

SUPPORT VISUAL BASIC SCRIPT

TABLE DES MATIERES

1. GENERALITES.....	2
1.1. DEFINITION D'UN PROGRAMME.....	2
1.2. LES COMMANDES, FONCTIONS, PROCEDURES	2
1.3. LES OBJETS : METHODES ET EVENEMENTS	3
1.3.1. <i>Propriétés</i>	3
1.3.2. <i>Méthodes</i>	3
1.4. ECRITURE D'UN SCRIPT	3
1.5. WMI	4
2. LES VARIABLES.....	5
2.1. DECLARATIONS	5
2.1.1. <i>Déclarations</i>	5
2.1.2. <i>Portée</i>	6
3. LES FONCTIONS	7
3.1. CONVERSION DE VALEURS.....	7
3.2. TRAITEMENT DE CHAINES.....	7
3.3. DATE ET HEURE	8
3.4. MATHEMATIQUES.....	9
3.5. TABLEAUX	10
3.6. OBJETS	10
3.7. TESTS	10
3.8. DIALOGUES	11
3.9. INFORMATIONS.....	11
3.10. DIVERS	11
4. ENTREES/SORTIES DE DONNEES.....	12
4.1. SORTIES.....	12
4.2. ENTREES :.....	13
5. CALCULS ET COMPARAISONS.....	14
5.1. LES OPERATEURS DE COMPARAISON.....	14
5.2. LES CALCULS.....	14
6. BOUCLE ET TEST	15
6.1. IF...THEN	15
6.2. SELECT...CASE	15
6.3. FOR...NEXT	16
6.4. DO...LOOP	17
7. LES ERREURS.....	18
8. MODULES ET PROCEDURES.....	18
8.1. LES PROCEDURES :	18
8.2. PROCEDURES FUNCTION :	19
8.3. PROCEDURES SUB :	19
9. LES OBJETS :	20
9.1. OBJET GROUPE	20
9.2. OBJET USER	20
9.3. LES AUTRES OBJETS.....	20

1. GENERALITES

Visual Basic Script vous offre la possibilité de développer vos propres programmes sous l'environnement Windows un peu comme les fichiers bat sous Ms-Dos mais avec des possibilités plus importante. Le cadre de ce support est de faire des scripts permettant de gérer un réseau Microsoft à base de serveur et de clients Windows NT, 2000 et XP.

Visual Basic Scripting Edition, (**VBScript** en abrégé), est un langage de programmation dérivé de **Visual Basic for Applications** (**VBA**), lui-même issu de **Visual Basic**.

Il est destiné à la conception de **scripts**, c'est à dire des suites de **commandes**, destinés à être utilisés dans différents **environnements** :

- **Active Server Pages (ASP)**, qui désigne la technologie de pages dynamiques sur les serveurs HTTP Microsoft (IIS - Internet Information Server).
- **Formulaires Outlook**
- **Windows Script Host (WSH)**, sous-ensemble de Windows utilisant ces scripts pour réaliser diverses tâches automatisées.
- **Microsoft Internet Explorer (IE)**, où il est utilisé dans des scripts cotés **client**. Cette dernière utilisation présente peu d'intérêt, car les navigateurs non-Microsoft ne supporte pas les scripts.

VBScript est utilisable sous toute plate-forme **Windows 32 bits** (Windows 95,98, ME, NT4, 2000, XP)

1.1. Définition d'un programme

Suite logique et séquentiel d'instructions qui commande le traitement des diverses données ainsi que la gestion de l'environnement (clavier, écran, imprimante, ...)

La façon d'écrire les instructions dépend du langage de programmation utilisé. Chaque langage à ses spécificités d'écriture et notamment de syntaxe, mais beaucoup se ressemblent. A vrai dire, l'ordinateur ne comprend qu'un seul langage qui est le " langage machine ". Par conséquent le langage de programmation va jouer le rôle de traducteur entre vous et la machine de manière à rendre l'approche beaucoup plus facile et pratique.

1.2. Les commandes, fonctions, procédures

Les commandes sont des instructions connu de VB avec lesquelles vous lui passez des ordres. On parle de "mots-clé".

Les fonctions servent à calculer des nombres ou à manipuler du texte et retournent un résultat.

La syntaxe est **Function** nomfonction()

Vous pouvez combiner commandes et fonctions.

Par exemple vous demandez à VB : "d'afficher le calcul racine carré de 4"

commande fonction

Les procédures sont comme les fonctions mais ne retournent pas forcément un résultat.
La syntaxe est **Sub** nomprocédure()

1.3. Les objets : méthodes et événements

Dans VBS on utilise des classes d'objets comme la classe Group qui désigne les groupes d'utilisateurs. Avec cette classe Group on associe une méthode par exemple Create qui permettra d'ajouter un groupe. Les méthodes ne sont pas propre à une classe d'objet.

Chaque objet dispose de propriétés, de méthodes. Chaque objet porte un nom pour l'identifier.

1.3.1. Propriétés

Les propriétés sont par exemple, la couleur ou la légende des objets.
Pour les modifier par programme, vous pouvez donner le nom de l'objet, ainsi que la désignation de la propriété.

Voici la syntaxe générale :

NomObjet.Nompropriété=" "

1.3.2. Méthodes

Une commande est une instruction pour un processus interne : traitement de variable, test, boucle,...

Une méthode est une instruction relative à un processus au système tel que le traitement des entrées, des sorties.

En résumé, toutes les manipulations d'objet que ce soit des fenêtres, des contrôles, des imprimantes, sont des méthodes. On peut résumé ceci en disant que toutes les instructions adressés à un objet bien défini sont des méthodes.

1.4. Ecriture d'un script

Le plus simple est d'utiliser le bloc note mais en cas d'erreur de fonctionnement vous n'avez pas la numérotation de ligne. Attention il faut donner l'extension VBS pour pouvoir l'exécuter.

Je vous conseille de télécharger un éditeur de script, beaucoup plus pratique (numérotation des lignes). Personnellement j'utilise "VBS Factory" que l'on trouve sur www.telecharger.com.

1.5. WMI

Certains scripts **VBS** peuvent faire appel à **WMI** (*Windows Management Instrumentation*).

C'est un **fournisseur d'informations** diverses sur une machine **locale** ou **distante** (matériel - BIOS compris -, disques, mémoire, comptes, réseau, ...), constitué d'une (gigantesque) **base de données standardisée** (gérée par le processus **WinMgmt.exe**).

Elle repose sur un modèle commun **CIM** (*Common Information Model*) lequel est un **standard**, créé par le **DMTF** (*Distributed Management Task Force*), organisme **indépendant** de Microsoft (mais Microsoft en fait partie, au même titre que 3Com, AMD, Cisco, Compaq, DELL, Fujitsu, HP, IBM, Intel, Nec, Novell, Oracle, SAP, Siemens, Sun, Symantec, ...).

