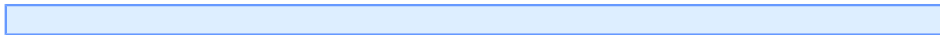


Tutoriel 9 : Utiliser des objets systèmes de fichiers

par Gilbert Miralles (gilmir.developpez.com)

Date de publication : Lundi 18 mars 2003

Dernière mise à jour : Lundi 28 janvier 2008



I - Introduction

II - Application Explorateur

Prochain tutorial :

I - Introduction

Visual Basic propose trois objets très utiles permettant d' accéder au système de fichiers. Il s' agit des contrôles "DriveListBox"(liste de lecteurs), "DirListBox"(liste de répertoires), et "FileListBox"(liste de fichiers).



La liste de lecteur permet de parcourir les lecteurs valides de votre système.



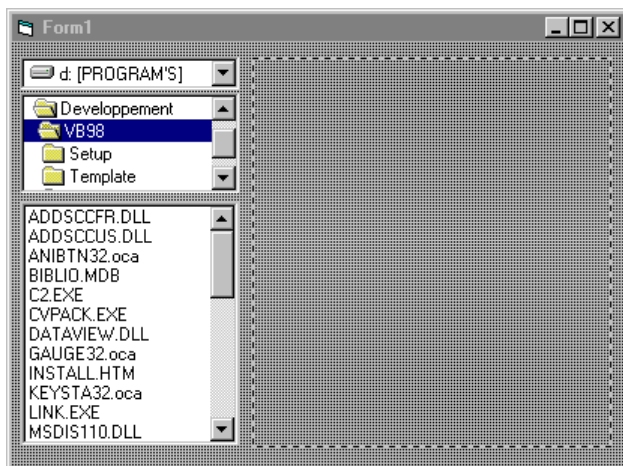
La liste de répertoires permet de naviguer parmi les dossiers d' un lecteur particulier.



La liste de fichiers permet de sélectionner un fichier spécifique dans un dossier.

II - Application Explorateur

Vous pourrez télécharger en option le fichier compressé de l'application "Explorateur" qui reprend les contrôles que nous venons d'évoquer.



Voici le prototype de l' application "**Explor**"

Les dimensions n' ont pas grande importance, à vous d' ajuster les paramètres en fonction de vos aptitudes créatrices.

Vous pouvez passer en mode Exécution pour voir ce qui se passe ?

Un objet "FileListBox"(liste de fichiers) permet à l' utilisateur de sélectionner un fichier spécifique dans le système de fichier.

Visual basic insère alors le nom du fichier sélectionné dans la propriété "FileName" de l' objet.

Les sélections appropriées effectuées par l' utilisateur sont récupérées également à l' identique par les objets :

DriveListBox qui sera inséré dans la propriété "Drive" de l' objet et, DirListBox sera inséré dans la propriété "Path" de l' objet.

Nous utiliserons ces trois propriétés dans le programme Explorateur.

L' utilisateur change le paramètre d' un objet durant l' exécution d' un programme, cette modification engendre un événement qui est traité par Visual Basic dans le code du programme.

Nota - Les propriétés "Drive", "Path" et "FileName" ne sont accessibles qu' en phase d' exécution. Il est impossible de les définir à l' aide de la fenêtre de propriétés.

Nous allons définir les propriétés suivantes en utilisant la fenêtre "Propriétés"

Objets	Propriétés	Paramètres
File1	Pattern	*.bmp ; *.* ; *.wmf ; *.ico
Image1	Stretch	True
Image1	BorderStyle	1 - Fixed Single

Si nous n' avons pas utilisé la propriété "Pattern" la "**FileListBox**" aurait affichée tous les types de fichiers contenus dans un dossier avec le risque que l' utilisateur sélectionne un fichier qui n' est pas supporté par **Visual Basic**. Ce qui aurait pour effet de provoquer une erreur. Il est préférable dans la mesure du possible d' essayer d' éviter tout risque d' erreurs, tout au moins celles qui sont prévisibles.

Nous allons ajouter quelques lignes de code aux procédures qui sont associées aux objets. Ces procédures sont appelées "**Procédures d' événement**" car elles sont engendrées par un événement, par exemple *un clic de souris*.

On affiche par un double clic sur l' objet "**DriveListBox**" la fenêtre de code de l' objet (Drive1) et particulièrement dans la procédure "Change" on écrit :

```
Private Sub Drive1_Change ( )  
    'L' utilisateur à cliqué sur l' objet  
    Dir1.Path = Drive1.Drive  
End Sub
```

Cette instruction actualise la propriété "**Path**" dans la liste de lecteurs lorsque l'utilisateur sélectionne un lecteur de disque dans la zone.

