("WScript.Network")
PrinterPath = "\\printserv\DefaultPrinter"
PrinterDriver = "Lexmark Optra S 1650"
WshNetwork.AddWindowsPrinterConnection PrinterPath, PrinterDriver
```

b) EnumPrinterConnections

(1) Syntaxe

objPrinters = object.EnumPrinterConnections

(2) Arguments

objPrinters Contient les informations de connexion

(3) Exemple

```
Set WshNetwork = WScript.CreateObject("WScript.Network")
Set oPrinters = WshNetwork.EnumPrinterConnections
AFFICHE= "Système local d'impression"
For i = 0 to oPrinters.Count - 1 Step 2
  AFFICHE=AFFICHE & Chr(13) & "Port " & oPrinters.Item(i) & " = " & oPrinters.Item(i+1)
Next
WScript.Echo AFFICHE
```

c) RemovePrinterConnection

(1) Intérêt

Retire une imprimante partagée

(2) Syntaxe

object.RemovePrinterConnection(strName, [bForce], [bUpdateProfile])

(3) Arguments

strName	Supprime un port de type UNC ou un port local (LPT1)
bForce	Valeur logique qui force la suppression alors des utilisateurs sont connectés
bUpdateProfile	Si la valeur est à True, les modifications sont apportées au profil de l'utilisateur. La valeur par défaut est False.

(4) Exemple

```
Set WshNetwork = WScript.CreateObject("WScript.Network")
Set PrinterPath = "\\printserv\DefaultPrinter"
Set PrinterDriver = "Lexmark Optra S 1650"
On Error Resume Next
WshNetwork.AddWindowsPrinterConnection(PrinterPath, PrinterDriver)
If Err.Number <> 0 then
    WScript.Echo("Printer Connection Failed!")
```

```
End If
WshNetwork.RemovePrinterConnection (PrinterPath, true, true)
AddPrinterConnection
```

d) AddPrinterConnection

(1) Intérêt

Ajoute un port d'impression MS-DOS

(2) Syntaxe

```
object.AddPrinterConnection(strLocalName, strRemoteName[,bUpdateProfile][,strUser][,strPassword])
```

(3) Arguments

strLocalName Nom du port local assigné à l'imprimante distante
strRemoteName Nom de l'impirmante distante (chemin UNC)
bUpdateProfile True permet la mise à jour du profil de l'utilisateur
strUser Nom de l'utilisateur
strPassword Mot de passe

(4) Exemple

```
Set WshNetwork = WScript.CreateObject("WScript.Network")
WshNetwork.AddPrinterConnection "LPT1", "\\\
```

e) SetDefaultPrinter

(1) Intérêt

Définit l'imprimante par défaut

(2) Syntaxe

```
object.SetDefaultPrinter(strPrinterName)
```

(3) Arguments

strPrinterName Imprimante distante en dénomination UNC

(4) Exemple

```
Set WshNetwork = WScript.CreateObject("WScript.Network")
Set PrinterPath = "\\research\library1"
Set rc = WshNetwork.AddWindowsPrinterConnection(PrinterPath)
If Not rc then
    WScript.Echo("Printer Connection Failed!")
End If
WshNetwork.SetDefaultPrinter PrinterPath
```

Returns the current network drive mapping information.

f) EnumNetworkDrives

(1) Intérêt

Liste les lecteurs mappés

(2) Syntaxe

```
objDrives = object.EnumNetworkDrivesArguments
```

(3) Argument

objDrives Variable qui stocke les valeurs des lecteurs mappés

(4) Exemple

```
Set WshNetwork = WScript.CreateObject("WScript.Network")
Set oDrives = WshNetwork.EnumNetworkDrives
AFFICHE="Lecteurs réseaux:"
For i = 0 to oDrives.Count - 1 Step 2
        AFFICHE=AFFICHE & Chr(13) & "Lecteur" & oDrives.Item(i) & " = " & oDrives.Item(i+1)
Next
WScript.Echo AFFICHE
```

g) MapNetWorkDrive

(1) Syntaxe

```
object.MapNetworkDrive(strLocalName, strRemoteName, [bUpdateProfile], [strUser], [strPassword])
```

(2) Arguments

strLocalName	Nom du port lecteur logique assigné au chemin UNC
strRemoteName	Chemin UNC du répertoire partagé
bUpdateProfile	True permet la mise à jour du profil de l'utilisateur
strUser	Nom de l'utilisateur
strPassword	Mot de passe

(3) Exemple

```
Set WshNetwork = WScript.CreateObject("WScript.Network")
WshNetwork.MapNetworkDrive ("E:", "\\\\Server\\Public")
WshNetwork.RemoveNetworkDrive ("E:");
```

h) RemoveNetWorkDrive

(1) Intérêt

Enlève un lecteur mappé

(2) Syntaxe

```
object.RemoveNetworkDrive(strName, [bForce], [bUpdateProfile])
```

(3) Arguments

strName	Supprime le lecteur logique spécifié
bForce	Valeur logique qui force la suppression alors des utilisateurs sont connectés
bUpdateProfile	Si la valeur est à True, les modifications sont apportées au profil de l'utilisateur. La valeur par défaut est False.

(4) Exemple

```
Set WshNetwork = WScript.CreateObject("WScript.Network");
WshNetwork.MapNetworkDrive ("E:", "\\\\Server\\Public")
WshNetwork.RemoveNetworkDrive ("E:");
```

i) ComputerName, UserName,UserDomain

(1) Remarque

UserDomain n'est valable que sur Windows Nt/2000. Il reprend le contenu de la variable

(2) Exemple

```
'Donne le nom de la machine et le nom d'utilisateur
Set OBJ_WSH_NETWORK=CreateObject("Wscript.Network")
with OBJ_WSH_NETWORK
    Wscript.Echo .ComputerName & " | " & .UserName
end with
```

K. Objet WshControler

1. Condition

HKLM\Software\Microsoft\Windows Script Host\Settings\Remote to 1

2. La méthode CreateScript

a) Syntaxe

object.CreateScript(CommandLine,[MachineName])

b) Paramètres

Commandline Chemin d'accès complet au script
MachineName Nom UNC de l'ordinateur distant

3. La méthode Execute

ObjetWshControler.Execute

4. La propriété Status

ObjetWshControler.Status

Valeur retournée	Sens	Description
0	NoTask	The remote script object has been created but has not yet executed.
1	Running	The remote script object is currently running.
2	Finished	The remote script object has finished running.

5. La propriété Error

ObjetWshControler.Error

6. La méthode Terminate

ObjetWshControler.Terminate

7. Les événements Error, Start, End

Ils sont définis par des procédures ayant le même préfixe :

```
Sub PROC_Error()

End Sub
Sub PROC_Start()

End Sub
Sub PROC_End()

End Sub
```

8. Exemple

```
Dim Controller, RemoteScript
Set Controller = WScript.CreateObject("WSHController")
Set RemoteScript = Controller.CreateScript("test.js", "remoteserver")
WScript.ConnectObject RemoteScript, "remote_ "
RemoteScript.Execute
Do While RemoteScript.Status <> 2
   WScript.Sleep 100
Loop

Sub remote_Error()
   Dim theError
   Set theError = RemoteScript.Error
   WScript.Echo "Error - Line: " & theError.Line & ", Char: " & theError.Character & vbCrLf & "Description: " &
theError.Description
```

```
   WScript.Quit -1
End Sub
```

L. Objet FileSystemObject

1. Propriétés

AtEndOfLine	Returns true if the file pointer is positioned immediately before the end-of-line marker in a TextStream file; false if it is not.
AtEndOfStream	Returns true if the file pointer is at the end of a TextStream file; false if it is not.
Attributes	Sets or returns the attributes of files or folders.
AvailableSpace	Returns the amount of space available to a user on the specified drive or network share.
Column	Returns the column number of the current character position in a TextStream file.
CompareMode	Sets and returns the comparison mode for comparing string keys in a Dictionary object.
Count	Returns the number of items in a collection or Dictionary object.
DateCreated	Returns the date and time that the specified file or folder was created. Read-only.
DateLastAccessed	Returns the date and time that the specified file or folder was last accessed.
DateLastModified	Returns the date and time that the specified file or folder was last modified.
Drive	Returns the drive letter of the drive on which the specified file or folder resides.
DriveLetter	Returns the drive letter of a physical local drive or a network share.
Drives	Returns a Drives collection consisting of all Drive objects available on the local machine.
DriveType	Returns a value indicating the type of a specified drive.
Files	Returns a Files collection consisting of all File objects contained in the specified folder, including those with hidden and system file attributes set.
FileSystem	Returns the type of file system in use for the specified drive.
FreeSpace	Returns the amount of free space available to a user on the specified drive or network share.
IsReady	Returns true if the specified drive is ready; false if it is not.
IsRootFolder	Returns true if the specified folder is the root folder; false if it is not.
Item	Sets or returns an item for a specified key in a Dictionary object. For collections, returns an item based on the specified key.
Key	Sets a key in a Dictionary object.
Line	Returns the current line number in a TextStream file.
Name	Sets or returns the name of a specified file or folder.
ParentFolder	Returns the folder object for the parent of the specified file or folder.
Path	Returns the path for a specified file, folder, or drive.
RootFolder	Returns a Folder object representing the root folder of a specified drive.
SerialNumber	Returns the decimal serial number used to uniquely identify a disk volume.
ShareName	Returns the network share name for a specified drive.
ShortName	Returns the short name used by programs that require the earlier 8.3 naming convention.
ShortPath	Returns the short path used by programs that require the earlier 8.3 file naming convention.
Size	For files, returns the size, in bytes, of the specified file. For folders, returns the size, in bytes, of all files and subfolders contained in the folder.
SubFolders	Returns a Folders collection consisting of all folders contained in a specified folder, including those with hidden and system file attributes set.
TotalSize	Returns the total space, in bytes, of a drive or network share.
Type	Returns information about the type of a file or folder.
VolumeName	Sets or returns the volume name of the specified drive.

2. Les méthodes

Add	Adds a new folder to a Folders collection.
BuildPath	Appends a name to an existing path.
Close	Closes an open TextStream file.
Copy	Copies a specified file or folder from one location to another.
CopyFile	Copies one or more files from one location to another.
CopyFolder	Recursively copies a folder from one location to another.
CreateFolder	Creates a folder.
CreateTextFile	Creates a specified file name and returns a TextStream object that can be used to read from or write to the file.
Delete	Deletes a specified file or folder.
DeleteFile	Deletes a specified file.

DeleteFolder	Deletes a specified folder and its contents.
DrivesExists	Returns true if the specified drive exists; false if it does not.
Exists	Returns true if a specified key exists in the Dictionary object, false if it does not.
FileExists	Returns true if a specified file exists; false if it does not.
FolderExists	Returns true if a specified folder exists; false if it does not.
GetAbsolutePathName	Returns a complete and unambiguous path from a provided path specification.
GetBaseName	Returns a string containing the base name of the last component, less any file extension, in a path.
GetDrive	Returns a Drive object corresponding to the drive in a specified path.
GetDriveName	Returns a string containing the name of the drive for a specified path.
GetExtensionName	Returns a string containing the extension name for the last component in a path.
GetFile	Returns a File object corresponding to the file in a specified path.
GetFileName	Returns the last component of specified path that is not part of the drive specification.
GetFileVersion	Returns the version number of a specified file.
GetFolder	Returns a Folder object corresponding to the folder in a specified path.
GetParentFolderName	Returns a string containing the name of the parent folder of the last component in a specified path.
GetSpecialFolder	Returns the special folder object specified.
GetTempName	Returns a randomly generated temporary file or folder name that is useful for performing operations that require a temporary file or folder.
Items	Returns an array containing all the items in a Dictionary object.
Keys	Returns an array containing all existing keys in a Dictionary object.
Move	Moves a specified file or folder from one location to another.
MoveFile	Moves one or more files from one location to another.
MoveFolder	Moves one or more folders from one location to another.
OpenAsTextStream	Opens a specified file and returns a TextStream object that can be used to read from, write to, or append to the file.
OpenTextFile	Opens a specified file and returns a TextStream object that can be used to read from, write to, or append to the file.
Read	Reads a specified number of characters from a TextStream file and returns the resulting string.
ReadAll	Reads an entire TextStream file and returns the resulting string.
ReadLine	Reads an entire line (up to, but not including, the newline character) from a TextStream file and returns the resulting string.
Remove	Removes a key, item pair from a Dictionary object.
RemoveAll	Removes all key, item pairs from a Dictionary object.
Skip	Skips a specified number of characters when reading a TextStream file.
SkipLine	Skips the next line when reading a TextStream file.
Write	Writes a specified string to a TextStream file.
WriteBlankLines	Writes a specified number of newline characters to a TextStream file.
WriteLine	Writes a specified string and newline character to a TextStream file.

3. Collections

Drive	Provides access to the properties of a particular disk drive or network share.
File	Provides access to all the properties of a file.
FileSystemObject	Provides access to a computer's file system.
Folder	Provides access to all the properties of a folder.
TextStream	Facilitates sequential access to file.

4. Exemples

a) Exemple 1 : OpenTextFile et ReadAll

```
'Lecture du fichier boot.ini
Set OBJ_FSO=CreateObject("Scripting.FileSystemObject")
Set OBJ_FICHIER=OBJ_FSO.OpenTextFile("c:\boot.ini")
wscript.echo OBJ_FICHIER.ReadAll
```

b) Exemple 2 : DateLastAcessed

```
'Date de modification du fichier boot.ini
```

```
Set OBJ_FSO=CreateObject("Scripting.FileSystemObject")
Set OBJ_FICHIER=OBJ_FSO.GetFile("c:\boot.ini")
MsgBox OBJ_FICHIER.DateLastAccessed
```

c) Exemple 3 : Files et DateLastAcessed

```
'Liste des fichiers du répertoire racine
Set OBJ_FSO=CreateObject("Scripting.FileSystemObject")
Set OBJ_REP=OBJ_FSO.GetFolder("c:\")
Set OBJ_FICHIERS=OBJ_REP.Files
For Each OBJ_FICHIER in OBJ_FICHIERS
    V_LISTE=V_LISTE & Chr(13) & OBJ_FICHIER & Chr(9) & OBJ_FICHIER.DateLastAccessed
Next
WScript.Echo V_LISTE
```

d) Exemple 4 : OpenTextFile, ReadLine, AtEndOfStream