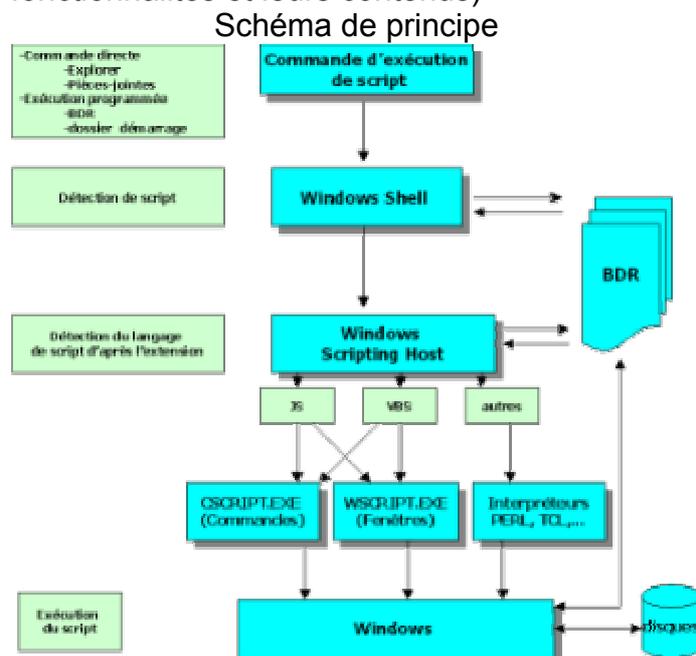
A l'aide de scripts ou programmes écrits en VBScript, Jscript, Perl, VB, C#, ..., on peut ainsi accéder à **n'importe quelle information** concernant le système (local ou distant), sans devoir aller explorer la Base de Registres ou autre source.

WMI ne remplace pas la BDR, mais va chercher dedans la plupart de ses informations. Le "plus" est que la collecte d'informations est **standardisée**, essentiellement sous la forme de requêtes **WQL** (*Windows Management Instrumentation Query Language*, analogue à **SQL**) .

WMI est installé **d'origine** sous **Windows ME**, **Windows 2000**, **Windows XP** et **Windows 2003**.

Dans le cas de **Windows 95/98/98SE** et **Windows NT4**, il faut l'installer séparément, à partir d'un fichier auto extractible **téléchargeable** librement sur le site Microsoft.

Pour ceux qui veulent **aller plus loin** dans la connaissance de **WMI**, un outil (fortement **documenté**, y compris en français) est le **WMI SDK**, qui se présente sous la forme d'interface WEB, et permet d'explorer **toutes** les classes (leurs fonctionnalités et leurs contenus)



2. LES VARIABLES

2.1. Déclarations

Toutes les données qui surgissent en cours de programme doivent être stockées en mémoire par l'intermédiaire d'une variable, mais il faut identifier la nature de la valeur stocker.

Pour conserver un nombre en mémoire, il suffit d'écrire simplement :

son nom = à son contenu

On distingue nombre et texte

Attention le séparateur décimal sous VB est le point et non la virgule.

Pour identifier la nature des données à stocker, on utilise des symboles.

Symbole	Type	Contenu
%	INTEGER	Nombre entier entre ± 32768
&	LONG	Nombre entier entre ± 2147483648
!	SINGLE	Décimale avec 7 chiffres significatifs
#	DOUBLE	Décimale avec 15 chiffres significatifs
@	CURRENCY	Décimale avec 19 chiffres significatifs
\$	STRING	Texte avec 65400 caractères
Aucun	VARIANT	Valeur numérique, chaîne ou date
	DATE	
	Booléen	Vrai ou Faux

2.1.1. Déclarations

Une variable est définie par son nom et son symbole. Le nom peut être un mot d'une longueur de 255 caractères, mais il doit commencer par une lettre suivie de lettre, chiffres et -. Attention au mot-clé.

Minuscule et majuscule n'ont pas d'importance.

Après le nom, on doit indiquer le symbole identifiant la nature du contenu puis le symbole "=" suivi du contenu. Si on a à faire à du texte on doit utiliser des guillemets.

```
Article$="programmer"   STRING
Nombre%=5               INTEGER
Prix=49.80              VARIANT
```

```
Total=Nombre% * Prix
```

2.1.2. Portée

Toute variable possède une certaine durée de vie, ce que l'on appelle sa portée. Celle-ci dépend de deux éléments : le mode et l'emplacement de ces variables :

- Si vous déclarez une variable à l'aide de l'instruction **Dim** sa portée est limitée à cette procédure. On dit que la variable est locale.
- Si vous déclarez une variable à l'aide de **Dim** dans la section Générale, elle est limitée au module dans toutes les procédures. On peut utiliser l'instruction **Private** à la place de **Dim**
- Si vous déclarez une variable toujours dans la section Générale en utilisant le mot clé **Public** à la place de **Dim**, la variable est accessible dans toutes les feuilles et tous les modules de l'application. On dit que la variable est globale.

3. LES FONCTIONS

3.1. Conversion de valeurs

CBool	Retourne une expression convertie en VARIANT de sous-type Boolean .
CByte	Retourne une expression convertie en VARIANT de sous-type Byte .
CCur	Retourne une expression convertie en VARIANT de sous-type Currency .
CDate	Retourne une expression convertie en VARIANT de sous-type Date .
CDbl	Retourne une expression convertie en VARIANT de sous-type Double .
CInt	Retourne une expression convertie en VARIANT de sous-type Integer .
CLng	Retourne une expression convertie en VARIANT de sous-type Long .
CSng	Retourne une expression convertie en VARIANT de sous-type Single .
CStr	Retourne une expression convertie en VARIANT de sous-type String .
FormatCurrency	Retourne une expression formatée sous forme de valeur de type Currency utilisant le symbole monétaire défini dans le Panneau de configuration du système.
FormatDateTime	Retourne une expression formatée sous forme de date ou d'heure.
FormatNumber	Retourne une expression formatée sous forme de nombre.
FormatPercent	Retourne une expression formatée sous forme de pourcentage (multiplié par 100) avec un caractère de fin %.

3.2. Traitement de chaînes

Asc	Retourne le code de caractère ANSI correspondant à la première lettre d'une chaîne.
Chr	Retourne le caractère associé au code de caractère ANSI spécifié.
Hex	Retourne une chaîne représentant la valeur hexadécimale d'un nombre.
InStr	Retourne la position de la première occurrence d'une chaîne à l'intérieur d'une autre.
InStrRev	Retourne la position d'une occurrence d'une chaîne dans une autre, à partir de la fin de la chaîne.

