

Tutoriel 31 : Les attributs de fichiers

par Gilbert Miralles (gilmir.developpez.com)

Date de publication : Lundi 18 mars 2003

Dernière mise à jour : Lundi 4 février 2008



Les attributs de fichiers (lecture seule- Fichier caché - Archives) La modification des attributs système
Prochain tutoriel :

Les attributs de fichiers (lecture seule- Fichier caché - Archives) La modification des attributs système

J'ai vainement cherché dans mes archives si je trouvais une documentation relative aux attributs de fichier, mais en vain. J'ai bien trouvé la propriété "Attributs de fontes" mais ce n'est pas l'expression que je recherchais, aussi nous allons nous contenter des propriétés directes qu'il est possible de modifier lorsque nous rencontrons des attributs de fichiers.

Certains en lisant ce commentaire doivent penser, mais ou veut -il en venir ? Il suffit de modifier par un clic de souris le fichier dont les caractéristiques possèdent l'attribut "Fichier caché" en attribut "Archives".

Oui, c'est vrai, mais n'oublions pas que pour nous programmeur cela est facile, mais supposons que nous voulons écrire un programme qui doit manipuler des fichiers et surtout des fichiers qui possèdent l'Attribut caché et que notre programme veuille modifier les caractéristiques de ce type de fichier.

Cela ne fonctionnera pas et risque d'engendrer une erreur fatale avec l'explication affichée dans une fenêtre de texte du genre, erreur caractéristique N° xx. Nous ne sommes pas plus avancé et pour solutionner notre problème nous devons avant de pouvoir modifier le fichier transformer l'attribut en lui donnant par exemple l'attribut "lecture simple", exécuter les modifications s'il y a lieu, et ensuite le retransformer en un fichier caché.

Le tour est joué et cela marche, aussi pour appuyer la logique de mon raisonnement nous allons tenter une petite expérience en créant un condensé de l'application "Savereg" © 2000 GilMirSoft Développement

Vous pouvez télécharger le projet ici Télécharger [Disk1.zip](#) et [Disk2.zip](#) (1.337Ko - 1.088Ko)

Dans ce projet, nous voulons récupérer des fichiers sensibles pour les mettre dans un lieu sur connu de nous même pour pouvoir en cas de problème grave écraser les fichiers existants qui ont été endommagés par une mauvaise manipulation ou peut être par un virus.

Vous comprenez que cette opération ne peut être envisagée par une entité humaine qui voudrait triturer la base de registre sans avoir l'expérience requise en la matière et qui au lieu de dépanner sa machine risquerait au contraire d'aggraver les problèmes existants au

risque de la rendre irréparable(j'exagère un peu).

Nous allons passer sur l'écriture de l'initialisation des paramètres du PC de l'utilisateur qui a fait l'objet d'un précédent Tutoriel.

Quels sont les fichiers qui peuvent nous intéresser ? (Si j'en oublie, vous pouvez les ajouter)

Autoexec.bat, Config.sys, Bootlog.txt, Win.ini, System.ini, User.dat, System.dat

Important :

Je vous conseille avant de manipuler ces fichiers de les sauvegarder dans un répertoire nouvellement créé à la racine du DOS. Exemple : C:\FileSystem\Fichiers à sauvegarder(voir liste)

En ce qui concerne la modification d'attributs système seul deux fichiers sont protégés et feront l'objet de modifications, le fichier "System.dat" et le fichier "User.dat" ces fichiers sauvegardent votre base de registre et feront l'objet de la plus grande attention.

Nous avons activé dans notre générateur de programme la déclaration "Option Explicit" ce qui nous oblige à déclarer toute procédure nouvellement créée.

Pour simplifier, je n'affiche pas les déclarations de procédures dans les instructions ou fonctions car elles ont été déclarées dans un module Global.

Si vous ne prenez pas la peine de les déclarer Globalement VB va engendrer un erreur et va vous demander de déclarer toute nouvelle procédure.

