

# INTRODUCTION

Cet article décrit comment utiliser VBA pour apporter des modifications dans les formulaires utilisateur dans Excel.

## Plus d'informations

Microsoft fournit des exemples de programmation à titre d'illustration uniquement, sans garantie expresse ou implicite. Ceci inclut, mais n'est pas limité à, les garanties implicites de qualité marchande ou d'adéquation à un usage particulier. Cet article suppose que vous êtes familiarisé avec le langage de programmation présenté et les outils utilisés pour créer et déboguer des procédures. Les ingénieurs du support technique Microsoft peuvent vous expliquer les fonctionnalités d'une procédure particulière, mais ils ne modifieront pas ces exemples pour fournir des fonctionnalités supplémentaires ou créer des procédures répondant à vos besoins spécifiques.

## Notions de base sur UserForm

### Comment faire pour afficher un UserForm

La syntaxe qui est utilisée pour afficher un UserForm par programme est le suivant :

**UserFormName.Show**

Pour afficher un UserForm nommé UserForm1, utilisez le code suivant :

```
UserForm1.Show
```

Vous pouvez charger un objet UserForm dans la mémoire sans l'afficher réellement. Il peut s'écouler plusieurs secondes avant qu'un UserForm complexe ne s'affiche. Dans la mesure où vous pouvez précharger un objet UserForm dans la mémoire, vous pouvez décider du moment à subir ce traitement. Pour charger la feuille UserForm1 en mémoire sans l'afficher, utilisez le code suivant :

```
Load UserForm1
```

Pour afficher l'objet UserForm, vous devez utiliser la méthode Show qui a été précédemment affichée.

### Comment faire pour masquer temporairement un UserForm

Si vous souhaitez masquer temporairement un objet UserForm, utilisez la méthode Hide . Voulez-vous masquer un objet UserForm, si votre application

implique le déplacement entre les UserForms. Pour masquer un objet UserForm, utilisez le code suivant :

```
UserForm1.Hide
```

## **Comment faire pour supprimer un objet UserForm à partir de la mémoire**

Pour supprimer un objet UserForm à partir de la mémoire, utilisez l'instruction Unload . Pour décharger un UserForm nommé UserForm1, utilisez le code suivant :

```
Unload UserForm1
```

Si vous déchargez un objet UserForm dans une procédure événementielle qui est associé à un objet UserForm ou qui est associé à un contrôle sur un formulaire utilisateur (par exemple, vous cliquez sur un contrôle bouton de commande ), vous pouvez utiliser « Me » mot clé au lieu du nom de l'objet UserForm. Pour utiliser le « Me » mot clé pour décharger un objet UserForm, utilisez le code suivant :

```
Unload Me
```

## **Comment utiliser des événements de l'objet UserForm**

UserForms prend en charge plusieurs événements prédéfinis que vous pouvez joindre des procédures VBA. Lorsque l'événement se produit, la procédure que vous avez associé à l'événement s'exécute. Une seule action qui est effectuée par un utilisateur peut lancer plusieurs événements. Parmi les plus fréquemment utilisés d'événements pour un objet UserForm sont l'événement Initialize , l'événement Click et l'événement Terminate .

Remarque Un module de Visual Basic qui contient une procédure d'événement peut être appelé un module « derrière » le UserForm. Un module qui contient les procédures d'événement n'est pas visible dans la collection de Modules de la fenêtre de l'Explorateur de projet de Microsoft de l'éditeur Visual Basic. Vous devez double-cliquer sur le corps d'un objet UserForm pour afficher le module de Code de l'objet UserForm.

## **Comment intercepter des événements de l'objet UserForm**

Pour intercepter des événements de l'objet UserForm, procédez comme suit :

1. Créer un nouveau classeur dans Excel.
2. Dans le menu **Outils** , pointez sur **Macro** , puis cliquez sur **Visual Basic Editor**.

3. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
4. Double-cliquez sur l' **objet UserForm** pour afficher la fenêtre Code pour l'objet UserForm.
5. Dans le module, tapez le code suivant :
6. Private Sub UserForm\_Click()
- 7.
8.     Me.Height = Int(Rnd \* 500)
9.     Me.Width = Int(Rnd \* 750)
- 10.
11. End Sub
- 12.
13. Private Sub UserForm\_Initialize()
- 14.
15.     Me.Caption = "Events Events Events!"
16.     Me.BackColor = RGB(10, 25, 100)
- 17.
18. End Sub
- 19.
20. Private Sub UserForm\_Resize()
- 21.
22.     msg = "Width: " & Me.Width & Chr(10) & "Height: " & Me.Height
23.     MsgBox prompt:=msg, Title:="Resize Event"
- 24.
25. End Sub
- 26.
- 27.
28. Private Sub UserForm\_QueryClose(Cancel As Integer, CloseMode As Integer)
- 29.
30.     msg = "Now Unloading " & Me.Caption
31.     MsgBox prompt:=msg, Title:="QueryClose Event"
- 32.
33. End Sub
- 34.
35. Private Sub UserForm\_Terminate()
- 36.
37.     msg = "Now Unloading " & Me.Caption
38.     MsgBox prompt:=msg, Title:="Terminate Event"
- 39.
- End Sub
40. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

Lors du premier chargement de l'objet UserForm, la macro utilise l'événement Initialize pour modifier la propriété Caption de l'objet UserForm à

« Événements événements événements ! » et la propriété BackColor bleu foncé.

Lorsque vous cliquez sur l'objet UserForm, vous déclenchez l'événement Click . L'événement Click redimensionne l'objet UserForm. Étant donné que vous avez créé une procédure pour l'événement Resize , vous recevez deux messages une fois que vous cliquez sur l'objet UserForm. L'événement Resize apparaît deux fois, car le code derrière l'événement Click modifie la propriété de largeur et de la propriété Height de l'objet UserForm.

Fermeture de l'objet UserForm déclenche l'événement QueryClose . L'événement QueryClose affiche une boîte de message qui contient la légende que vous avez donné à l'objet UserForm dans le code pour l'événement Initialize . Vous pouvez utiliser l'événement QueryClose lorsque vous souhaitez effectuer un ensemble d'actions si l'utilisateur ferme l'objet UserForm.

L'événement Terminate génère ensuite une boîte de message indiquant que la légende de l'objet UserForm est UserForm1. L'événement Terminate se produit après que l'objet UserForm est supprimé de la mémoire et la légende de l'objet UserForm retourne à son état d'origine.

### **Comment faire pour empêcher la fermeture en utilisant le bouton de fermeture d'un objet UserForm**

Lorsque vous exécutez un objet UserForm, un bouton de **fermeture** est ajouté dans le coin supérieur droit de la fenêtre UserForm. Si vous souhaitez empêcher la fermeture à l'aide du bouton **Fermer** l'objet UserForm, vous devez intercepter l'événement QueryClose .

L'événement QueryClose se produit juste avant que l'objet UserForm est déchargé de la mémoire. Pour déterminer la façon dont l'objet UserForm est fermé, utilisez l'argument *CloseMode* de l'événement QueryClose . La valeur de vbFormControlMenu pour l'argument *CloseMode* indique qu'un utilisateur a cliqué sur le bouton **Fermer** . Pour conserver l'objet UserForm actif, définir l'argument *Cancel* de l'événement QueryClose à True. Pour utiliser l'événement QueryClose pour empêcher la fermeture à l'aide d'un objet UserForm le

**Fermer** bouton, procédez comme suit :

1. Créer un nouveau classeur dans Excel.
2. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.

4. Ajoutez un contrôle CommandButton sur l'objet UserForm.
5. Double-cliquez sur l' **objet UserForm** pour afficher la fenêtre Code pour l'objet UserForm.
6. Dans la fenêtre Code, tapez le code suivant :
7. Private Sub CommandButton1\_Click()  
8.  
9.     Unload Me  
10.  
11. End Sub  
12.  
13. Private Sub UserForm\_QueryClose(Cancel As Integer,  
    CloseMode As Integer)  
14.  
15.     IF CloseMode = vbFormControlMenu Then  
16.         Cancel = True  
17.         Me.Caption = "Click the CommandButton to close  
    Me!"  
18.     End If  
19.  
    End Sub
20. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

L'objet UserForm n'est pas fermé lorsque vous cliquez sur le bouton **Fermer** . Vous devez cliquer sur le contrôle **CommandButton** pour fermer l'objet UserForm.