```
'Lecture du fichier boot.ini
Set OBJ_FSO=CreateObject("Scripting.FileSystemObject")
Set OBJ_FICHIER=OBJ_FSO.OpenTextFile("c:\boot.ini")
Do
    I=I+1
    v_LIGNE=OBJ_FICHIER.ReadLine
    V_AFFICHE = V_AFFICHE & Chr(13) & I & Chr(9) & v_LIGNE
Loop Until OBJ_FICHIER.AtEndOfStream
Wscript.Echo V_AFFICHE
```

M. Excution automatique à l'aide des commutateurs de la commande CSCRIPT.EXE

Commande	Description
//B	Mode silencieux : supprime l'affichage des erreurs de scripts et des lignes de commande de demande d'information pour l'utilisateur.
//I	Mode interactif : mode par défaut
//T :nn	Spécifie au bout de combien de secondes le script s'arrête. Par défaut, il n'y a pas de limite.
//logo	Affiche une bannière à l'exécution du script dans une fenêtre DOS
//nologo	N'affiche pas de bannière.
//?	Affiche l'aide sur les commandes
//H:Cscript ou Wscript	Spécifie Cscript ou Wscript comme l'application par défaut pour lancer des scripts.
//S	Enregistre les options de lancement pour l'utilisateur courant.
//E :<moteur>	Spécifie le moteur à utiliser
//X	Lance le programme dans le débogueur
//D	Active le débogueur
//Job :<JobID>	Lance le job spécifié dans le fichier .ws.

N. Les tableaux associatifs Scripting.Dictionary

```
<html>
    <head>
        <SCRIPT LANGUAGE = VBScript>
            Set Temperatures = CreateObject("Scripting.Dictionary")
            Temperatures.Add "Rouen", 23
            Temperatures.Add "Toulouse", 28
            Temperatures.Add "Nice", 31
            If Not Temperatures.Exists("Paris") Then
                Temperatures.Add "Paris", 82
            End If
            If Temperatures.Exists("Rouen") Then
                Temperatures.Remove "Rouen"
```

```
            End If
            MsgBox Date()
        </SCRIPT>
    </head>
    <body>
        <SCRIPT LANGUAGE = VBScript>
            document.write Temperatures("Paris")
            DItems = Temperatures.Items
            DKeys = Temperatures.Keys
            For i = 0 To Temperatures.Count - 1
                Document.Write DKeys(i) & " : " & DItems(i) & "<br/>"
            Next
            For Each Ville In Temperatures
                Document.write Ville & " : " & Temperatures(Ville) & "<br/>"
            Next
            Temperatures.RemoveAll
        </script>
    </body>
</html>
```

IV. WMI Windows Management Instrumentation

A. Définition

WMI est un ensemble de composants fournis depuis Windows NT4 (SP3) pour obtenir des informations sur votre machine et les machines Windows fonctionnant dans le réseau. Il fait partie intégrante de l'OS. Il s'agit de l'implémentation Microsofot Web-Based Enterprise Management (WBEM) proposée par le Desktop Management Task Force's (DMTF). Il s'agit d'une extension du standard CIM (Common Information Model) créé par le DTMF. La représentation est assurée par le CIM (Common Information Model). Si vous développez sur .Net, vous pouvez utiliser le composant System.Management.

B. Fournisseurs WMI

Fournisseur	DLL	Espace de nom	Description
Fournisseur Active Directory	dsprov.dll	root\directory\ldap	Établit la correspondance entre les objets Active Directory et WMI.
Fournisseur Event Log	ntevt.dll	root\cimv2	Gère les journaux d'événements Windows, par exemple, lit, sauvegarde, efface, copie, supprime, surveille, renomme, compresse, décompresse et modifie les paramètres des journaux d'événements.
Fournisseur Performance Counter	wbemperf.dll	root\cimv2	Donne accès aux données brutes de performance.
Fournisseur Registry	stdprov.dll	root\default	Lit, écrit, énumère, surveille, crée et supprime les clés et valeurs de bases de registres.
Fournisseur SNMP	snmpincl.dll	root\snmp	Donne accès aux données SNMP MIB et aux interruptions issues de périphériques gérés par SNMP.
Fournisseur WDM	wmiprov.dll	root\wmi	Donne accès aux informations sur les pilotes de périphériques WDM.
Fournisseur Win32	cimwin32.dll	root\cimv2	Donne des informations sur l'ordinateur, les disques, les périphériques, les fichiers, les dossiers, les systèmes de fichiers, les composants de mise en réseau, le système d'exploitation, les imprimantes, les processus, la sécurité, les services, les partages, les utilisateurs et groupes SAM et bien plus encore.
Fournisseur Windows Installer	msiprov.dll	root\cimv2	Donne accès aux informations sur le logiciel installé.

C. Installation

A partir de la ligne de commandes, veillez à ce que le service winmgmt (Infrstructure de Gestion Windows) soit exécuté :

```
Sc config winmgmt start= auto
Sc start winmgmt
```

D. Utilisation de la commande WMIC

1. Obtenir la liste des alias

ALIAS	Accès aux alias disponibles sur l'ordinateur local
BASEBOARD	Gestion de la carte de base (également appelée carte mère ou carte système).
BIOS	Gestion des services d'entrées/sorties (E/S) de base (BIOS).
BOOTCONFIG	Gestion de la configuration du démarrage.
CDROM	Gestion des CD-ROM.
COMPUTERSYSTEM	Gestion de systèmes informatiques.
CPU	Gestion de l'unité centrale.
CSPRODUCT	Informations sur l'ordinateur issues du SMBIOS.
DATAFILE	Gestion des fichiers de données. he ÉCHAP pour arrêter

DCOMAPP	Gestion d'applications.
DESKTOP	Gestion du Bureau de l'utilisateur.
DESKTOPMONITOR	Gestion du moniteur de bureau.
DEVICEMEMORYADDRESS	Gestion des adresses mémoire pour périphériques.
DISKDRIVE	Gestion des disques durs physiques.
DISKQUOTA	Gestion de l'utilisation de l'espace disque sur les volumes NTFS.
DMACHANNEL	Gestion du canal DMA (Accès direct à la mémoire).
ENVIRONMENT	Gestion des paramètres d'environnement système.
FSDIR	Gestion des entrées de répertoires du système de fichiers.
GROUP	Gestion des comptes de groupes.
IDECONTROLLER	Gestion de contrôleurs IDE.
IRQ	Gestion des requêtes d'interruption (IRQ).
JOB	Permet l'accès aux tâches planifiées à l'aide du service de planification.
LOADORDER	Gestion des services système définissant les dépendances d'exécution.
LOGICALDISK	Gestion des périphériques de stockage locaux.
LOGON	Sessions LOGON.
MEMCACHE	Gestion de la mémoire cache.
MEMLOGICAL	Gestion de la mémoire système (configuration, disposition et disponibilité de la mémoire).
MEMPHYSICAL	Gestion de la mémoire physique d'un ordinateur.
NETCLIENT	Gestion des clients réseau.
NETLOGIN	Gestion des informations d'ouverture de session réseau (d'un utilisateur précis).
NETPROTOCOL	Gestion des protocoles et de leurs caractéristiques réseau.
NETUSE	Gestion des connexion réseau actives.
NIC	Gestion des contrôleurs réseau NIC (Network Interface Controller).
NICCONFIG	Gestion des cartes réseau.
NTDOMAIN	Gestion de l'arborescence du domaine.
NTEVENT	Entrées dans le journal d'événements NT.
NTEVENTLOG	Gestion du fichier journal d'événements NT.
ONBOARDDEVICE	Gestion des périphériques carte communs intégrés dans la carte mère.
OS	Gestion des systèmes d'exploitation installés.
PAGEFILE	Gestion des paramètres du fichier d'échange de mémoire virtuelle.
PAGEFILESET	Gestion des paramètres de fichier d'échange.
PARTITION	Gestion des zones partitionnées d'un disque physique.
PORT	Gestion des ports d'E/S. ouche ÉCHAP pour arrêter
PORTCONNECTOR	Gestion des ports de connexion physique.
PRINTER	Gestion des périphériques d'impression.
PRINTERCONFIG	Gestion de la configuration des périphériques d'impression.
PRINTJOB	Gestion des tâches d'impression.
PROCESS	Gestion des processus.
PRODUCT	Gestion des tâches des packages d'installation.
QFE	Ingénierie de correctifs à chaud.
QUOTASETTING	Gestion des informations de quotas de disque sur un volume.
RECOVEROS	Informations recueillies en mémoire en cas de dysfonctionnement du système d'exploitation.
REGISTRY	Gestion du Registre système.
SCSICONTROLLER	Gestion de contrôleurs SCSI.
SERVER	Gestion des informations sur le serveur.
SERVICE	Gestion des applications de services.
SHARE	Gestion des ressources partagées.
SOFTWAREELEMENT	Gestion des éléments d'un logiciel installé sur un ordinateur.
SOFTWAREFEATURE	Gestion des logiciels sous-ensembles de SoftwareElement.
SOUNDDEV	Gestion des périphériques audio.
STARTUP	Gestion des commandes qui s'exécutent dès que l'utilisateur ouvre une session sur l'ordinateu
SYSACCOUNT	Gestion des comptes système.
SYSDRIVER	Gestion du pilote système pour un service de base.
SYSTEMENCLOSURE	Gestion de la mise en armoire du système.
SYSTEMSLOT	Gestion de des points de connexion physiques : ports, connecteurs et périphériques, et points

TAPEDRIVE	Gestion de lecteurs de bandes.
TEMPERATURE	Gestion d'un capteur de température (thermomètre électronique).
TIMEZONE	Gestion des données de fuseau horaire.
UPS	Gestion de l'alimentation de secours (UPS).
USERACCOUNT	Auditer la gestion des comptes.
VOLTAGE	Gestion des données de capteurs de tension (tensiomètre électronique).
VOLUMEQUOTASETTING	Associe le paramètre de quota de disque à un volume précis.
WMISET	Gestion des paramètres opérationnels du service WMI.

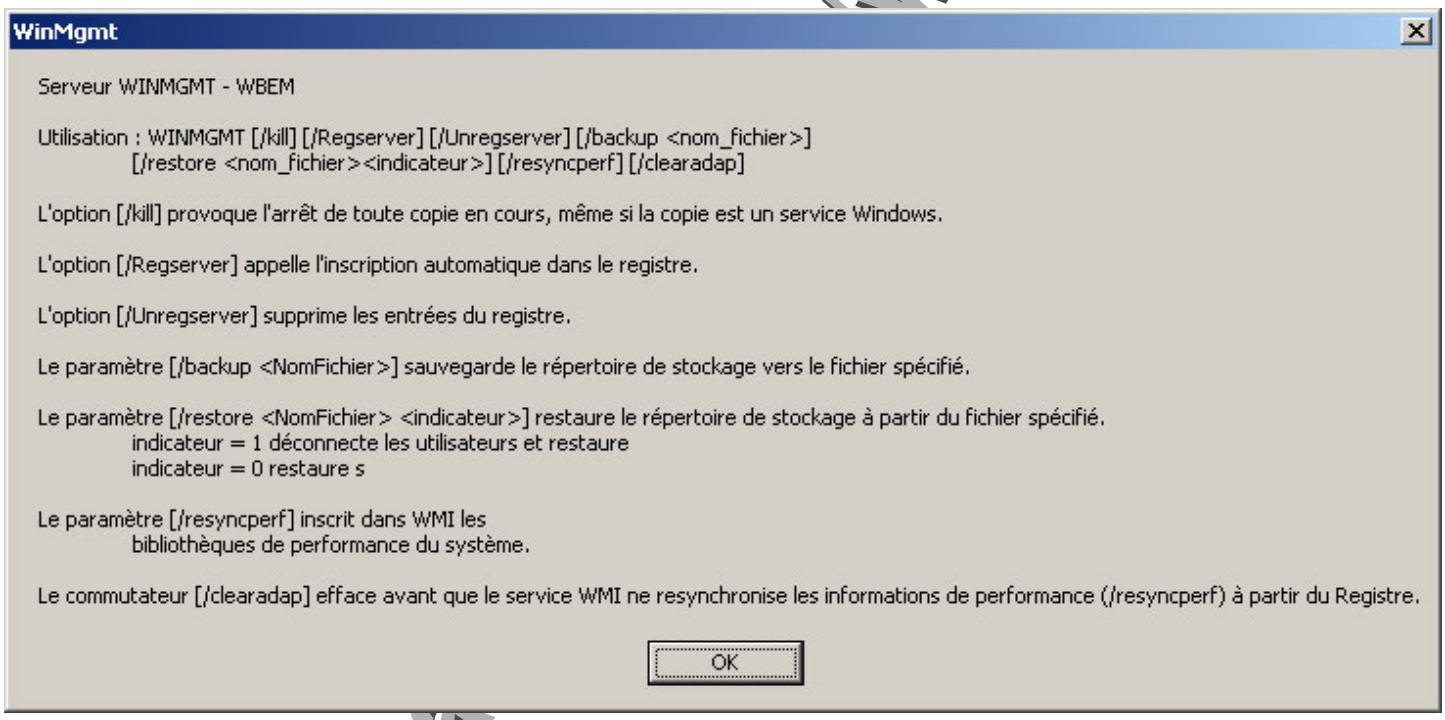
2. Exemples de commandes

