

Tutoriel 18 : Utiliser le contrôle Threed32.ocx et Spin32.ocx dans ses applications

par Gilbert Miralles (gilmir.developpez.com)

Date de publication : Lundi 18 mars 2003

Dernière mise à jour : Lundi 4 février 2008



I - Introduction

II - Le contrôle Threed

III - Le contrôle "SPIN32.OCX"

Prochain tutoriel :

I - Introduction

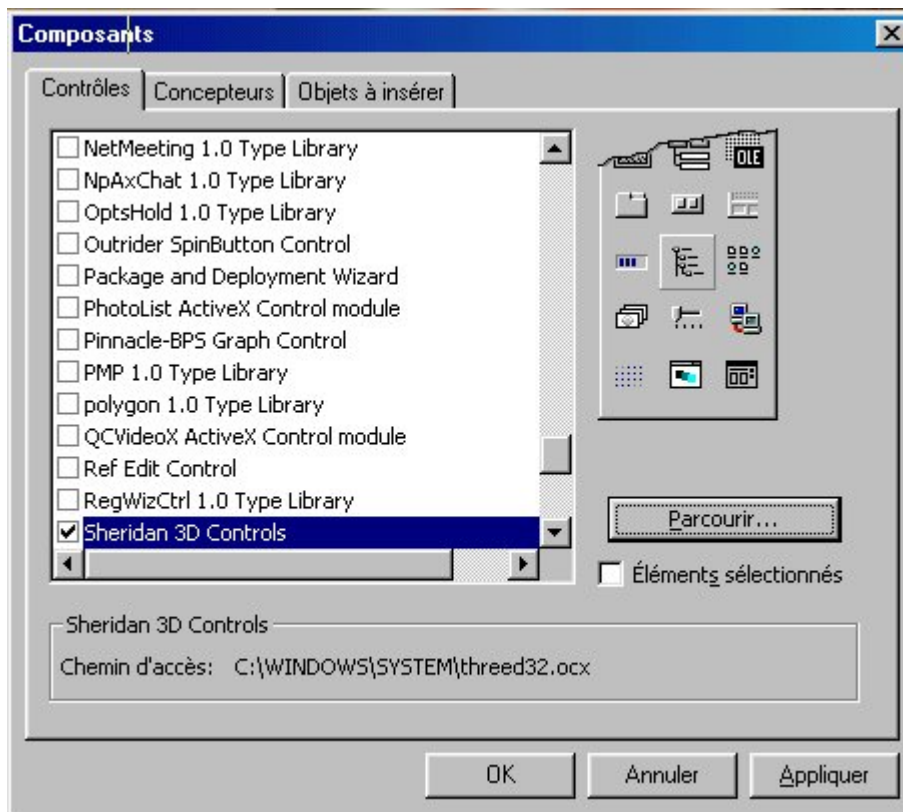
Il m' est difficile d' être catégorique dans mes affirmations car possédant toutes les versions de Visual Basic, je possède en effet les contrôles "Threed" et "Spin", mais je crois que toutes les versions de VB ne fournissent pas ce type d' objet, à me confirmer!

Par contre si vous ne possédez pas en standard l' objet en question, vous ne pourrez pas l' incorporer à votre projet, car ne possédant pas de droits acquis légalement, celui-ci refusera de fonctionner avec votre application.

Visual Basic 4.0 version professionnelle fournissait en standard ce type d' objet, et si vous possédez cette version dans un tiroir, réinstallez la, ce qui vous permettra de pouvoir utiliser ces contrôles dans vos futures applications.

II - Le contrôle Threed

Le contrôle "Threed" issu du fichier "Threed32.ocx" que vous pouvez récupérer depuis le répertoire de "Windows/system" avec la méthode apprise précédemment c'est à dire depuis la barre de menu et en sélectionnant "Projet", "Composants" et parcourir, ce qui vous permet d'afficher la boîte de dialogue "Ajouter un contrôle ActiveX".



Nous venons de sélectionner le contrôle "Threed32.ocx" de la firme "Sheridan"

Il ne nous reste plus qu'à cliquer sur le bouton "Appliquer" et lancer l'insertion du contrôle en appuyant sur le bouton "OK"

A quoi peut nous servir ce contrôle ? Il peut en effet nous servir à beaucoup de choses néanmoins pour ceux qui ne le possèdent pas, laissez moi vous dire qu'il n'est pas nécessaire et que l'on peut s'en passer avec un peu d'ingéniosité.



Voyons d'abord comment se présentent ces nouveaux objets dans la boîte à outils.