## Code VBA

Excel comprend quinze différents contrôles que vous pouvez utiliser la commande. Cette section contient plusieurs exemples qui utilisent ces contrôles par programme.

Remarque Le code VBA qui est inclus dans cet article ne contient pas d'exemples qui affectent toutes les propriétés et événements pour les contrôles. Si c'est nécessaire, vous pouvez utiliser la fenêtre Propriétés pour afficher la liste des propriétés disponibles pour un contrôle. Pour afficher la liste des propriétés, dans le menu **affichage** , cliquez sur **Fenêtre Propriétés**.

## Comment faire pour utiliser le mode Création pour modifier des contrôles

Lorsque vous utilisez l'éditeur de Visual Basic pour créer une boîte de dialogue, vous utilisez le mode Création. En mode Création, vous pouvez modifier des contrôles, et vous pouvez modifier les propriétés d'un contrôle sur un formulaire

utilisateur dans la fenêtre Propriétés. Pour afficher la fenêtre Propriétés, dans le menu **affichage** , cliquez sur **Fenêtre Propriétés**.

Remarque Les contrôles ne répondent pas aux événements alors que vous êtes en mode Création. Lorsque vous exécutez une boîte de dialogue pour l'afficher comme les utilisateurs la verront, le programme est en cours d'exécution. Les modifications que vous apportez aux propriétés d'un contrôle en mode d'exécution ne sont pas conservées lorsque l'objet UserForm est déchargé de la mémoire.

Remarque Des contrôles en réponse à des événements en cours d'exécution.

### **Comment faire référence aux contrôles dans un UserForm**

Manière de faire référence à des contrôles par programme dépend du type de la feuille de module Visual Basic lorsque vous exécutez le code. Si le code est exécuté à partir d'un module en général, la syntaxe est la suivante :

**UserFormName.Controlname.Property = valeur**

Par exemple, si vous souhaitez définir la propriété Text d'un contrôle TextBox nommé TextBox1 sur un objet UserForm nommé UserForm1 à celle de Bob, utilisez le code suivant :

```
UserForm1.TextBox1.Text = "Bob"
```

Si le code est dans une procédure qui est lancée par un événement d'un contrôle ou l'objet UserForm, vous n'avez pas à faire référence au nom de l'objet UserForm. Au lieu de cela, utilisez le code suivant :

```
TextBox1.Text = "Bob"
```

Lorsque vous associez un code à un objet, le code est associé à l'un des événements de cet objet. Dans la plupart des exemples dans cet article, vous attachez du code pour l'événement Click de l'objet du contrôle CommandButton .

### **Contrôles Label**

Contrôles Label sont principalement utilisés pour décrire d'autres contrôles sur un formulaire utilisateur. Un contrôle Label ne peut pas être modifié par l'utilisateur pendant l'exécution de l'objet UserForm. Utilisez la propriété Caption pour définir ou renvoyer le texte dans un contrôle Label . D'autres propriétés fréquemment utilisées pour la mise en forme d'un contrôle Label : la propriété de police et de la propriété ForeColor .

### **Comment faire pour utiliser une instruction WITH pour mettre en forme un contrôle Label**

Pour utiliser l'instruction WITH pour modifier les propriétés d'un contrôle Label , procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
4. Ajoutez un contrôle Label à l'objet UserForm.
5. Ajoutez un contrôle CommandButton sur l'objet UserForm.
6. Double-cliquez sur le contrôle **CommandButton** pour ouvrir la fenêtre Code pour l'objet UserForm.
7. Dans la fenêtre Code, tapez le code suivant pour l'événement Click du contrôle CommandButton1 :

```

8. Private Sub CommandButton1_Click()
9.
10.     With Label1
11.         ' Set the text of the label.
12.         .Caption = "This is Label Example 1"
13.         ' Automatically size the label control.
14.         .AutoSize = True
15.         .WordWrap = False
16.         ' Set the font used by the Label control.
17.         .Font.Name = "Times New Roman"
18.         .Font.Size = 14
19.         .Font.Bold = True
20.         ' Set the font color to blue.
21.         .ForeColor = RGB(0, 0, 255)
22.     End With
23.
24. End Sub

```

24. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.
25. Cliquez sur le **bouton de commande**.

Le texte « Ceci est l'exemple étiquette 1 » s'affiche sur le contrôle d'étiquette en gras Times New Roman d'une taille de police de 14.

## Contrôles de zone de texte

Contrôles TextBox sont souvent utilisés pour collecter les entrées d'un utilisateur. La propriété Text contient l'entrée effectuée dans un contrôle TextBox .

## Comment utiliser un contrôle TextBox à valider un mot de passe

Si vous définissez la propriété PasswordChar d'un contrôle de zone de texte, il devient un contrôle « masque de modification ». Chaque caractère est tapé dans le contrôle TextBox est remplacé visuellement par le caractère que vous spécifiez. Pour utiliser un contrôle TextBox à valider un mot de passe, procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans le menu **Outils**, pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion**, cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
4. Ajoutez un contrôle TextBox à l'objet UserForm.
5. Dans le menu **affichage**, cliquez sur **Propriétés** pour afficher la fenêtre Propriétés.
6. Dans la propriété PasswordChar du contrôle TextBox, tapez \*.

Remarque Vous modifiez la valeur d'un astérisque.

7. Ajoutez un contrôle CommandButton sur l'objet UserForm.
8. Double-cliquez sur le contrôle **CommandButton** pour ouvrir la fenêtre Code pour l'objet UserForm.
9. Dans la fenêtre Code, tapez le code suivant pour l'événement Click du contrôle CommandButton1 :

```
10. Private Sub CommandButton1_Click()  
11.  
12.     If TextBox1.Text <> "userform" Then  
13.         MsgBox "Password is Incorrect. Please reenter."  
14.         TextBox1.Text = ""  
15.         TextBox1.SetFocus  
16.     Else  
17.         MsgBox "Welcome!"  
18.         Unload Me  
19.     End If  
20.  
21. End Sub  
22.
```



23. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

24. Tapez mot de passe userform dans le contrôle TextBox .

25. Cliquez sur le contrôle **CommandButton**.

Pour cet exemple, le mot de passe est « objet userform ». Si vous tapez un mot de passe, vous recevez une boîte de message indiquant que votre mot de passe est incorrect, le contrôle TextBox est désactivé, et vous pouvez puis retapez le mot de passe. Lorsque vous tapez un mot de passe correct, vous recevez un message d'accueil, et l'objet UserForm est fermé.

## Contrôles CommandButton

Vous pouvez utiliser un contrôle CommandButton pour démarrer une procédure VBA. La procédure VBA est généralement associée à l'événement Click du contrôle CommandButton . Pour utiliser un contrôle de bouton de commande qui exécute une procédure lorsque l'événement Click se produit, procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
4. Ajoutez un contrôle CommandButton sur l'objet UserForm.
5. Double-cliquez sur le contrôle **CommandButton** pour afficher la fenêtre Code pour l'objet UserForm.
6. Dans la fenêtre Code, tapez le code suivant :

```
7. Private Sub CommandButton1_Click()  
8.  
9.     red = Int(Rnd * 255)  
10.    green = Int(Rnd * 255)  
11.    blue = Int(Rnd * 255)  
12.    CommandButton1.BackColor = RGB(red, green, blue)  
13.  
    End Sub
```

14. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

La couleur d'arrière-plan du contrôle **CommandButton1** change chaque fois que vous cliquez dessus.

## Contrôles ListBox

L'objectif du contrôle ListBox est à présenter à l'utilisateur une liste d'éléments à sélectionner à partir de. Vous pouvez stocker la liste d'éléments d'un contrôle de zone de liste sur une feuille de calcul Excel. Pour remplir un contrôle ListBox avec une plage de cellules dans une feuille de calcul, utilisez la propriété contenu (RowSource) . Lorsque vous utilisez la propriété MultiSelect , vous pouvez configurer un contrôle ListBox pour accepter plusieurs sélections.

