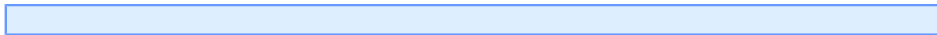


Tutoriel 24 : Les bases de la gestion de fichiers

par Gilbert Miralles (gilmir.developpez.com)

Date de publication : Lundi 18 mars 2003

Dernière mise à jour : Lundi 4 février 2008



I - Les fichiers à accès séquentiels

Prochain tutoriel :

I - Les fichiers à accès séquentiels

Dans certains Tutoriels que je diffuse sur Internet, je fais référence à la création de fichiers afin de stocker des données en mémoire inérantes aux applications développées.

Aussi, je vous propose d'étudier les bases de la gestion des fichiers pouvant vous servir dans la plupart des applications que vous aurez à développer.

Visual Basic possède les fonctions fondamentales de gestion de fichiers séquentiels, à accès direct et binaires.

Quel que soit le mode de gestion adopté, nous ouvrirons les fichiers à l'aide de l'instruction "Open" et nous les fermerons avec l'instruction "Close".

Recherchons dans Visual Basic les caractéristiques de l'instruction "Open"

Description : Ouvre un fichier

Syntaxe : Open fichier\$ [For Mode] [Access accès] [Look] As [#] identificateur fichier% [Len=longueur%] (Cette chaîne de caractère doit être écrite sur la même ligne)

fichier\$: Chaîne de caractère dont la valeur décrit le nom du fichier à ouvrir.

mode : Vous pouvez entrer ici une chaîne de caractères dont la valeur définit le mode de transfert des données et doit contenir l'une des expressions suivantes :

- **Random** : Mode d'accès direct
- **Random** : Mode d'accès direct
- **Binary** : Mode binaire. Tout octet est lu ou écrit par "Get" et "Put"
- **Input** : Entrée séquentielle.
- **Output** : Sortie séquentielle.
- **Append** : Sortie séquentielle ajoutée à la fin d'un fichier existant.

Le mot clé **Access** est réservé aux versions du DOS supérieures ou égales à 3.1

En paramètre, on peut passer une chaîne de caractères contenant l'une des expressions suivantes.(Attribut)

- **Read** : Lecture seulement
- **Write** : Ecriture seulement

- **Read Write** : Lecture et écriture. Réservé aux fichiers ouverts en mode Binary ou Append
- **Lock** : Le fichier peut, avec l'un des mots-clés suivants, être doté de certains droits d'accès autres que Lock :
- **Shared** : Tout programme peut accéder au fichier et lire ou écrire des données.
- **Lock Read** : Aucun autre programme ne peut lire des données dans ce fichier.
- **Lock Write** : Aucun autre programme ne peut écrire de données dans ce fichier.

Identificateur fichier : Expression numérique comprise entre 1 et 255 donnant le numéro de fichier.

Longueur : Expression numérique comprise entre 1 et 32767. Le fichier est ouvert en accès libre avec des enregistrements de longueur "longueur".

Exemple

```
Open "C:\WINDOWS\SYSTEM\DATA.DAT" For Input As #2 Lock Read
```

En principe nous écrivons notre fichier dans le répertoire de Windows ou de System, mais pour le cas où vous souhaiteriez le créer dans un répertoire nous allons étudier de prime abord la création du répertoire devant accueillir notre fichier



L'application "CREREP"

Je vous ai concocté une petite application pour vous permettre de créer vos répertoires favoris.

Télécharger ici!

Le listing complet est disponible ci-dessous (1) .

```
Private Sub Command2_Click()
'Contrôler si répertoire existe?
Dim MyPath As String 'Définition des variables
Dim MyName As String
Dim RepertName As String
'
RepertName$ = Text1.Text
MyPath = "C:\windows\" & RepertName$ 'On initialise la variable
If Text1.Text = "" Then
Title$ = "Création d'un répertoire"
msg = "Veuillez saisir le nom du répertoire à créer!"
MsgBox msg, 64, Title$
Text1.SetFocus
Else
'Instructions
End If
'Si l'utilisateur n'a pas saisi un nom... alors sortie du programme
If Text1.Text = "" Then Exit Sub
'Puisqu'il a saisi un nom, nous traitons sa demande!
If (MyName = Dir(MyPath, vbDirectory)) = vbEmpty Then
'On teste l'existence du répertoire
MsgBox "Le répertoire " & Chr(34) & MyPath & Chr(34) & " existe bien!"
'Si le répertoire saisi existe... alors sortons....
Exit Sub
```

```
Else
  MsgBox "Le répertoire " & Chr(34) & MyPath & Chr(34) & " n 'existe pas!", vbExclamation
  'Alors...créons le répertoire
  RepertName$ = Text1.Text
  Mkdir "c:\windows\" & RepertName$ 'Indiquez le chemin ou doit être stocké le répertoire
End If
'Ouverture d'une fenêtre de dialogue
MsgBox "Le répertoire " & RepertName$ & " vient d'être créé!"
End Sub
```

Pour pouvoir écrire dans un fichier, nous devons d'abord le créer, quelles sont les fonctions que nous offrent VB pour nous permettent de créer un fichier sous Windows ?

