

Utiliser les UserForm en VBA Excel

par SilkyRoad (silkyroad.developpez.com)

Date de publication : 13/08/2006

Dernière mise à jour : 22/06/2007

Ce document est une initiation à l'objet UserForm.

Les UserForm (USF) servent à créer des boîtes de dialogue personnalisées. Vous pouvez y ajouter des contrôles afin de mettre en place une Interface utilisateur adaptée à votre projet. Il sera ainsi possible d'effectuer des saisies depuis ce support préformaté ou y visualiser des informations très diversifiées (Textes, données numériques, images, vidéos...)

Toutes les procédures de ce document ont été testées en utilisant Excel2002.

- I - Préambule
- II - Créer un UserForm
- III - Quelques méthodes et propriétés
 - III-A - Afficher l'UserForm
 - III-A-1 - Affichage Modal
 - III-A-2 - Affichage non Modal
 - III-A-3 - Afficher un UserForm dont le nom est contenu dans une variable
 - III-B - Masquer l'UserForm
 - III-C - Fermer l'UserForm
 - III-D - Définir la Position de l'UserForm à l'écran
 - III-E - Imprimer un UserForm
 - III-F - Effectuer l'aperçu avant impression d'une feuille de calcul depuis un UserForm
 - III-G - Mettre à jour l'affichage du UserForm
 - III-H - Insérer une image de fond
 - III-I - Modifier durablement les propriétés d'un UserForm
- IV - Les événements dans un UserForm
 - IV-A - UserForm_Initialize
 - IV-B - UserForm_Activate
 - IV-C - UserForm_KeyPress
 - IV-D - UserForm_KeyDown et UserForm_KeyUp
 - IV-E - UserForm_MouseMove
 - IV-F - UserForm_MouseDown et UserForm_MouseUp
 - IV-G - UserForm_Click
 - IV-H - UserForm_AddControl
 - IV-I - UserForm_RemoveControl
 - IV-J - UserForm_Layout
 - IV-K - UserForm_QueryClose
 - IV-L - UserForm_Terminate
- V - Quelques exemples particuliers
 - V-A - Personnaliser l'affichage des UserForm
 - V-B - Afficher un UserForm temporairement
 - V-C - Lister le nom des USF contenus dans le classeur
 - V-D - Récupérer le Handle d'un Userform
 - V-E - Déclencher une procédure événementielle
 - V-F - Sauvegarder les paramètres des contrôles
- VI - Conclusion
- VII - Liens
- VIII - Téléchargement

I - Préambule

Ce document propose une description générale des UserForm.

Il s'agit avant tout d'une introduction pour:

La création et la mise en forme de la boîte de dialogue dans votre projet.

Une présentation des méthodes et propriétés accessibles par macro.

Une description des événements principaux.

Quelques exemples particuliers d'utilisation.

II - Créer un UserForm

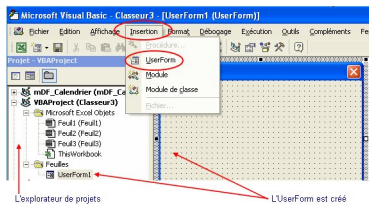
Pour insérer un UserForm dans votre projet:

Vous devez tout d'abord afficher l'éditeur de macros (Alt + F11), puis

Cliquez sur votre fichier dans l'explorateur de projets.

Ensuite,

Sélectionnez le Menu Insertion / UserForm



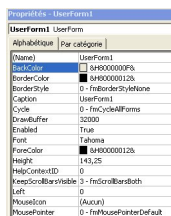
Dès que l'UserForm est créé, il est possible d'effectuer un affichage rapide en appuyant sur le bouton "Exécuter Sub/UserForm", ou sur la touche raccourci **F5**.

Vous pouvez modifier la taille de la boîte de dialogue en utilisant les poignées de redimensionnement.

Utilisez la fenêtre de propriétés pour personnaliser l'USF manuellement: Changer le nom, une fonctionnalité, l'apparence du formulaire...etc...