Join	Retourne une chaîne créée par la jonction de plusieurs sous-chaînes contenues dans un tableau.
LCase	Retourne une chaîne convertie en minuscules.
Left	Retourne un nombre spécifié de caractères à partir de la gauche d'une chaîne.
Len	Retourne le nombre de caractères contenus dans une chaîne, ou le nombre d'octets requis pour stocker une variable.
LTrim	Retourne une copie d'une chaîne sans espaces à gauche.
Mid	Retourne un nombre spécifié de caractères d'une chaîne.
Oct	Retourne une chaîne représentant la valeur octale d'un nombre.
Replace	Retourne une chaîne dans laquelle une sous-chaîne donnée a été remplacée par une autre sous-chaîne le nombre de fois spécifié.
Right	Retourne un nombre spécifié de caractères à partir de la droite d'une chaîne.
RTrim	Retourne une copie d'une chaîne sans espaces à droite.
Space	Retourne une chaîne composée d'un nombre spécifié d'espaces.
StrComp	Retourne une valeur indiquant le résultat d'une comparaison de chaîne.
String	Retourne une chaîne constituée d'un caractère répété sur la longueur spécifiée.
StrReverse	Retourne une chaîne contenant des caractères dont l'ordre a été inversé par rapport à une chaîne donnée.
Trim	Retourne une copie d'une chaîne sans espaces ni à gauche ni à droite.
UCase	Retourne une chaîne convertie en majuscules.

3.3. *Date et heure*

Date	Retourne la date système courante.
DateAdd	Retourne une date à laquelle un intervalle spécifique a été ajouté.
DateDiff	Retourne le nombre d'intervalles entre deux dates.
DatePart	Retourne la partie spécifiée d'une date donnée.
DateSerial	Retourne un VARIANT de sous-type Date pour une année, un mois et un jour spécifiés.
DateValue	Retourne un VARIANT de sous-type Date .
Day	Retourne un nombre entier compris entre 1 et 31 inclus, représentant le jour du mois.
Hour	Retourne un nombre entier compris entre 0 et 23

	inclus, représentant l'heure du jour.
Minute	Retourne un nombre entier compris entre 0 et 59 inclus, représentant la minute de l'heure.
Month	Retourne un nombre entier compris entre 1 et 12 inclus, représentant le mois de l'année.
MonthName	Retourne une chaîne indiquant le mois spécifié.
Now	Retourne la date et l'heure en cours en fonction de leur paramétrage dans le système de votre ordinateur.
Second	Retourne un nombre entier compris entre 0 et 59 inclus, représentant la seconde de la minute.
Time	Retourne un Variant de sous-type Date indiquant l'heure système en cours.
Timer	Retourne le nombre de secondes qui se sont écoulées depuis 00:00 (minuit).
TimeSerial	Retourne un Variant de sous-type Date contenant l'heure correspondant à des éléments spécifiques d'heure, de minute et de seconde.
TimeValue	Retourne un Variant de sous-type Date contenant l'heure.
Weekday	Retourne un nombre entier représentant le jour de la semaine.
WeekdayName	Retourne une chaîne indiquant le jour de la semaine spécifié.
Year	Retourne un nombre entier représentant l'année.

3.4. *Mathématiques*

Abs	Retourne la valeur absolue d'un nombre.
Atn	Retourne l'arc tangente d'un nombre.
Cos	Retourne le cosinus d'un angle.
Exp	Retourne e (la base des logarithmes népériens) élevé à une puissance.
Eval	Évalue une expression et Retourne le résultat.
Fix	Retourne la partie entière d'un nombre.
Int	Retourne la partie entière d'un nombre.
Log	Retourne le logarithme népérien d'un nombre.
RGB	Retourne un nombre entier représentant une valeur de couleur RVB.
Rnd	Retourne un nombre aléatoire.
Round	Retourne un nombre arrondi à un nombre spécifié de positions décimales.
Sgn	Retourne un entier indiquant le signe d'un nombre.
Sin	Retourne le sinus d'un angle.
Sqr	Retourne la racine carrée d'un nombre.
Tan	Retourne la tangente d'un angle.

3.5. Tableaux

Array	Retourne une variable de type Variant contenant un tableau.
Filter	Retourne un tableau commençant par zéro contenant un sous-ensemble d'un tableau de chaîne basé sur des critères de filtre spécifiés.
LBound	Retourne le plus petit indice disponible pour la dimension indiquée d'un tableau.
UBound	Retourne le plus grand indice disponible pour la dimension indiquée d'un tableau.
Split	Retourne un tableau à une dimension commençant par zéro contenant le nombre spécifié de sous-chaînes.

3.6. Objets

CreateObject	Crée et Retourne une référence à un objet Automation.
GetObject	Retourne une référence à l'objet Automation d'un fichier.
GetRef	Retourne une référence à une procédure éventuellement liée à un événement.

3.7. Tests

IsArray	Retourne une valeur booléenne indiquant si la variable est un tableau.
IsDate	Retourne une valeur booléenne indiquant si une expression peut être convertie en date.
IsEmpty	Retourne une valeur booléenne indiquant si une variable a été initialisée.
IsNull	Retourne une valeur booléenne indiquant si une expression contient des données valides ou non (Null).
IsNumeric	Retourne une valeur booléenne indiquant si une expression peut être évaluée sous la forme d'un nombre.
IsObject	Retourne une valeur booléenne indiquant si une expression fait référence à un objet Automation valide.

3.8. Dialogues

InputBox	Affiche une invite dans une boîte de dialogue, attend que l'utilisateur entre du texte ou choisisse un bouton et Retourne le contenu de la zone de texte.
MsgBox	Affiche un message dans une boîte de dialogue, attend que l'utilisateur clique sur un bouton et Retourne une valeur indiquant le bouton choisi par l'utilisateur.

3.9. Informations

GetLocale	Renvoie la valeur d'ID des paramètres régionaux en cours.
ScriptEngine	Retourne une chaîne représentant le langage de script utilisé.
ScriptEngineBuildVersion	Retourne le numéro de compilation du moteur de script employé.
ScriptEngineMajorVersion	Retourne le numéro de version principal du moteur de script employé.
ScriptEngineMinorVersion	Retourne le numéro de version secondaire du moteur de script employé.
SetLocale	Définit les paramètres régionaux globaux et renvoie les paramètres régionaux précédents.
TypeName	Retourne une chaîne qui fournit des informations de sous-type Variant sur une variable.
VarType	Retourne une valeur indiquant le sous-type d'une variable.

3.10. Divers

LoadPicture	Retourne un objet image. Disponible seulement sur les plates-formes 32 bits.
--------------------	--

4. ENTREES/SORTIES DE DONNEES

4.1. Sorties

C'est une information qui informe l'utilisateur. Vous avez plusieurs manières pour le faire :

Print Date\$	Renvoie la date du jour
Print 123	Renvoie 123
Print	Renvoie une ligne vide
Print "coucou"	Renvoie coucou
Print "Nous sommes le" Date\$	Renvoie Nous sommes le "date du jour"

Mais la meilleur façon pour le faire est d'utiliser une boite de message. Une boite de message permet également de fournir une réponse.