Pour les initiés, je pense que je n'avais pas besoin de le leur dire, désolé, mais...pour le cas ou !

Synoptique de l'ordinogramme

- 1 Contrôler si les fichiers qui nous intéressent existes
- 2 Désactiver l'attribut "Lecture seule"
- 3 Exécuter les instructions recherchées
- 4 Réactiver l'attribut "Lecture seule"
- 5 Sauvegarder les modifications

Ecriture du code

```
Private Sub SSCommand1_Click ( )           'Bouton de commande
Dim FileExist As String
'
On Error GoTo FileNotExist                 'Si erreur sortie discrète sans provoquer
d'erreur
FileExist = "C:\autoexec.bat"             'Nom du fichier à rechercher
If Dir (FileExist, vbNormal Or vbReadOnly Or vbHidden Or vbArchives = "" Then
    Call MsgBox("Ce fichier n'existe pas!")
    Exit Sub                               'Sortie de la procédure
Else
    'Il existe, alors...instructions suivantes 'Pour la forme
    'msg = "OK"
    'MsgBox msg
End If
```

Nous écrivons exactement les mêmes routines pour les autres fichiers à savoir que certains fichiers se trouvent dans le répertoire de Windows et qu'il faudra afficher le chemin complet.

Lorsque vous avez écrit toutes les routines afférentes aux fichiers recherchés, vous saisissez la routine d'erreur suivante:

```
FileNotExist :
'Instructions
msg = "La sauvegarde des fichiers systèmes est impossible, désolé !"
MsgBox msg, vbExclamation, Title$
End sub
```

Sauvegarde des fichiers systemes

```

Private Sub SSCommand5_Click( )
    '
    Dim MyPath As String
    Dim MyName As String
    Dim Reponse As Integer
    Dim A As String
    Dim B As String
    '
    'Contrôle de l'existence du répertoire de sauvegarde
    Title$ = "Restauration des fichiers systèmes"
    msg = "Voulez-vous vraiment restaurer le système ?"
    Reponse% = MsgBox(msg, vbYesNo, Title$)
    If Reponse% = 7 Then
        msg = "La sauvegarde a été annulée!"
        MsgBox msg, vbExclamation, Title$
        'L'utilisateur a répondu "Non" alors sortie!
        Exit Sub
    End If
    'L'utilisateur a répondu "Oui"
    'Avant de sauvegarder le système contrôler si le répertoire existe
    MyPath = "C:\windows\filesystem"
    If (MyName = Dir(MyPath, vbDirectory)) = vbEmpty Then
        'On teste l'existence du répertoire, il existe alors...
        Frame1.Caption = "Restore Files System"
        List1.Visible = True
        List1.AddItem "Restauration des fichiers du registre système! Veuillez patienter SVP!"
        MousePointer = 11
    End If
End Sub
    
```

Le système affiche une ProgressBar (qui fera l'objet d'un prochain Tutoriel)

```

'===== Restauration du systeme =====
'Récupérer et transférer les fichiers dans le répertoire de sauvegarde
A = "c:\windows\filesystem\autoexec.bat"
B = "c:\autoexec.bat"
FileCopy A, B
'
A = "c:\windows\filesystem\config.sys"
B = "c:\config.sys"
FileCopy A, B
'
A = "c:\windows\filesystem\bootlog.txt"
B = "c:\bootlog.txt"
FileCopy A, B
'
A = "c:\windows\filesystem\win.ini"
B = "c:\windows\win.ini"
FileCopy A, B
'
A = "c:\windows\filesystem\system.ini"
B = "c:\windows\system.ini"
FileCopy A, B
'
'Désactiver les attributs systèmes
ModifAttribMoins
A = "c:\windows\filesystem\system.dat"
B = "c:\windows\system.dat"
FileCopy A, B
'
A = "c:\windows\filesystem\user.dat"
B = "c:\windows\user.dat"
FileCopy A, B
'
'Réactiver les attributs systèmes
ModifAttribPlus
'
'===== Fin de la Restauration =====
    
```

```
Title$ = "Restauration des fichiers systèmes"
msg = "La restauration de la base de registre s'est bien déroulée!"
MsgBox msg, vbOKOnly, Title$
EradicationPicture
Frame1.Caption = "Description"
MousePointer = 0
End If
End Sub
'==== Nous créons les procédures de modifications d'attributs system insérés dans un module ===
```