Si cette fenêtre n'est pas affichée par défaut:

Sélectionnez le Menu Afficher/Fenêtre propriétés ou Sélectionnez votre UserForm et appuyez sur la touche raccourci **F4**.



Le tableau suivant décrit succinctement chaque propriété.

Propriété	Description
(Name)	Renvoie le nom de l'objet
BackColor	Spécifie la couleur de fond
BorderColor	Spécifie la couleur de bordure
BorderStyle	Spécifie le style de bordure
Caption	Spécifie le texte affiché dans la barre de titre
Cycle	Spécifie l'action à effectuer lorsque l'utilisateur quitte le dernier contrôle
DrawBuffer	Spécifie le nombre de pixels de la mémoire tampon vidéo utilisée pour le rafraichissement de l'image
Enabled	Spécifie si l'objet peut recevoir le focus et répondre aux événements générés par l'utilisateur
DrawBuffer	Spécifie le nombre de pixels de la mémoire tampon vidéo utilisée pour le rafraichissement de l'image
ForeColor	Définit la couleur de la police
Height	Définit la dimension verticale
HelpContextID	Spécifie la rubrique d'un fichier d'aide
KeepScrollBarsVisible	Spécifie si les barres de défilement sont visibles
Left	Définit la position par rapport au bord gauche de l'application
Mouselcon	Affecte un icône personnalisé
MousePointer	Spécifie le type de pointeur
Picture	Spécifie l'image de fond dans l>UserForm
PictureAlignment	Spécifie l'alignement de l'image de fond
PictureSizeMode	Spécifie comment afficher une image de fond
PictureTiling	Permet d'afficher l'image en mosaïque
RightToLeft	Indique le sens d'affichage du texte et contrôle l'apparence visuelle sur un système bidirectionnel.
ScrollBars	Indique si les barres de défilement verticales et horizontales doivent être affichées
ScrollHeight	Spécifie la hauteur de la zone totale pouvant être affichée en déplaçant la barre de défilement
ScrollLeft	Spécifie la distance, du bord gauche de la partie visible, à partir du bord gauche du UserForm
ScrollTop	Spécifie la distance, du bord supérieur de la partie visible, à partir du bord supérieur du UserForm
ScrollWidth	Spécifie la largeur de la zone totale pouvant être affichée en déplaçant la barre de défilement
ShowModal	Indique si la boîte de dialogue est modale
SpecialEffect	Spécifie l'aspect du UserForm à l'écran
StartUpPosition	

	Indique la position du UserForm lors de sa première apparition
Tag	Permet de stocker des informations supplémentaires
Top	Définit la position par rapport au bord supérieur de l'application
WhatsThisButton	Indique si le bouton d'aide "Qu'est-ce que c'est" apparaît sur la barre de titre
WhatsThisHelp	Indique si l'aide contextuelle utilise la fenêtre automatique fournie par l'aide de Windows ou la fenêtre d'aide principale
Width	Définit la dimension horizontale
Zoom	Spécifie le changement de taille de l'objet
Visible	Spécifie si l'objet est masqué ou affiché.

Vous pouvez ensuite insérer des contrôles dans votre projet:

 ***[L'article pour ajouter et utiliser les contrôles dans un UserForm](#)***

III - Quelques méthodes et propriétés

III-A - Afficher l'UserForm

III-A-1 - Affichage Modal

Pour afficher la boîte de dialogue par macro, utilisez:

```
Vba  
  
Private Sub CommandButton1_Click()  
    UserForm1.Show  
End Sub
```

Lorsque l'UserForm est affiché, vous ne pouvez plus accéder aux feuilles de calcul. Dans ce cas l'UserForm est dit modal.

`UserForm1` est le nom défini dans la propriété **(Name)**.

III-A-2 - Affichage non Modal

Pour que les feuilles de calcul restent accessibles, vous devez passer la boîte de dialogue en mode non modal.

La propriété **ShowModal** doit être égale à `False`.