```
WMIC PROCESS
WMIC PROCESS WHERE (Description="explorer.exe")
WMIC PROCESS GET /?
WMIC PROCESS GET Name, Handle, PageFaults
wmic SERVICE WHERE (Started=1) GET Name,Started
```

E. L'utilitaire Scriptomatic

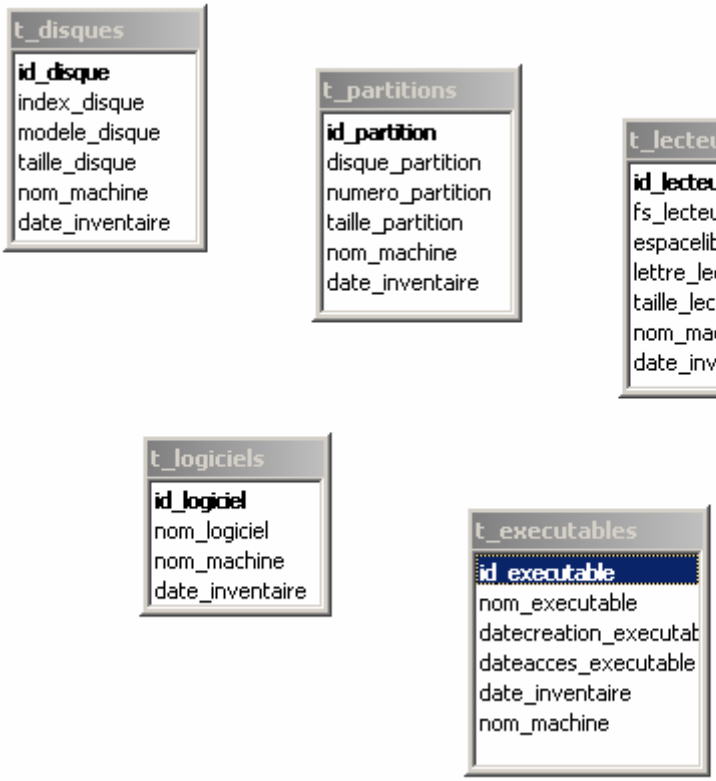
Afin d'éviter à taper de fastidieuses lignes de code, vous pouvez télécharger l'utilitaire scriptomatic qui vous permettra par des operations Copier/Coller à extraire les informations don't vous avez besoin pour votre application

F. Winmgmt : le service WMI



G. Une application complète

1. La base Access INVENTAIRE.MDB



2. Le script de connexion

```
Dim CHAINE_CONNECT,SQL,TEMP,DEBUT
DEBUT=Now
On Error Resume Next
Dim
O_SHELL,O_CONNECT,O_WMI,O_ELEMENTS,O_NETWORK,O_FSO,O_FICHIER,ELEMENT,O_REP,O_REPS,O_FICHIERS,FICHIE
R,REP

Set O_SHELL=WScript.CreateObject( "WScript.Shell" )
'
'Connexion à la base de données
'
Set O_CONNECT=CreateObject("ADODB.Connection")
CHAINE_CONNECT="DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" &  "e:\INVENTAIRE.mdb"
O_CONNECT.Open CHAINE_CONNECT
'
'Récupération infos partition WMI
'
strComputer = "."
Set O_WMI = GetObject("winmgmts:\\" & strComputer & "\root\cimv2")
Set O_ELEMENTS = O_WMI.ExecQuery("Select * from Win32_DiskPartition",,48)
For Each ELEMENT in O_ELEMENTS
    With ELEMENT
        SQL="INSERT INTO
t_partitions(disque_partition,numero_partition,taille_partition,nom_machine,date_inventaire) " & _
        "VALUES(" & .DiskIndex & "," & .Index & "," & Int(.Size/1048576) & ",'" & .SystemName & "','" & Now & "')"
        O_CONNECT.Execute(SQL)
        If Err.Number<>0 Then
            SQL="UPDATE t_partitions set taille_partition=" & Int(.Size/1048576) & ",date_inventaire='" & Now & "'" & _
```

```
            " WHERE disque_partition=" & .DiskIndex & " AND numero_partition=" & .Index & " AND nom_machine='" &
.SystemName & "'"
            O_CONNECT.Execute(SQL)
        End If
    End With
Next
'
'Récupération infos disque WMI
'

Set O_ELEMENTS = O_WMI.ExecQuery("Select * from Win32_DiskDrive",,48)
For Each ELEMENT in O_ELEMENTS
    With ELEMENT
        SQL="INSERT INTO t_disques(modele_disque,index_disque,taille_disque,nom_machine,date_inventaire) " & _
        "VALUES('" & .Model & "'," & .Index & "," & Int(.Size/1048576) & ",'" & .SystemName & "','" & Now & "')"
        O_CONNECT.Execute(SQL)
        If Err.Number<>0 Then
            SQL="UPDATE t_disques set date_inventaire='" & Now & "' " & _
            " WHERE modele_disque='" & .Model & "' AND nom_machine='" & .SystemName & "' AND index_disque=" &
.Index
            O_CONNECT.Execute(SQL)
        End If
    End With
Next

Set O_ELEMENTS = O_WMI.ExecQuery("Select * from Win32_LogicalDisk",,48)
For Each ELEMENT in O_ELEMENTS
    With ELEMENT
        'If .Description="Disque fixe local" Then
            SQL="INSERT INTO
t_lecteurs(fs_lecteur,espacelibre_lecteur,taille_lecteur,lettre_lecteur,nom_machine,date_inventaire) " & _
            "VALUES('" & .FileSystem & "'," & Int(.FreeSpace/1048576) & "," & Int(.Size/1048576) & ",'" & .Name & "','" &
.SystemName & "','" & Now & "')"
            O_CONNECT.Execute(SQL)
            If Err.Number<>0 Then
                SQL="UPDATE t_lecteurs set fs_lecteur='" & .FileSystem  & "',taille_lecteur =" & Int(.Size/1048576) &
",espacelibre_lecteur=" & Int(.FreeSpace/1048576) & ",date_inventaire='" & Now & "' " & _
                " WHERE lettre_lecteur='" & .Name & "' AND nom_machine='" & .SystemName & "'"
                O_CONNECT.Execute(SQL)
            End If
        'Enf If
    End With
Next
'
'Inventaire des logiciels installés Sur l'ordinateur
'
TEMP=O_SHELL.ExpandEnvironmentStrings("%TEMP%")
Set O_NETWORK=CreateObject("Wscript.Network")
MACHINE=O_NETWORK.ComputerName
o_SHELL.Run "reg export HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall " & TEMP & "\softs.txt",0,True
Set O_FSO=CreateObject("Scripting.FileSystemObject")
Set O_FICHIER=O_FSO.OpenTextFile(TEMP & "\softs.txt",1,False,-1)
Do While Not O_FICHIER.AtEndOfStream
    v_LIGNE=O_FICHIER.ReadLine
    If Left(V_LIGNE,13)="""DisplayName""" Then
        V_LIGNE=Mid(V_LIGNE, 16,Len(V_LIGNe)-16)
        SQL="INSERT INTO t_logiciels(nom_logiciel,nom_machine,date_inventaire) " & _
        "VALUES('" & V_LIGNE & "','" & MACHINE & "','" & Now & "')"
        O_CONNECT.Execute(SQL)
        If Err.Number<>0 Then
```

```
            SQL="UPDATE t_logiciels set date_inventaire='" & Now & "' " & _
            " WHERE nom_logiciel='" & V_LIGNE & "' AND nom_machine='" & MACHINE & "'"
            O_CONNECT.Execute(SQL)
        End If
    End IF
Loop
O_FSO.DeleteFile(TEMP & "\softs.txt")
'
'Scanner des exécutables
'

Sub ScanFolder(folderSpec)
    Dim Fc,thisFolder,AllFiles
    Set thisFolder = O_FSO.GetFolder(folderSpec)
    Set fc = thisFolder.SubFolders
    Set AllFiles=thisFolder.Files
    For Each iFile in AllFiles
        With iFile
            If .Type="Application" Then
                SQL="INSERT INTO
t_executables(nom_executable,datecreation_executable,dateacces_executable,nom_machine,date_inventaire) " & _
                "VALUES('" & .Name & "','" & .DateCreated & "','" & .DateLastAccessed & "','" & MACHINE & "','" & Now & "')"
                O_CONNECT.Execute(SQL)
                If Err.Number<>0 Then
                    SQL="UPDATE t_executables set date_inventaire='" & Now & "',dateacces_executable='" &
.DateLastAccessed & "' " & _
                    " WHERE nom_executable='" & .Name & "' AND nom_machine='" & MACHINE & "' AND
datecreation_executable='" & .DateCreated & "'"
                    O_CONNECT.Execute(SQL)
                    Continue
                End If
            End If
        End With
    Next
    For Each iFolder In Fc
        ScanFolder(iFolder.path)
    Next
End Sub

ScanFolder("C:\Program Files")
ScanFolder("C:\WINDOWS")



Set FICHIER=Nothing
Set O_FICHIERS=Nothing
Set O_REPS=Nothing
Set O_REP=Nothing
Set O_WMI=Nothing
Set O_ELEMENTS=Nothing
Set O_NETWORK=Nothing
Set ELEMENT=Nothing
Set O_CONNECT=Nothing
Set O_FICHIER=Nothing
Set O_FSO=Nothing
Set O_SHELL=Nothing
MsgBox (Now-DEBUT)*86400
Wscript.quit
```

H. Autres Exemples

1. La liste des propriétés d'une classe (ListePropriétéss.vbs)

```
Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\" & strComputer & "\root\cimv2")
Set objClass = objWMIService.Get("Win32_BIOS")
PROPERTIES=""
For Each objProperty in objClass.properties_
    PROPERTIES=PROPERTIES &  objProperty.name & Chr(10)
Next
WScript.Echo PROPERTIES
Set objClass = objWMIService.Get("Win32_GroupUser")
PROPERTIES=""
For Each objProperty in objClass.properties_
    PROPERTIES=PROPERTIES &  objProperty.name & Chr(10)
Next
WScript.Echo PROPERTIES
```

2. Lancement et arrêt d'une application

a) Lancement

```
Option Explicit
Dim PROCESS,RES,PID
Set PROCESS=GetObject("winmgmts://.").Get("Win32_Process")
RES=PROCESS.Create("calc.exe",Null,Null,PID)
MsgBox "Processus n° : " & PID
Set PID=Nothing
Set RES=Nothing
Set PROCESS=Nothing
WScript.Quit
```

b) Arrêt

```
Option Explicit
Dim PROCESS
For Each PROCESS In GetObject("winmgmts://.").ExecQuery("SELECT * FROM Win32_Process WHERE Name='calc.exe'")
   PROCESS.Terminate(0)
Next
Set PROCESS=Nothing
```

3. Affiche la liste des processus

```
Dim FLAG
FLAG=true
For Each PROCESS in GetObject("winmgmts:{impersonationLevel=impersonate}").InstancesOf ("Win32_process")
'Variante : affiche la liste des processus sur une machine distante
' For Each PROCESS in GetObject("winmgmts:{impersonationLevel=impersonate}!//ServerName").InstancesOf
("Win32_process")
'
   If FLAG Then
      MESSAGE=PROCESS.Name
      FLAG=false
   Else
      MESSAGE=MESSAGE & Chr(10) & PROCESS.Name
   End If
Next
Wscript.Echo MESSAGE
```

4. La liste des services

```
'Nom donné à la machine courante
'
```

```
'strComputer = "."
'
strComputer = "."
'
'Création d'un objet Wmi
'
'Set objWMIService = GetObject("winmgmts:\\" & strComputer & "\root\cimv2")
'
'Création d'un fichier sur le disque contenant la liste des classes
'
Set Obj_Fso=CreateObject("Scripting.FileSystemObject")
Set OBJ_FICHIER=OBJ_FSO.CreateTextFile("e:\classeswmi.txt")
CLASSES=""
'For Each objclass in objWMIService.SubclassesOf()
'    CLASSES=CLASSES & objClass.Path_.Class & Chr(11)
'Next
'OBJ_FICHIER.Write CLASSES
'
'Fermeture et destruction des références au fichier
'
OBJ_FICHIER.Close
Set OBJ_FICHIER=Nothing
Set OBJ_FSO=Nothing
'wScript.echo CLASSES
'
'Chargement des élements de la classe Win32_Service à l'aide du langage WQL
'
'Set colItems = objWMIService.ExecQuery("Select * from Win32_Service where State='Running'")
'Set colItems = GetObject("Winmgmts:").ExecQuery("Select * from Win32_Service")
'
'Autre méthode pour lire les élements de la classe
'
Set wbemServices = GetObject("winmgmts:\\" & strComputer)
Set colItems = wbemServices.InstancesOf("Win32_Service")
'AFFICHE="Name" & Chr(9) & "State" & chr(13)
For Each objItem in colItems
    'AFFICHE=AFFICHE & objItem.Name & Chr(9) & objItem.State & chr(13)
     AFFICHE=AFFICHE & objItem.Name(0) & chr(13)
Next
wScript.echo AFFICHE
```

5. Autre exemple : la liste des services

```
Dim FLAG
FLAG=true
For Each SERVICE in GetObject("winmgmts:{impersonationLevel=impersonate}").InstancesOf ("Win32_service")
'Variante : affiche la liste des processus sur une machine distante
' For Each SERVICE in GetObject("winmgmts:{impersonationLevel=impersonate}!//ServerName").InstancesOf
("Win32_service")
'
    If FLAG Then
        MESSAGE=SERVICE.Name
        FLAG=false
    Else
        MESSAGE=MESSAGE & Chr(10) & SERVICE.Name
    End If
Next
Wscript.Echo MESSAGE
```

6. Affiche la mémoire

```
On Error Resume Next
Dim o_WMISERVICE,o_COLLECTION, o_ELEMENT, str_MESSAGE,str_ORDI,str_REQ
str_ORDI = "."
Set o_WMISERVICE= GetObject("winmgmts:\\" & str_ORDI & "\root\cimv2")
str_REQ = "Select TotalPhysicalMemory From Win32_LogicalMemoryConfiguration"
Set o_COLLECTION = o_WMISERVICE.ExecQuery(str_REQ)
For Each o_ELEMENT in o_COLLECTION
    If Err.Number Then
        Err.Clear
    Else
        If o_ELEMENT is nothing Then
            Exit For
        Else
            str_MESS = o_ELEMENT.TotalPhysicalMemory & " KB"
            Wscript.Echo str_MESS
        End If
    End If
Next
```

7. Autre méthode : affichage de la mémoire

```
strComputer = "."
Set wbemServices = GetObject("winmgmts:\\" & strComputer)
Set wbemObjectSet = wbemServices.InstancesOf("Win32_LogicalMemoryConfiguration")
For Each wbemObject In wbemObjectSet
    MsgBox "Mémoire physique totale (ko): " & wbemObject.TotalPhysicalMemory
Next
```

15. Changement du mon associé à une lecteur logique

