

# VBA - Un contrôle listbox transparent

Juin 2014

## VBA - Un contrôle listbox transparent

Lorsque l'on regarde les propriétés des listbox, on s'aperçoit que la propriété BackStyle n'existe pas. Or cette propriété, disponible par exemple pour les contrôles label, textbox etc., nous permet, en la réglant sur fmBackStyleTransparent, de rendre notre contrôle transparent. Or, j'aime bien mon image de fond et je voudrais qu'elle apparaisse dans le fond de ma liste. Nous allons donc tricher en utilisant un textbox à la place de la listbox.



- I - Prérequis :
- II - Astuce :
- III - Les codes :
  - A l'initialisation de l'UserForm :
  - Lors de l'événement MouseDown du textbox :
- IV - Supplément :
- V - Téléchargement :

## I - Prérequis :

- Dessinons tout d'abord un UserForm auquel nous appliquons (grâce à sa propriété Picture), une image de fond.
- Dessinons, dans cet userform, un contrôle textbox

## II - Astuce :

Elle consiste tout simplement à utiliser les propriétés MultiLine, BackStyle et ScrollBars des textbox pour lui donner l'apparence d'une listbox. Au niveau du « chargement » des données, il suffit de délimiter chacune de ses lignes par un caractère invisible (ici : Chr(1)) pour les rendre sélectionnable. Pour utiliser ces données, il suffira de boucler sur tous les caractères de cette sélection.

## III - Les codes :

### A l'initialisation de l'UserForm :

Option Explicit

```
Private Sub UserForm_Initialize()
```

```
Dim i As Integer, texto As String
```

```
For i = 1 To 100
```

```
'On remplit le texte amené à figurer dans le textbox comme une liste,
```

```
'avec à chaque début de ligne le caractère invisible Chr(1)
```

```
If i = 1 Then texto = Chr(1) & "Valeur de liste 1" Else texto = texto & Chr(10) & Chr(1) & "Valeur de liste " & i
```

```
Next i
```

```
With TextBox1
```

```
.BackStyle = fmBackStyleTransparent
```

```
.MultiLine = True
```

```
.ScrollBars = fmScrollBarsVertical
```

```
.Move 5, 5, Me.Width - 16, Me.Height - 40
```

```
'On ajoute, en fin de dernière ligne, encore un caractère invisible pour délimiter la dernière ligne
```

```
.Text = texto & Chr(1)
```

```
'Si vous voulez, au départ, que la ligne sélectionnée soit la 1ère, décochée les deux prochaines lignes de code :
```

```
'.SetFocus
```

```
'.CurLine = 0
```

```
End With
```

```
End Sub
```

## Lors de l'événement MouseDown du textbox :

```
Private Sub TextBox1_MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single)
```

```
Dim debSel As Long, finSel As Long, texto As String, txtSel As String, i As Integer
```

```
'Les caractères Chr(10) (sauts de lignes) utilisés lors du remplissage  
'du textbox comptent pour 1 caractère.
```

```
'Par conséquent, il ne faut pas en tenir compte dans cette procédure
```

```
'on les supprime donc dans notre variable
```

```
texto = Replace(TextBox1.Text, Chr(10), "")
```

```
'on se positionne là où l'utilisateur clique
```

```
debSel = TextBox1.SelStart
```

```
finSel = TextBox1.SelStart
```

```
'Les lignes commencent systématiquement par le caractère : "Chr(1)", on va donc le chercher :
```

```
' 1- A reculons => va nous donner la position du premier caractère de la ligne
```

```
Do While Mid(texto, debSel, 1) <> Chr(1)
```

```
debSel = debSel - 1
```

```
Loop
```

```
' 2- en avançant => va nous donner la position du premier caractère de la ligne suivante
```

```
If Mid(texto, finSel, 1) = Chr(1) Then finSel = finSel + 1
```

```
Do While Mid(texto, finSel, 1) <> Chr(1)
```

```
finSel = finSel + 1
```

```
Loop
```

```
'Boucle pour stocker dans une variable le contenu sélectionné
```

```
For i = debSel + 1 To finSel - 1
```

```
txtSel = txtSel & Mid(texto, i, 1)
```

```
Next i
```

```
'Positionnement du curseur au début de la ligne
```

```
TextBox1.SelStart = debSel
```

```
'Sélection de la ligne
```

```
TextBox1.SelLength = finSel - debSel - 1
```

```
'Envoi de la valeur sélectionnée dans une cellule de la feuille
```

```
Sheets("Feuil1").Range("A1") = Trim(txtSel)
```

```
End Sub
```

## IV - Supplément :

Pour plus de maniabilité, on peut également placer, dans l'userform, une vraie listbox que l'on rendra invisible (ListBox1.Visible = False). Ceci vous permettra plus de souplesse en profitant de toutes les propriétés des listbox. Il vous suffira d'utiliser la listbox pour le traitement des données et le textbox uniquement pour l'affichage.

## V - Téléchargement :

[Lien vers fichier exemple](#). Si toutefois celui-ci n'était plus disponible sur cjoint, merci de me le faire savoir en m'envoyant un MP [ici](#), cliquez sur « [Lui écrire un message](#) »

Ce document intitulé « [\\_VBA - Un contrôle listbox transparent](#) » issu de **CommentCaMarche** ([www.commentcamarche.net](http://www.commentcamarche.net)) est mis à disposition sous les termes de la licence [Creative Commons](#). Vous pouvez copier, modifier des copies de cette page, dans les conditions fixées par la licence, tant que cette note apparaît clairement.