Les 3 exemples suivants sont équivalents et utilisables à partir d'Excel2000.

```
Vba  
  
UserForm1.Show 0
```

```
Vba  
  
UserForm1.Show False
```

```
Vba  
  
UserForm1.Show vbModeless
```

Nota:

Vous devrez utiliser des APIs pour afficher les UserForms non modaux dans Excel97.

III-A-3 - Afficher un UserForm dont le nom est contenu dans une variable

UserForms est l'ensemble des objets UserForm placés dans le classeur.

la méthode **Add** permet de placer un nouvel élément UserForm dans la collection.

Vba

```
Sub lancementUSF()  
    Dim sVariable As String  
  
    sVariable = "UserForm1"  
    VBA.UserForms.Add(sVariable).Show  
End Sub
```

III-B - Masquer l'UserForm

Pour masquer un UserForm sans le décharger, utilisez:

Vba

```
Private Sub CommandButton1_Click()  
    UserForm1.Hide  
End Sub
```

Lorsqu'un UserForm est masqué, l'utilisateur n'a plus aucune interaction avec celui-ci mais la boîte de dialogue et les objets qu'elle contient restent accessibles par programmation.

III-C - Fermer l'UserForm

Vba

```
'Ferme l'UserForm nommé UserForm1  
Unload UserForm1
```

Lorsqu'un objet est déchargé, il est supprimé de la mémoire et toute la mémoire qui lui était associée est libérée.

Si la procédure de fermeture est placée dans l'UserForm, il est aussi possible d'écrire:

Vba

```
Unload Me
```

Pour fermer en une seule fois tous les UserForm en cours d'affichage, utilisez:

Vba

```
Private Sub CommandButton1_Click()  
    End  
End Sub
```

III-D - Définir la Position de l'UserForm à l'écran

La propriété **StartPosition** définit la position des USF lors de l'affichage.

Ce tableau décrit les constantes disponibles:

Valeur	Paramètre	Description
Manual	0	Aucune valeur initiale n'est indiquée.
CenterOwner	1	Centre l'objet UserForm au sein de l'élément auquel il appartient.
CenterScreen	2	Centre l'objet par rapport à l'écran.
WindowsDefault	3	Place l'objet dans l'angle supérieur gauche de l'écran.

Cet exemple affiche un USF en haut et dans le coin supérieur gauche de l'écran.

Vba

```
Private Sub UserForm_Initialize()  
    Me.StartPosition = 3  
End Sub
```

Vous pouvez aussi utiliser les propriétés **Left** et **Top** pour personnaliser la position de la boîte de dialogue.

Left: Définit la distance du bord gauche du UserForm par rapport au bord gauche de l'application.

Top: Définit la distance du bord supérieur du UserForm par rapport au bord supérieur de l'application.

Quelques exemples particuliers:

(L'application Excel est supposée préalablement affichée en mode "Aggrandi".)

Afficher un USF en haut et dans le coin droit de l'écran.

```
Vba

Private Sub userForm_Initialize()
    With Me
        .startUpPosition = 3
        .Left = Application.Width - Me.Width
    End With
End Sub
```

Positionner l'USF en bas et à droite.

```
Vba

Private Sub UserForm_Activate()
    With Me 'affiche l'USF en bas à droite
        .Top = Application.Height - Me.Height
        .Left = Application.Width - Me.Width
    End With
End Sub
```

Afficher un USF en plein écran.

```
Vba

Private Sub UserForm_Activate()
    With Me
        .StartUpPosition = 3
        .Width = Application.Width
        .Height = Application.Height
        .Left = 0
        .Top = 0
    End With
End Sub
```

III-E - Imprimer un UserForm

Remarque préalable:

Ce chapitre présente quelques solutions d'impression, mais un UserForm n'est pas bien adapté pour les éditions (consommation d'encre importante, long temps d'impression, rendu sur le papier ne correspondant pas toujours à ce qui est vu à l'écran...)

Plus généralement, il faut toujours avoir une réflexion sur l'utilisation du papier pour ne pas le gaspiller.

Vba

```
Private Sub CommandButton1_Click()  
    Me.PrintForm  
End Sub
```

L'image du UserForm est automatiquement placée dans le coin supérieur gauche de la page imprimée.