```
set disk = GetObject ("winmgmts:{impersonationLevel=impersonate}!//./root/cimv2:Win32_LogicalDisk=""C:""")
disk.VolumeName = "My C Drive"
disk.Put_
```

17. Autre exemple : afficher la mémoire vive

```
str_REQ = "Select TotalPhysicalMemory From Win32_LogicalMemoryConfiguration"
On Error Resume Next
Dim o_COLLECTION, o_ELEMENT, str_MESSAGE,o_SERVICE
Set o_COLLECTION = o_SERVICE.ExecQuery(str_REQ)
For Each o_ELEMENT in o_COLLECTION
    If Err.Number Then
        Err.Clear
    Else
        If o_ELEMENT is nothing Then
            Exit Sub
        Else
            str_MESS = Space(6) & o_ELEMENT.TotalPhysicalMemory & " KB"
            Wscript.Echo str_MESS
        End If
    End If
Next
```

18. Autre exemple : affiche la liste des processus locaux

```
Set oCIM = GetObject("cim:")
for each Process in oCIM.CreateInstanceEnum("Win32_process")
    WScript.Echo Process.Name
Next
```

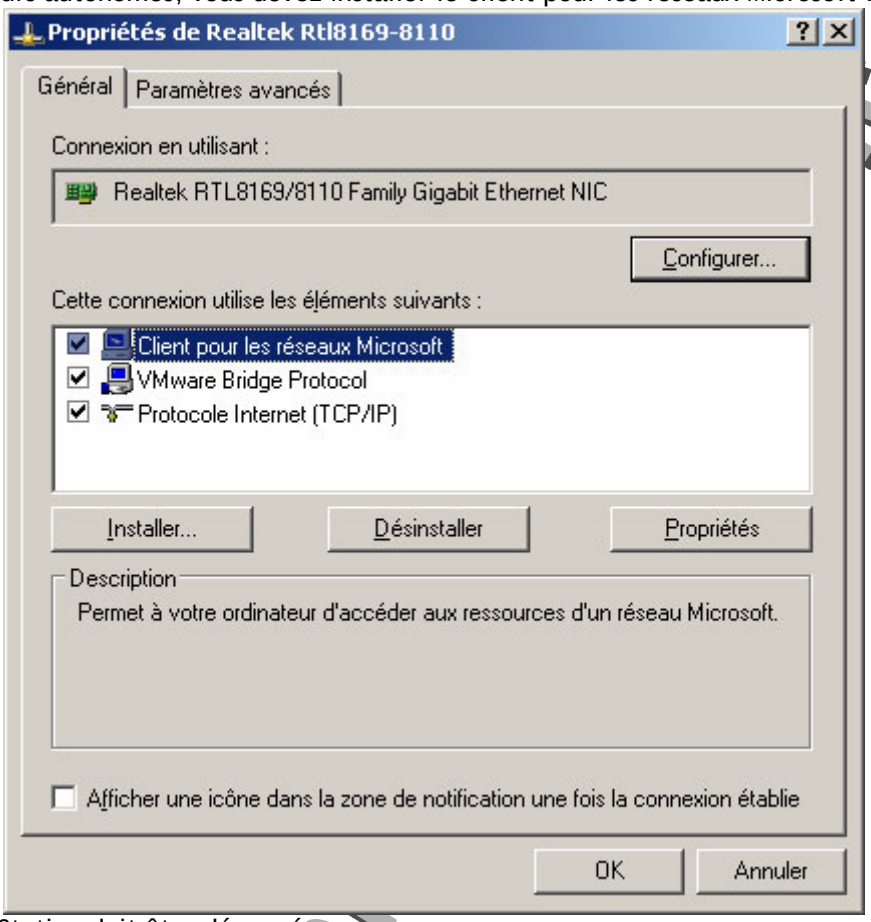
19. Affiche la liste des processus d'une machine distante

```
Set oCIM = GetObject("cim://marketing1")
for each Service in oCIM.CreateInstanceEnum("Win32_service")
   WScript.Echo Service.DisplayName
Next
```

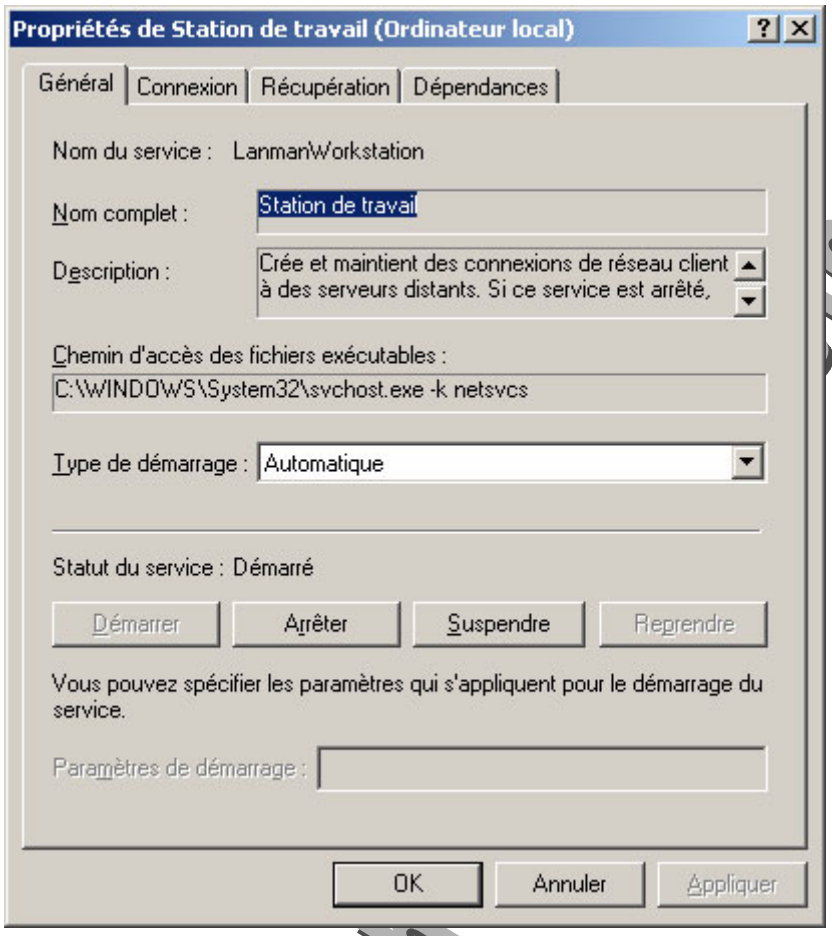
V. ADSI Active Directory Service Interfaces

A. Conditions d'utilisation de ADSI

Sur les stations et serveurs autonomes, vous devez installer le client pour les réseaux Microsoft :



Le service LanmanWorkStation doit être démarré :



B. Création d'un utilisateur sur une station de travail ou un serveur autonome

```
Option Explicit
Dim LOGIN,PWD,DOMAINE,UTILISATEUR
LOGIN=InputBox("Tapez le login :","Login du nouvel utilisateur")
PWD=InputBox("Tapez le mot de passe :","Mot de passe de l'utilisateur")
Set DOMAINE=GetObject("WinNT://.")
Set UTILISATEUR=DOMAINE.Create("USER",LOGIN)
UTILISATEUR.SetInfo
UTILISATEUR.SetPassword PWD
UTILISATEUR.SetInfo
Set UTILISATEUR=Nothing
Set DOMAINE=Nothing
WScript.Quit
```

C. Propriétés des utilisateurs

ADSI Property	Active Directory Property
AccountDisabled	userAccountControl Mask
AccountExpirationDate	accountExpires
BadLoginAddress	Not Supported
BadLoginCount	badPwdCount
Department	department
Description	description
Division	division
EmailAddress	mail
EmployeeID	employeeID
FaxNumber	facsimileTelephoneNumber

ADSI Property	Active Directory Property
FirstName	givenName
FullName	displayName
GraceLoginsAllowed	Not Supported
GraceLoginsRemaining	Not Supported
HomeDirectory	homeDirectory
HomePage	wWWHomePage
IsAccountLocked	userAccountControl
Languages	language
LastFailedLogin	badPasswordTime
LastLogin	lastLogon
LastLogoff	lastLogoff
LastName	sn
LoginHours	logonHours
LoginScript	scriptPath
LoginWorkstations	userWorkstations
Manager	manager
MaxLogins	Not Supported
MaxStorage	maxStorage
NamePrefix	personalTitle
NameSuffix	generationQualifier
OfficeLocations	physicalDeliveryOfficeName
OtherName	middleName
PasswordExpirationDate	Set using Group Policy Editor*
PasswordLastChanged	pwdLastSet
PasswordMinimumLength	Set using Group Policy Editor*
PasswordRequired	userAccountControl mask
Picture	thumbnailPhoto
PostalAddresses	postalAddress
PostalCodes	postalCode
Profile	profilePath
RequireUniquePassword	Set using Group Policy Editor*
SeeAlso	seeAlso
TelephoneHome	homePhone
TelephoneMobile	mobile
TelephoneNumber	telephoneNumber
TelephonePager	pager
Title	title

D. Interface Asp

1. La saisie d'un nouvel utilisateur : creation_user.html

```
<html>
<body>
   <form method="post" action="creation_user.asp">
      <table border="0">
         <tr>
            <td>Login</td>
            <td><input type="text" size="50" name="LOGIN"/></td>
         </tr>
```

```
        <tr>
            <td>Nom</td>
            <td><input type="text" size="50" name="NOM"/></td>
        </tr>
        <tr>
            <td>Pwd</td>
            <td><input type="password" size="50" name="PWD"/></td>
    </tr>
        <td><input type="submit" value="Ok"/></td>
        <td><input type="button" value="Retour" onclick="location.href='liste.asp'"></td>
    </table>
</form>
</body>
</html>
```

2. Le script Asp de Création de l'utilisateur : creation_user.asp

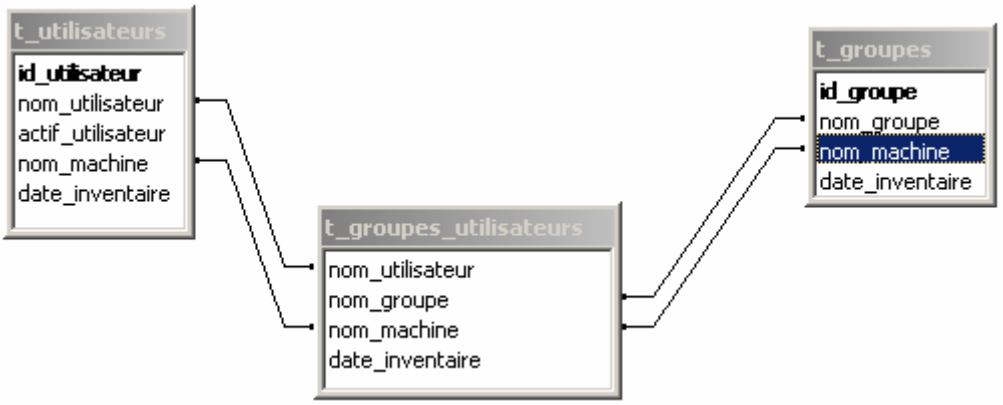
```
<%@ language="VBSCRIPT"%>
<%
Option Explicit
Dim LOGIN,NOM,PWD,DOMAINE,UTILISATEUR
LOGIN=Request.Form("LOGIN")
NOM=Request.Form("NOM")
PWD=Request.Form("PWD")
'Response.Write LOGIN & " " & NOM & " " & PWD
Set DOMAINE=GetObject("WinNT://.")
Set UTILISATEUR=DOMAINE.Create("USER",LOGIN)
UTILISATEUR.SetPassword PWD
UTILISATEUR.SetInfo
Set UTILISATEUR=Nothing
Set DOMAINE=Nothing
Response.Redirect("liste.asp")
%>
```

3. La liste des utilisateurs : liste.asp

```
<%@ language="VBSCRIPT" %>
<html>
    <body>
    <%
        Dim DOMAIN,USER
        Set DOMAIN=GetObject("WinNT://.,computer")
        DOMAIN.Filter=Array("User")
        For Each USER In DOMAIN
            Response.Write (USER.Name & "<br/>")
        Next
        Set USER=Nothing
        Set DOMAIN=Nothing
    %>
</body>
</html>
```

E. Une application consignant les utilisateurs et les groupes

1. La base de données



2. Le script

```
Option Explicit

Dim CONN,DSN,SQL
Set CONN=CreateObject("ADODB.Connection")
DSN="DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" &  "D:\SUPPORTS\WINDOWS\WshWmiAdsi\INVENTAIRE.mdb"
CONN.Open DSN

Dim NETWORK,COMPUTER
Set NETWORK=CreateObject("WScript.Network")
COMPUTER=NETWORK.ComputerName

On Error Resume Next
SQL="DELETE FROM t_utilisateurs WHERE nom_machine='" & COMPUTER & "'"
CONN.Execute(SQL)
SQL="DELETE FROM t_groupes WHERE nom_machine='" & COMPUTER & "'"
CONN.Execute(SQL)
SQL="DELETE FROM t_groupes_utilisateurs WHERE nom_machine='" & COMPUTER & "'"
CONN.Execute(SQL)

Dim DOMAINE,GROUPE,USER,ACTIF
Set DOMAINE= GetObject("WinNT://.")