Syntaxe :

MsgBox(Texte Message, Numéro, Titre)

message à afficher

Titre de la fenêtre

Le numéro est une addition de différentes valeurs dont voici la signification

Première valeur :

Bouton	Valeur
OK (par défaut)	0
Ok et Annuler	1
Abandon, Répéter, Ignorer	2
Oui, Non, Annuler	3
Oui, Non	4
Répéter et Annuler	5

Deuxième valeur :

Icones	Valeur
Panneau Stop	16
Point d'interrogation	32
Point d'exclamation	48
Information	64

Troisième valeur :

Bouton par défaut	Valeur
1 ^{er} Bouton	0
2 ^{ème} Bouton	256
3 ^{ème} Bouton	512

Exemple :

```
MsgBox "Vous avez appris quelques choses",36,"Question"
```

Ou

```
MsgBox Date$
```

Attention quand vous cliquez sur un bouton, cette instruction renvoie la valeur du bouton cliqué à condition de transformer votre instruction en fonction. Pour cela vous utilisez les parenthèses mais vous devez stocker la réponse dans une variable.

```
Réponse = MsgBox ("Vous avez appris quelques choses",36,"Question")  
Print Réponse
```

Voici la valeur des boutons :

Ok=1, Annuler=2, Abandon=3, Répéter=4, Ignorer=5, Oui=6, Non=7

Exemple :

On veut savoir la valeur des boutons renvoyée par une boite de message :

```
Print MsgBox("Vous avez cliqué sur" & Réponse)
```

ou

```
MsgBox "Vous avez cliqué sur" & Réponse
```

4.2. Entrées :

Permet de saisir des données de l'utilisateur. On utilise également une boite que l'on appelle Boite de dialogue. Attention l'information saisie par l'utilisateur sera toujours du texte.

Syntaxe :

```
Nomvariable = InputBox$ (message, titre, défaut )
```

Exemple :

```
Reponse = InputBox$("SVP, Entrez votre nom")
```

```
Print Reponse
```

Vous pouvez aussi utiliser naturellement une zone de saisie dans un objet Forms.

5. CALCULS ET COMPARAISONS

5.1. Les opérateurs de comparaison

>	Supérieur
<	Inférieur
<>	Différent
>=	Supérieur ou égal
<=	Inférieur ou égal
Not	Négation
And	Et logique
OR	Ou logique
XOR	Ou exclusif

5.2. Les calculs

Bien sûr vous trouvez les opérateurs standards qui sont :

+	Addition
-	Soustraction
*	Multiplication
/	Division
^	Puissance
\	Division entière
MOD	Donne le reste de la division
&	Concaténation de chaînes

Mais aussi les fonctions mathématique suivante :

Sin(x)	Sinus de x
Cos(x)	Cosinus de x
Tan(x)	Tangente de x
Atn(x)	Arc tangente de x
Sqr(x)	Racine Carrée de x
Log(x)	Logarithme de x

6. BOUCLE ET TEST

6.1. *If...Then*

Permet de renvoyer à une suite d'instructions en fonction d'une condition.

Syntaxe :

```
If Condition Then
    'Instructions si condition est vraie
Else
    'Instructions si condition est fausse
End If
```

Else n'est pas obligatoire si vous avez qu'une seule instruction. If Condition Then instruction1

Vous pouvez aussi ajouter Else If Condition2 Then, mais cela devient lourd et il est préférable de faire une instruction Select Case.

6.2. *Select....case*

C'est une structure qui permet de faire des tests comme la structure IF.

Si vous devez écrire une série de conditions, il est préférable de faire appel à Select Case plutôt qu'à If car l'écriture est plus simple et plus courte.

Le principe de la syntaxe est que l'on énonce avant le nom de la variable.

Voici un exemple qui test les différentes saisons :

```
Select Case NuméroMois
    Case 1,2,12
        Saison="Hiver"
    Case 3,4,5
        Saison="Printemps"
    Case 6
        Saison="Eté"
    Case 7,8
        Saison="Eté-Vacances"
    Case 9,10,11
        Saison="Automne"
    Case Else
        Saison="Inconnue"
End Select
```

6.3. For...Next

Ici la structure For permet de faire une boucle compteur et permet de faire Xfois une suite d'instructions.

Syntaxe :

```
For nomvariable = début To Fin Step pas
  Instructions
Next
```

Utile pour compter un nombre de ligne pour une facture, une page, etc....

Exemple :

On veut remplir une variable tableau de la manière suivante :

```
0  1  2  3
1  2  3  4
2  3  4  5
9
```

Voici la solution :

```
Dim Essai(9,4) as Integer
For I=0 To 9
  For J=0 to 3
    Essai (I,J)=I+J
  Next
```

| *Pour sortir de cette boucle on utilise l'instruction Exit For.*

6.4. **Do...Loop**

Cette structure permet de faire une boucle conditionnelle.
Permet d'exécuter une suite d'instruction en fonction d'une condition.
Il existe plusieurs variante.

Syntaxe 1 :

```
Do Until condition
    Instructions
Loop
```

On boucle jusqu'à ce que la condition devienne vraie.

Syntaxe 2 :

```
Do While condition
    Instructions
Loop
```

On boucle tant que la condition reste vraie ou jusqu'à ce que la condition soit fausse.

Syntaxe 3 :

```
Do
    Instructions
Loop Until Condition
```

Idem que la syntaxe 1 mais on passe au moyen une fois.

Syntaxe 3 :

```
Do
    Instructions
Loop While condition
```

On peut sortir d'une boucle avec Exit Do

On peut remplacer Do While ...Loop par While Condition Instructions ...Wend.

7. LES ERREURS

Lorsqu'une erreur se produit, la variable Err de VB contient le numéro de l'erreur et une boîte de message vous informe mais le programme s'arrête. Attention il faut que la bibliothèque aderror.dll soit installé sur votre ordinateur pour voir la description de votre erreur.

Pour éviter cela, on peut indiquer une séquence d'instruction en cas d'erreur. Ce déroutage est effectué par l'instruction On error ou par un renvoi à une procédure.

Syntaxe :
On Error resume next

*Attention placer cette instruction en 1er ligne (avant les futurs erreurs).
Pour revenir dans le déroulement normal de votre programme il faut avoir l'instruction On error goto 0.*

Syntaxe de la procédure :
On test la valeur de la variable Err

On error Resume Next

Exemple :
Sub procerreur()
 On Error Resume Next
 Instructions
 If Err= ... Then
 Instructions
 End If
End Sub

8. MODULES ET PROCEDURES

8.1. Les procédures :

Dans un module standard, il est possible de créer des procédures de 3 types :

Il existe une procédure dite de démarrage : Sub Main

Procédures Function : Elles reçoivent des arguments et retournent toujours une valeur par l'intermédiaire de leur nom.