Il n'existe pas d'option pour centrer ou repositionner l'impression. La seule solution palliative consiste à faire une copie d'écran de la boîte de dialogue. Ensuite vous faites un collage dans la feuille de calcul en tant qu'image, pour la manipuler à votre guise.

Cet exemple imprime l'UserForm centré dans la page:

Vba

```
Option Explicit  
  
Private Declare Sub keybd_event Lib "user32" ( _  
    ByVal bVk As Byte, ByVal bScan As Byte, ByVal dwFlags As Long, _  
    ByVal dwExtraInfo As Long)  
  
Private Sub CommandButton1_Click()  
    Dim Ws As Worksheet  
  
    'Copie d'écran de la forme active  
    keybd_event vbKeySnapshot, 1, 0&, 0&  
    DoEvents  
  
    'Ajoute une feuille pour coller l'image de la forme  
    Set Ws = Sheets.Add  
    Ws.Paste  
  
    'impression centrée dans la page  
    With Ws  
        .PageSetup.CenterHorizontally = True  
        .PageSetup.CenterVertically = True  
        .PrintOut  
    End With  
End Sub
```

III-F - Effectuer l'aperçu avant impression d'une feuille de calcul depuis un UserForm

Lorsque vous souhaitez effectuer un aperçu avant impression d'une feuille de calcul, depuis un UserForm, utilisez:

Vba

```
Private Sub CommandButton1_Click()  
    Me.Hide  
    'Aperçu avant impression de la Feuille  
    Feuill.PrintPreview  
    Me.Show
```

```
Vba  
End Sub
```

III-G - Mettre à jour l'affichage du UserForm

La méthode Repaint est utile si le contenu ou l'aspect d'un objet change de façon significative, et si vous ne voulez pas attendre que le système redessine automatiquement la zone.

```
Vba  
UserForm1.Repaint
```

III-H - Insérer une image de fond

La fonction **LoadPicture** permet de spécifier un objet Image dans la propriété **Picture** du UserForm.

Cette procédure charge une image de fond dans l'UserForm lors de l'évènement Click sur un CommandButton.

L'image sera automatiquement déchargée lors de la fermeture de la boîte de dialogue.

```
Vba  
  
Private Sub CommandButton1_Click()  
    UserForm1.Picture = LoadPicture("C:\monFichier.jpg")  
End Sub
```

Remarque: Certains formats d'image ne sont pas reconnus: PNG, TIF...

III-I - Modifier durablement les propriétés d'un UserForm

Vous avez constaté dans les exemples précédents qu'il est possible de modifier les propriétés de la boîte de dialogue par macro. Lorsque vous refermez l'UserForm, les propriétés sont réinitialisées.

Il est aussi possible de changer de façon "définitive" et dynamique les propriétés, à condition que la boîte de dialogue ne soit pas en cours d'affichage.

L'exemple suivant montre comment modifier la couleur de fond.

Vba

```
'Change la couleur de fond du UserForm  
ThisWorkbook.VBProject.VBComponents("UserForm1"). _  
    Properties("BackColor") = RGB(125, 125, 125)
```

IV - Les évènements dans un UserForm

Ce chapitre décrit les évènements principaux, sans se vouloir totalement exhaustif dans le domaine.

Vous pouvez accéder aux procédures événementielles en double cliquant sur l'UserForm depuis l'éditeur de macros.

Le menu déroulant de droite vous donne ensuite accès à la liste des évènements disponibles.



IV-A - UserForm_Initialize

Cet évènement est important car il permet de définir les propriétés des objets et les valeurs par défaut des variables, lors du lancement du Userform.

Exemple :

Vba

```
Private Sub UserForm_Initialize()  
    maVariable = "x"  
    CheckBox1 = False  
    CheckBox2 = True  
    TextBox1 = "mon texte"  
    TextBox2 = Range("A1")  
End Sub
```

IV-B - UserForm_Activate

Cet évènement est déclenché lorsque l'UserForm est activé, dès que la boîte de dialogue prend le focus.