Avec cette simple ligne de code nous constatons que nous pouvons changer de lecteur et lire dans le contrôle DirListBox les répertoires se trouvant dans le lecteur sélectionné. Essayez de passer en mode Exécution pour visualiser le fonctionnement.

Nous allons effectuer la même opération avec la "**DirListBox**"

```
Private Sub Dir1_Change ( )  
    'L' utilisateur à cliqué sur l' objet  
    File1.Path = Dir1.Path  
End Sub
```

Cette instruction lie la liste des fichiers à celle des répertoires de manière à ce que les fichiers présentés correspondent à ceux du dossier sélectionné.

Nous allons effectuer la même opération avec la "**FileListBox**" et ajouter le code suivant :

```
Private Sub File1_Click()  
    Dim SelectedFile As String  
    'L'utilisateur a sélectionné l' objet  
    SelectedFile = File1.Path & "\" & File1.FileName ' (&) Voir opérateurs  
    Image1.Picture = LoadPicture(SelectedFile)  
End Sub
```

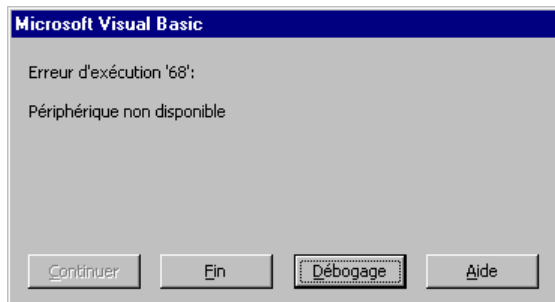
Nous avons déclaré la variable SelectedFile, concaténé les instructions afférentes au fichier sélectionné puis rajouté l' anti-slash pour le cas ou ? et avons donné au programme le chemin du fichier qui nous intéresse en l' occurrence celui que l' utilisateur a sélectionné dans la liste des fichiers valides.

Le paramètre 1(Fixed Single) de la propriété BorderStyle de l' objet "Image" nous permet de cadrer l' image sélectionnée en fonction de ces propres dimensions(pour l' esthétique).

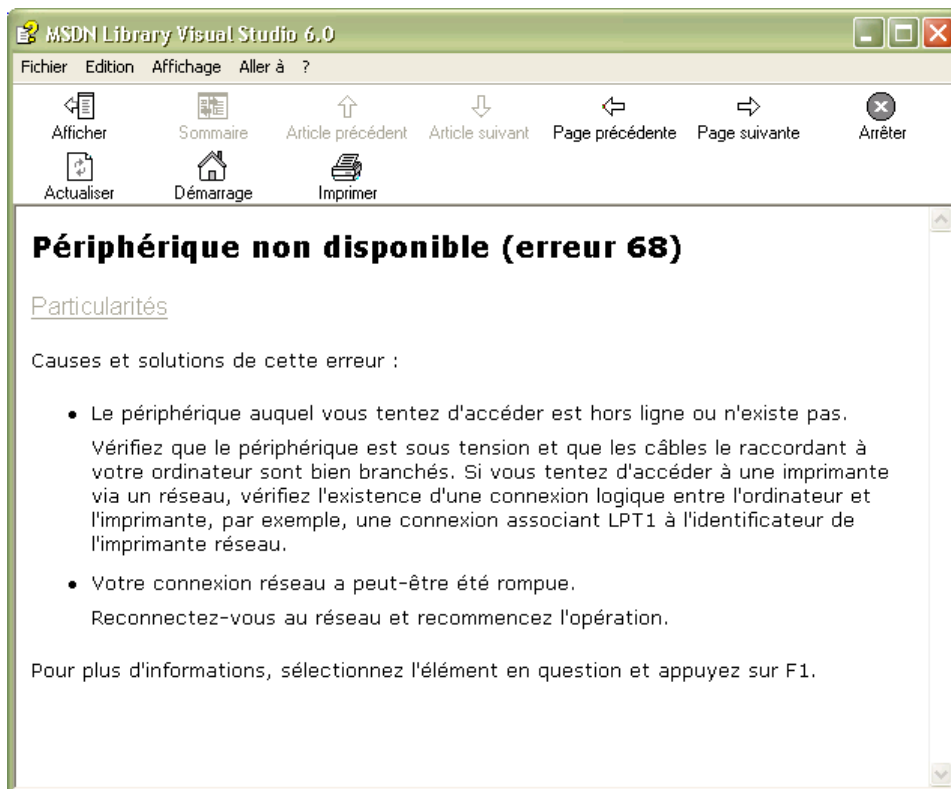
Exécutons le programme en procédant comme d' habitude.