Procédures Sub : Elles reçoivent des arguments et peuvent renvoyer une valeur mais à la différence des fonctions ne retournent pas une valeur en lieu et en place de leur nom.

Toutes ces procédures évitent la répétition de certaines lignes de codes, simplifient la lecture et le développement, réutilisable dans d'autres programmes.

8.2. Procédures Function :

Pour créer une fonction, vous devez taper la syntaxe suivante :

```
Function Nomfonction (argument1, argument2,...) As typefonction
```

Maintenant que vous avez créé une fonction il faut savoir l'appeler. Pour cela, vous devez avoir une ligne dans votre module du type :

```
Nomvariable = Nomfonction(valeur1, valeur2, ....)
```

Par exemple :

```
Function TaxeTotale (Montant)
    TaxeEtat = Montant * 5.5%
    TaxeLocale = Montant* 1.5%
    Taxetotale = TaxeEtat + Taxelocale
End Function
```

Pour appeler la fonction dans votre module :

```
PbTaxes.Caption = TaxeTotale (500)
```

8.3. Procédures Sub :

Pour créer une procédure, vous devez taper :

```
SUB NomProcédure (arguments, ...)
```

Par exemple :

```
SUB Ajout (Personne$)
    If Personne$ <> " " then
        Form1.List1.AddItem Personne$
        Msg$ = Personne $ " a été ajouté à la liste."
    Else
        Msg$ = " Aucun nom entré "
    End if
End Sub
```

Dans un module standard, le nom des objets doit être précédé par le nom de la feuille.

Maintenant pour appeler la procédure il faut taper la ligne suivante

```
nomprocédure listeargument
```

Par exemple la procédure "Ajout" permet d'ajouter un nom dans un fichier et voici la syntaxe Ajout "Jack London"

Dans cet exemple, les arguments sont passés par référence c'est-à-dire si ils sont modifier dans la procédure, ils sont pris en compte dans le programme.

9. LES OBJETS :

9.1. *Objet groupe*

Concerne les groupes de votre machine.

Tableau des propriétés

<i>NOM</i>	<i>EXPLICATION</i>
Members	Renvoie le nom de l'utilisateur

Tableau des méthodes

<i>NOM</i>	<i>EXPLICATION</i>
Setinfo	Exécute les opérations ou modification
Add	Ajoute un utilisateur à un groupe
Remove	Détruit un utilisateur du groupe
Create	Créer un groupe
Delete	Détruit le groupe

9.2. *Objet User*

Concerne les utilisateurs de votre machine.

Tableau des propriétés

<i>NOM</i>	<i>EXPLICATION</i>
Lastlogin	Indique la date de la dernière utilisation d'un compte.

Tableau des méthodes

<i>NOM</i>	<i>EXPLICATION</i>
Setinfo	Exécute les opérations ou modification
Create	Créer un utilisateur
Delete	Détruit l'utilisateur
ChangePassword	Change de mot de passe
MoveHere	Pour renommer un compte

9.3. *Les autres objets*

La liste étant très grande, je me permets de faire un copier, coller du site de Microsoft.

La légende :

O : objet
C : classe
M : Méthode
P : Propriété

▲		
O	<u>Err</u>	Objet contenant l'information relative aux erreurs d'exécution.
M	<u>Clear</u>	Réinitialise les propriétés de l'objet Err .
P	<u>Description</u>	Renvoie ou définit une chaîne descriptive associée à une erreur.
P	<u>HelpContext</u>	Définit ou renvoie un identificateur de contexte pour une rubrique dans un fichier d'aide.
P	<u>HelpFile</u>	Définit ou renvoie le chemin complet d'un fichier d'aide.
P	<u>Number</u>	Renvoie ou définit une valeur numérique spécifiant une erreur. Number est la propriété par défaut de l'objet Err .
M	<u>Raise</u>	Génère une erreur d'exécution.
P	<u>Source</u>	Renvoie ou définit le nom de l'objet ou de l'application qui est à l'origine de l'erreur.
▲		
O	<u>RegExp</u>	Objet permettant la gestion des expressions régulières
M	<u>Execute</u>	Exécute une recherche d'expression régulière dans une chaîne spécifiée.
P	<u>Global</u>	Définit ou renvoie une valeur booléenne indiquant si toutes les occurrences d'une chaîne de recherche ou seulement la première chaîne doivent satisfaire aux critères.
P	<u>IgnoreCase</u>	Définit ou renvoie une valeur booléenne indiquant si les critères de recherche distinguent les minuscules et les majuscules.
P	<u>Pattern</u>	Définit ou renvoie les critères de recherche de l'expression régulière.
M	<u>Replace</u>	Remplace le texte trouvé dans une recherche d'expression régulière.
M	<u>Test</u>	Lance une recherche d'expression régulière dans une chaîne spécifiée.
▲		
C	<u>Matches</u>	<u>Collection</u> d'objets Match sous forme d'expressions régulières.
P	<u>Count</u>	Renvoie le nombre d'éléments de la collection
P	<u>Item</u>	Renvoie un <i>item</i> basé sur la <i>key</i> spécifiée.
▲		
O	<u>Match</u>	Objet donnant accès aux propriétés d'une correspondance d'expression régulière.
P	<u>FirstIndex</u>	Indique la position dans une chaîne de recherche où une correspondance a été

trouvée.

P	Length	Renvoie la longueur d'une correspondance trouvée dans une chaîne de recherche.
P	Value	Renvoie la valeur ou le texte d'une correspondance trouvée dans une chaîne de recherche.
▲		
O	Dictionary	Objet qui stocke des paires clé/élément de données.
M	Add	Ajoute une paire clé/élément à un objet Dictionary
P	CompareMode	Définit et renvoie le mode de comparaison pour comparer les clés de chaîne dans un objet Dictionary .
P	Count	Renvoie le nombre d'éléments d'un objet Dictionary
M	Exists	Renvoie la valeur True si une clé spécifiée existe dans l'objet Dictionary
P	Item	Définit ou renvoie un <i>élément</i> pour un argument <i>key</i> spécifié dans un objet Dictionary
M	Items	Renvoie un tableau contenant tous les éléments d'un objet Dictionary .
P	Key	Définit un argument <i>key</i> dans un objet Dictionary .
M	Keys	Renvoie un tableau contenant toutes les clés existantes dans un objet Dictionary .
M	Remove	Retire une paire clé/élément d'un objet Dictionary .
M	RemoveAll	Retire toutes les paires clé/élément d'un objet Dictionary .
▲		
O	Wscript	Expose des propriétés qui fournissent le chemin d'accès de l'hôte de script en cours d'exécution (WScript.exe ou CScript.exe), ses arguments, ainsi que le mode de travail (interactif ou par lots). L'objet WScript fournit également des méthodes pour créer et lire des objets.
P	Application	Fournit l'interface IDispatch sur l'objet WScript .
P	Arguments	Renvoie un pointeur vers la collection WshArguments ou identifie des arguments pour le raccourci pointant vers la collection.
P	FullName	Renvoie une chaîne contenant le chemin d'accès complet du fichier exécutable de l'hôte ou d'un objet de raccourci.
P	Name	Renvoie une chaîne contenant le nom de l'objet WScript .
P	Path	Fournit une chaîne contenant le nom du répertoire où réside WScript.exe ou CScript.exe.
P	ScriptFullName	Fournit le chemin d'accès complet du script en