IV-C - UserForm_KeyPress

Cet évènement est déclenché lorsque vous appuyez sur une touche du clavier alors que l'UserForm a le focus.

L'argument **KeyAscii** renvoie un nombre représentant la touche du clavier dans le jeu de caractères alpha numérique.

Vba

```
Private Sub UserForm_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    MsgBox Chr(KeyAscii)
End Sub
```

IV-D - UserForm_KeyDown et UserForm_KeyUp

UserForm_KeyDown est déclenché lorsqu'une touche du clavier est enfoncée.

UserForm_KeyUp est déclenché lorsqu'une touche du clavier est relâchée.

```
Private Sub UserForm_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)
```

La description des arguments:

KeyCode: Renvoie un nombre représentant la touche du clavier dans le jeu de caractères alpha numérique.

Shift: Indique le type de touche manipulé lors de l'évènement.

0 = Touche alpha numérique

1 = Touche Shift

2 = Touche Ctrl

4 = Touche Alt

Vous pouvez aussi obtenir des associations de touches:

6 = Touches Ctrl & Alt

...etc...

Vba

```
Private Sub UserForm_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)
    If Shift = 0 Then
        Range("A1") = Chr(KeyCode)
    Else
```

Vba

```
Select Case Shift
Case 1: Range("A1") = "Shift"
Case 2: Range("A1") = "Ctrl"
Case 3: Range("A1") = "Touches Ctrl & Shift"
Case 4: Range("A1") = "Alt"
'
'
'
End Select
End If
End Sub
```

IV-E - UserForm_MouseMove

Cet évènement permet d'intercepter les mouvements de la souris dans l'UserForm.

La description des arguments:

Button: Indique quel bouton est activé lors de l'évènement MouseMove:

xlNoButton

xlPrimaryButton

xlSecondaryButton

xlMiddleButton

Shift: Indique quelle touche du clavier est enfoncée lors de l'évènement:

0 = Pas de touche

1 = Touche Shift

2 = Touche Ctrl

4 = Touche Alt

Vous pouvez aussi obtenir des associations de touches:

6 = Touches Ctrl & Alt

...etc...

X: Renvoie la position horizontale dans l'USF.

Y: Renvoie la position verticale dans l'USF.

Voici un exemple, juste pour le fun ...

Placez cette procédure dans un UserForm nommé UserForm1.

```
Vba

Option Explicit

Private Sub UserForm_Activate()
    Feuille.Activate
    Feuille.Range("A1").Select
End Sub

Private Sub UserForm_MouseMove(ByVal Button As Integer, _
    ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single)

    Dim i As Byte, j As Byte, k As Byte
    Dim Tableau(10, 2) As Double

    On Error Resume Next

    For i = 1 To 10 'initialisation
        Feuille.Shapes.AddShape msoShapeHeart, X, Y, 10, 10
        With Feuille.Shapes(i + 1)
            .Fill.ForeColor.RGB = RGB(255, Int(25.5 * i), Int(25.5 * i))
            .Line.Visible = msoFalse
        End With
        'DoEvents

        Tableau(j, 0) = X
        Tableau(j, 1) = Y
    Next i

    '*****
    Tableau(2, 0) = Tableau(1, 0)
    Tableau(2, 1) = Tableau(1, 1)

    For j = 2 To 10
        Tableau(j, 0) = Feuille.Shapes(j + 1).Left
        Tableau(j, 1) = Feuille.Shapes(j + 1).Top
    Next j

    With Feuille.Shapes(2)
        .Left = X
        .Top = Y
    End With

    Tableau(1, 0) = Feuille.Shapes(2).Left
    Tableau(1, 1) = Feuille.Shapes(2).Top
    '*****

    For j = 2 To 10
        With Feuille.Shapes(j + 1)
            .Left = Tableau(j, 0)
```

Vba

```
.Top = Tableau(j, 1)
End With
'DoEvents
Next j

For k = 2 To 16
    Feuille.Shapes(k).Delete
Next k
End Sub

Private Sub UserForm_Terminate()
    Dim k As Integer

    For k = Feuille.Shapes.Count To 1 Step -1
        Feuille.Shapes(k).Delete
    Next k
End Sub
```

Ensuite, placez cette macro dans un module standard.