### Comment faire pour obtenir l'élément actuellement sélectionné dans le contrôle ListBox

Utilisez la propriété Value d'un contrôle ListBox pour renvoyer l'élément actuellement sélectionné. Pour renvoyer l'élément actuellement sélectionné dans une contrôle de zone de liste de sélection unique, procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans les cellules a1 : a5 de la feuille Sheet1, tapez les valeurs que vous souhaitez utiliser pour remplir le contrôle ListBox .
3. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
4. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
5. Ajoutez un contrôle ListBox à l'objet UserForm.
6. Double-cliquez sur le contrôle **ListBox** pour afficher la fenêtre de Code pour le contrôle ListBox .
7. Dans la fenêtre Code, tapez le code suivant pour l'événement Click de ListBox1 :
8. Private Sub ListBox1\_Click()  
9.  
10.       MsgBox ListBox1.Value  
11.  
    End Sub

12. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

Lorsque vous cliquez sur un élément dans la liste, un message s'affiche avec l'élément actuellement sélectionné.

### **Comment obtenir les éléments sélectionnés dans un contrôle de zone de liste Sélectionner plusieurs**

Pour déterminer les éléments sélectionnés dans une contrôle de zone de liste de sélection multiple, vous devez exécuter une boucle sur tous les éléments dans la liste et ensuite interroger la propriété Selected . Pour retourner les éléments actuellement sélectionnés dans une contrôle ListBox , sélectionnez plusieurs, procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans les cellules a1 : a5 de la feuille Sheet1, tapez les valeurs que vous souhaitez utiliser pour remplir le contrôle ListBox .
3. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
4. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
5. Ajoutez un contrôle ListBox à l'objet UserForm.
6. Dans le menu **affichage** , cliquez sur **Propriétés** pour afficher la fenêtre Propriétés.
7. Tapez les valeurs indiquées pour les propriétés suivantes du contrôle ListBox :
8.     Property    Value
9.     -----        -----
10.     MultiSelect    1 - frmMultiSelectMulti  
       RowSource     Sheet1!A1:A8
11. Ajoutez un contrôle CommandButton sur l'objet UserForm.
12. Double-cliquez sur le contrôle **CommandButton** pour afficher la fenêtre Code pour l'objet UserForm.
13. Dans la fenêtre Code, tapez le code suivant pour l'événement Click du contrôle CommandButton1 :

```

14. Sub CommandButton1_Click ()
15.
16.     ' Loop through the items in the ListBox.
17.     For x = 0 to ListBox1.ListCount - 1
18.
19.         ' If the item is selected...
20.         If ListBox1.Selected(x) = True Then
21.
22.             ' display the Selected item.
23.             MsgBox ListBox1.List(x)
24.         End If
25.     Next x
26.
    End Sub

```

27. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

28. Sélectionnez un ou plusieurs éléments dans la liste.

29. Cliquez sur le **contrôle CommandButton1**.

Une fois que vous cliquez sur le **contrôle CommandButton1**, chaque élément que vous avez sélectionné dans le contrôle de zone de liste s'affiche dans un message séparé. Une fois que tous les éléments sélectionnés s'affichent dans une boîte de message, l'objet UserForm est automatiquement fermé.

### **Comment utiliser la propriété RowSource pour remplir un contrôle ListBox avec des cellules dans une feuille de calcul**

Pour remplir un contrôle ListBox à partir d'une plage de cellules dans une feuille de calcul à l'aide de la propriété contenu (RowSource) , procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans les cellules a1 : a5 de la feuille Sheet1, tapez les valeurs que vous souhaitez utiliser pour remplir le contrôle ListBox .
3. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
4. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
5. Ajoutez un contrôle ListBox à l'objet UserForm.
6. Ajoutez un contrôle CommandButton sur l'objet UserForm.

7. Double-cliquez sur le contrôle **CommandButton** pour afficher la fenêtre Code pour l'objet UserForm.

8. Dans la fenêtre Code, tapez le code suivant pour l'événement Click du contrôle CommandButton1 :

```
9. Private Sub CommandButton1_Click()  
10.     ListBox1.RowSource = "=Sheet1!A1:A5"  
    End Sub
```

11. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

Remarque ListBox1 ne contient pas de valeurs.

12. Cliquez sur le **contrôle CommandButton1**.

ListBox1 est rempli avec les valeurs dans les cellules a1 : a5 de la feuille Sheet1.

### **Comment faire pour remplir un contrôle ListBox avec les valeurs dans un tableau**

Cet exemple vous montre comment remplir un contrôle ListBox avec une variable de tableau. Vous devez affecter les valeurs du tableau pour l'élément contrôle ListBox à la fois. En général, ce processus requiert l'utilisation d'une structure en boucle, comme un pour... Suivant boucle. Pour remplir un contrôle ListBox avec une variable de tableau, procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
4. Ajoutez un contrôle ListBox à l'objet UserForm.
5. Dans le menu **Insertion** , cliquez sur **Module** pour insérer une feuille de module.
6. Dans la fenêtre Code, tapez le code suivant :

```
7. Sub PopulateListBox()  
8.  
9.     Dim MyArray As Variant  
10.     Dim Ctr As Integer
```

```

11.     MyArray = Array("Apples", "Oranges", "Peaches",
    "Bananas", "Pineapples")
12.
13.     For Ctr = LBound(MyArray) To UBound(MyArray)
14.         UserForm1.ListBox1.AddItem MyArray(Ctr)
15.     Next
16.
17.     UserForm1.Show
18.
    End Sub

```

19. Dans le menu **Outils**, cliquez sur **Macros**, cliquez sur **PopulateListBox**, puis cliquez sur **exécuter**.

La procédure PopulateListBox génère un tableau simple et puis ajoute les éléments dans le tableau au contrôle ListBox à l'aide de la méthode AddItem . Ensuite, l'objet UserForm s'affiche.

### **Comment faire pour utiliser une plage de cellules horizontale sur une feuille de calcul pour remplir un contrôle ListBox**

Si vous définissez la propriété contenu (RowSource) d'un contrôle ListBox à une plage de cellules horizontale, uniquement la première valeur s'affiche dans le contrôle ListBox .

Pour remplir un contrôle ListBox à partir d'une plage horizontale de cellules à l'aide de la méthode AddItem , procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans les cellules a1 à E1 de la feuille Sheet1, tapez les valeurs que vous souhaitez utiliser pour remplir le contrôle ListBox .
3. Dans le menu **Outils**, pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
4. Dans le menu **Insertion**, cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
5. Ajoutez un contrôle ListBox à l'objet UserForm.
6. Dans le menu **Insertion**, cliquez sur **Module** pour insérer une feuille de module.
7. Dans la fenêtre Code, tapez le code suivant :

```

8. Sub PopulateListWithHorizontalRange()
9.
10.     For Each x In Sheet1.Range("A1:E1")
11.         UserForm1.ListBox1.AddItem x.Value
12.     Next
13.
14.     UserForm1.Show
15.
    End Sub

```

16. Dans le menu **Outils**, cliquez sur **Macros**, cliquez sur **PopulateListWithHorizontalRange**, puis cliquez sur **exécuter**.

La procédure de macro boucle sur les cellules a1 : E5 dans la feuille Sheet1, ajoutez les valeurs à ListBox1 une à la fois.

Remarque ListBox1 n'est pas lié aux cellules a1 : E5 dans la feuille Sheet1.

### **Comment faire pour retourner plusieurs valeurs à partir d'un contrôle de zone de liste qui est lié à plusieurs colonnes de données**

Vous pouvez mettre en forme les contrôles de zone de liste pour afficher plus d'une colonne de données. Cela signifie que le contrôle ListBox affiche plus d'un élément sur chaque ligne de la liste. Pour retourner plusieurs valeurs à partir de l'élément sélectionné dans la liste, procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Tapez les données suivantes dans les cellules qui sont indiqués dans la feuille Sheet1 :

A1 : année	B1 : région	C1 : ventes
A2 : 1996	B2 : nord	C2 : 140
A3 : 1996	B3 : sud	C3 : 210
A4 : 1997	B4 : nord	C4 : 190
A5 : 1997	B5 : sud	C5 : 195

3. Dans le menu **Outils**, pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
4. Dans le menu **Insertion**, cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
5. Ajoutez un contrôle Label à l'objet UserForm.