DOMAINE.Filter = Array("User")
For Each USER In DOMAINE
        ACTIF=CBool(USER.AccountDisabled)
        If ACTIF=-1 Then
            ACTIF=0
        Else
            ACTIF=-1
        End If
        SQL="INSERT INTO t_utilisateurs(nom_utilisateur,actif_utilisateur,nom_machine,date_inventaire) " & _
        " VALUES('" & USER.Name & "'," & ACTIF  & ",'" & COMPUTER & "','" & Now & "')"
        CONN.Execute(SQL)
Next

DOMAINE.Filter = Array("Group")
For Each GROUPE In DOMAINE
        SQL="INSERT INTO t_groupes(nom_groupe,nom_machine,date_inventaire) " & _
        " VALUES('" & GROUPE.Name & "','" & COMPUTER & "','" & Now & "')"
        CONN.Execute(SQL)
        For Each USER In GROUPE.Members
```

```
        SQL="INSERT INTO t_groupes_utilisateurs(nom_utilisateur,nom_groupe,nom_machine,date_inventaire) " & _
        " VALUES('" & USER.Name & "','" & GROUPE.Name & "','" & COMPUTER & "','" & Now & "')"
        CONN.Execute(SQL)
    Next
Next
```

F. Les Chaînes de connexion

Fournisseur	Domaine
`WinNT:`	Permet l'administration de serveurs NT4 (PDC & BDC), de stations de travails et de serveurs autonomes
`LDAP:`	S'appliquent aux serveurs LDAP et Active Directory
`NDS:`	Accès aux serveurs Netware disposant de l'annuaire NDS
`NWCOMPAT:`	Accès aux serveurs Netware

1. WinNT

Chemin	Intérêt
`WinNT:`	Connexion à l'espace de nom
`WinNT://domaine`	Connexion au domaine "domaine"
`WinNT://domaine/polo,user`	Connexion à l'utilisateur polo du domaine
`WinNT://domaine/dc/informaticiens, group`	Connexion au groupe des informaticiens du contrôleur de domaine dc appartenant au domaine
`WinNT://serveur,computer`	Connexion à la machine "serveur"

2. Ldap

Chemin	Signification
`LDAP://dc`	Connexion au contrôleur de domaine dc
`LDAP://machine.domaine.com`	Connexion au domaine
`LDAP://exchsvr:390`	Connexion à un serveur avec specification du port
`LDAP://server/cn=users,ou=info,dc=dsfc,dc=fr`	Connexion à l'objet représentant la liste des utilisateurs

3. Exemples Ldap

a) Liste des objets d'une organisation (ListeOu.vbs)

```
Option Explicit
Dim OU, ELEMENT,MSG
Set OU=GetObject("LDAP://127.0.0.1:389/ou=info,dc=dsfc,dc=fr")
For ELEMENT In OU
   MSG=MSG & ELEMENT.Class & Chr(9) & ELEMENT.Name & Chr(13)
Next
Set ELEMENT=Nothing
Set OU=Nothing
```

b) Créer un utilisateur dans une organisation (CreationUsrOu.vbs)

```
Option Explicit
Dim OU, USER
Set OU=GetObject("LDAP://127.0.0.1:389/ou=info,dc=dsfc,dc=fr")
Set USER=OU.Create("User","cn=Essai")
With USER
   .Put "sAMAccountName","Essai"
   .Put "givenName","Roger"
   .Put "sn","Rabbit"
   .Put "description","Créature féroce"
   .Put "streetAddress","12 boulevard Hollywood"
```

```
    .Put "l","Paris"
    .Put "st","Ile de France"
    .Put "postalCode","75000"
    .Put "telephoneNumber","01.47.47.47.47"
    .SetInfo
    .SetPassword "Choucroute"
    .AccountDisabled=False
    .SetInfo
End With
Set USER=Nothing
Set OU=Nothing
```

G. Ado

```
Dim Con As New Connection

Set con = CreateObject("ADODB.Connection")
con.Provider = "ADsDSOObject"

Dim command As New Command

Set command = CreateObject("ADODB.Command")
Set command.ActiveConnection = con
command.CommandText =
"SELECT AdsPath, cn, FROM '<LDAP://DC=ArcadiaBay,DC=com>'
 WHERE objectCategory='person' AND objectClass='user' AND sn = 'Johnson' ORDER BY sn"

Set command.ActiveConnection = con
command.CommandText =

"<LDAP://DC=ArcadiaBay,DC=com(&(objectClategory=person)(objectClass=user)(sn=Johnson));Ad
sPath, cn; subTree"

Dim Com As New Command

Com.Properties("Page Size") = 100
Com.Properties("Timeout") = 30    'seconds
Com.Properties("searchscope") = ADS_SCOPE_ONELEVEL     'Define in ADS_SCOPEENUM
Com.Properties("Chase referrals") = ADS_CHASE_REFERRALS_EXTERNAL
Com.Properties("Cache Results") = False   'do not cache the result set

Set rs = Com.Execute

For i = 0 To rs.Fields.Count - 1
  Debug.Print rs.Fields(i).Name, rs.Fields(i).Type
Next i

'---------------------------------------
' Navigate the record set
'---------------------------------------
rs.MoveFirst
lstResult.Clear      'Clear the user interface
While Not rs.EOF
For i = 0 To rs.Fields.Count - 1
   'For Multi Value attribute
    If rs.Fields(i).Type = adVariant And Not (IsNull(rs.Fields(i).Value)) Then
      Debug.Print rs.Fields(i).Name, " = "
      For j = LBound(rs.Fields(i).Value) To UBound(rs.Fields(i).Value)
         Debug.Print rs.Fields(i).Value(j), " # "
         lstResult.AddItem rs.Fields(i).Value(j)
      Next j
```

```
    Else
      'For Single Value attribute
       Debug.Print rs.Fields(i).Name, " = ", rs.Fields(i).Value
       lstResult.AddItem rs.Fields(i).Value
    End If
Next i
rs.MoveNext
Wend

'Replacing department for all users in OU=sales
Set con = Server.CreateObject("ADODB.Connection")
con.Provider = "ADsDSOObject"

Set command = CreateObject("ADODB.Command")
Set command.ActiveConnection = con

command.CommandText =
"SELECT AdsPath, cn, FROM '<LDAP://OU=Sales,DC=ArcadiaBay,DC=com>'
 WHERE objectClass = 'user'"

Com.Properties("searchscope") = ADS_SCOPE_ONELEVEL
Set rs = Com.Execute
While Not rs.EOF
  Set usr = GetObject(rs.Fields('AdsPath').Value)
  usr.Put "department", "1001"
  usr.SetInfo
  rs.MoveNext
Wend
```

1. Affichage des champs

```
Dim X as IADs
Dim con As New Connection, rs As New Recordset
Dim MyUser As IADsUser

con.Provider = "ADsDSOObject"
con.Open "Active Directory Provider", "CN=Foobar,CN=Users,DC=ARCADIABAY,DC=COM,O=INTERNET", "Password"
Set rs =
con.Execute("<LDAP://MyMachine/DC=MyDomain,DC=ArcadiaBay,DC=com>;(objectClass=User);ADsPath;onelevel")

While Not rs.EOF
    ' Bind to the object to make changes
    ' to it since ADO is currently read-only
    MyUser = GetObject(rs.Fields(0).Value)
    MyUser.AccountDisabled = True
    rs.MoveNext
Wend
```

H. Méthodes

1. Get

```
Dim MyUser as IADs
Dim MyCommonName as String

' Bind to a specific user object.
set MyUser = GetObject("LDAP://CN=JamesSmith,OU=MyOrgUnit")

' Get property.
MyCommonName = MyUser.Get("CN")
```

```
Dim MyUser as IADs
Dim MyName as String

' Bind to a specific user object.
Set MyUser = GetObject("LDAP://MyMachine/CN=JamesSmith,DC=ArcadiaBay,DC=COM")

' Get property
MyName = MyUser.Name

Dim MyUser as IADs
Dim MyName as String

' Bind to a specific user object.
set MyUser = GetObject("LDAP://CN=JamesSmith,OU=MyOrgUnit")

' Get property
MyName = MyUser.Get("distinguishedName")
```

2. GetEx

```
Dim obj As IADs
Dim objList As Variant

Set obj = GetObject("LDAP://MyMachine/CN=Administrator,CN=Users,DC=ArcadiaBay,DC=com")

objList = obj.GetEx("description")

For Each Desc In objList
    ' Print the descriptions
    Debug.Print (Desc)
Next
```

3. GetInfo

```
Dim MyUser as IADsUser
'MyUser will be used to demonstrate implicit GetInfo
Dim MyUser2 as IADsUser
'Myuser2 will show the explicit GetInfo

' Bind to a specific user object.
set MyUser = GetObject("LDAP://MyMachine/CN=JamesSmith,DC=ArcadiaBay,DC=COM")
set MyUser2 = GetObject("LDAP//MyMachine/CN=JamesSmith2,DC=ArcadiaBay,DC=COM");

'Perform some time consuming operations.

'Code assumes that the property description has a single value in the directory
' Note that this will IMPLICITLY call GetInfo as at the point this call is made GetInfo
' has not yet been called (implicitly or explicitly) on the MyUser object.
Debug.print "MyUser's description value is "; MyUser.Get("Description")

'Since the GetInfo has already been called implicitly this call is satisfied from
' the value in the cache.
Debug.print "MyUser's sAMAccountName is "; MyUser.Get("sAMAccountName")

' Refresh the cache explicitly so the most current value is available
MyUser2.GetInfo

'Perform time consuming operations

'Note that this call is satisfied from the cache has GetInfo has already been called
'explicitly for this object.
```

```
Debug.print "MyUser2 has the description set to "; MyUser.Get("Description")
```

4. GetInfoEx

```
GetInfoEx
```

5. Put

```
Dim Namespace As IADsOpenDSObject
Dim User As IADsUser
Dim NewName As Variant

Set Namespace = GetObject("LDAP:")

Set User =
Namespace.OpenDSObject("LDAP://MyMachine/CN=Administrator,CN=Users,DC=MyDomain,DC=ArcadiaBay,DC=COM",
"Administrator", "", ADS_SECURE_AUTHENTICATION)

NewName = "James Smith"

' Set using IADs::PutMethod
User.Put "FullName", NewName
User.SetInfo
```

6. PutEx

```
Dim x As IADs
Set x = GetObject("LDAP://CN=Administrator,CN=Users,DC=ArcadiaBay,DC=com")
'-----------------------------------------------
' Assume the otherHomePhoneNumber has the following values:
' 111-1111, 222-2222
'-----------------------------------------------
x.PutEx ADS_PROPERTY_APPEND, "OtherhomePhone", Array("333-3333" )
x.SetInfo          'Now the values are 111-1111,222-222,333-3333

x.PutEx ADS_PROPERTY_DELETE, "OtherHomePhone", Array("111-1111", "222-2222")
x.SetInfo          'Now the values are 333-3333
x.PutEx ADS_PROPERTY_UPDATE, "OtherHomePhone", Array("888-8888", "999-9999")
x.SetInfo          'Now the values are 888-8888,999-9999
x.PutEx ADS_PROPERTY_CLEAR, "OtherHomePhone",  vbNull
x.SetInfo          'Now the property has no value
```

I. Propriétés

```
Dim propList As IADsPropertyList
Dim propEntry As IADsPropertyEntry
Dim propValue As IADsPropertyValue
Dim rootDSE As IADs

'Get the rootDSE entry that contains configuration information
Set rootDSE = GetObject("LDAP://RootDSE")

'Bind to the domain
Set propList = GetObject("LDAP://" & rootDSE.Get("defaultNamingContext"))

'Get the Property Entries

Set propEntry = propList.GetPropertyItem("allowedChildClassesEffective", ADSTYPE_CASE_IGNORE_STRING)

'Get more info about Property Entry
Debug.Print propEntry.ADsType
Debug.Print propEntry.ControlCode
Debug.Print propList.PropertyCount
```

```
'Values contain pointers to IADsPropertyValues interfaces
For Each v In propEntry.Values
    Set propValue = v
    Debug.Print propValue.CaseIgnoreString
Next
```

J. Générer une feuille LDIF

LGETATTCLS.VBS

```
On Error Resume Next

''''''''''''''''''''''''''''''''''''''''
'Bind to the rootDSE
''''''''''''''''''''''''''''''''''''''''
sPrefix = "LDAP://"
Set root= GetObject(sPrefix & "rootDSE")
If (Err.Number <> 0) Then
   BailOnFailure Err.Number, "on GetObject method"
End If

''''''''''''''''''''''''''''''''''''''''
'Get the DN for the Schema
''''''''''''''''''''''''''''''''''''''''
sSchema = root.Get("schemaNamingContext")
If (Err.Number <> 0) Then
   BailOnFailure Err.Number, "on Get method"
End If

''''''''''''''''''''''''''''''''''''''''
'Bind to the Schema container
''''''''''''''''''''''''''''''''''''''''
Set Schema= GetObject(sPrefix & sSchema )
If (Err.Number <> 0) Then
   BailOnFailure Err.Number, "on GetObject method to bind to Schema"
End If
''''''''''''''''''''''''''''''''''''''''
'Read the fsmoRoleOwner attribute to see which server is the schema master.
''''''''''''''''''''''''''''''''''''''''
sMaster = Schema.Get("fsmoRoleOwner")
If (Err.Number <> 0) Then
   BailOnFailure Err.Number, "on IADs::Get method for fsmoRoleOwner"
End If
''''''''''''''''''''''''''''''''''''''''
'fsmoRoleOwner attribute returns the nTDSDSA object.
'The parent is the server object.
'Bind to NTDSDSA object and get parent
''''''''''''''''''''''''''''''''''''''''
Set NTDS = GetObject(sPrefix & sMaster)
If (Err.Number <> 0) Then
   BailOnFailure Err.Number, "on GetObject method for NTDS"
End If
sServer = NTDS.Parent
If (Err.Number <> 0) Then
   BailOnFailure Err.Number, "on IADs::get_Parent method"
End If
''''''''''''''''''''''''''''''''''''''''
'Bind to server object
'and get the reference to the computer object.
''''''''''''''''''''''''''''''''''''''''
```

```
Set Server = GetObject(sServer)
If (Err.Number <> 0) Then
   BailOnFailure Err.Number, "on GetObject method for " & sServer
End If
''''''''''''''''''''''''''''''''''''''''
'Display the DN for the computer object.
''''''''''''''''''''''''''''''''''''''''
sComputerDNSName = Server.Get("DNSHostName")
strText = "Schema Master has the following DNS Name: "& sComputerDNSName
WScript.echo strText

sFile = "myschemaext.ldf"
sFromDN = sSchema
sToDN = "CN=Schema,CN=Configuration,DC=myorg,DC=com"
sAttrPrefix = "My-Test"
sFilter = "(&((cn=" & sAttrPrefix & "*)(|(objectCategory=classSchema)(objectCategory=attributeSchema))))"
sRetAttr =
"dn,adminDescription,adminDisplayName,governsID,cn,mayContain,mustContain,systemMayContain,systemMustContain,lDAPDisplayName,objectClassCategory,distinguishedName,objectCategory,objectClass,possSuperiors,systemPossSuperiors,subClassOf,defaultObjectCategory,name,schemaIDGUID,auxiliaryClass,auxiliaryClass,systemAuxiliaryClass,description,defaultHidingValue,rDNAttId,defaultSecurityDescriptor,attributeID,attributeSecurityGUID,attributeSyntax,isMemberOfPartialAttributeSet,isSingleValued,mAPIID,oMSyntax,rangeLower,rangeUpper,searchFlags,oMObjectClass,linkID"