Vba

```
Sub afficher_USF()
    UserForm1.Show
End Sub
```

Lancez la procédure "afficher_USF" **depuis la feuille de calcul**, puis déplacez le curseur de la souris sur le bord gauche de la boîte de dialogue...;o)

IV-F - UserForm_MouseDown et UserForm_MouseUp

UserForm_MouseDown est déclenché lorsqu'un des boutons de la souris est enfoncé.

UserForm_MouseUp est déclenché lorsqu'un des boutons de la souris est relâché.

Les arguments des deux événements sont similaires à **UserForm_MouseMove**.

Cet exemple identifie le moment où vous relâchez le bouton, suite à un clic droit dans un UserForm:

Vba

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

    If Button And 2 Then MsgBox "Bouton droit relâché"
End Sub
```

IV-G - UserForm_Click

L'évènement **Click** est déclenché lorsque vous cliquez sur l'UserForm.

IV-H - UserForm_AddControl

L'évènement **AddControl** survient lorsque vous ajoutez un contrôle dynamiquement dans l'UserForm.

Il est déclenché immédiatement après la ligne [Me.Controls.Add](#).

```
Private Sub UserForm_AddControl(ByVal Control As MSForms.Control)
```

L'argument **Control** correspond au nouvel objet créé.

Vba

```
'--- Cette procédure permet d'ajouter un TextBox dans l'USF. ---  
Private Sub CommandButton1_Click()  
    Dim Obj As Control  
  
    Set Obj = Me.Controls.Add("forms.TextBox.1")  
    With Obj  
        .Name = "monTextBox"  
        .Left = 140  
        .Top = 30  
        .Width = 50  
        .Height = 20  
    End With  
  
    MsgBox Obj.Name & " ajouté dans l'UserForm."  
End Sub  
'---  
  
'--- Identifie la création du contrôle ---  
Private Sub UserForm_AddControl(ByVal Control As MSForms.Control)  
    MsgBox TypeName(Control) & ": " & Control.Name  
End Sub
```

Pour plus d'informations au sujet des contrôles créés dynamiquement, consultez l'article [Manipuler l'éditeur de macros ... par macro](#).

IV-I - UserForm_RemoveControl

L'évènement **RemoveControl** identifie la suppression des contrôles dans l'UserForm.

Il est déclenché immédiatement après la ligne `Me.Controls.Remove`.

```
Private Sub UserForm_RemoveControl(ByVal Control As MSForms.Control)
```

L'argument `Control` correspond à l'objet supprimé.

Cette procédure permet de supprimer un TextBox dans l'USF.

La boîte de dialogue doit contenir un TextBox créé dynamiquement et nommé "monTextBox" (Voir le chapitre précédent).

Vba

```
'Suppression du contrôle.
Private Sub CommandButton2_Click()
    Me.Controls.Remove "monTextBox"
End Sub

'--- Identifie la suppression du contrôle ---
Private Sub UserForm_RemoveControl(ByVal Control As MSForms.Control)
    MsgBox Control.Name & " supprimé."
End Sub
```

IV-J - UserForm_Layout

L'évènement **Layout** est déclenché lorsque vous changez la position du UserForm.

Cet exemple définit la position de la boîte de dialogue et empêche de le déplacer à l'écran.

Vba

```
Private Sub UserForm_Layout()
    Application.ScreenUpdating = False
    'Définit la position horizontale de l'USF
    Me.Left = 200
    'Définit la position verticale de l'USF
    Me.Top = 50
    Application.ScreenUpdating = True
End Sub
```


IV-K - UserForm_QueryClose

Cet évènement est déclenché lors de la fermeture du UserForm. A ce stade il est encore possible d'annuler l'opération de fermeture grâce à l'argument `Cancel = True`. L'argument `CloseMode` permet d'identifier l'action qui est à l'origine de la fermeture.

Ce tableau décrit les constantes disponibles:

Valeur	Paramètre	Description
vbFormControlMenu	0	L'ordre de fermeture provient de la Croix du UserForm
vbFormCode	1	L'ordre de fermeture provient d'une macro
vbAppWindows	2	L'ordre provient de la fermeture de la session Windows
vbAppTaskManager	3	L'ordre de fermeture provient d'une instruction du Task Manager

Exemple: Désactiver la croix de fermeture du UserForm.

 **IMPORTANT:** Pensez à ajouter un bouton de sortie (contenant une procédure "Unload Me") pour ne pas bloquer l'application.

Vba