6. Ajoutez un contrôle ListBox à l'objet UserForm.
7. Avec le bouton droit de la **zone de liste**, puis cliquez sur **Propriétés**.
8. Tapez ou sélectionnez les valeurs qui sont indiquées pour les propriétés suivantes du contrôle ListBox , comme répertoriés dans le tableau suivant :

9.	Property	Value
10.	-----	
11.	BoundColumn	1
12.	ColumnCount	3
13.	ColumnHeads	True
	RowSource	Sheet1!A2:A5

14. Double-cliquez sur le contrôle **ListBox** pour afficher la fenêtre de Code pour le contrôle ListBox .

15. Dans la fenêtre Code, tapez le code suivant :

```

16. Private Sub ListBox1_Change()
17.
18.     Dim SourceData As Range
19.     Dim Val1 As String, Val2 As String, Val3 As String
20.
21.     Set SourceRange = Range(ListBox1.RowSource)
22.
23.     Val1 = ListBox1.Value
24.     Val2 = SourceRange.Offset(ListBox1.ListIndex,
25.     1).Resize(1, 1).Value
26.     Val3 = SourceRange.Offset(ListBox1.ListIndex,
27.     2).Resize(1, 1).Value
28.
29.     Label1.Caption = Val1 & " " & Val2 & " " & Val3
30.
31. End Sub

```

29. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

Lorsque vous cliquez sur une entrée dans le contrôle ListBox , l'étiquette change pour afficher les trois éléments dans cette entrée.

### **Comment faire pour supprimer tous les éléments d'un contrôle de zone de liste qui est lié à une feuille de calcul**

Pour supprimer tous les éléments d'un contrôle de zone de liste qui est lié à une feuille de calcul, effacez la valeur qui est stockée dans la propriété contenu (RowSource) . Pour supprimer des éléments d'un contrôle de zone de liste qui est lié à une feuille de calcul, procédez comme suit :



1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans les cellules a1 : a5 de la feuille Sheet1, tapez les valeurs que vous souhaitez utiliser pour remplir le contrôle ListBox .
3. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
4. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
5. Ajoutez un contrôle ListBox à l'objet UserForm.
6. Cliquez sur le contrôle **zone de liste** , puis cliquez sur **Propriétés**.
7. Dans la propriété contenu (RowSource) , tapez la feuille Sheet1 ! A1 : a5.
8. Ajoutez un contrôle CommandButton sur l'objet UserForm.
9. Double-cliquez sur le contrôle **CommandButton** pour afficher la fenêtre de Code du contrôle CommandButton .
10. Dans la fenêtre Code, tapez le code suivant pour l'événement Click du contrôle CommandButton1 :
11. Private Sub CommandButton1\_Click()  
12.  
13.       ListBox1.RowSource = ""  
14.  
15.       End Sub

15. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

Le contrôle de zone de liste que vous avez ajouté à l'objet UserForm est rempli avec les valeurs que vous avez entré dans la feuille Sheet1.

16. Cliquez sur le contrôle CommandButton1.

Tous les éléments sont supprimés de ListBox1.

### **Comment faire pour supprimer tous les éléments d'un contrôle ListBox qui n'est pas lié à une feuille de calcul**

Il n'y a pas de commande VBA simple qui supprime tous les éléments d'un contrôle de zone de liste si la liste n'est pas liée à une feuille de calcul. Pour

supprimer tous les éléments d'un contrôle ListBox qui est rempli à partir d'un tableau Visual Basic, procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
4. Ajoutez un contrôle ListBox à l'objet UserForm.
5. Dans le menu **Insertion** , cliquez sur **Module** pour insérer une feuille de module.

6. Dans la fenêtre Code, tapez le code suivant :

```
7. Sub PopulateListBox()  
8.  
9.     Dim MyArray As Variant  
10.    Dim Ctr As Integer  
11.    MyArray = Array("Apples", "Oranges", "Peaches",  
    "Bananas", "Pineapples")  
12.  
13.    For Ctr = LBound(MyArray) To UBound(MyArray)  
14.        UserForm1.ListBox1.AddItem MyArray(Ctr)  
15.    Next  
16.  
17.    UserForm1.Show  
18.  
    End Sub
```

19. Ajoutez un contrôle CommandButton sur l'objet UserForm.

20. Double-cliquez sur le contrôle **CommandButton** pour afficher la fenêtre de Code du contrôle CommandButton .

21. Dans la fenêtre Code, tapez le code suivant pour l'événement Click du contrôle CommandButton1 :

```
22. Private Sub CommandButton1_Click()  
23.  
24.     For i = 1 To ListBox1.ListCount  
25.         ListBox1.RemoveItem 0  
26.     Next I  
27.  
    End Sub
```

28. Dans le menu **Outils** , cliquez sur **Macros**, cliquez sur **PopulateListBox**, puis cliquez sur **exécuter**.

Le contrôle ListBox est rempli, et l'objet UserForm s'affiche alors.

29. Cliquez sur le contrôle CommandButton1.

Tous les éléments sont supprimés de ListBox1.

## **Contrôles ComboBox**

Vous pouvez utiliser le contrôle de zone de liste déroulante sous la forme d'une zone de liste déroulante ou une zone de liste déroulante dans laquelle vous pouvez sélectionner une valeur dans une liste ou taper une nouvelle valeur. La propriété Style détermine si le contrôle ComboBox se comporte comme une zone de liste déroulante ou une zone de liste déroulante.

Remarque Tous les exemples dans la section précédente pour le contrôle ListBox peuvent également être appliqués sur le contrôle ComboBox , à l'exception de l'exemple « Comment obtenir les éléments sélectionnés dans un contrôle ListBox, sélectionnez plusieurs ».

## **Comment faire pour ajouter un nouvel élément à la liste si le contrôle de zone de liste déroulante n'est pas lié à une feuille de calcul**

Lorsque vous tapez une valeur qui n'est pas déjà dans la liste dans le contrôle ComboBox , vous souhaitez peut-être ajouter la nouvelle valeur à la liste. Pour ajouter la nouvelle valeur que vous avez tapé dans le contrôle de zone de liste déroulante si le contrôle de zone de liste déroulante n'est pas lié à la feuille de calcul, procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
4. Ajoutez un contrôle ComboBox à l'objet UserForm.
5. Dans le menu **Insertion** , cliquez sur **Module** pour insérer une feuille de module.
6. Dans la fenêtre Code, tapez le code suivant :

```

7. Sub PopulateComboBox()
8.
9.     Dim MyArray As Variant
10.    Dim Ctr As Integer
11.    MyArray = Array("Apples", "Oranges", "Peaches",
    "Bananas", "Pineapples")
12.
13.    For Ctr = LBound(MyArray) To Ubound(MyArray)
14.        UserForm1.ComboBox1.AddItem MyArray(Ctr)
15.    Next
16.
17.    UserForm1.Show
18.
    End Sub

```

19. Ajoutez un contrôle **CommandButton** sur l'objet UserForm.

20. Double-cliquez sur le contrôle **CommandButton** pour afficher la fenêtre de Code du contrôle **CommandButton** .

21. Dans la fenêtre Code, tapez le code suivant pour l'événement Click du contrôle **CommandButton1** :

```

22. Private Sub CommandButton1_Click()
23.
24.     Dim listvar As Variant
25.
26.     listvar = ComboBox1.List
27.
28.     On Error Resume Next
29.     ' If the item is not found in the list...
30.     If IsError(WorksheetFunction.Match(ComboBox1.Value,
    listvar, 0)) Then
31.         ' add the new value to the list.
32.         ComboBox1.AddItem ComboBox1.Value
33.     End If
34.
    End Sub

```

35. Dans le menu **Outils** , cliquez sur **Macros**, cliquez sur **PopulateListBox**, puis cliquez sur **exécuter**.

Le contrôle de zone de liste déroulante est rempli, et l'objet UserForm s'affiche alors.

36. Dans le contrôle zone de liste déroulante , tapez mangues (ou toute valeur qui n'est pas déjà dans la liste).

37. Cliquez sur le **contrôle CommandButton1**.

La nouvelle valeur que vous avez tapé maintenant apparaît à la fin de la liste.

### **Comment faire pour ajouter un nouvel élément à la liste si le contrôle de zone de liste déroulante est lié à une feuille de calcul**

Lorsqu'un utilisateur tape une valeur qui n'est pas déjà dans la liste dans le contrôle ComboBox, vous souhaitez peut-être ajouter la nouvelle valeur à la liste. Pour ajouter la nouvelle valeur que vous avez tapé dans le contrôle ComboBox à la liste, procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans les cellules a1 : a5 de la feuille Sheet1, tapez les valeurs que vous souhaitez utiliser pour remplir le contrôle zone de liste déroulante .
3. Sélectionnez les cellules a1 : a5 de la feuille Sheet1.
4. Dans le menu **Insertion**, pointez sur **nom**, puis cliquez sur **définir**.