'Add flag rootDN--we want schema.
sCommand = "ldifde -d " & sSchema
sCommand = sCommand & " -c " & sFromDN & " " & sToDN
'Add flag schema master
sCommand = sCommand & " -s " & sComputerDNSName
'Add flag filename
sCommand = sCommand & " -f " & sFile
'Add flag filter to search for my attributes
sCommand = sCommand & " -r " & sFilter
'Add flag for attributes to return
sCommand = sCommand & " -l " & sRetAttr

WScript.echo sCommand
Set WshShell = Wscript.CreateObject("Wscript.Shell")
WshShell.Run (sCommand)


''''''''''''''''''''''''''''''''''''''''
'Display subroutines
''''''''''''''''''''''''''''''''''''''''

Sub BailOnFailure(ErrNum, ErrText)    strText = "Error 0x" & Hex(ErrNum) & " " & ErrText
    MsgBox strText, vbInformation, "ADSI Error"
    WScript.Quit
End Sub
```

K. Enumératyion des ACL

```
Dim X as IADs
Dim Namespace As IADsOpenDSObject
Dim SecurityDescriptor As IADsSecurityDescriptor
Dim Dacl As IADsAccessControlList

' First get access to the LDAP directory service
Set Namespace = GetObject("LDAP:")
' Establish your credentials with no password
Set X = Namespace.OpenDSObject("LDAP://MyLdapSvr/O=Internet/DC=MS",
    "cn=administrator,DC=MS,O=Internet", "", 1)
```

```
' Retrieve the contents of the ntSecurityDescriptor field which is
'    an interface pointer on the security descriptor object
'    for the X object
Set SecurityDescriptor = X.Get("ntSecurityDescriptor")
' Print out the owner of the object
' Print out the group
Debug.Print SecurityDescriptor.Owner
Debug.Print SecurityDescriptor.Group
'
Set Dacl = SecurityDescriptor.DiscretionaryAcl
' Enumerate the ACEs in the Dacl, first printing out the count
Debug.Print Dacl.AceCount

For Each Obj In Dacl
   Debug.Print Obj.Trustee
   Debug.Print Obj.AccessMask
   Debug.Print Obj.AceFlags
   Debug.Print Obj.AceType
Next
```

*L. Propriétés de LDAP://rootDSE ou **LDAP://servername/rootDSE***

Property	Description
currentTime	Current time set on this directory server.
subschemaSubentry	Distinguished name for the subSchema object. The subSchema object contains properties that expose the supported attributes (in the attributeTypes property) and classes (in the objectClasses property). The subschemaSubentry property and subschema are defined in LDAP 3.0 (see RFC 2251).
dsServiceName	The distinguished name of the NTDS settings object for this directory server.
namingContexts	Multi-valued. DISTINGUISHED NAMEs for all naming contexts stored on this directory server. By default, a Windows 2000 domain controller contains at least three namespaces: Schema, Configuration, and one for the domain of which the server is a member.
defaultNamingContext	By default, the distinguished name for the domain of which this directory server is a member.
schemaNamingContext	Distinguished name for the schema container.
configurationNamingContext	Distinguished name for the configuration container.
RootDomainNamingContext	Distinguished name for the first domain in the forest that contains the domain of which this directory server is a member.
SupportedControl	Multi-valued. OIDs for extension controls supported by this directory server.
SupportedLDAPVersion	Multi-valued. LDAP versions (specified by major version number) supported by this directory server.
HighestCommittedUSN	Highest USN used on this directory server. Used by directory replication.
SupportedSASLMechanisms	Security mechanisms supported for SASL negotiation (see LDAP RFCs). By default, GSSAPI is supported.
DnsHostName	DNS address for this directory server.
LdapServiceName	Service Principal Name (SPN) for the LDAP server. Used for mutual authentication.
ServerName	Distinguished name for the server object for this directory server in the configuration container.

M. Connexions NDS

1. Connexion.

```
Dim dso
Dim obj
Dim usrName
Dim password
Dim serverName
```

```
servername = "ntmarst2"
userName = "supervisor.ntmarst2"
password = "secretpwd"

'--bind to the server
Set dso = GetObject("NDS:")
Set cont = dso.OpenDSObject("NDS://" & serverName,userName,password,0)

'--enumerate the server's objects
For Each obj In cont
   Debug.Print obj.Name & " (" & obj.Class & ")"
Next
```

2. Propriétés d'un utilisateur

```
Path = "O=NTMARST2/CN=benny"
ADsPath = "NDS://" & serverName & "/" & Path
Set usr = dso.OpenDSObject(ADsPath, userName, password, 0)
Debug.Print usr.Get("Surname")
usr.Put "SurName", "Johnson"
usr.SetInfo
```

3. Création d'un utilisateur

```
Path = "O=NTMARST2"
ADsPath = "NDS://" & serverName & "/" & Path
Set cont = dso.OpenDSObject(ADsPath, userName, password, 0)
Set usr = cont.Create("user", "bagheeraw")
usr.Put "cn", "Bagheera"
usr.Put "Surname", "White"
usr.SetInfo
```

4. Recherche d'un utilisateur

```
ADsPath = "NDS://" & serverName
Set con = CreateObject("ADODB.Connection")
con.Provider = "ADsDSOObject"
con.Properties("User ID") = userName
con.Properties("Password") = password
con.Open "ADSI"

Set com = CreateObject("ADODB.Command")
Set com.ActiveConnection = con
com.CommandText = "SELECT ADsPath, 'Object Class' FROM '" & ADsPath & "' WHERE Surname='Bagheera'"
Set rs = com.Execute

While Not (rs.EOF)
  Debug.Print rs.Fields("ADsPath")
  rs.MoveNext
Wend
```

N. Connexion Netware

1. Tester une connexion à un serveur Netware

```
net use \\nwserver /U:mydom\jsmith
```

2. Exemple de Code

```
servername = "ntmarst2"
adsPathName = "NWCOMPAT://" & serverName
Set cont = GetObject(adsPathName)
```

```
'--enumerate the server's objects
For Each obj In cont
   Debug.Print obj.Name & " (" & obj.Class & ")"
Next
```

This example creates a new user with a username of "alicew".

```
adsPath = "NWCOMPAT://" & serverName
Set cont = GetObject(adsPath)
Set usr = cont.Create("user", "alicew")
usr.SetInfo
```

This example changes a particular user's full name to "Alice I. Wonderland".

```
objPath = "alicew,user"
adsPath = "NWCOMPAT://" & serverName & "/" & objPath
Set usr = GetObject(adsPath)
usr.FullName = "Alice I. Wonderland"
usr.SetInfo
```

Searching is not supported in NWCOMPAT. You can, however, use the IADsContainer::put_Filter method to limit the type of object classes to be included in the enumeration.

```
adsPath = "NWCOMPAT://" & serverName
Set con = GetObject(adsPath)
con.Filter = Array("user", "group") 'Show user and group

For Each acct In con
   Debug.Print acct.Name & " (" & acct.Class & ")"
Next
```

VI. Annexe : objets et mots-clés Microsoft Scripting

Objet	Mot-clé
Collections	Drives Files Folders
Data Storage	Dictionary
Dictionary	Add Exists Items, Keys Remove, RemoveAll Count Item, Key
File System	Drive File FileSystemObject Folder TextStream
FileSystemObject	BuildPath CopyFile, CopyFolder CreateFolder,CreateTextFile DeleteFile,DeleteFolder DriveExists, FileExists, FolderExists GetAbsolutePathName, GetBaseName GetDrive, GetDriveName GetFile, GetExtensionNameGetFileName GetFolder, GetParentFolderName GetSpecialFolder GetTempName MoveFile, MoveFolder OpenTextFile Drives
Drive, Drives	AvailableSpace Count DriveLetter DriveType FileSystem FreeSpace IsReady Item RootFolder SerialNumber ShareName TotalSize VolumeName
File, Files Folder, Folders	Add Attributes Copy, Delete, Move Count OpenAsTextStream DateCreated, DateLastAccessed, DateLastModified Drive Item ParentFolder Name, Path ShortName, ShortPath Size
TextStream	Close

	Read, ReadAll, ReadLine Skip, SkipLine Write, WriteBlankLines, WriteLine AtEndOfLine, AtEndOfStream Column, Line

VII. Annexe : les classes Wmi

__AbsoluteTimerInstruction
__AggregateEvent
__ClassCreationEvent
__ClassDeletionEvent
__ClassModificationEvent
__ClassOperationEvent
__ClassProviderRegistration
__ConsumerFailureEvent
__Event
__EventConsumer
__EventConsumerProviderRegistration
__EventDroppedEvent
__EventFilter
__EventGenerator
__EventProviderRegistration
__EventQueueOverflowEvent
__ExtendedStatus
__ExtrinsicEvent
__FilterToConsumerBinding
__IndicationRelated
__InstanceCreationEvent
__InstanceDeletionEvent
__InstanceModificationEvent
__InstanceOperationEvent
__InstanceProviderRegistration
__IntervalTimerInstruction
__MethodInvocationEvent
__MethodProviderRegistration
__NAMESPACE
__NamespaceCreationEvent
__NamespaceDeletionEvent
__NamespaceModificationEvent
__NamespaceOperationEvent
__NotifyStatus
__NTLMUser9X
__ObjectProviderRegistration
__PARAMETERS
__PropertyProviderRegistration
__Provider
__ProviderRegistration
__QOSFailureEvent
__SecurityRelatedClass
__SystemClass
__SystemEvent
__SystemSecurity
__thisNAMESPACE
__TimerEvent
__TimerInstruction
__TimerNextFiring
__Win32Provider
__WmiMappedDriverNamespace
CIM_Action
CIM_ActionSequence
CIM_ActsAsSpare
CIM_AdjacentSlots
CIM_AggregatePExtent
CIM_AggregatePSExtent
CIM_AggregateRedundancyComponent
CIM_AlarmDevice
CIM_AllocatedResource
CIM_ApplicationSystem
CIM_ApplicationSystemSoftwareFeature
CIM_AssociatedAlarm
CIM_AssociatedBattery
CIM_AssociatedCooling
CIM_AssociatedMemory
CIM_AssociatedProcessorMemory
CIM_AssociatedSensor
CIM_AssociatedSupplyCurrentSensor
CIM_AssociatedSupplyVoltageSensor
CIM_BasedOn
CIM_Battery
CIM_BinarySensor
CIM_BIOSElement
CIM_BIOSFeature
CIM_BIOSFeatureBIOSElements
CIM_BIOSLoadedInNV
CIM_BootOSFromFS
CIM_BootSAP
CIM_BootService
CIM_BootServiceAccessBySAP
CIM_CacheMemory
CIM_Card
CIM_CardInSlot
CIM_CardOnCard
CIM_CDROMDrive
CIM_Chassis
CIM_ChassisInRack
CIM_Check
CIM_Chip
CIM_ClusteringSAP
CIM_ClusteringService
CIM_ClusterServiceAccessBySAP
CIM_CollectedCollections
CIM_CollectedMSEs
CIM_CollectionOfMSEs
CIM_CollectionOfSensors
CIM_CollectionSetting
CIM_CompatibleProduct
CIM_Component
CIM_ComputerSystem
CIM_ComputerSystemDMA
CIM_ComputerSystemIRQ
CIM_ComputerSystemMappedIO
CIM_ComputerSystemPackage
CIM_ComputerSystemResource
CIM_Configuration
CIM_ConnectedTo
CIM_ConnectorOnPackage
CIM_Container
CIM_ControlledBy
CIM_Controller
CIM_CoolingDevice
CIM_CopyFileAction
CIM_CreateDirectoryAction
CIM_CurrentSensor
CIM_DataFile
CIM_Dependency
CIM_DependencyContext
CIM_DesktopMonitor
CIM_DeviceAccessedByFile
CIM_DeviceConnection
CIM_DeviceErrorCounts
CIM_DeviceFile
CIM_DeviceSAPImplementation
CIM_DeviceServiceImplementation
CIM_DeviceSoftware
CIM_Directory
CIM_DirectoryAction
CIM_DirectoryContainsFile
CIM_DirectorySpecification
CIM_DirectorySpecificationFile
CIM_DiscreteSensor
CIM_DiskDrive
CIM_DisketteDrive
CIM_DiskPartition
CIM_DiskSpaceCheck
CIM_Display
CIM_DMA
CIM_Docked
CIM_ElementCapacity
CIM_ElementConfiguration
CIM_ElementSetting
CIM_ElementsLinked
CIM_ErrorCountersForDevice
CIM_ExecuteProgram
CIM_Export
CIM_ExtraCapacityGroup
CIM_Fan
CIM_FileAction
CIM_FileSpecification
CIM_FileStorage
CIM_FileSystem
CIM_FlatPanel
CIM_FromDirectoryAction
CIM_FromDirectorySpecification
CIM_FRU
CIM_FRUIncludesProduct
CIM_FRUPhysicalElements
CIM_HeatPipe
CIM_HostedAccessPoint
CIM_HostedBootSAP
CIM_HostedBootService
CIM_HostedFileSystem
CIM_HostedJobDestination
CIM_HostedService
CIM_InfraredController
CIM_InstalledOS
CIM_InstalledSoftwareElement
CIM_IRQ