```
Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
    If CloseMode = 0 Then Cancel = True
End Sub
```

IV-L - UserForm_Terminate

Terminate est le dernier évènement, déclenché lors de la fermeture d'un UserForm. A ce stade, l'annulation de la fermeture est impossible contrairement à l'évènement QueryClose.

V - Quelques exemples particuliers

V-A - Personnaliser l'affichage des UserForm

Cet exemple permet d'afficher l'USF en agrandissant progressivement la fenêtre du centre vers les bords extérieurs.

Créez un Userform nommé UserForm1 et placez y un CommandButton nommé "CommandButton1".

Collez ces procédures dans la boîte de dialogue:

```
Vba

Option Explicit

Private Declare Function FindWindow Lib "user32" Alias "FindWindowA" _
    (ByVal lpClassName As String, ByVal lpWindowName As String) As Long

Private Declare Function AnimateWindow Lib "user32" (ByVal hWnd As Long, _
    ByVal dwTime As Long, ByVal dwFlags As Long) As Long

Dim Valeur As Long
Dim strFormClassName As String

Private Sub UserForm_Initialize()

    'saisir par défaut les mêmes valeurs dans les propriétés de l'USF
    Me.Left = 200
    Me.Top = 200
    Me.Caption = "Usf animé"

    'récupère le Handle de l'USF : Daniel Klann, mpep
    If Val(Application.Version) < 9 Then 'Excel 2000
        strFormClassName = "ThunderXFrame"
    Else
        strFormClassName = "ThunderDFrame" 'Excel 2000/2002
    End If

    Valeur = FindWindow(strFormClassName, "Usf animé")

    AnimateWindow Valeur, 2000, &H10
    '2000 : le temps en millisecondes pour l'effet

    '-----
    '&H1 Anime la fenêtre de gauche à droite.
    '&H2 Anime la fenêtre de droite à gauche.
    '&H4 Anime la fenêtre du haut vers le bas.
    '&H5 Anime la fenêtre de en haut à gauche vers en bas à droite.
    '&H6 Anime la fenêtre en haut à droite vers en bas à gauche.
    '&H8 Anime la fenêtre du bas vers le haut.
    '&H9 Anime la fenêtre de en bas à gauche vers en haut à droite.
    '&h10 Agrandissement de la fenêtre du centre vers les bords.
    '&H10000 Cache la fenêtre.
    '&H20000 Active la fenêtre.
    '&H40000 Fait "rouler" la fenêtre.( à combiner avec autre constante )
    '&H80000 Effet de "fondu" sur la fenêtre (si elle est au premier niveau).
    '-----

End Sub
```

Vba

```
Private Sub UserForm_Activate()  
    Me.Repaint 'pour afficher les objets de l'USF  
End Sub  
  
Private Sub CommandButton1_Click()  
    AnimateWindow Valeur, 2000, &H8 Or &H40000 Or &H10000  
    Unload Me  
End Sub
```

La procédure doit impérativement être lancée depuis la feuille de calcul et en mode non modal. Un message d'erreur survient si l'USF est lancé depuis l'éditeur de macros.

Placez cette dernière procédure dans un module standard puis lancez la macro:

Vba

```
Sub LancerUSF()  
    'Le mode non modal est impératif  
    'sinon vous obtiendrez un message d'erreur  
    UserForm1.Show 0  
End Sub
```

V-B - Afficher un UserForm temporairement

Cette procédure referme la boîte de dialogue 5 secondes après son activation.

Vba