P	<u>ScriptName</u>	cours d'exécution. Fournit le nom de fichier du script en cours d'exécution.
P	<u>StdErr</u>	Expose le flux de sortie d'erreur en lecture seule du script actuel.
P	<u>StdIn</u>	Expose le flux de données d'entrée en lecture seule du script actuel.
P	<u>StdOut</u>	Expose le flux de sortie en lecture seule du script actuel.
P	<u>Version</u>	Renvoie la version de Microsoft Windows Script Host.
M	<u>CreateObject</u>	Crée un objet spécifié par l'argument <i>strProgID</i> .
M	<u>ConnectObject</u>	Connecte les sources d'événements d'un objet à des fonctions possédant des préfixes donnés.
M	<u>DisconnectObject</u>	Déconnecte un objet préalablement connecté à Windows Script Host.
M	<u>Echo</u>	Envoie une sortie à une boîte de dialogue ou à la console.
M	<u>GetObject</u>	Récupère un objet Automation depuis un fichier ou depuis un objet spécifié par l'argument <i>strProgID</i> .
M	<u>Quit</u>	Arrête l'exécution en affichant un code d'erreur spécifié.
M	<u>Sleep</u>	Place le traitement de script dans un état d'inactivité pendant le nombre de millisecondes spécifié, avant d'en poursuivre l'exécution.
▲		
C	<u>WshArguments</u>	<u>Collection</u> des paramètres de ligne de commande.
P	<u>Item</u>	Renvoie un élément spécifié en provenance d'une collection.
P	<u>Count</u>	Renvoie le nombre d'éléments énumérés.
P	<u>Length</u>	Renvoie le nombre d'éléments énumérés. (identique à la propriété Count et est fournie pour assurer une compatibilité avec Microsoft® JScript®.)
▲		
O	<u>FileSystemObject</u>	Donne accès au système de fichiers d'un ordinateur. (voir <u>Hiérarchie de FileSystemObject</u>)
M	<u>BuildPath</u>	Ajoute un nom à un chemin existant.
M	<u>CopyFile</u>	Copie un ou plusieurs fichiers d'un emplacement vers un autre.
M	<u>CopyFolder</u>	Copie récursivement un dossier d'un emplacement vers un autre.
M	<u>CreateFolder</u>	Crée un dossier.
M	<u>CreateTextFile</u>	Crée un nom de fichier spécifié et renvoie un objet TextStream pouvant être utilisé pour lire ou y écrire.

M	<u>DeleteFile</u>	Supprime un fichier spécifié.
M	<u>DeleteFolder</u>	Supprime un dossier spécifié et son contenu.
M	<u>DriveExists</u>	Renvoie la valeur True si le lecteur spécifié existe, False dans le cas contraire.
M	<u>FileExists</u>	Renvoie la valeur True si un fichier spécifié existe La valeur False est renvoyée dans le cas contraire.
M	<u>FolderExists</u>	Renvoie la valeur True si le dossier spécifié existe et la valeur False dans le cas contraire.
M	<u>GetAbsolutePathName</u>	Renvoie un chemin complet et non ambigu à partir d'une spécification de chemin fournie.
M	<u>GetBaseName</u>	Renvoie une chaîne contenant le nom de base (sans extension) du fichier ou du dossier dans une spécification de chemin fournie.
M	<u>GetDrive</u>	Renvoie un objet Drive correspondant au lecteur figurant dans un chemin spécifié.
M	<u>GetDriveName</u>	Renvoie une chaîne contenant le nom du lecteur pour un chemin spécifié.
M	<u>GetExtensionName</u>	Renvoie une chaîne contenant le nom de l'extension du dernier composant d'un chemin.
M	<u>GetFile</u>	Renvoie un objet File correspondant au fichier dans un chemin spécifié.
M	<u>GetFileName</u>	Renvoie le dernier nom de fichier ou dossier d'un chemin spécifié qui ne fait pas partie de la spécification de lecteur.
M	<u>GetFileVersion</u>	Renvoie le numéro de version d'un fichier spécifié.
M	<u>GetFolder</u>	Renvoie un objet Folder correspondant au dossier d'un chemin spécifié.
M	<u>GetParentFolderName</u>	Renvoie une chaîne contenant le nom du dossier parent du dernier fichier ou dossier dans un chemin spécifié.
M	<u>GetSpecialFolder</u>	Renvoie le dossier spécial spécifié.
M	<u>GetTempName</u>	Renvoie un nom de fichier ou de dossier temporaire généré aléatoirement utile pour les opérations qui nécessitent un fichier ou un dossier temporaire.
M	<u>MoveFile</u>	Change l'emplacement d'un ou plusieurs fichiers.
M	<u>MoveFolder</u>	Change l'emplacement d'un ou plusieurs dossiers.
M	<u>OpenTextFile</u>	Ouvre un fichier spécifié et renvoie un objet TextStream pouvant être utilisé pour lire le fichier, écrire ou insérer un élément dans le fichier.
P	<u>Drives</u>	Renvoie une collection Drives constituée de tous les objets Drive disponibles sur la machine locale.