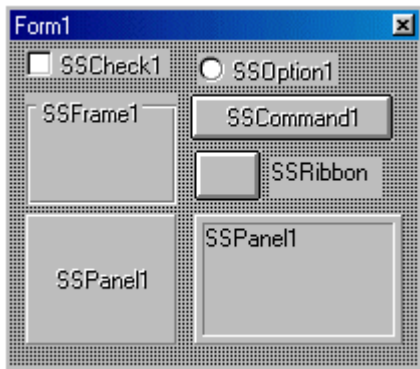
(de haut en bas) La petite croix en haut et à gauche qui porte le nom de "SSCheck1" est une "CheckBox".

Le carré qui porte intérieurement le caractère "F" est une "Frame" et porte le nom de SSFrame1 L'objet de droite est un bouton de commande qui se nomme "SSCommand1"

Dans la ligne du bas, nous avons l'objet "SSPanel1" qui est une sorte de "Panneau conteneur d'objets", vous pouvez y insérer pratiquement tous les objets de votre boîte à outils.

Au centre nous avons un bouton d' options nommé "SSOption1"

Et sur la droite un bouton de commande qui se nomme "SSRibbon1"



Tous ces objets ont la particularité de pouvoir écrire en caractère 3D, en plus le bouton "SSCommand" peut afficher une image en couleur.

Les panels peuvent avoir un affichage graphique différent, j' ai représenté dans la Form1 deux panels identiques , mais représentés graphiquement de façon différente.

Tous ces objets ont leur homologues qui sont livrés en standard par Visual Basic et ne sont donc pas indispensables pour réaliser une interface graphique.

L' avantage si vous les possédez, est que vous allez pouvoir enjoliver votre présentation par une approche plus esthétique de la présentation graphique de votre page d' accueil..

L' inconvénient est que vous allez être obligé de livrer dans votre application le fichier "Threed.ocx" Mais... pourquoi pas ?

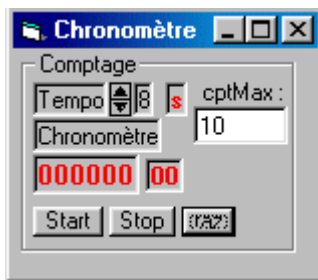
III - Le contrôle "SPIN32.OCX"



Le Contrôle "SPIN32.OCX" double bouton poussoir(supérieur et inférieur)

est très intéressant car il permet de créer toute sorte d' applications qui offrent la possibilité de faire modifier une valeur par l' utilisateur.

Prenons par exemple la réalisation d' un chronomètre dont le pas d' incrémentation doit pouvoir être modifié par l' utilisateur.



Dans cet exemple le bouton "Spin" sert à régler la durée de la temporisation qui est fixé à 8 secondes, elle peut-être modifiée en plus ou en moins par l' utilisateur.

Nous allons voir comment écrire le code nous permettant d' utiliser ce contrôle dans nos applications.

L'objet "SPIN" est un double bouton poussoir qui comprend deux procédures qui génèrent chacune d' entre-elles un événement lorsque vous appuyez sur un des deux boutons.

Le premier événement se produit lorsque vous appuyez sur le bouton supérieur pour incrémenter la valeur par défaut (incrémenter veut simplement dire ajouter 1 à la valeur initiale)

Le deuxième événement se produit lorsque vous appuyez sur le bouton inférieur pour décrémenter la valeur par défaut (décrémenter veut simplement dire retirer 1 à la valeur initiale) (La valeur 1 est un exemple et peut varier selon le vouloir du développeur)

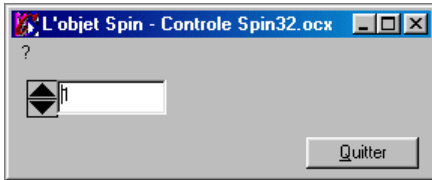
Création des lignes de codes afférentes à cette procédure, nous écrivons :

```

Private Sub SpinButton1_SpinDown( )
    'Jours débutent à 1
    Dim Jour1 As Integer
    Jour1% = Val(ctlJour1.Text)
    If Jour1% > 1 Then
        'Décrémenter
        ctlJour1.Text = Trim$(Str$(Jour1% - 1))
    End If
End Sub

Private Sub SpinButton1_SpinUp( )
    'Jours terminent à 31
    Dim Jour1 As Integer
    Jour1% = Val(ctlJour1.Text)
    If Jour1% < 31 Then
        ' Jour1 est le nom de la variable qui va stocker
        ' la valeur affichée dans une case mémoire
        'ctlJour1 est la Propriété Name de la TextBox
    End If
End Sub
    
```

```
'Incrémenter  
ctlJour1.Text = Trim$(Str$(Jour1% + 1))  
End If  
End Sub
```



L'application "Spin32" avec les sources de l'exemple décrites dans cette page.

Télécharger l'application "Spin"

Télécharger l'application "AppSpin"

Prochain tutoriel :

 ***Accès aux bases de données "ADO" avec Visual Basic 6.0***