ECRIRE DANS UN FICHIER

Tous les fichiers devant comporter un numéro, nous utiliserons la fonction "FreeFile" qui retourne le premier numéro de fichier libre.

Quelles sont les informations que nous donne VB sur cette Fonction :

Description : Retourne l'identificateur de fichier suivant et disponible.

Remarque : Cette fonction est particulièrement utile quand vous désirez ouvrir un fichier sans savoir avec certitude si l'identificateur de fichier n'est pas déjà occupé.

Nous allons le conserver dans la variable NumFich(Numéro de fichier), nous écrivons

NumFich = FreeFile

Création du fichier "Protect.org" (vous pouvez saisir le nom de votre choix)

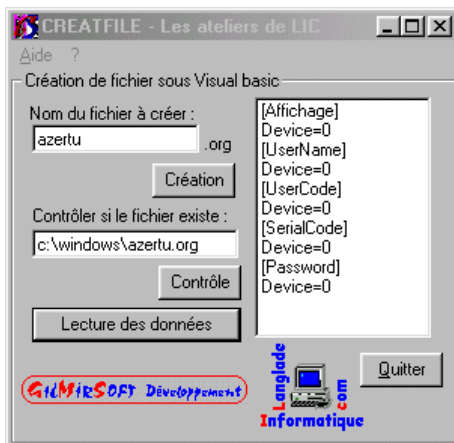
```
Sub CreatSerialNumberCode()
  Dim UserCode As String
  Dim SerialCode As String
  Dim PasswordNumberCode As String
  Dim Valeur As String
  Dim UserName As String
  Dim msg As String
  Dim MonFichier As String
  '
  'Rechercher si le fichier existe?
  NumFich = FreeFile
  MonFichier$ = "c:\windows\protect.org"
  If Dir(MonFichier$, vbNormal Or vbReadOnly Or vbHidden Or vbArchive) = "" Then
    'Le fichier n'existe pas, alors...le créer!
    Open "c:\windows\protect.org" For Output As #NumFich
    '=====[Affichage]=====
    'Ecriture des valeurs dans la clé [Affichage]
    Print #NumFich, "[Affichage]"
    Valeur$ = "0" 'Récupérer la valeur dans le champ de saisie approprié - Ex
    Protect.lblAffichage.Caption
    'Afficher la valeur récupérée
    Print #NumFich, "Device="; Valeur$
    '=====[UserName ]=====
    'Ecriture des valeurs dans la clé [UserName]= Nom de l'utilisateur
    Print #NumFich, "[UserName]"
  End If
End Sub
```

```

UserName$ = "0" 'Récupérer la valeur dans le champ de saisie approprié - Ex
Protect.lblUsername.Caption
'Afficher la valeur récupérée
Print #NumFich, "Device="; UserName$
'===== [UserCode] =====
'Ecriture des valeurs dans la clé [UserCode] = Nom de l'ordinateur
Print #NumFich, "[UserCode]"
UserCode$ = "0" 'Récupérer la valeur dans le champ de saisie approprié - Ex
Protect.lblUserCode.Caption
'Afficher la valeur récupérée
Print #NumFich, "Device="; UserCode$
'===== [NumberSerie] =====
'Ecriture des valeurs dans la clé [Serial Code]Code de controle
SerialCode$ = "0" 'Récupérer la valeur dans le champ de saisie approprié - Ex
Protect.lblSerialCode.Caption
Print #NumFich, "[SerialCode]"
'Afficher la valeur récupérée
Print #NumFich, "Device="; SerialCode$
'===== [Password] =====
'Ecriture des valeurs dans la clé [Password]
Print #NumFich, "[Password]"
PasswordNumberCode$ = "0" 'Récupérer la valeur dans le champ de saisie approprié - Ex
Protect.lblPasswordNumberCode.Caption
'Afficher la valeur récupérée
Print #NumFich, "Device="; PasswordNumberCode$
'Ferme le fichier
Close #1
Else
'Le fichier existe, alors...pas de commentaires, sortie...
End If
End Sub

```

Dans cet exemple si vous ne remplacez pas la valeur "0" par le contenu du champs de saisie de l'exemple cité, vous afficherez "0"(Zéro)



L'application CREATFILE

Une vue de l'application concoctée spécialement pour les internautes de www.developpez.com

Elle vous permettra de mieux comprendre le système employé dans ce module de conception originale

Si l'on veut laisser le choix à l'utilisateur du nom du fichier à créer, nous utiliserons les données suivantes, à savoir :

```

NomFile$ = ctlTextel.Text
NumFich = FreeFile
Open NomFile$ + ".org" For Output As #NumFich

```

L'extension est fantaisiste, libre choix à vous d'utiliser celle qui vous conviendra.

Télécharger l'exemple ci contre ici!

Vous savez dorénavant créer un répertoire, créer un fichier et saisir les données que vous souhaitez mettre en mémoire, il ne nous reste plus qu'à réaliser l'application qui va nous permettre de lire ce fichier.

Vous ajoutez une zone de liste et vous lisez dans celle-ci les données enregistrées afin de les traiter dans votre application.

Prochain tutoriel :

 ***Les applications réalisées sans interface utilisateur***

1 : Dans cet exemple, le dossier "RepertName\$" sera créé dans le répertoire de "Windows"