C

[Drives](#)

[Collection](#) en lecture seule de tous les lecteurs

P	<u>Count</u>	disponibles.
P	<u>Item</u>	Renvoie le nombre d'éléments de la collection
▲		Renvoie un <i>item</i> basé sur la <i>key</i> spécifiée.
O	<u>Drive</u>	Donne accès aux propriétés d'un lecteur de disque ou d'un partage réseau spécifié.
P	<u>AvailableSpace</u>	Renvoie la quantité d'espace libre disponible pour un utilisateur sur le lecteur ou le partage réseau.
P	<u>DriveLetter</u>	Renvoie la lettre d'un lecteur local physique ou d'un partage réseau.
P	<u>DriveType</u>	Renvoie une valeur indiquant le type d'un lecteur spécifié.
P	<u>FileSystem</u>	Renvoie le type de système de fichiers du lecteur spécifié.
P	<u>FreeSpace</u>	Renvoie la quantité d'espace disponible pour un utilisateur sur le lecteur ou le partage réseau. Lecture seule.
P	<u>IsReady</u>	Renvoie la valeur True si le lecteur spécifié est prêt et la valeur False dans le cas contraire.
P	<u>Path</u>	Renvoie le chemin d'un lecteur spécifié.
P	<u>RootFolder</u>	Renvoie un objet Folder correspondant au dossier racine d'un lecteur spécifié
P	<u>SerialNumber</u>	Renvoie le numéro de série décimal utilisé pour identifier de manière unique un volume de disque.
P	<u>ShareName</u>	Renvoie le nom du partage réseau pour un lecteur spécifié.
P	<u>TotalSize</u>	Renvoie l'espace total, en octets, d'un lecteur ou d'un partage réseau.
P	<u>VolumeName</u>	Définit ou renvoie le nom de volume du lecteur spécifié. Lecture/écriture.
▲		
C	<u>Folders</u>	<u>Collection</u> de tous les objets Folder d'un objet Folder .
M	<u>Add</u>	Ajoute un nouvel élément Folder à une collection Folders .
P	<u>Count</u>	Renvoie le nombre d'éléments de la collection
P	<u>Item</u>	Renvoie un <i>item</i> basé sur la <i>key</i> spécifiée.
▲		
O	<u>Folder</u>	Donne accès aux propriétés d'un dossier.
P	<u>Attributes</u>	Définit ou renvoie les attributs de dossiers.
M	<u>Copy</u>	Copie un dossier spécifié d'un emplacement vers un autre.
M	<u>Move</u>	Change l'emplacement d'un dossier spécifié.
P	<u>DateCreated</u>	Renvoie la date et l'heure de création du fichier ou du dossier spécifié.
P	<u>DateLastAccessed</u>	Renvoie la date et l'heure de dernier accès au

		fichier ou au dossier spécifié.
P	<u>DateLastModified</u>	Renvoie la date et l'heure à laquelle le fichier ou le dossier spécifié a été modifié pour la dernière fois.
M	<u>Delete</u>	Supprime un dossier spécifié.
P	<u>Drive</u>	Renvoie la lettre du lecteur sur lequel réside le dossier spécifié.
P	<u>Files</u>	Renvoie une collection Files constituée de tous les objets File contenus dans le dossier spécifié, y compris ceux qui sont cachés ou système.
P	<u>IsRootFolder</u>	Renvoie la valeur True si le dossier spécifié est le dossier racine et la valeur False dans le cas contraire.
P	<u>Name</u>	Définit ou renvoie le nom d'un fichier ou dossier spécifié.
P	<u>ParentFolder</u>	Renvoie l'objet dossier pour le parent du fichier ou dossier spécifié.
P	<u>Path</u>	Renvoie le chemin d'un fichier, dossier ou lecteur spécifié.
P	<u>ShortName</u>	Renvoie le nom abrégé utilisé par les programmes qui emploient la convention de dénomination 8.3.
P	<u>ShortPath</u>	Renvoie le chemin abrégé utilisé par les programmes qui emploient la convention de dénomination 8.3.
P	<u>Size</u>	Renvoie la taille en octets de tous les fichiers et sous-dossiers du dossier.
P	<u>Subfolders</u>	Renvoie une collection Folders comprenant tous les dossiers contenus dans le dossier spécifié, y compris ceux qui sont cachés ou système.
P	<u>Type</u>	Renvoie des informations sur le type d'un dossier.
▲		
C	<u>Files</u>	<u>Collection</u> de tous les objets File d'un dossier.
P	<u>Count</u>	Renvoie le nombre d'éléments de la collection
P	<u>Item</u>	Renvoie un <i>item</i> basé sur la <i>key</i> spécifiée.
▲		
O	<u>File</u>	Donne accès aux propriétés d'un fichier.
P	<u>Attributes</u>	Définit ou renvoie les attributs de fichiers.
M	<u>Copy</u>	Copie un fichier spécifié d'un emplacement vers un autre.
M	<u>CreateTextFile</u>	Crée un nom de fichier spécifié et renvoie un objet TextStream pouvant être utilisé pour lire ou y écrire.
P	<u>DateCreated</u>	Renvoie la date et l'heure de création du fichier.
P	<u>DateLastAccessed</u>	Renvoie la date et l'heure de dernier accès au fichier.

P	<u>DateLastModified</u>	Renvoie la date et l'heure à laquelle le fichier spécifié a été modifié pour la dernière fois.
M	<u>Delete</u>	Supprime un fichier spécifié.
P	<u>Drive</u>	Renvoie la lettre du lecteur sur lequel réside le fichier spécifié.
M	<u>Move</u>	Change l'emplacement d'un fichier spécifié.
P	<u>Name</u>	Définit ou renvoie le nom d'un fichier spécifié.
M	<u>OpenAsTextStream</u>	Ouvre un fichier spécifié et renvoie un objet TextStream pouvant être utilisé pour lire le fichier, écrire ou insérer un élément dans le fichier.
P	<u>ParentFolder</u>	Renvoie l'objet dossier pour le parent du fichier spécifié.
P	<u>Path</u>	Renvoie le chemin d'un fichier spécifié.
P	<u>ShortName</u>	Renvoie le nom abrégé utilisé par les programmes qui emploient la convention de dénomination 8.3.
P	<u>ShortPath</u>	Renvoie le chemin abrégé utilisé par les programmes qui emploient la convention de dénomination 8.3.
P	<u>Size</u>	Renvoie la taille en octets du fichier spécifié.
P	<u>Type</u>	Renvoie des informations sur le type d'un fichier.
▲		
O	<u>TextStream</u>	Donne l'accès séquentiel à un fichier texte.
P	<u>AtEndOfLine</u>	Renvoie la valeur True si le pointeur de fichier est positionné immédiatement après le marqueur de fin de ligne d'un fichier TextStream .
P	<u>AtEndOfStream</u>	Renvoie la valeur True si le pointeur de fichier se trouve à la fin d'un fichier TextStream .
M	<u>Close</u>	Ferme un fichier TextStream ouvert.
P	<u>Column</u>	Renvoie le numéro de colonne de la position de caractère courante dans un fichier TextStream .
P	<u>Line</u>	Renvoie le numéro de ligne courant d'un fichier TextStream .
M	<u>Read</u>	Lit un nombre spécifié de caractères dans un fichier TextStream et renvoie la chaîne résultante.
M	<u>ReadAll</u>	Lit la totalité d'un fichier TextStream et renvoie la chaîne résultante.
M	<u>ReadLine</u>	Lit toute une ligne (caractère de nouvelle ligne exclus) d'un fichier TextStream et renvoie la chaîne résultante.
M	<u>Skip</u>	Omet un nombre spécifié de caractères lors de la lecture d'un fichier TextStream .
M	<u>SkipLine</u>	Omet la ligne suivante lors de la lecture d'un fichier TextStream .
M	<u>Write</u>	Écrit une chaîne spécifiée dans un fichier TextStream .

