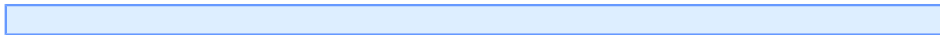


Tutoriel 25 : Les applications réalisées sans interface utilisateur - la feuille Sub Main

par Gilbert Miralles (gilmir.developpez.com)

Date de publication : Lundi 18 mars 2003

Dernière mise à jour : Lundi 4 février 2008



Un programme sans interface utilisateur graphique

Application "Close Win"

Application "RunProg"

Application Gestion d'erreurs

Conclusion

Prochain tutoriel :

Un programme sans interface utilisateur graphique

Rien ne s'oppose aux premiers pas sous Visual Basic, si vous êtes débutants, sachez rester humble pour vos débuts. Avant de vous lancer dans une super production à la Métro Goldwin Mayer, apprenez à vous servir de l'environnement de programmation, à créer des feuilles et à saisir du code.

L'interface graphique n'est pas impérative pour tous les utilitaires de Visual Basic, ce qui permet de les écrire exclusivement sous l'éditeur de texte.

Le sous programme "Sub Main" que je vous propose d'étudier a une signification particulière car Visual Basic ne connaît pas de programme principal. Il peut servir à lancer l'exécution du programme à la place d'une feuille.

Les quelques exemples qui vont être développés vont vous montrer comment écrire des programmes dépourvus d'interface utilisateur graphique.

Ouvrez un nouveau projet par "Fichier"/"NouveauProjet", choisissez l'icône "EXE" standard et affichez la fenêtre de projet "Projet1".

Dans celle-ci vous devez avoir un fichier d'inscrit "Form1", en cliquant avec le bouton droit de votre souris sur l'étiquette, vous ouvrez une nouvelle fenêtre, recherchez l'étiquette "Supprimez Form1"

et cliquez sur celle-ci pour provoquer l'effacement de la feuille "Form1" dans la fenêtre de projet.

Dans la barre de menu de VB cliquez sur "Projet"/"Ajoutez un module

Le fichier "Module1" vient d'être créé !

Dans l'objet "Général" - "Déclarations", tapez :

Sub Main et appuyez sur le bouton <Return> ou <Entrée> de votre clavier. Vous venez de créer la procédure :

```
Sub Main ( )  
  
End Sub
```

Application "Close Win"

Nous allons écrire une petite application qui permet d'arrêter la machine.

Dans l'objet "Général" - "Déclaration" on écrit :

```
Declare Function ExitWindows Lib "user32" Alias "ExitWindowsEx" (ByVal dwReserved As Long, ByVal  
uReturnCode As Long) As Long
```

(mettez cette instruction sur une seule ligne)

Et dans la procédure Main précédemment créée on saisi :

```
Sub Main( )  
  'Dim X As Long  
  Dim y As Long  
  '  
  'X = ExitWindows(4, EWX_FORCE)  
  y = ExitWindows(1, EWX_SHUTDOWN)  
End Sub
```

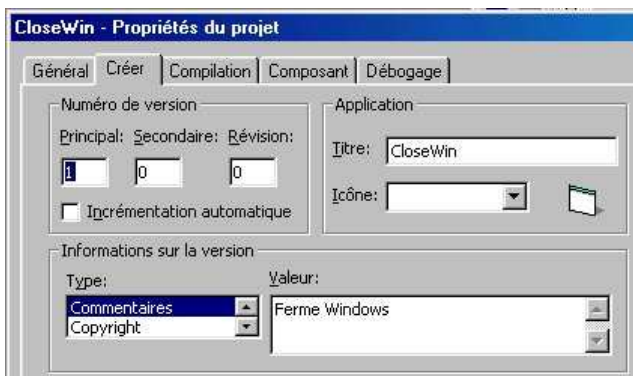
Vous avez vu que j'ai mis en commentaire une instruction(X)

En effet j'ai découvert en feuilletant la documentation de Windows que cette instruction permet de forcer la fermeture d'une application qui a buggé, qui vous affiche parfois un panneau de couleur qu'il est souvent difficile de faire disparaître.

Je ne l'ai pas essayé ? manque de temps désolé, je vous suggère de créer la même procédure en mettant cette-fois l'instruction "Y" en commentaire.

Et si possible me faire savoir si cela fonctionne ? (Petit exercice)

Voilà c'est tout ! Votre programme est terminé, il ne nous reste plus qu'à créer un fichier exécutable en utilisant dans la barre de menu l'étiquette adéquate, mais, je vais broder un tout petit peu en vous demandant d'insérer dans votre programme les renseignements relatifs à la réalisation de votre application, cela fait un peu plus "Pro".



Plusieurs onglets se présentent à nous, celui qui nous intéresse se nomme, "Créer". Nous l'ouvrons et dans la "Frame" "Informations sur la version", nous découvrons une ComboBox qui liste différentes étiquettes, à savoir :

Commentaires :

Copyright :

Description du fichier :

Marques légales :

Nom du produit :

Organisation :

Remplissez le champ de saisie qui se trouve à droite des étiquettes en fonction de vos critères respectifs.

Pour pouvoir récupérer le nom de notre réalisation, nous la sauvegardons sous le nom de "CloseWin", ensuite nous pouvons compiler le fichier en allant dans la barre de menu et en cliquant sur :

"Fichier"/"Créer CloseWin.exe" Ce fichier pourra être stocké où bon vous semble, d'autant que nous allons l'ouvrir à partir d'un raccourci clavier.

