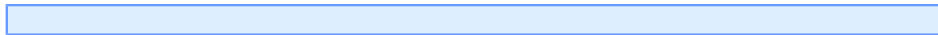


Tutoriel 10 : La case à cocher, le bouton d'options et la zone de liste

par Gilbert Miralles (gilmir.developpez.com)

Date de publication : Lundi 18 mars 2003

Dernière mise à jour : Lundi 28 janvier 2008



I - Le contrôle "CheckBox" (case à cocher)

II - Le contrôle "OptionButton"

III - La zone de liste et zone modifiable

Prochain tutoriel :

I - Le contrôle "CheckBox" (case à cocher)

La plupart des applications laissent une certaine marge de choix à l'utilisateur.

Le contrôle "**CheckBox**"(case à cocher)



permet d'affecter à une option ou un paramètre une valeur qui peut être Vrai ou Faux (True ou False) Les cases à cocher peuvent être rassemblées en groupe, chaque case étant activable individuellement et sans incidence pour les autres cases du même groupe.

Vous pouvez utiliser la propriété "**ToolTip**" qui permet d'adjoindre au contrôle un court texte explicatif sous forme d'une Info-bulle lorsque le pointeur de la souris survole le contrôle un court instant. Une nouveauté dans la version 6.0 de Visual Basic qui propose un contrôle "CheckBox" graphique.

Utilisez pour cela la propriété "**Picture**" pour affecter à la case à cocher une icône pour l'état "non enfoncé, puis de la propriété "**DownPicture**" pour l'état "enfoncé" vous pouvez utiliser également la propriété "**DisabledPicture**" pour l'état désactivé.

Contrairement aux contrôles d'options boutons, la case à cocher est dotée de 3 états, l'état désactivé qui correspond à la valeur "0", l'état activé = 1 et l'état grisé = 2.

C'est la raison pour laquelle, nous ne pourrions pas utiliser des valeurs booléennes pour coder ce contrôle, nous utiliserons les valeurs cités précédemment.

Déterminer l'état de la case à cocher

Le code à utiliser est :

```

If CheckBox1.Value = 0 Then
    'Instructions
    Beep                               'Emet un son sur le haut parleur du PC lorsque le "Beep" est
    activé
Else
    'Instructions
    Beep
End If
    
```

Les états possibles des contrôles Case à cocher		Valeur	Description	Constante
0	Désactivée(par défaut)			vbUnchecked
1	Activée			vbChecked
2	État intermédiaire			vbGrayed

En règle générale, et si vous ne savez pas quelle instruction écrire ou dans le cas d'un traitement conditionnel, ou si vous ne savez pas quelle direction va prendre votre pointeur(surtout si vous utilisez "**Else**" dans votre procédure),

saisissez l'instruction "**Beep**" qui n'aura aucune influence sur votre programme mais qui vous indiquera exactement où se dirige votre pointeur, puis, si vous lancez l'application et que vous entendez le **Beep**, vous aurez la confirmation que l'instruction aura été prise en compte.

Vous n'aurez plus qu'à remplacer "**Beep**" par les instructions définitives.

Si vous voulez fixer en interne l'état d'une case à cocher comme étant "non cochée", c'est à dire en écrivant du code, vous écrivez l'instruction suivante :

```
CheckBox1.Value = 0
```

Pour quelle soit cochée, vous écrivez :

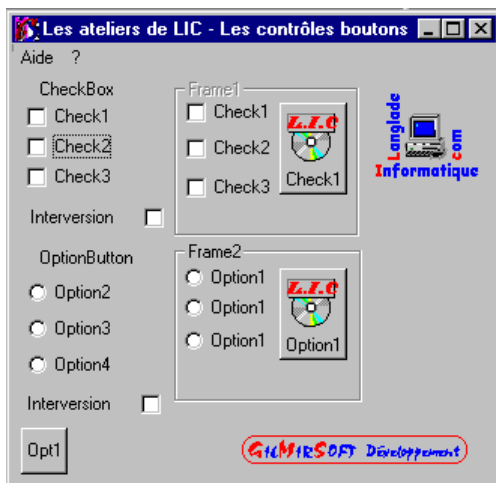
```
CheckBox1.Value = 1
```

Pour quelle apparaisse en grisé, vous écrivez :

```
CheckBox1.Value = 2
```

Le texte affiché à coté de la case à cocher est déterminé par la propriété "**Caption**".