Dans la zone **noms dans le classeur**, tapez ListRange, puis cliquez sur OK. Cette opération crée le nom défini ListRange. Vous pouvez utiliser le nom défini ListRange pour lier la propriété contenu (RowSource) du contrôle ComboBox à la feuille de calcul.

5. Dans le menu **Outils**, pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
6. Dans le menu **Insertion**, cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
7. Ajoutez un contrôle ComboBox à l'objet UserForm.
8. Dans les **Propriétés** de ComboBox1, tapez la feuille Sheet1 ! ListRange en tant que la propriété **contenu (RowSource)**.
9. Ajoutez un contrôle CommandButton sur l'objet UserForm.
10. Double-cliquez sur le contrôle **CommandButton** pour afficher la fenêtre de Code du contrôle CommandButton .
11. Dans la fenêtre Code, tapez le code suivant pour l'événement Click du contrôle CommandButton1 :
12. `Private Sub CommandButton1_Click()`
- 13.

```

14.     Dim SourceData As Range
15.     Dim found As Object
16.
17.     Set SourceData = Range("ListRange")
18.     Set found = Nothing
19.     ' Try to find the value on the worksheet.
20.     Set found = SourceData.Find(ComboBox1.Value)
21.
22.     ' If the item is not found in the list...
23.     If found Is Nothing Then
24.         ' redefine ListRange.
25.         SourceData.Resize(SourceData.Rows.Count + 1,
26. 1).Name = "ListRange"
27.         ' Add the new item to the end of the list on the
28.         worksheet.
29.         SourceData.Offset(SourceData.Rows.Count,
30. 0).Resize(1, 1).Value _
31.         = ComboBox1.Value
32.         ' Reset the list displayed in the ComboBox.
33.         ComboBox1.RowSource =
34. Range("listrange").Address(external:=True)
35.     End If
36.
37. End Sub

```

33. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

L'objet UserForm s'affiche dans la feuille Sheet1.

34. Dans le contrôle zone de liste déroulante , tapez une valeur qui n'est pas déjà dans la liste.

35. Cliquez sur le **contrôle CommandButton1**.

Le nouvel élément que vous avez tapé dans le contrôle ComboBox est ajouté à la liste, et que le contrôle de zone de liste déroulante est lié à la liste est étendue pour inclure les cellules a1 : a6.

### **Comment faire pour afficher la liste d'un contrôle de zone de liste déroulante lorsque l'objet UserForm s'affiche.**

Parfois, il peut être utile d'afficher la liste d'un contrôle ComboBox , apparition d'un objet UserForm. L'exemple suivant utilise l'événement Activate de l'objet UserForm. Pour afficher la liste d'un contrôle ComboBox , procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.

2. Dans les cellules a1 : a5 de la feuille Sheet1, tapez les valeurs que vous souhaitez utiliser pour remplir le contrôle zone de liste déroulante .
3. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
4. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
5. Ajoutez un contrôle ComboBox à l'objet UserForm.
6. Dans les **Propriétés** de ComboBox1, tapez la feuille Sheet1 ! A1 : a5 que la propriété **contenu (RowSource)** .
7. Double-cliquez sur l' **objet UserForm** pour afficher la fenêtre Code pour l'objet UserForm.
8. Dans la fenêtre Code, tapez le code suivant pour l'événement Click du bouton de commande :
9. Private Sub UserForm\_Activate()  
 10.  
 11.                   ComboBox1.DropDown  
 12.  
                   End Sub
13. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

L'objet UserForm s'affiche dans la feuille Sheet1, et vous pouvez voir la liste de ComboBox1.

### **Comment faire pour afficher la liste d'un contrôle de zone de liste déroulante lorsque vous effectuez une sélection dans un autre contrôle ComboBox**

Pour afficher automatiquement la liste d'un contrôle de zone de liste déroulante lorsqu'un choix est effectué dans un autre contrôle de zone de liste déroulante , procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans les cellules a1 : a10 dans la feuille Sheet1, tapez les valeurs que vous souhaitez utiliser pour remplir le contrôle zone de liste déroulante .
3. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.

4. Dans le menu **Insertion**, cliquez sur **Module**.
5. Dans la fenêtre Code du module, tapez le code suivant :
6. Sub DropDown\_ComboBox()
- 7.
8.     UserForm1.ComboBox2.DropDown
- 9.
- End Sub
10. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
11. Ajoutez un contrôle ComboBox à l'objet UserForm.
12. Dans les **Propriétés** de ComboBox1, tapez la feuille Sheet1 ! A1 : a5 que la propriété **contenu (RowSource)** .
13. Double-cliquez sur le contrôle **ComboBox** pour ouvrir la fenêtre de Code pour le contrôle ComboBox .
14. Dans la fenêtre de Code pour le contrôle de zone de liste déroulante , tapez le code suivant pour l'événement Click du contrôle ComboBox :
15. Private Sub ComboBox1\_Click()
- 16.
17.     Application.OnTime Now, "DropDown\_ComboBox"
- 18.
- End Sub
19. Ajoutez un deuxième contrôle de zone de liste déroulante à l'objet UserForm.
20. Dans les **Propriétés** ComboBox2, tapez la feuille Sheet1 ! A6:a10 en tant que la propriété **contenu (RowSource)** .
21. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

Lorsque vous cliquez sur un élément dans la liste ComboBox1 , la liste de ComboBox2 s'affiche automatiquement.

## Contrôle Frame

Utiliser un contrôle Frame pour regrouper des éléments logiquement associés dans un objet UserForm. Contrôles Frame sont fréquemment utilisés pour regrouper des contrôles OptionButton.



## Comment effectuer une boucle sur tous les contrôles dans un contrôle Frame

Pour utiliser un pour chaque... Suivant boucle pour accéder à tous les contrôles dans un contrôle Frame , procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
4. Ajoutez un contrôle de cadre à l'objet UserForm.
5. Ajouter un contrôle OptionButton dans le contrôle Frame .

Répétez cette étape pour ajouter deux contrôles OptionButton plus dans le contrôle Frame .

6. Double-cliquez sur le contrôle **Frame** pour ouvrir la fenêtre de Code pour le contrôle de l'image .
7. Dans la fenêtre Code, tapez le code suivant pour l'événement Click de cadre :

```
8. Private Sub Frame1_Click()  
9.  
10.     Dim Ctrl As Control  
11.  
12.     For Each Ctrl In Frame1.Controls  
13.         Ctrl.Enabled = Not Ctrl.Enabled  
14.     Next  
15.  
    End Sub
```

16. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

17. Dans l'objet UserForm, cliquez sur le contrôle **Frame**.

La première fois que vous cliquez sur le contrôle Frame , tous les contrôles dans le contrôle Framene sont pas disponibles. Si vous cliquez de nouveau sur le contrôle **Frame** , les contrôles sont à nouveau disponibles.

## Contrôle OptionButton

Vous pouvez utiliser des groupes de contrôles `OptionButton` pour effectuer une sélection parmi un groupe d'options. Vous pouvez utiliser une ou l'autre des techniques suivantes pour regrouper des contrôles `OptionButton` :

- Contrôle `Frame`
- `GroupName` , propriété

Remarque La valeur sur la valeur `Oui` et la valeur `True` indiquent qu'un contrôle `OptionButton` est sélectionné. La valeur `Off` , la valeur `No` et la valeur `False` indiquent qu'un contrôle `OptionButton` n'est pas sélectionnée.

### **Comment faire pour déterminer le contrôle `OptionButton` est sélectionné lorsque les contrôles `OptionButton` sur un contrôle `Frame`**

Lorsque vous regroupez des contrôles de boutons d'option à l'aide d'un contrôle `Frame` , vous pouvez déterminer le contrôle `OptionButton` est sélectionné en effectuant une boucle sur tous les contrôles dans le contrôle `Frame` et en vérifiant la valeur de propriété de chaque contrôle. Pour déterminer le contrôle `OptionButton` est sélectionné, procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet `UserForm` dans votre classeur.
4. Ajoutez un contrôle de cadre à l'objet `UserForm`.
5. Ajouter un contrôle `OptionButton` dans le contrôle `Frame` .

Répétez cette étape pour ajouter deux contrôles `OptionButton` plus dans le contrôle `Frame` .