M	<u>WriteBlankLines</u>	Écrit un nombre spécifié de caractères de nouvelle ligne dans un fichier TextStream .
M	<u>WriteLine</u>	Écrit une chaîne spécifiée et un caractère de nouvelle ligne dans un fichier TextStream .
<hr/>		
O	<u>WshShell</u>	Démarre un nouveau traitement, crée des raccourcis et fournit la collection <u>WshEnvironment</u> pour la gestion de variables d'environnement telles que WINDIR, PATH ou PROMPT.
P	<u>Environment</u>	Renvoie la collection <u>WshEnvironment</u> .
P	<u>SpecialFolders</u>	Renvoie la collection <u>WshSpecialFolders</u> pour l'accès aux dossiers environnement de Windows tels que le dossier du bureau, le dossier du menu Démarrer et le dossier de documents personnels.
M	<u>AppActivate</u>	Active une fenêtre d'application.
M	<u>CreateShortcut</u>	Crée une référence d'objet à un <u>raccourci</u> ou à un <u>raccourci d'URL</u> .
M	<u>ExpandEnvironmentStrings</u>	Développe la variable d'environnement requise depuis le traitement d'exécution et renvoie la chaîne de résultat.
M	<u>LogEvent</u>	Consigne un événement dans le journal d'événements de Windows NT ou dans le fichier WSH.log.
M	<u>Popup</u>	Affiche une fenêtre de message contextuelle qui contient le message contenu dans <i>strText</i> .
M	<u>RegDelete</u>	Supprime du registre la clé ou la valeur nommée par <i>strName</i> .
M	<u>RegRead</u>	Renvoie la clé ou la valeur du registre nommée par <i>strName</i> .
M	<u>RegWrite</u>	Définit la clé ou la valeur de registre nommée par <i>strName</i> .
M	<u>Run</u>	Crée un nouveau traitement qui exécute <i>strCommand</i> .
M	<u>SendKeys</u>	Envoie une ou plusieurs entrées clavier vers la fenêtre active comme si ces dernières étaient saisies depuis le clavier.
<hr/>		
O	<u>WshShortcutArguments</u>	Crée une référence d'objet à un raccourci . Renvoie un pointeur vers la collection WshArguments ou identifie des arguments pour le raccourci pointant vers la collection.
P	<u>Description</u>	Fournit une description d'un objet de raccourci.
P	<u>FullName</u>	Renvoie une chaîne contenant le chemin d'accès complet du fichier exécutable de l'hôte ou d'un objet de raccourci.
P	<u>Hotkey</u>	Fournit la touche de raccourci clavier d'un objet raccourci.
P	<u>IconLocation</u>	Fournit l'emplacement de l'icône d'un objet

P	<u>TargetPath</u>	raccourci. Définit le chemin d'accès du fichier exécutable d'un objet de raccourci.
P	<u>WindowStyle</u>	Fournit le style de fenêtre d'un objet raccourci.
P	<u>WorkingDirectory</u>	Fournit le répertoire de travail d'un objet raccourci.
M	<u>Save</u>	Enregistre un raccourci vers l'emplacement spécifié.
<hr/>		
O	<u>WshUriShortcut</u>	Crée une référence d'objet vers un raccourci d'URL.
P	<u>FullName</u>	Renvoie une chaîne contenant le chemin d'accès complet du fichier exécutable de l'hôte ou d'un objet de raccourci.
P	<u>TargetPath</u>	Définit le chemin d'accès du fichier exécutable d'un objet de raccourci.
M	<u>Save</u>	Enregistre un raccourci vers l'emplacement spécifié.
<hr/>		
C	<u>WshEnvironment</u>	<u>Collection</u> des variables d'environnement système Microsoft Windows.
P	<u>Item</u>	Renvoie un élément spécifié en provenance d'une collection.
P	<u>Count</u>	Renvoie le nombre d'éléments énumérés.
P	<u>Length</u>	Renvoie le nombre d'éléments énumérés. (identique à la propriété Count et est fournie pour assurer une compatibilité avec Microsoft® JScript®.)
M	<u>Remove</u>	Supprime la variable d'environnement définie par <i>strName</i> .
<hr/>		
C	<u>WshSpecialFolders</u>	<u>Collection</u> des chemins d'accès pour les dossiers environnement de Windows tels que le dossier du bureau, le dossier du menu Démarrer et le dossier des documents personnels.
P	<u>Item</u>	Renvoie un élément spécifié en provenance d'une collection.
P	<u>Count</u>	Renvoie le nombre d'éléments énumérés.
P	<u>Length</u>	Renvoie le nombre d'éléments énumérés. (identique à la propriété Count et est fournie pour assurer une compatibilité avec Microsoft® JScript®.)
<hr/>		
O	<u>WshNetwork</u>	Expose le réseau Microsoft Windows, facilitant la connexion et la déconnexion des lecteurs et des imprimantes distants.
P	<u>ComputerName</u>	Fournit une représentation de la chaîne correspondant au nom de l'ordinateur.
P	<u>UserDomain</u>	Fournit une représentation de chaîne du nom du

		domaine de l'utilisateur.
P	<u>UserName</u>	Fournit une représentation de chaîne du nom de l'utilisateur.
M	<u>AddPrinterConnection</u>	Mappe l'imprimante locale spécifiée par <i>strRemoteName</i> sur le nom de ressource locale <i>strLocalName</i> .
M	<u>AddWindowsPrinterConnection</u>	Ajoute une connexion d'imprimante à Windows. Contrairement à la méthode <u>AddPrinterConnection</u> , cette méthode permet de créer une connexion à une imprimante en la dirigeant vers un port spécifique tel que LPT1.
M	<u>EnumNetworkDrives</u>	Renvoie les mappages de lecteurs de réseau actuels.
M	<u>EnumPrinterConnections</u>	Renvoie les mappages d'imprimantes de réseau actuels.
M	<u>MapNetworkDrive</u>	Mappe le point de partage spécifié <i>strRemoteName</i> sur le nom de la ressource locale <i>strLocalName</i> .
M	<u>RemoveNetworkDrive</u>	Supprime la connexion à une ressource lecteur réseau.
M	<u>RemovePrinterConnection</u>	Supprime la connexion à la ressource imprimante réseau.
M	<u>SetDefaultPrinter</u>	Définit l'imprimante distante spécifiée comme imprimante par défaut.