

Utiliser le contrôle ImageList en VBA Excel

par SilkyRoad (silkyroad.developpez.com)

Date de publication : 18/08/2006

Dernière mise à jour : 18/08/2006

Ce document décrit l'utilisation des contrôles ImageList dans un UserForm.

Le contrôle ImageList fait partie des Common Controls Visual Basic 6.0 (mscomctl.ocx).

Toutes les procédures ont été testées à partir Excel2002.

I - Introduction

II - Description

III - Exemples

III-A - Manipuler les images

III-A-1 - charger des images dans l'ImageList

III-A-2 - Contrôler le nombre d'images stockées

III-A-3 - Supprimer une image

III-A-4 - Supprimer toutes les images

III-A-5 - Coller une image dans la feuille de calcul

III-A-6 - Exporter les images sur le disque dur

III-B - Utilisation associée à un contrôle Standard

III-C - Utilisation associée à un contrôle ListView

III-D - Utilisation associée à un contrôle TreeView

IV - Téléchargement

I - Introduction

Le contrôle ImageList permet de stocker et gérer les images (jpg ,bmp ,ico ,gif) dans le classeur. Cet objet est souvent utilisé en complément des contrôles:

Treeview, pour gérer la sélection des noeuds

ListView, pour gérer les images associées aux modes d'affichage (Icon, SmallIcon).

Remarque:

Il est toujours préférable de charger des images externes. Evitez (si possible) de sauvegarder les images dans le classeur pour ne pas en augmenter le poids.

II - Description

Tout d'abord, sélectionnez "Microsoft ImageList Control, version 6.0" (ou l'équivalent) dans la liste des contrôles supplémentaires.

Vous pouvez placer le contrôle n'importe où dans l'UserForm: L'objet ne sera pas visible lors de l'affichage du UserForm.

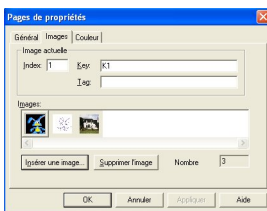
Ensuite, si vous souhaitez ajouter des images manuellement dans l'objet ImageList :

Sélectionnez l'option "Personnalisé" dans la fenêtre propriétés

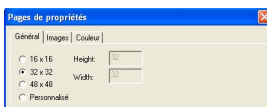
Puis l'onglet "Images"

Cliquez sur le bouton "Insérer une image"

Choisissez les icônes ou les images qui vous intéressent



Vous pouvez spécifier la dimension des images (en pixels) dans l'onglet "Général".



Vous avez le choix entre les dimensions prédéfinies (16x16, 32x32, 48x48) et Personnalisé. Faites attention à bien paramétrer les dimensions en fonction du support d'affichage, notamment pour les contrôles ListView et TreeView (Voir les exemples des chapitres III-C et III-D). Si la taille est trop élevée, les images risquent d'être disproportionnées par rapport aux autres objets.

III - Exemples

III-A - Manipuler les images

III-A-1 - charger des images dans l'ImageList

Cette procédure permet de charger 3 images:

Vba

```
Me.ImageList1.ListImages.Add , "Cle1", LoadPicture("C:\lapin4.gif")  
Me.ImageList1.ListImages.Add , "Cle2", LoadPicture("C:\fourmiz.JPG")  
Me.ImageList1.ListImages.Add , "Cle3", LoadPicture("C:\slcplappl.ico")
```

Description des arguments:

[Me.ImageList1.ListImages.Add \[Index\], \[Key\], \[Image\]](#)

L'argument [\[Index\]](#) est facultatif. Si vous ne spécifiez pas de valeur, un numéro s'incrémente automatiquement lors de chaque ajout d'image.

L'argument [\[Key\]](#) est facultatif. Cette clé est unique et fiabilise l'attribution des images, surtout quand vous devez gérer des ajouts et des suppressions. Cet argument est souvent utilisé dans les contrôles ListView et TreeView (voir les exemples des chapitres III-C et III-D).

L'argument [\[Image\]](#) spécifie le chemin du fichier à charger ou le conteneur d'image (voir les exemples du chapitre III-B).

III-A-2 - Contrôler le nombre d'images stockées

Cet exemple compte le nombre d'images contenues dans l'ImageList.