6. Ajoutez un contrôle `CommandButton` sur l'objet `UserForm` à l'extérieur du cadre de contrôle.
7. Double-cliquez sur le contrôle **`CommandButton`** pour afficher la fenêtre Code pour l'objet `UserForm`.
8. Dans la fenêtre Code, tapez le code suivant pour l'événement `Click` du contrôle `CommandButton1` :

```

9. Private Sub CommandButton1_Click()
10.
11.     For Each x In Frame1.Controls
12.         If x.Value = True Then
13.             MsgBox x.Caption
14.         End If
15.     Next
16.
    End Sub

```

17. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

18. Dans l' **objet UserForm**, cliquez sur un contrôle OptionButton , puis cliquez sur **CommandButton1**.

Une boîte de message s'affiche qui contient la légende du contrôle OptionButton actuellement sélectionné.

### **Comment faire pour déterminer le contrôle OptionButton est sélectionné**

L'objectif de l'exemple suivant est de déterminer le contrôle OptionButton est sélectionné dans le groupe 1. Pour créer un UserForm qui a deux groupes de contrôles OptionButton , procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
4. Ajoutez un contrôle de cadre à l'objet UserForm.
5. Ajouter un contrôle OptionButton dans le contrôle Frame .

Répétez cette étape pour ajouter deux contrôles OptionButton plus dans le contrôle Frame .

6. Pour chaque contrôle OptionButton , tapez Groupe1 dans la propriété GroupName .
7. Répétez les étapes 4 et 5 pour créer un deuxième contrôle de cadre qui contient trois contrôles OptionButton .

8. Pour chaque contrôle **OptionButton** dans le deuxième contrôle **Frame** , tapez **groupe2** dans la propriété **GroupName** .
9. Ajoutez un contrôle **CommandButton** sur l'objet **UserForm** à l'extérieur des contrôles **Frame**.
10. Double-cliquez sur le contrôle **CommandButton** pour afficher la fenêtre Code pour l'objet **UserForm**.
11. Dans la fenêtre Code, tapez le code suivant pour l'événement **Click** du contrôle **CommandButton1** :

```

12. Private Sub CommandButton1_Click()
13.
14.     Dim x As Control
15.
16.     ' Loop through ALL the controls on the UserForm.
17.     For Each x In Me.Controls
18.         ' Check to see if "Option" is in the Name of each
control.
19.         If InStr(x.Name, "Option") Then
20.             ' Check Group name.
21.             If x.GroupName = "Group1" Then
22.                 ' Check the status of the OptionButton.
23.                 If x.Value = True Then
24.                     MsgBox x.Caption
25.                     Exit For
26.                 End If
27.             End If
28.         End If
29.     Next
30. End Sub

```

31. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.
32. Dans l'objet **UserForm**, cliquez sur un contrôle **OptionButton** dans **Group1**, puis cliquez sur **CommandButton1**.

Une boîte de message s'affiche qui contient la légende du contrôle **OptionButton** qui est actuellement sélectionné.

## Contrôle **CheckBox**

Vous pouvez utiliser un contrôle de case à cocher pour indiquer une valeur **true** ou **false**. Un contrôle de case à cocher qui apparaît avec une coche qu'il indique une valeur de

True. Une case à cocher qui apparaît sans coche indique la valeur False. Si la valeur de la propriété TripleState est True, un contrôle de case à cocher peut également avoir la valeur Null. Un contrôle de case à cocher qui a une valeur Null semble ne pas être disponibles.

Remarque La valeur sur la valeur Oui et la valeur True indiquent qu'un contrôle de case à cocher est activée. La valeur Off , la valeur No et la valeur False indiquent qu'un contrôle de case à cocher est désactivé.

### **Comment faire pour vérifier la valeur d'un contrôle de case à cocher**

Pour utiliser la propriété Value pour renvoyer la valeur actuelle d'un contrôle de case à cocher , procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
4. Ajoutez un contrôle de case à cocher à l'objet UserForm.
5. Dans la liste de propriétés pour CheckBox1, sélectionnez True la propriété TripleState .
6. Double-cliquez sur le contrôle **case à cocher** pour afficher la fenêtre de Code du contrôle CheckBox .
7. Dans la fenêtre Code, tapez le code suivant pour l'événement de Modification de CheckBox1 :

```
8. Private Sub CheckBox1_Change()  
9.  
10.     Select Case CheckBox1.Value  
11.         Case True  
12.             CheckBox1.Caption = "True"  
13.         Case False  
14.             CheckBox1.Caption = "False"  
15.         Case Else  
16.             CheckBox1.Caption = "Null"  
17.     End Select  
18.  
End Sub
```

19. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

Lorsque vous cliquez sur le contrôle **case à cocher** , la légende du contrôle CheckBox change pour refléter la valeur actuelle.

## **ToggleButton, contrôle**

Un contrôle ToggleButton a la même apparence qu'un contrôle de bouton de commande jusqu'à ce que vous cliquez dessus. Lorsque vous cliquez sur un contrôle ToggleButton , elle semble être enfoncée ou enclenchée. La propriété Value d'un contrôle ToggleButton est True lorsque le bouton est sélectionné et False lorsque le bouton n'est pas sélectionné. Si la valeur de la propriété TripleState est True, un contrôle ToggleButton peut également avoir la valeur Null. Un contrôle ToggleButton qui a une valeur Null semble ne pas être disponibles.

Remarque La valeur sur la valeur Oui et la valeur True indiquent qu'un contrôle ToggleButton est sélectionné. La valeur Off , la valeur No et la valeur False indiquent qu'un contrôle ToggleButton n'est pas sélectionné.

### **Comment faire pour obtenir la valeur d'un contrôle ToggleButton**

Pour obtenir la valeur d'un contrôle ToggleButton , procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans le menu **Outils** , pointez sur **Macro** , puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
4. Ajoutez un contrôle ToggleButton dans l'objet UserForm.
5. Ajoutez un contrôle Label à l'objet UserForm.
6. Double-cliquez sur le contrôle **ToggleButton** pour ouvrir la fenêtre de Code pour le contrôle ToggleButton .
7. Dans la fenêtre Code, tapez le code suivant pour l'événement ToggleButton1Click :  

```
8. Private Sub ToggleButton1_Click()  
9.
```

```

10.     If ToggleButton1.Value = True Then
11.         ' Set UserForm background to Red.
12.         Me.BackColor = RGB(255, 0, 0)
13.     Else
14.         ' Set UserForm background to Blue.
15.         Me.BackColor = RGB(0, 0, 255)
16.     End If
17.
    End Sub

```

18. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

Lorsque vous cliquez sur le contrôle ToggleButton , la couleur d'arrière-plan des modifications UserForm.

### **Comment faire pour créer un groupe de contrôles ToggleButton s'excluent**

Cet exemple utilise l'événement MouseUp pour définir une variable et appelle la procédure ExclusiveToggleButtons . La procédure ExclusiveToggleButtons détermine le contrôle bouton bascule qui est sélectionné, puis annule les autres. Pour créer un groupe de contrôles ToggleButton s'excluent, procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion**, cliquez sur **Module**.
4. Dans la fenêtre Code du module, tapez le code suivant :
5. ' Variable that holds the name of the ToggleButton that was clicked.
6. Public clicked As String
- 7.
8. Sub ExclusiveToggleButtons()
- 9.
10. Dim toggle As Control
- 11.
12. ' Loop through all the ToggleButtons on Frame1.
13. For Each toggle In UserForm1.Frame1.Controls
- 14.
15. ' If Name of ToggleButton matches name of  
ToggleButton
16. ' that was clicked...
17. If toggle.Name = clicked Then

```

18.         '...select the button.
19.         toggle.Value = True
20.     Else
21.         '...otherwise clear the selection of the button.
22.         toggle.Value = False
23.     End If
24. Next
25.
    End Sub

```

26. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.

27. Ajoutez un contrôle de cadre à l'objet UserForm.

28. Ajoutez un contrôle ToggleButton dans le contrôle Frame .

Répétez cette étape pour ajouter deux autres contrôles ToggleButton dans le contrôle Frame .