```
CIM_Job
CIM_JobDestination
CIM_JobDestinationJobs
CIM_Keyboard
CIM_LinkHasConnector
CIM_LocalFileSystem
CIM_Location
CIM_LogicalDevice
CIM_LogicalDisk
CIM_LogicalDiskBasedOnPartition
CIM_LogicalDiskBasedOnVolumeSet
CIM_LogicalElement
CIM_LogicalFile
CIM_LogicalIdentity
CIM_MagnetoOpticalDrive
CIM_ManagedSystemElement
CIM_ManagementController
CIM_MediaAccessDevice
CIM_MediaPresent
CIM_Memory
CIM_MemoryCapacity
CIM_MemoryCheck
CIM_MemoryMappedIO
CIM_MemoryOnCard
CIM_MemoryWithMedia
CIM_ModifySettingAction
CIM_MonitorResolution
CIM_MonitorSetting
CIM_Mount
CIM_MultiStateSensor
CIM_NetworkAdapter
CIM_NFS
CIM_NonVolatileStorage
CIM_NumericSensor
CIM_OperatingSystem
CIM_OperatingSystemSoftwareFeature
CIM_OSProcess
CIM_OSVersionCheck
CIM_PackageAlarm
CIM_PackageCooling
CIM_PackagedComponent
CIM_PackageInChassis
CIM_PackageInSlot
CIM_PackageTempSensor
CIM_ParallelController
CIM_ParticipatesInSet
CIM_PCIController
CIM_PCMCIAController
CIM_PCVideoController
CIM_PExtentRedundancyComponent
CIM_PhysicalCapacity
CIM_PhysicalComponent
CIM_PhysicalConnector
CIM_PhysicalElement
CIM_PhysicalElementLocation
CIM_PhysicalExtent
CIM_PhysicalFrame
CIM_PhysicalLink
CIM_PhysicalMedia
CIM_PhysicalMemory
CIM_PhysicalPackage
CIM_PointingDevice
CIM_PotsModem
CIM_PowerSupply
CIM_Printer
CIM_Process
CIM_ProcessExecutable
CIM_Processor
CIM_ProcessThread
CIM_Product
CIM_ProductFRU
CIM_ProductParentChild
CIM_ProductPhysicalElements
CIM_ProductProductDependency
CIM_ProductSoftwareFeatures
CIM_ProductSupport
CIM_ProtectedSpaceExtent
CIM_PSExtentBasedOnPExtent
CIM_Rack
CIM_Realizes
CIM_RealizesAggregatePExtent
CIM_RealizesDiskPartition
CIM_RealizesPExtent
CIM_RebootAction
CIM_RedundancyComponent
CIM_RedundancyGroup
CIM_Refrigeration
CIM_RelatedStatistics
CIM_RemoteFileSystem
CIM_RemoveDirectoryAction
CIM_RemoveFileAction
CIM_ReplacementSet
CIM_ResidesOnExtent
CIM_RunningOS
CIM_SAPSAPDependency
CIM_Scanner
CIM_SCSIController
CIM_SCSIInterface
CIM_Sensor
CIM_SerialController
CIM_SerialInterface
CIM_Service
CIM_ServiceAccessBySAP
CIM_ServiceAccessPoint
CIM_ServiceSAPDependency
CIM_ServiceServiceDependency
CIM_Setting
CIM_SettingCheck
CIM_SettingContext
CIM_Slot
CIM_SlotInSlot
CIM_SoftwareElement
CIM_SoftwareElementActions
CIM_SoftwareElementChecks
CIM_SoftwareElementVersionCheck
CIM_SoftwareFeature
CIM_SoftwareFeatureSAPImplementation
CIM_SoftwareFeatureServiceImplementation
CIM_SoftwareFeatureSoftwareElements
CIM_SpareGroup
CIM_StatisticalInformation
CIM_Statistics
CIM_StorageDefect
CIM_StorageError
CIM_StorageExtent
CIM_StorageRedundancyGroup
CIM_SupportAccess
CIM_SwapSpaceCheck
CIM_System
CIM_SystemComponent
CIM_SystemDevice
CIM_SystemResource
CIM_Tachometer
CIM_TapeDrive
CIM_TemperatureSensor
CIM_Thread
CIM_ToDirectoryAction
CIM_ToDirectorySpecification
CIM_UninterruptiblePowerSupply
CIM_UnitaryComputerSystem
CIM_USBController
CIM_USBControllerHasHub
CIM_USBDevice
CIM_USBHub
CIM_UserDevice
CIM_VersionCompatibilityCheck
CIM_VideoBIOSElement
CIM_VideoBIOSFeature
CIM_VideoBIOSFeatureVideoBIOSElements
CIM_VideoController
CIM_VideoControllerResolution
CIM_VideoSetting
CIM_VolatileStorage
CIM_VoltageSensor
CIM_VolumeSet
CIM_WORMDrive
CmdTriggerConsumer
MSFT_ForwardedEvent
MSFT_ForwardedMessageEvent
MSFT_NCProvAccessCheck
MSFT_NCProvCancelQuery
MSFT_NCProvClientConnected
MSFT_NCProvEvent
MSFT_NCProvNewQuery
MSFT_NetBadAccount
MSFT_NetBadServiceState
MSFT_NetBootSystemDriversFailed
MSFT_NetCallToFunctionFailed
MSFT_NetCallToFunctionFailedII
MSFT_NetCircularDependencyAuto
MSFT_NetCircularDependencyDemand
MSFT_NetConnectionTimeout
MSFT_NetDependOnLaterGroup
MSFT_NetDependOnLaterService
```

MSFT_NetFirstLogonFailed
MSFT_NetFirstLogonFailedII
MSFT_NetInvalidDriverDependency
MSFT_NetReadfileTimeout
MSFT_NetRevertedToLastKnownGood
MSFT_NetServiceConfigBackoutFailed
MSFT_NetServiceControlSuccess
MSFT_NetServiceCrash
MSFT_NetServiceCrashNoAction
MSFT_NetServiceExitFailed
MSFT_NetServiceExitFailedSpecific
MSFT_NetServiceNotInteractive
MSFT_NetServiceRecoveryFailed
MSFT_NetServiceStartFailed
MSFT_NetServiceStartFailedGroup
MSFT_NetServiceStartFailedII
MSFT_NetServiceStartFailedNone
MSFT_NetServiceStartHung
MSFT_NetServiceStatusSuccess
MSFT_NetSevereServiceFailed
MSFT_NetTakeOwnership
MSFT_NetTransactInvalid
MSFT_NetTransactTimeout
Msft_Providers
MSFT_SCMEvent
MSFT_SCMEventLogEvent
MSFT_WMI_GenericNonCOMEvent
MSFT_WmiCancelNotificationSink
MSFT_WmiConsumerProviderEvent
MSFT_WmiConsumerProviderLoaded
MSFT_WmiConsumerProviderSinkLoaded
MSFT_WmiConsumerProviderSinkUnloaded
MSFT_WmiConsumerProviderUnloaded
MSFT_WmiCoreEvent
MSFT_WmiCoreLogoffEvent
MSFT_WmiCoreLogonEvent
MSFT_WmiCoreObject
MSFT_WmiCoreStatus
MSFT_WmiCoreTaskFailure
MSFT_WmiCoreUser
MSFT_WmiEssEvent
MSFT_WmiFilterActivated
MSFT_WmiFilterDeactivated
MSFT_WmiFilterEvent
Msft_WmiProvider_AccessCheck_Post
Msft_WmiProvider_AccessCheck_Pre
Msft_WmiProvider_CancelQuery_Post
Msft_WmiProvider_CancelQuery_Pre
Msft_WmiProvider_ComServerLoadOperationEvent
Msft_WmiProvider_ComServerLoadOperationFailureEvent
Msft_WmiProvider_Counters
Msft_WmiProvider_CreateClassEnumAsyncEvent_Post
Msft_WmiProvider_CreateClassEnumAsyncEvent_Pre
Msft_WmiProvider_CreateInstanceEnumAsyncEvent_Post
Msft_WmiProvider_CreateInstanceEnumAsyncEvent_Pre
Msft_WmiProvider_DeleteClassAsyncEvent_Post
Msft_WmiProvider_DeleteClassAsyncEvent_Pre
Msft_WmiProvider_DeleteInstanceAsyncEvent_Post
Msft_WmiProvider_DeleteInstanceAsyncEvent_Pre
Msft_WmiProvider_ExecMethodAsyncEvent_Post
Msft_WmiProvider_ExecMethodAsyncEvent_Pre
Msft_WmiProvider_ExecQueryAsyncEvent_Post
Msft_WmiProvider_ExecQueryAsyncEvent_Pre
Msft_WmiProvider_GetObjectAsyncEvent_Post
Msft_WmiProvider_GetObjectAsyncEvent_Pre
Msft_WmiProvider_InitializationOperationEvent
Msft_WmiProvider_InitializationOperationFailureEvent
Msft_WmiProvider_LoadOperationEvent
Msft_WmiProvider_LoadOperationFailureEvent
Msft_WmiProvider_NewQuery_Post
Msft_WmiProvider_NewQuery_Pre
Msft_WmiProvider_OperationEvent
Msft_WmiProvider_OperationEvent_Post
Msft_WmiProvider_OperationEvent_Pre
Msft_WmiProvider_ProvideEvents_Post
Msft_WmiProvider_ProvideEvents_Pre
Msft_WmiProvider_PutClassAsyncEvent_Post
Msft_WmiProvider_PutClassAsyncEvent_Pre
Msft_WmiProvider_PutInstanceAsyncEvent_Post
Msft_WmiProvider_PutInstanceAsyncEvent_Pre
Msft_WmiProvider_UnLoadOperationEvent
MSFT_WmiProviderEvent
MSFT_WmiRegisterNotificationSink
MSFT_WmiSelfEvent
MSFT_WmiThreadPoolEvent
MSFT_WmiThreadPoolThreadCreated
MSFT_WmiThreadPoolThreadDeleted
NetDiagnostics
NTEventlogProviderConfig
ScriptingStandardConsumerSetting
Win32_1394Controller
Win32_1394ControllerDevice
Win32_Account
Win32_AccountSID
Win32_ACE
Win32_ActionCheck
Win32_ActiveRoute
Win32_AllocatedResource
Win32_ApplicationCommandLine
Win32_ApplicationService
Win32_AssociatedBattery
Win32_AssociatedProcessorMemory
Win32_AutochkSetting
Win32_BaseBoard
Win32_BaseService
Win32_Battery
Win32_Binary
Win32_BindImageAction
Win32_BIOS
Win32_BootConfiguration
Win32_Bus
Win32_CacheMemory
Win32_CDROMDrive
Win32_CheckCheck
Win32_CIMLogicalDeviceCIMDataFile
Win32_ClassicCOMApplicationClasses
Win32_ClassicCOMClass
Win32_ClassicCOMClassSetting
Win32_ClassicCOMClassSettings
Win32_ClassInfoAction
Win32_ClientApplicationSetting
Win32_CodecFile
Win32_CollectionStatistics
Win32_COMApplication
Win32_COMApplicationClasses
Win32_COMApplicationSettings
Win32_COMClass
Win32_ComClassAutoEmulator
Win32_ComClassEmulator
Win32_CommandLineAccess
Win32_ComponentCategory
Win32_ComputerShutdownEvent
Win32_ComputerSystem
Win32_ComputerSystemEvent
Win32_ComputerSystemProcessor
Win32_ComputerSystemProduct
Win32_ComputerSystemWindowsProductActivationSetting
Win32_COMSetting
Win32_Condition
Win32_ConnectionShare
Win32_ControllerHasHub
Win32_CreateFolderAction
Win32_CurrentProbe
Win32_CurrentTime
Win32_DCOMApplication
Win32_DCOMApplicationAccessAllowedSetting
Win32_DCOMApplicationLaunchAllowedSetting
Win32_DCOMApplicationSetting
Win32_DependentService
Win32_Desktop
Win32_DesktopMonitor
Win32_DeviceBus
Win32_DeviceChangeEvent
Win32_DeviceMemoryAddress
Win32_DeviceSettings
Win32_Directory
Win32_DirectorySpecification
Win32_DiskDrive
Win32_DiskDrivePhysicalMedia
Win32_DiskDriveToDiskPartition
Win32_DiskPartition
Win32_DiskQuota
Win32_DisplayConfiguration
Win32_DisplayControllerConfiguration
Win32_DMAChannel
Win32_DriverForDevice
Win32_DriverVXD
Win32_DuplicateFileAction
Win32_Environment
Win32_EnvironmentSpecification
Win32_ExtensionInfoAction
Win32_Fan
Win32_FileSpecification
Win32_FloppyController