Vba

```
MsgBox ImageList1.ListImages.Count
```

III-A-3 - Supprimer une image

Cette procédure supprime une image du conteneur. "1" correspond à l'index de l'image supprimée.

Vba

```
ImageList1.ListImages.Remove 1
```

III-A-4 - Supprimer toutes les images

Cette procédure permet de supprimer l'ensemble des images stockées dans l'ImageList.

Vba

```
Me.ImageList1.ListImages.Clear
```

III-A-5 - Coller une image dans la feuille de calcul

Cet exemple colle la 1ere image dans la feuille active.

Remarque:

La procédure ne fonctionne pas si l'image stockée est de type .ico.

Vba

```
Private Declare Function OpenClipboard& _  
Lib "user32" (ByVal hwnd As Long)  
Private Declare Function EmptyClipboard Lib "user32" () As Long  
Private Declare Function SetClipboardData& _  
Lib "user32" (ByVal wFormat&, ByVal hMem&)  
Private Declare Function CloseClipboard& Lib "user32" ()  
Private Declare Function DestroyIcon& Lib "user32" (ByVal hIcon&)  
  
Private Sub CommandButton1_Click()  
    Dim iPic As StdPicture  
    Dim hCopy&  
  
    'Spécifie l'image cible  
    Set iPic = ImageList1.ListImages(1).Picture  
  
    'Ouverture du presse papier  
    OpenClipboard 0&: EmptyClipboard  
    'Place l'image dans le presse papier  
    hCopy = SetClipboardData(2, iPic.Handle)  
  
    'Fermeture du presse papier  
    CloseClipboard  
  
    'Vérifie s'il y a quelques chose à coller  
    If hCopy Then  
        ActiveSheet.Range("A1").Select  
        'Collage dans la feuille active  
        ActiveSheet.Paste  
    End If  
  
    'Ménage  
    DestroyIcon iPic.Handle  
    Set iPic = Nothing  
End Sub
```

III-A-6 - Exporter les images sur le disque dur

Cette macro boucle sur l'ensemble des images et les exporte sur le disque dur.

Toutes les images sont supposées être au format .jpg.

Vba

```
Dim Img As ListImage
For Each Img In ImageList1.ListImages
SavePicture Img.Picture, "C:\export_image_" & Format(Img.Index, "00") & ".jpg"
Next Img
```

III-B - Utilisation associée à un contrôle Standard

Il est possible de visualiser les éléments de l'ImageList dans **les contrôles standards** s'ils possèdent une propriété "Picture".

Cet exemple affiche le 2eme élément de l'ImageList dans un contrôle Image.

Vba

```
Me.Image1.Picture = ImageList1.ListImages(2).Picture
```

Cet autre exemple affiche une image dans un label nommé Label1.

Vba

```
Me.Label1.Picture = ImageList1.ListImages(1).Picture
```

Il est bien entendu possible d'effectuer l'opération inverse.

Ce dernier exemple récupère une image affichée sur un CommandButton afin de la stocker dans l'ImageList.

Vba

```
'Récupération de l'image du CommandButton
ImageList1.ListImages.Add , "CLeX", Me.CommandButton1.Picture
```

III-C - Utilisation associée à un contrôle ListView

les contrôles ListView proposent plusieurs modes d'affichage (dans le style de la fenêtre de droite dans l'explorateur Windows). En fonction du mode sélectionné vous pouvez afficher des images différentes (Icon, SmallIcon).

Placez Une ImageList, Une ListBox et Une ListView dans un UserForm afin de tester l'exemple suivant. Sélectionnez un élément dans la ListBox pour définir le mode d'affichage.