29. Double-cliquez sur le contrôle **image** pour afficher la fenêtre Code pour l'objet UserForm.

30. Dans la fenêtre Code du module, tapez le code suivant pour l'événement MouseUp du contrôle ToggleButton :

```

31. Private Sub ToggleButton1_MouseUp(ByVal Button As Integer,
32.     _
33.         ByVal Shift As Integer, ByVal X As Single,
34.         ByVal Y As Single)
35.     clicked = ToggleButton1.Name
36.     Application.OnTime Now, "ExclusiveToggleButtons"
37. End Sub
38.
39. Private Sub ToggleButton2_MouseUp(ByVal Button As
40.     Integer, _
41.         ByVal Shift As Integer, ByVal X As Single,
42.         ByVal Y As Single)
43.     clicked = ToggleButton2.Name
44.     Application.OnTime Now, "ExclusiveToggleButtons"
45. End Sub
46.
47. Private Sub ToggleButton3_MouseUp(ByVal Button As
    Integer, _

```



```

48.           ByVal Shift As Integer, ByVal X As Single,
           ByVal Y As Single)
49.
50.           clicked = ToggleButton3.Name
51.           Application.OnTime Now, "ExclusiveToggleButtons"
52.
           End Sub

```

53. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

Lorsque vous cliquez sur un contrôle ToggleButton ,  
contrôle ToggleButton précédemment sélectionné est annulé.

## Contrôle TabStrip

Utilisez un contrôle TabStrip pour visualiser différents ensembles d'informations d'un ensemble de contrôles.

### Comment faire pour contrôler un contrôle TabStrip par programme

Pour modifier la propriété BackColor d'un contrôle d'Image en fonction de l'onglet est sélectionné, procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
4. Ajoutez un contrôle TabStrip à l'objet UserForm.
5. Ajoutez un contrôle Image qui couvre la base du contrôle TabStrip , mais qui ne couvre pas les onglets.
6. Dans le volet Propriétés Image1, tapez & H000000FF & de la propriété BackColor .
7. Double-cliquez sur le contrôle **TabStrip** pour ouvrir la fenêtre Code pour le contrôle TabStrip .
8. Dans la fenêtre Code, tapez le code suivant pour l'événement de Modification de TabStrip1 :

```

9. Private Sub TabStrip1_Change()
10.
11.     Dim i As Integer
12.
13.     i = TabStrip1.SelectedItem.Index
14.     Select Case i
15.         Case 0
16.             ' If Tab1 is selected, change the color of
Image control to Red.
17.             Image1.BackColor = RGB(255, 0, 0)
18.         Case 1
19.             ' If Tab2 is selected, change the color of
Image control to Green.
20.             Image1.BackColor = RGB(0, 255, 0)
21.     End Select
22.
End Sub

```

23. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

La couleur du contrôle Image change en fonction de la page dans le contrôle TabStrip est actif.

## Contrôle multiPage

Utiliser un contrôle MultiPage pour travailler avec un grand nombre d'informations qui peuvent être triées en plusieurs catégories. Un contrôle MultiPage est constitué par une ou plusieurs Pageobjets chacun contient un ensemble différent de contrôles. Vous pouvez définir la page active par programme en définissant la propriété Value du contrôle MultiPage .

### Comment des contrôles par programme un contrôle MultiPage

Pour ajouter un contrôle MultiPage et le contrôle à l'aide d'une macro, procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
4. Ajouter un contrôle MultiPage à l'objet UserForm.

5. Ajoutez un contrôle Label à la page 1 sur le contrôle MultiPage .
6. Ajoutez un contrôle de zone de texte « Page1 » sur le contrôle MultiPage .
7. Sur le contrôle MultiPage , cliquez sur **page 2** et répétez les étapes 5 et 6 pour ajouter un contrôle Label et un contrôle TextBox .
8. Double-cliquez sur le contrôle **MultiPage** pour ouvrir la fenêtre de Code du contrôle MultiPage .
9. Dans la fenêtre Code, tapez le code suivant pour l'événement de Modification de MultiPage1 :

```

10. Private Sub MultiPage1_Change()
11.
12.     Select Case MultiPage1.Value
13.         ' If activating Page1...
14.         Case 0
15.             Label1.Caption = TextBox2.Text
16.             TextBox1.Text = ""
17.         ' If activating Page2...
18.         Case 1
19.             Label2.Caption = TextBox1.Text
20.             TextBox2.Text = ""
21.     End Select
22.
23. End Sub

```

23. Dans la fenêtre Code, tapez le code suivant pour l'événement Initialize de l'objet UserForm :

```

24. Private Sub UserForm_Initialize()
25.
26.     ' Force Page1 to be active when UserForm is displayed.
27.     MultiPage1.Value = 0
28.     Label1.Caption = ""
29.
30. End Sub

```

30. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

Dans le contrôle zone de texte sur la page 1, tapez Test. Lorsque vous cliquez sur l'onglet de la **page 2** , TextBox2 est désactivée et la légende de Label2 change à l'entrée que vous avez apportées dans TextBox1 de la Page1 (« Test »).

## **Comment créer une interface d'Assistant à l'aide d'un contrôle MultiPage**

Lorsqu'une tâche nécessite plusieurs étapes incrémentielles, une interface peut être très efficace. Le contrôle MultiPage vous permet de créer une interface d'Assistant au lieu d'utiliser plusieurs UserForms. Cet exemple montre comment manipule un contrôle MultiPage qui comporte trois pages. Une procédure qui est associée à la Événement Initialize de l'objet UserForm désactive Page2 et Page3 et force Page1 du contrôle MultiPage soit active.

Remarque Lorsque vous indexez les pages d'un contrôle MultiPage à l'aide de la collection de Pages , la première page dans la collection est zéro. En outre, cette procédure définit la légende des contrôles CommandButton et désactive le < retour bouton.

Remarque La procédure qui est assignée à l'événement Click du contrôle CommandButton1 contrôle les fonctionnalités de la < retour bouton. La procédure qui est assignée à l'événement Click du contrôle CommandButton2, il contrôle les fonctionnalités de la suivante > bouton. Pour créer une interface d'Assistant à l'aide d'un contrôle MultiPage , procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
4. Ajouter un contrôle MultiPage à l'objet UserForm.
5. Cliquez sur l'onglet de la **page 1** , puis cliquez sur **Nouvelle Page** pour ajouter le contrôle MultiPage Page3.
6. Ajoutez un contrôle CommandButton sur l'objet UserForm qui ne se trouve pas sur le contrôle MultiPage .  
  
Répétez cette étape pour ajouter un deuxième contrôle CommandButton sur l'objet UserForm.
7. Double-cliquez sur **formulaire utilisateur** pour ouvrir la fenêtre Code pour l'objet UserForm.
8. Dans la fenêtre Code, tapez le code suivant pour l'événement Initialize de l'objet UserForm :
9. Private Sub UserForm\_Initialize()  
10.

```

11.     With MultiPage1
12.         ' The next 2 lines disable Page2 & Page3.
13.         .Pages(1).Enabled = False
14.         .Pages(2).Enabled = False
15.         ' Make Page1 the active page.
16.         .Value = 0
17.     End With
18.
19.     ' Set the caption on the CommandButtons.
20.     CommandButton1.Caption = "<Back"
21.     CommandButton1.Enabled = False
22.     CommandButton2.Caption = "Next>"
23.
24. End Sub
25.
26. ' Procedure for the "<Back" button
27. Private Sub CommandButton1_Click()
28.     Select Case MultiPage1.Value
29.         Case 1                                     ' If Page2
   is active...
30.             With MultiPage1
31.                 .Pages(0).Enabled = True         ' Enable
   Page1.
32.                 .Value = MultiPage1.Value - 1   ' Move back
   1 page.
33.                 .Pages(1).Enabled = False       ' Disable
   Page2.
34.             End With
35.             CommandButton1.Enabled = False     ' Disable
   Back button.
36.
37.         Case 2                                     ' If Page3
   is active...
38.             With MultiPage1
39.                 .Pages(1).Enabled = True         ' Enable
   Page2.
40.                 .Value = MultiPage1.Value - 1   ' Move back
   1 page.
41.                 .Pages(2).Enabled = False       ' Disable
   Page3.
42.             CommandButton2.Caption = "Next>"
43.             End With
44.         End Select
45.
46. End Sub
47.
48. ' Procedure for the "Next>" button
49. Private Sub CommandButton2_Click()
50.
51.     Select Case MultiPage1.Value

```

```

52.          Case 0                                ' If Page1
   is active...
53.          With MultiPage1
54.          .Value = MultiPage1.Value + 1        ' Move
   forward 1 page.
55.          .Pages(1).Enabled = True            ' Enable
   Page2.
56.          .Pages(0).Enabled = False          ' Disable
   Page1.
57.          End With
58.          CommandButton1.Enabled = True        ' Enable
   Back button.
59.
60.          Case 1                                ' If Page2
   is active...
61.          With MultiPage1
62.          .Value = MultiPage1.Value + 1        ' Move
   forward 1 page.
63.          .Pages(2).Enabled = True            ' Enable
   Page3.
64.          .Pages(1).Enabled = False          ' Disable
   Page2.
65.          End With
66.          CommandButton2.Caption = "Finish"    ' Change
   Next button to Finish.
67.
68.          Case 2                                ' If Page3
   is active...
69.          MsgBox "Finished!"                  ' User is
   Finished.
70.          Unload Me                            ' Unload
   the UserForm.
71.      End Select
72.
73. End Sub
74. Dans le menu exécuter , cliquez sur Exécuter Sub/UserForm.