Désactivation des attributs systèmes

```
Sub ModifAttribMoins( )
Dim MyFile As String
'
MyFile = "c:\windows\system.dat"
If Dir(MyFile, vbNormal Or vbReadOnly Or vbHidden Or vbSystem Or vbArchive) = "" Then
Call MsgBox("Erreur, désolé! Le fichier de sauvegarde n'existe pas!")
Exit Sub
Else
'Le fichier existe, Affiche l'attribut fichier caché
MyFile = GetAttr("c:\windows\system.dat")
NomProg.Label7.Caption = MyFile
'
'Modifier l'attribut système
MyFile = "c:\windows\system.dat"
SetAttr MyFile, vbNormal
MyFile = GetAttr("c:\windows\system.dat")
NomProg.Label7.Caption = MyFile
End If
'
MyFile = "c:\windows\user.dat"
If Dir(MyFile, vbNormal Or vbReadOnly Or vbHidden Or vbSystem Or vbArchive) = "" Then
Call MsgBox("Erreur, désolé! Le fichier de sauvegarde n'existe pas!")
Exit Sub
Else
'Le fichier existe, Affiche l'attribut fichier caché
MyFile = GetAttr("c:\windows\user.dat")
NomProg..Label7.Caption = MyFile
'
'Modifier l'attribut système
MyFile = "c:\windows\user.dat"
SetAttr MyFile, vbNormal
'Afficher à nouveau pour contrôle!
MyFile = GetAttr("c:\windows\user.dat")
NomProg.Label7.Caption = MyFile
End If
End Sub
```

Réactivation des attributs systèmes

```
'Il suffit simplement de faire l'opération à l'envert
Sub ModifAttribPlus( )
Dim MyFile As String
'
MyFile = "c:\windows\system.dat"
If Dir(MyFile, vbNormal Or vbReadOnly Or vbHidden Or vbSystem Or vbArchive) = "" Then
Call MsgBox("Erreur, désolé! Le fichier de sauvegarde n'existe pas!")
Exit Sub
Else
'Le fichier existe, Affiche l'attribut!
MyFile = GetAttr("c:\windows\system.dat")
NomProg.Label7.Caption = MyFile
'
'Modifier l'attribut système en fichier caché et en lecture seule
```

```
MyFile = "c:\windows\system.dat"
SetAttr MyFile, vbHidden + vbReadOnly
'Afficher à nouveau pour contrôle
MyFile = GetAttr("c:\windows\system.dat")
NomProg.Label7.Caption = MyFile
End If
'
MyFile = "c:\windows\user.dat"
If Dir(MyFile, vbNormal Or vbReadOnly Or vbHidden Or vbSystem Or vbArchive) = "" Then
Call MsgBox("Erreur, désolé! Le fichier de sauvegarde n'existe pas!")
Exit Sub
Else
'Le fichier existe, Affiche l'attribut !
MyFile = GetAttr("c:\windows\user.dat")
NomProg.Label7.Caption = MyFile
'
'Modifier l'attribut système en fichier caché et en lecture seule
MyFile = "c:\windows\user.dat"
SetAttr MyFile, vbHidden + vbReadOnly
'Afficher à nouveau pour contrôle
MyFile = GetAttr("c:\windows\user.dat")
NomProg.Label7.Caption = MyFile
End If
End Sub
```

Prochain tutoriel :

 ***Les applications MDI - Avant Propos***