```
Private Sub UserForm_Activate()  
    Application.Wait Now + TimeValue("00:00:05") '5 secondes  
    Unload UserForm1  
End Sub
```

V-C - Lister le nom des USF contenus dans le classeur

Cette procédure boucle sur les UserForm du classeur contenant cette macro, et affiche leur nom.

Vba

```
Sub listeUserFormClasseur()  
    'nécessite d'activer la référence  
    'Microsoft Visual Basic For Applications Extensibility 5.3  
    Dim VBCmp As VBComponent  
  
    For Each VBCmp In ThisWorkbook.VBProject.VBComponents  
        If VBCmp.Type = 3 Then MsgBox VBCmp.Name  
    Next VBCmp
```

Vba

`End Sub`

V-D - Récupérer le Handle d'un Userform

Le Handle est une valeur entière unique définie par l'environnement d'exploitation et utilisée par un programme pour identifier un objet et y accéder.

Cette solution à été donnée par **Bbil** sur le site Developpez.com.

Vba

```
Private Declare Function FindWindowA Lib "user32" (ByVal lpClassName As String, _
    ByVal lpWindowName As String) As Long

Private Sub UserForm_Initialize()
    Dim MeHwnd As Long

    MeHwnd = FindWindowA(vbNullString, Me.Caption)
    MsgBox "le Handle de l'Userform " & Me.Caption & " est : 0x" & Hex(MeHwnd)
End Sub
```

V-E - Déclencher une procédure événementielle

Cette macro déclenche la procédure événementielle "CommandButton1_Click" contenue dans une boîte de dialogue nommée UserForm1.

Le préfixe "**Private**" de la procédure "CommandButton1_Click" doit être préalablement ôté dans l'UserForm.

Vba

```
Sub EssaiDeclenchementProcedureUSF()
    CallByName UserForm1, "CommandButton1_Click", VbMethod
End Sub
```

V-F - Sauvegarder les paramètres des contrôles

Lorsque vous fermez un UserForm, il est parfois utile de sauvegarder les paramètres d'un contrôle (par exemple le dernier contenu d'un TextBox) afin de réutiliser ces informations ultérieurement, lors de la prochaine ouverture de la boîte de dialogue.

Le moyen le plus simple consiste à enregistrer ces données dans les cellules d'une feuille masquée.

Il est aussi possible d'utiliser l'instruction **SaveSetting** pour sauvegarder les paramètres des contrôles (contenu, position ...) en créant une entrée dans la base de registres et ensuite d'utiliser **GetSetting** pour lire la valeur de la clé dans la base de registres.

Dans cet exemple, la procédure sauvegarde automatiquement le contenu des TextBox1 et 2 lors de la fermeture du UserForm.

Fermez votre classeur puis ré-ouvrez le. Les dernières données des TextBox apparaissent.

Vba

```
'Sauvegarde le contenu des TextBox lors de la fermeture du UserForm
Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
    SaveSetting "Mes parametres", "TextBox1", "Valeur TextBox1", TextBox1.Value
    SaveSetting "Mes parametres", "Textbox2", "Valeur TextBox2", TextBox2.Value
End Sub
```

Vba

```
'Récupère la valeur de la clé lors de l'ouverture du UserForm
Private Sub UserForm_Initialize()
    TextBox1.Value = GetSetting("Mes parametres", "TextBox1", "Valeur TextBox1")
    TextBox2.Value = GetSetting("Mes parametres", "TextBox2", "Valeur TextBox2")
End Sub
```

Pour supprimer une entrée, utilisez:

Vba

```
DeleteSetting "Mes parametres", "TextBox2"
```

VI - Conclusion

Les exemples présentés dans ce document ne sont qu'une partie infime des options offertes par l'objet UserForm. Les possibilités d'utilisation sont très diversifiées.

Un Userform est très simple à manipuler et facilement modulable, ce qui en fait un outil indispensable pour de nombreux projets VBA.

VII - Liens

FAQ [La FAQ](#)

Source [La page Sources](#)

VIII - Téléchargement