Vous pouvez modifier l'alignement du texte à droite ou à gauche avec la propriété "**Alignment**".



Tous nos Tutoriels sont accompagnés d'exemples que vous pouvez télécharger sur notre site. L'interactivité du module vous permet de comprendre aisément le fonctionnement des exemples fournis [ici](#)

II - Le contrôle "OptionButton"

Le contrôle "**OptionButton**" accepte deux états pour l'option Vrai ou Faux(True ou False). Ces "**OptionsButtons**"



ou boutons d'options peuvent être rassemblés en groupes de contrôles dans une "**Frame**" ce type de contrôle bénéficie d'une sélection exclusive. Vous pouvez également comme le contrôle décrit précédemment bénéficier de l'option graphique avec la propriété "**Picture**" en insérant une image en position bouton sélectionné et d'une autre image en position bouton désactivée.

Déterminer l'état du bouton d'option

Les boutons d'options connaissent deux états, l'état cochée qui correspond à la valeur booléenne "True" et l'état non cochée qui correspond à la valeur booléenne "False". Dans un groupe de contrôle seul un bouton peut être coché.

Exploitation d'un événement utilisateur avec 2 boutons d'option :

```
Sub Option1_Click ( )
    If Option1.Value = True Then
        'Instructions si l'option1 est activée
    Else
        'Instructions si l'option2 est activée
    End If
End Sub
```

Exploitation d'un événement utilisateur avec 3 boutons d'option :

```
Sub Option1_Click ( )
    If Option1.Value = True Then
        'Instructions si l' option1 est activée
    ElseIf
        'Instructions si l' option2 est activée
    ElseIf
        'Instructions si l' option3 est activée
    End If
End Sub
```

Vous pouvez également comme le bouton précédent utiliser l' option "InfoBulle" avec la propriété "**ToolTip**".

On peut également affecter une valeur par l' intermédiaire du code, comme par exemple : Option1.Value = 1 ou dans l' hypothèse d' affecter une valeur dans une zone de liste de fichiers pour sélectionner une extension de fichier bien déterminée nous utiliserons le masque de recherche:

```
Sub Option1_Click ( )
    Fichier1.Pattern = "*.exe"
End Sub
```

III - La zone de liste et zone modifiable

Le contrôle zone de liste affiche une liste contenant



des entrées dont l'utilisateur peut sélectionner un ou plusieurs éléments. Si le nombre d'éléments contenus dans la liste dépasse celui qui peut être affiché, une barre de déplacement (ascenseur) s'affiche automatiquement.



Vous pouvez **télécharger le module FileList** qui explique le fonctionnement des zones de liste. Téléchargez le module et saisissez un certain nombre d'entrées pour voir apparaître la barre de déplacement.

Une zone de liste peut être créée en mode "Conception" comme décrit dans l'exemple exo2, dans ce cas le remplissage de la zone de liste est réalisé par la procédure événementielle "Load" qui est appelée lors du chargement de la feuille.

En mode "Exécution" nous utiliserons la méthode "**AddItem**".

```
Objet.AddItem "Element"
```

L'argument "Objet" désigne le nom du contrôle, c'est à dire celui de la zone de liste.

L'argument "Element" est l'entrée qui doit être ajoutée.

Ex : lstZonedeliste.AddItem "**Help System Restorer**"

L'application décrite montre l'utilisation d'une zone de liste.

Elle comprend une "**TextBox**", une zone de liste, deux "**Label**", 4 boutons poussoirs.

Nous avons utilisé dans ce module les "**Méthodes**" "**RemoveItem**" et "**Clear**". La méthode "**RemoveItem**" nous permet après avoir sélectionné la ligne qui nous intéresse de pouvoir sélectivement la supprimer.

La méthode "**Clear**" quant à elle efface complètement le contenu de la liste.

Nous avons également utilisé la propriété "**ListCount**"

La propriété "List" est liée à la propriété "ListCount", qui renvoie le nombre d'entrées de la liste. Dans l'exercice Exo2 la propriété "ListCount" renvoie le nombre d'entrée que vous saisissez et rectifie le nombre indiqué lorsque vous effacez une ou plusieurs entrées. Vous pourrez étudier tous les autres contrôles dans les cours que nous diffusons sur notre site.

Prochain tutoriel :

 ***MsgBox, Instruction ou Fonction ?***