Vba

```

Private Sub UserForm_Initialize()
    Dim X As Integer

    'Supprime toutes les images de la liste
    Me.ImageList1.ListImages.Clear

    'Définit la dimension des images
    Me.ImageList1.ImageHeight = 32 'Hauteur
    Me.ImageList1.ImageWidth = 32 'Largeur

    'Charge les nouvelles images
    Me.ImageList1.ListImages.Add , "Im2", LoadPicture("C:\fourmiz.JPG")
    Me.ImageList1.ListImages.Add , "Im3", LoadPicture("C:\slcplappl.ico")

    'Associe les images à la ListView
    Set Me.ListView1.SmallIcons = Me.ImageList1
    Set Me.ListView1.Icons = Me.ImageList1

    'Ajoutez une ListBox qui permettra de spécifier le
    'type d'affichage dans la ListView
    With Me.ListBox1
        .AddItem "lvwIcon"
        .AddItem "lvwSmallIcon"
        .AddItem "lvwList"
        .AddItem "lvwReport"
    End With

    '----- remplissage ListView-----
    With ListView1
        'définit le nombre de colonnes et Entêtes (titres et tailles des colonnes)
        With .ColumnHeaders
            .Clear
            .Add , , "Nom", 80
            .Add , , "Ville", 50
            .Add , , "Age", 50
        End With
        'Remplissage lere colonne
        With .ListItems
            .Add , , "Riri"
            .Add , , "Fifi"
            .Add , , "Loulou"
        End With
        'Remplissage colonnes 2 et 3
        .ListItems(1).ListSubItems.Add , , "Ville01"
        .ListItems(1).ListSubItems.Add , , "30"
        .ListItems(2).ListSubItems.Add , , "Ville02"
        .ListItems(2).ListSubItems.Add , , "27"
        .ListItems(3).ListSubItems.Add , , "Ville03"
        .ListItems(3).ListSubItems.Add , , "41"
    End With

    '-----

    'Boucle sur chaque ligne de la ListView
    For X = 1 To .ListItems.Count
        'Associe l'image ayant la clé (Key) "Im2". Cette image sera affichée
        'lorsque la ListView sera en mode Icône
        .ListItems(X).Icon = "Im2"

        'Associe l'image ayant la clé (Key) "Im3". Cette image sera affichée
        'lorsque la ListView sera en mode petite Icône
        .ListItems(X).SmallIcon = "Im3"
    Next

    End With

    'Spécifie l'affichage en mode Icône lors du lancement du UserForm

```


Vba

```
Listview1.View = lvwIcon
End Sub

Private Sub ListBox1_Click()
'Spécifie le type d'affichage dans la ListView
Me.ListView1.View = Me.ListBox1.ListIndex
End Sub
```

III-D - Utilisation associée à un contrôle TreeView

Les contrôles TreeView possèdent des arguments facultatifs pour afficher des images dans chaque noeud et modifier cette image lorsque le noeud est sélectionné.

Syntaxe pour l'association d'images dans un TreeView:

`TreeView1.Nodes.Add [Relative], [Relationship], [Key], [Text], [Image], [SelectedImage]`

L'argument `[Image]` facultatif, spécifie l'image qui doit s'afficher par défaut dans le TreeView.

L'argument `[SelectedImage]` facultatif, spécifie l'image qui doit s'afficher lorsqu'un noeud est sélectionné dans le TreeView.

Pour cet exemple, Placez Une ImageList et un contrôle TreeView dans un UserForm.

Vba

```
Private Sub UserForm_Initialize()
Dim k As Byte, j As Byte
Dim Img1 As String, Img2 As String

'Supprime toutes les images de la liste
Me.ImageList1.ListImages.Clear

'Définit la dimension des images
Me.ImageList1.ImageHeight = 16 'Hauteur
Me.ImageList1.ImageWidth = 16 'Largeur

'Charge les images
Img1 = "C:\lapin4.gif"
Img2 = "C:\fourmiz.JPG"
Me.ImageList1.ListImages.Add , "Image1", LoadPicture(Img1)
Me.ImageList1.ListImages.Add , "Image2", LoadPicture(Img2)

'Associe les images au TreeView
Set Me.TreeView1.ImageList = Me.ImageList1

'----- création structure TreeView -----
'Noeuds principaux
For k = 1 To 5
'Ajoute un noeud et associe les images qui devront s'afficher
TreeView1.Nodes.Add , , "maClé1" & k, "NoeudParent" & k, "Image1", "Image2"
Next
```

Vba

```
'Noeuds enfants
For j = 1 To 5
    For k = TreeView1.Nodes.Count To TreeView1.Nodes.Count + 2
        'Ajoute un noeud et associe les images qui devront s'afficher
        TreeView1.Nodes.Add "maClé1" & j, tvwChild, _
            "maClé2" & k, "NoeudEnfant" & k, "Image1", "Image2"
    Next
Next
'-----
End Sub
```

IV - Téléchargement