Je ne veux pas douter de vos compétences, néanmoins pour ceux qui ne connaîtraient pas la marche à suivre, voici comment faire un raccourci clavier depuis le bureau(l'écran) de votre computer.

Après la compilation de votre projet et la fermeture de Windows, procédez comme suit :

Cliquer du bouton droit sur le bureau (*) **'Selon la configuration de votre souris(*)**

Cliquez sur "Nouveau" (bouton Gauche) (*)

Cliquez sur "Raccourci"

Cliquez sur le bouton "Parcourir"

Sélectionnez le programme **"CloseWin.exe"**

Cliquez sur le bouton "Suivant", puis sur le bouton "Terminer"

Vous pouvez arrêter votre appareil depuis l'icône que vous venez d'installer sur le bureau!

Vous pouvez également modifier l'icône installé sur votre bureau, en cliquant sur le bouton droit de la souris, puis sur "Raccourci"/"Changez d'icône", Parcourir et rentrer dans le répertoire de Windows pour rechercher le fichier "Moricons.dll", choisissez l'icône qui vous convient.

Application "RunProg"

Je vous propose pour clôturer ce chapitre de réaliser une autre application basée sur le même principe.

Cela sera plus simple maintenant puisque nous en connaissons le fonctionnement.

Cette application sera sauvegardée sous le nom de **"RunProg.exe"**

Nous créons sous le même principe évoqué précédemment un fichier "Module1.bas" que nous renommerons sans plus attendre **"RunProg.bas"**.

Dans l'objet "Général" - "Déclarations", tapez :

```
Public Declare Function ShellExecute Lib "shell32.dll" Alias "ShellExecuteA" _
    (ByVal hwnd As Long, ByVal lpOperation As String, ByVal lpFile As String, _
    ByVal lpParameters As String, ByVal lpDirectory As String, ByVal nShowCmd As Long) As Long
```


Faites un copier/coller c'est plus simple et surtout plus sûr!

[Ctrl + c] pour copier et [Ctrl +V] pour coller, petit rappel, dès fois que...!

Et dans la procédure Main précédemment créée on saisi :

```
Sub Main()  
    Dim RetVal  
    Dim fName As String  
    fName = "Index.html" 'Ou le nom du fichier que vous voulez ouvrir  
    RetVal = ShellExecute(hwnd, "Open", fName, ByVal 0&, 0&, SW_Normal)  
End Sub
```

Il ne reste plus qu'à compiler notre nouvelle réalisation qui sera insérée dans les fichiers devant visualiser une composition appartenant à un Cd-Rom.

 *J'ai réalisé une variante de ce programme que je vous laisserai découvrir en téléchargement*

Télécharger "RunProg.exe" classique selon l'exposé décrit dans cette page

Télécharger "RunProg1.exe" variante que j'utilise depuis des années dans la diffusion de mes Cd-Rom

Ce programme a été écrit initialement en VB4.0 puis Transcrit en VB6.0

Application Gestion d'erreurs

Autre exemple :

LES PROPRIETES DE L'OBJET "ERREUR"

Une fois que l'exécution du programme a été transmise à votre routine de gestions d'erreurs, votre code peut déterminer l'erreur qui s'est produite en affichant son numéro et sa description.

Quand une erreur se produit, les informations la concernant sont stockées dans l'objet "Err"

Err.Number : indique le numéro de l'erreur

Err.Description : indique la description de l'erreur

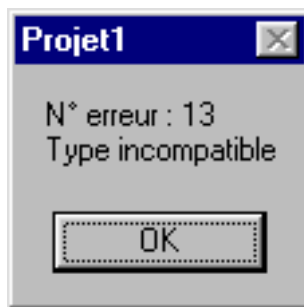
Voici le listing de cette application :

```
Sub Main ( )  
On Error GoTo SortieDiscrete ' ou l'étiquette de votre choix  
If Dir("c:\toto\danseaveclesloups.txt") = False Then  
    MsgBox "Ce fichier n'existe pas!"  
End If  
Exit Sub  
SortieDiscrete:  
MsgBox "Erreur N° : " & Err.Number & VBLF & Err.Description  
End Sub
```

Si le fichier existe, alors nous sortons de cette routine par l'instruction Exit Sub et nous ignorons l'étiquette "SortieDiscrete".

Finalement vous constatez que la programmation c'est un circuit ferroviaire avec des aiguillages qui vous dirigent selon le bon vouloir du concepteur du programme.

Si le fichier n'existe pas, le mot "SortieDiscrete" remplit sa fonction et vous renvoie à l'étiquette du même nom, c'est à dire "SortieDiscrete" qui demande à la boîte de messagerie MsgBox de vous indiquer le type d'erreur et sa description.



Lorsque l'on utilise un gestionnaire d'erreur, Visual basic affiche le numéro de l'erreur ainsi que le descriptif de l'erreur. Lorsque l'erreur n'est pas reconnue par Visual Basic un message vous informe que l'erreur est inconnue du système.

Conclusion

Voilà, c'est tout pour aujourd'hui ! Bon travail !

Prochain tutoriel :

 ***Les éléments de programmation "Les Variables"***