```

Lorsque vous cliquez sur **Suivant >**, Page2 est activé et le **< retour** bouton devient disponible. Lorsque vous cliquez sur **Suivant >** une seconde fois, Page3 est activé et la légende **CommandButton2** devient « Terminer ».

## Contrôle de barre de défilement

Vous pouvez utiliser un contrôle de barre de défilement lorsque vous souhaitez modifier la valeur affichée par un autre contrôle, tel qu'un contrôle d'étiquette .

**Comment faire pour modifier un contrôle Label qui est basé sur la valeur d'un contrôle de barre de défilement**

Pour modifier la propriété Caption d'un contrôle Label à la valeur actuelle de la propriété Value d'un contrôle de barre de défilement , procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
4. Ajoutez un contrôle de barre de défilement à l'objet UserForm.
5. Ajoutez un contrôle Label à l'objet UserForm.
6. Double-cliquez sur le contrôle de **barre de défilement** pour ouvrir la fenêtre de Code pour le contrôle de barre de défilement .
7. Dans la fenêtre Code, tapez le code suivant pour l'événement de Modification de ScrollBar1:
8. Private Sub ScrollBar1\_Change()  
9.  
10.       Label1.Caption = ScrollBar1.Value  
11.  
12.       End Sub
12. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

Lorsque vous faites défiler à l'aide du contrôle de barre de défilement , Label1 est mis à jour avec la valeur actuelle du contrôle de barre de défilement .

## **SpinButton, contrôle**

Un contrôle SpinButton , comme un contrôle de barre de défilement , est fréquemment utilisé pour incrémenter ou décrémenter la valeur d'un autre contrôle, tel qu'un contrôle d'étiquette . La propriété SmallChange détermine de combien la valeur d'un contrôle SpinButton change lorsqu'il est sélectionné.

### **Comment faire pour ajouter un contrôle SpinButton qui incrémente ou décrémente une date qui est stocké dans un contrôle TextBox**

Pour ajouter un contrôle SpinButton qui incrémente ou décrémente une date qui est stocké dans un contrôle de zone de texte , procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
4. Ajouter un contrôle SpinButton pour l'objet UserForm.
5. Ajoutez un contrôle TextBox à l'objet UserForm.
6. Double-cliquez sur le contrôle **SpinButton** pour ouvrir la fenêtre Code pour le contrôle SpinButton .
7. Dans la fenêtre Code, tapez le code suivant pour l'événement SpinUp de SpinButton1 :
8. Private Sub SpinButton1\_SpinUp()  
9.  
10.        TextBox1.Text = DateValue(TextBox1.Text) + 1  
11.  
12. End Sub
13. Dans la fenêtre Code, tapez le code suivant pour l'événement SpinDown décrémente la SpinButton1 :
14. Private Sub SpinButton1\_SpinDown()  
15.  
16.        TextBox1.Text = DateValue(TextBox1.Text) - 1  
17.  
18. End Sub
19. Dans la fenêtre Code, tapez le code suivant pour l'événement Initialize de l'objet UserForm :
20. Private Sub UserForm\_Initialize()  
21.  
22.        TextBox1.Text = Date  
23.  
24. End Sub
25. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

Lorsque l'objet UserForm s'affiche, la date du jour s'affiche dans TextBox1. Lorsque vous cliquez sur le contrôle SpinButton , la date est incrémentée ou décrémentée par un jour.

Dans cet exemple, si vous modifiez la propriété SmallChange de SpinButton1, vous n'affectez pas le nombre de jours de que l'entrée dans TextBox1 est



modifiée par lorsque vous cliquez sur SpinButton1. Le nombre de jours est déterminé uniquement par la procédure que vous avez associé à l'événement SpinUp et l'événement SpinDown décrémente de SpinButton1.

## Contrôle RefEdit

Le contrôle RefEdit imite le comportement des boîtes de référence qui sont intégrées dans Excel. Vous pouvez utiliser la propriété Value pour obtenir l'adresse de la cellule en cours qui sont stockés dans un contrôle RefEdit .

### Comment faire pour remplir une plage de cellules en fonction de la plage que vous sélectionnez à l'aide du contrôle RefEdit

Pour utiliser le contrôle RefEdit pour remplir les cellules, procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
4. Ajoutez un contrôle RefEdit pour l'objet UserForm.
5. Ajoutez un contrôle CommandButton sur l'objet UserForm.
6. Double-cliquez sur le contrôle **CommandButton** pour ouvrir la fenêtre de Code du contrôle CommandButton .
7. Dans la fenêtre Code, tapez le code suivant pour l'événement Click du contrôle CommandButton1 :

```
8. Private Sub CommandButton1_Click()  
9.  
10.     Dim MyRange As String  
11.     MyRange = RefEdit1.Value  
12.     Range(MyRange).Value = "test"  
13.     Unload Me  
14.  
     End Sub
```

15. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

Le UserForm apparaît.

16. Cliquez sur le bouton dans le contrôle RefEdit .

Notez que le UserForm est réduit.

17. Sélectionnez une plage de cellules comme a1 : a5, puis cliquez sur le bouton dans le contrôle RefEdit pour développer l'objet UserForm.

18. Cliquez sur le **contrôle CommandButton1**.

L'objet UserForm se ferme et les cellules que vous avez sélectionnées contiennent à présent le mot « test ».

## Contrôle de l'image

Il est l'objectif du contrôle Image pour afficher une image sur un formulaire utilisateur. Pour assigner une image à un contrôle Image au moment de l'exécution, utilisez la fonction LoadPicture .

### Le chargement d'une image dans un contrôle Image

Pour insérer un contrôle d'Image qui vous invite à sélectionner l'image à charger lorsque vous cliquez sur le contrôle Image , procédez comme suit :

1. Démarrez Excel et ouvrez un nouveau classeur vide.
2. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
3. Dans le menu **Insertion** , cliquez sur **UserForm** pour insérer un objet UserForm dans votre classeur.
4. Ajouter un contrôle Image sur l'objet UserForm.
5. Double-cliquez sur le contrôle **Image** pour ouvrir la fenêtre de Code du contrôle d'Image .
6. Dans la fenêtre Code, tapez le code suivant pour l'événement Click de Image1 :
7. Private Sub Image1\_Click()  
8.  
9. Dim fname As String  
10.  
11. ' Display the Open dialog box.  
12. fname = Application.GetOpenFilename(filefilter:= \_

13.                    "Bitmap Files(\*.bmp),\*.bmp", Title:="Select  
Image To Open")
- 14.
15.            ' If you did not click Cancel...
16.            If fname <> "False" Then
- 17.
18.            ' Load the bitmap into the Image control.
19.            Image1.Picture = LoadPicture(fname)
- 20.
21.            ' Refresh the UserForm.
22.            Me.Repaint
23.            End If
- 24.
25. End Sub
26. Dans le menu **exécuter** , cliquez sur **Exécuter Sub/UserForm**.

L'objet UserForm s'affiche.

27. Cliquez sur le contrôle **Image** .

Lorsque vous cliquez sur le contrôle d'Image , la boîte de dialogue **Sélectionner Image à ouvrir** s'affiche et puis, vous pouvez sélectionner un fichier bitmap à insérer dans le contrôle.

## Informations supplémentaires

### L'Explorateur d'objets

Une liste complète de toutes les propriétés et les méthodes pour une commande spécifique est disponible dans l' Explorateur d'objets. Pour trouver ces informations dans Excel, procédez comme suit :

1. Dans le menu **Outils** , pointez sur **Macro**, puis cliquez sur **Visual Basic Editor**.
2. Dans le menu **affichage** , cliquez sur **Explorateur d'objets**.
3. Dans la zone de **Texte de recherche** , tapez le nom du contrôle que vous souhaitez, puis cliquez sur **Rechercher**.