```
Win32_FloppyDrive
Win32_FontInfoAction
Win32_Group
Win32_GroupInDomain
Win32_GroupUser
Win32_HeatPipe
Win32_IDEController
Win32_IDEControllerDevice
Win32_ImplementedCategory
Win32_InfraredDevice
Win32_IniFileSpecification
Win32_InstalledSoftwareElement
Win32_IP4PersistedRouteTable
Win32_IP4RouteTable
Win32_IP4RouteTableEvent
Win32_IRQResource
Win32_JobObjectStatus
Win32_Keyboard
Win32_LaunchCondition
Win32_LoadOrderGroup
Win32_LoadOrderGroupServiceDependencies
Win32_LoadOrderGroupServiceMembers
Win32_LocalTime
Win32_LoggedOnUser
Win32_LogicalDisk
Win32_LogicalDiskRootDirectory
Win32_LogicalDiskToPartition
Win32_LogicalFileAccess
Win32_LogicalFileAuditing
Win32_LogicalFileGroup
Win32_LogicalFileOwner
Win32_LogicalFileSecuritySetting
Win32_LogicalMemoryConfiguration
Win32_LogicalProgramGroup
Win32_LogicalProgramGroupDirectory
Win32_LogicalProgramGroupItem
Win32_LogicalProgramGroupItemDataFile
Win32_LogicalShareAccess
Win32_LogicalShareAuditing
Win32_LogicalShareSecuritySetting
Win32_LogonSession
Win32_LogonSessionMappedDisk
Win32_LUID
Win32_LUIDandAttributes
Win32_ManagedSystemElementResource
Win32_MappedLogicalDisk
Win32_MemoryArray
Win32_MemoryArrayLocation
Win32_MemoryDevice
Win32_MemoryDeviceArray
Win32_MemoryDeviceLocation
Win32_MethodParameterClass
Win32_MIMEInfoAction
Win32_ModuleLoadTrace
Win32_ModuleTrace
Win32_MotherboardDevice
Win32_MoveFileAction
Win32_MSIResource
Win32_NamedJobObject
Win32_NamedJobObjectActgInfo
Win32_NamedJobObjectLimit
Win32_NamedJobObjectLimitSetting
Win32_NamedJobObjectProcess
Win32_NamedJobObjectSecLimit
Win32_NamedJobObjectSecLimitSetting
Win32_NamedJobObjectStatistics
Win32_NetworkAdapter
Win32_NetworkAdapterConfiguration
Win32_NetworkAdapterSetting
Win32_NetworkClient
Win32_NetworkConnection
Win32_NetworkLoginProfile
Win32_NetworkProtocol
Win32_NTDomain
Win32_NTEventlogFile
Win32_NTLogEvent
Win32_NTLogEventComputer
Win32_NTLogEventLog
Win32_NTLogEventUser
Win32_ODBCAttribute
Win32_ODBCDataSourceAttribute
Win32_ODBCDataSourceSpecification
Win32_ODBCDriverAttribute
Win32_ODBCDriverSoftwareElement
Win32_ODBCDriverSpecification
Win32_ODBCSourceAttribute
Win32_ODBCTranslatorSpecification
Win32_OnBoardDevice
Win32_OperatingSystem
Win32_OperatingSystemAutochkSetting
Win32_OperatingSystemQFE
Win32_OSRecoveryConfiguration
Win32_PageFile
Win32_PageFileElementSetting
Win32_PageFileSetting
Win32_PageFileUsage
Win32_ParallelPort
Win32_Patch
Win32_PatchFile
Win32_PatchPackage
Win32_PCMCIAController
Win32_Perf
Win32_PerfFormattedData
Win32_PerfFormattedData_ASP_ActiveServerPages
Win32_PerfFormattedData_ASPNET_114322_ASPNETAppsv11432
2
Win32_PerfFormattedData_ASPNET_114322_ASPNETv114322
Win32_PerfFormattedData_ASPNET_ASPNET
Win32_PerfFormattedData_ASPNET_ASPNETApplications
Win32_PerfFormattedData_InetInfo_InternetInformationSe
rvicesGlobal
Win32_PerfFormattedData_MSDTC_DistributedTransactionCo
ordinator
Win32_PerfFormattedData_NETFramework_NETCLRExceptions
Win32_PerfFormattedData_NETFramework_NETCLRInterop
Win32_PerfFormattedData_NETFramework_NETCLRJit
Win32_PerfFormattedData_NETFramework_NETCLRLoading
Win32_PerfFormattedData_NETFramework_NETCLRLocksAndThr
eads
Win32_PerfFormattedData_NETFramework_NETCLRMemory
Win32_PerfFormattedData_NETFramework_NETCLRRemoting
Win32_PerfFormattedData_NETFramework_NETCLRSecurity
Win32_PerfFormattedData_NTFSDRV_Pilotedebanquedinforma
tionsSMTPNTFS
Win32_PerfFormattedData_PerfDisk_LogicalDisk
Win32_PerfFormattedData_PerfDisk_PhysicalDisk
Win32_PerfFormattedData_PerfNet_Browser
Win32_PerfFormattedData_PerfNet_Redirector
Win32_PerfFormattedData_PerfNet_Server
Win32_PerfFormattedData_PerfNet_ServerWorkQueues
Win32_PerfFormattedData_PerfOS_Cache
Win32_PerfFormattedData_PerfOS_Memory
Win32_PerfFormattedData_PerfOS_Objects
Win32_PerfFormattedData_PerfOS_PagingFile
Win32_PerfFormattedData_PerfOS_Processor
Win32_PerfFormattedData_PerfOS_System
Win32_PerfFormattedData_PerfProc_FullImage_Costly
Win32_PerfFormattedData_PerfProc_Image_Costly
Win32_PerfFormattedData_PerfProc_JobObject
Win32_PerfFormattedData_PerfProc_JobObjectDetails
Win32_PerfFormattedData_PerfProc_Process
Win32_PerfFormattedData_PerfProc_ProcessAddressSpace_C
ostly
Win32_PerfFormattedData_PerfProc_Thread
Win32_PerfFormattedData_PerfProc_ThreadDetails_Costly
Win32_PerfFormattedData_PSched_PSchedFlow
Win32_PerfFormattedData_PSched_PSchedPipe
Win32_PerfFormattedData_RSVP_RSVPInterfaces
Win32_PerfFormattedData_RSVP_RSVPService
Win32_PerfFormattedData_SMTPSVC_ServeurSMTP
Win32_PerfFormattedData_Spooler_PrintQueue
Win32_PerfFormattedData_TapiSrv_Telephony
Win32_PerfFormattedData_Tcpip_ICMP
Win32_PerfFormattedData_Tcpip_IP
Win32_PerfFormattedData_Tcpip_NBTConnection
Win32_PerfFormattedData_Tcpip_NetworkInterface
Win32_PerfFormattedData_Tcpip_TCP
Win32_PerfFormattedData_Tcpip_UDP
Win32_PerfFormattedData_TermService_TerminalServices
Win32_PerfFormattedData_TermService_TerminalServicesSe
ssion
Win32_PerfFormattedData_W3SVC_WebService
Win32_PerfRawData
Win32_PerfRawData_ASP_ActiveServerPages
Win32_PerfRawData_ASPNET_114322_ASPNETAppsv114322
Win32_PerfRawData_ASPNET_114322_ASPNETv114322
Win32_PerfRawData_ASPNET_ASPNET
Win32_PerfRawData_ASPNET_ASPNETApplications
Win32_PerfRawData_InetInfo_InternetInformationServices
Global
Win32_PerfRawData_MSDTC_DistributedTransactionCoordina
tor
Win32_PerfRawData_NETFramework_NETCLRExceptions
Win32_PerfRawData_NETFramework_NETCLRInterop
```

```
Win32_PerfRawData_NETFramework_NETCLRJit
Win32_PerfRawData_NETFramework_NETCLRLoading
Win32_PerfRawData_NETFramework_NETCLRLocksAndThreads
Win32_PerfRawData_NETFramework_NETCLRMemory
Win32_PerfRawData_NETFramework_NETCLRRemoting
Win32_PerfRawData_NETFramework_NETCLRSecurity
Win32_PerfRawData_NTFSDRV_PilotedebanquedinformationsS
MTPNTFS
Win32_PerfRawData_PerfDisk_LogicalDisk
Win32_PerfRawData_PerfDisk_PhysicalDisk
Win32_PerfRawData_PerfNet_Browser
Win32_PerfRawData_PerfNet_Redirector
Win32_PerfRawData_PerfNet_Server
Win32_PerfRawData_PerfNet_ServerWorkQueues
Win32_PerfRawData_PerfOS_Cache
Win32_PerfRawData_PerfOS_Memory
Win32_PerfRawData_PerfOS_Objects
Win32_PerfRawData_PerfOS_PagingFile
Win32_PerfRawData_PerfOS_Processor
Win32_PerfRawData_PerfOS_System
Win32_PerfRawData_PerfProc_FullImage_Costly
Win32_PerfRawData_PerfProc_Image_Costly
Win32_PerfRawData_PerfProc_JobObject
Win32_PerfRawData_PerfProc_JobObjectDetails
Win32_PerfRawData_PerfProc_Process
Win32_PerfRawData_PerfProc_ProcessAddressSpace_Costly
Win32_PerfRawData_PerfProc_Thread
Win32_PerfRawData_PerfProc_ThreadDetails_Costly
Win32_PerfRawData_PSched_PSchedFlow
Win32_PerfRawData_PSched_PSchedPipe
Win32_PerfRawData_RSVP_RSVPInterfaces
Win32_PerfRawData_RSVP_RSVPService
Win32_PerfRawData_SMTPSVC_ServeurSMTP
Win32_PerfRawData_Spooler_PrintQueue
Win32_PerfRawData_TapiSrv_Telephony
Win32_PerfRawData_Tcpip_ICMP
Win32_PerfRawData_Tcpip_IP
Win32_PerfRawData_Tcpip_NBTConnection
Win32_PerfRawData_Tcpip_NetworkInterface
Win32_PerfRawData_Tcpip_TCP
Win32_PerfRawData_Tcpip_UDP
Win32_PerfRawData_TermService_TerminalServices
Win32_PerfRawData_TermService_TerminalServicesSession
Win32_PerfRawData_W3SVC_WebService
Win32_PhysicalMedia
Win32_PhysicalMemory
Win32_PhysicalMemoryArray
Win32_PhysicalMemoryLocation
Win32_PingStatus
Win32_PNPAllocatedResource
Win32_PnPDevice
Win32_PnPEntity
Win32_PnPSignedDriver
Win32_PnPSignedDriverCIMDataFile
Win32_PointingDevice
Win32_PortableBattery
Win32_PortConnector
Win32_PortResource
Win32_POTSModem
Win32_POTSModemToSerialPort
Win32_PowerManagementEvent
Win32_Printer
Win32_PrinterConfiguration
Win32_PrinterController
Win32_PrinterDriver
Win32_PrinterDriverDll
Win32_PrinterSetting
Win32_PrinterShare
Win32_PrintJob
Win32_PrivilegesStatus
Win32_Process
Win32_Processor
Win32_ProcessStartTrace
Win32_ProcessStartup
Win32_ProcessStopTrace
Win32_ProcessTrace
Win32_Product
Win32_ProductCheck
Win32_ProductResource
Win32_ProductSoftwareFeatures
Win32_ProgIDSpecification
Win32_ProgramGroup
Win32_ProgramGroupContents
Win32_ProgramGroupOrItem
Win32_Property
Win32_ProtocolBinding
Win32_Proxy
Win32_PublishComponentAction
Win32_QuickFixEngineering
Win32_QuotaSetting
Win32_Refrigeration
Win32_Registry
Win32_RegistryAction
Win32_RemoveFileAction
Win32_RemoveIniAction
Win32_ReserveCost
Win32_ScheduledJob
Win32_SCSIController
Win32_SCSIControllerDevice
Win32_SecurityDescriptor
Win32_SecuritySetting
Win32_SecuritySettingAccess
Win32_SecuritySettingAuditing
Win32_SecuritySettingGroup
Win32_SecuritySettingOfLogicalFile
Win32_SecuritySettingOfLogicalShare
Win32_SecuritySettingOfObject
Win32_SecuritySettingOwner
Win32_SelfRegModuleAction
Win32_SerialPort
Win32_SerialPortConfiguration
Win32_SerialPortSetting
Win32_ServerConnection
Win32_ServerSession
Win32_Service
Win32_ServiceControl
Win32_ServiceSpecification
Win32_ServiceSpecificationService
Win32_Session
Win32_SessionConnection
Win32_SessionProcess
Win32_SessionResource
Win32_SettingCheck
Win32_Share
Win32_ShareToDirectory
Win32_ShortcutAction
Win32_ShortcutFile
Win32_ShortcutSAP
Win32_SID
Win32_SIDandAttributes
Win32_SMBIOSMemory
Win32_SoftwareElement
Win32_SoftwareElementAction
Win32_SoftwareElementCheck
Win32_SoftwareElementCondition
Win32_SoftwareElementResource
Win32_SoftwareFeature
Win32_SoftwareFeatureAction
Win32_SoftwareFeatureCheck
Win32_SoftwareFeatureParent
Win32_SoftwareFeatureSoftwareElements
Win32_SoundDevice
Win32_StartupCommand
Win32_SubDirectory
Win32_SubSession
Win32_SystemAccount
Win32_SystemBIOS
Win32_SystemBootConfiguration
Win32_SystemConfigurationChangeEvent
Win32_SystemDesktop
Win32_SystemDevices
Win32_SystemDriver
Win32_SystemDriverPNPEntity
Win32_SystemEnclosure
Win32_SystemLoadOrderGroups
Win32_SystemLogicalMemoryConfiguration
Win32_SystemMemoryResource
Win32_SystemNetworkConnections
Win32_SystemOperatingSystem
Win32_SystemPartitions
Win32_SystemProcesses
Win32_SystemProgramGroups
Win32_SystemResources
Win32_SystemServices
Win32_SystemSetting
Win32_SystemSlot
Win32_SystemSystemDriver
Win32_SystemTimeZone
Win32_SystemTrace
Win32_SystemUsers
Win32_TapeDrive
Win32_TCPIPPrinterPort
Win32_TemperatureProbe
Win32_Terminal
```

```
Win32_TerminalError
Win32_TerminalService
Win32_TerminalServiceSetting
Win32_TerminalServiceSettingError
Win32_TerminalServiceToSetting
Win32_TerminalSetting
Win32_TerminalTerminalSetting
Win32_Thread
Win32_ThreadStartTrace
Win32_ThreadStopTrace
Win32_ThreadTrace
Win32_TimeZone
Win32_TokenGroups
Win32_TokenPrivileges
Win32_Trustee
Win32_TSAccount
Win32_TSClientSetting
Win32_TSClientSettingError
Win32_TSEnvironmentSetting
Win32_TSEnvironmentSettingError
Win32_TSGeneralSetting
Win32_TSGeneralSettingError
Win32_TSLogonSetting
Win32_TSLogonSettingError
Win32_TSNetworkAdapterListSetting
Win32_TSNetworkAdapterSetting
Win32_TSNetworkAdapterSettingError
Win32_TSPermissionsSetting
Win32_TSPermissionsSettingError
Win32_TSRemoteControlSetting
Win32_TSRemoteControlSettingError
Win32_TSSessionDirectory
Win32_TSSessionDirectoryError
Win32_TSSessionDirectorySetting
Win32_TSSessionSetting
Win32_TSSessionSettingError
Win32_TypeLibraryAction
Win32_UninterruptiblePowerSupply
Win32_USBController
Win32_USBControllerDevice
Win32_USBHub
Win32_UserAccount
Win32_UserDesktop
Win32_UserInDomain
Win32_UTCTime
Win32_VideoConfiguration
Win32_VideoController
Win32_VideoSettings
Win32_VoltageProbe
Win32_VolumeChangeEvent
Win32_VolumeQuotaSetting
Win32_WindowsProductActivation
Win32_WMIElementSetting
Win32_WMISetting
```