Lorsque le programme se trouve en mode Exécution, sélectionnez le contrôle "Drive1" et choisissez le lecteur "A:", vous constatez que le programme(VB) génère une erreur d' exécution en ouvrant et affichant une fenêtre du style :

Voici la fenêtre affichée par Visual Basic lorsque vous générez l' erreur d' exécution n° 68 qui stipule que le périphérique est non disponible.



Pour ceux qui possèdent le programme Visual basic d' origine Microsoft, vous pouvez cliquer sur le bouton d'aide, ce qui vous permet de savoir que vous disposez d' une aide en ligne bien ficelée qui vous aidera dans bien des cas à trouver la procédure ou l' instruction recherchée. Cliquez sur le bouton d' aide et vous devez voir apparaître la fenêtre suivante:



Périphérique non disponible (erreur 68)

PARTICULARITES, Causes et solutions de cette erreur :

- Le périphérique auquel vous tentez d'accéder est hors ligne ou n'existe pas.

Vérifiez que le périphérique est sous tension et que les câbles le raccordant à votre ordinateur sont bien branchés. Si vous tentez d'accéder à une imprimante via un réseau, vérifiez l'existence d'une connexion logique entre l'ordinateur et l'imprimante, par exemple, une connexion associant LPT1 à l'identificateur de l'imprimante réseau.

- Votre connexion réseau a peut-être été rompue.

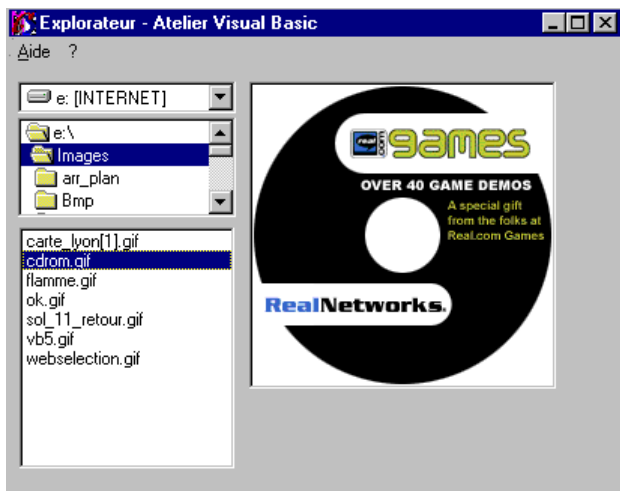
Reconnectez-vous au réseau et recommencez l'opération.

Pour plus d'informations, sélectionnez l'élément en question et appuyez sur F1.

Il est possible de générer d'autres erreurs avec ce programme, aussi sachez que pour résoudre ces problèmes, il faut utiliser des instructions qui contournent la cause de l'erreur, ou créer des routines spéciales nommées "Gestionnaires d'erreurs" qui vous permettront de contourner les problèmes rencontrés par cette application qui demande à être déboguée.

Nous apprendrons plus tard à écrire un gestionnaire d'erreurs et à rechercher les erreurs éventuelles.

Le programme "Explorer" est terminé, nous affichons le résultat de notre réalisation.



Le programme en lui-même est très simple, mais si vous le souhaitez, vous pouvez télécharger les sources [ici](#)

Nd'A- L'image affichée par le programme Visual Basic est un Logo de la firme Real.Com

Sauvegardez votre réalisation.

Dans le menu, cliquez sur l'étiquette "Enregistrez le projet sous" : vous pouvez modifier le nom de la feuille "Form1" par "Explorer" qui prendra aussitôt l'extension "Explorer.frm" et le projet que nous venons de réaliser "Projet1.vbp" sera sauvegardé sous le nom de "Explorer.vbp"

Vous pouvez également créer votre première application exécutable

Dans la barre de menu cliquez sur l'étiquette "Fichier" vous devez avoir dans la fenêtre qui vient de s'ouvrir une étiquette comportant la saisie : "Créer Projet1.exe" En cliquant sur cette étiquette vous ouvrez une fenêtre de dialogue de Windows qui comporte dans le champs de saisie le nom de votre projet.

Vous le renommez en lui donnant le nom de votre choix.

Dans l'exemple que je viens d'évoquer, je l'ai appelé "Explorat.exe" Vous avez pu constater que dans le libellé du nom, je n'ai saisi que 8 caractères.

Cela évite qu'il puisse être renommé par "Dos" (Si je l'avais affublé d'un nom Long) pour le cas où un utilisateur le lancerait depuis cette plate forme.

Prochain tutorial :

 ***La case à cocher et la zone de liste***

