

Compléter un document Word à l'aide d'une UserForm

Tutoriel et exemple par Anacoluthé

QU'EST-CE QU'UNE USERFORM ?

EXEMPLE DE USERFORM POUR COMPLETER UN DOCUMENT

CREER UNE NOUVELLE USERFORM

PLACER DES CONTROLES

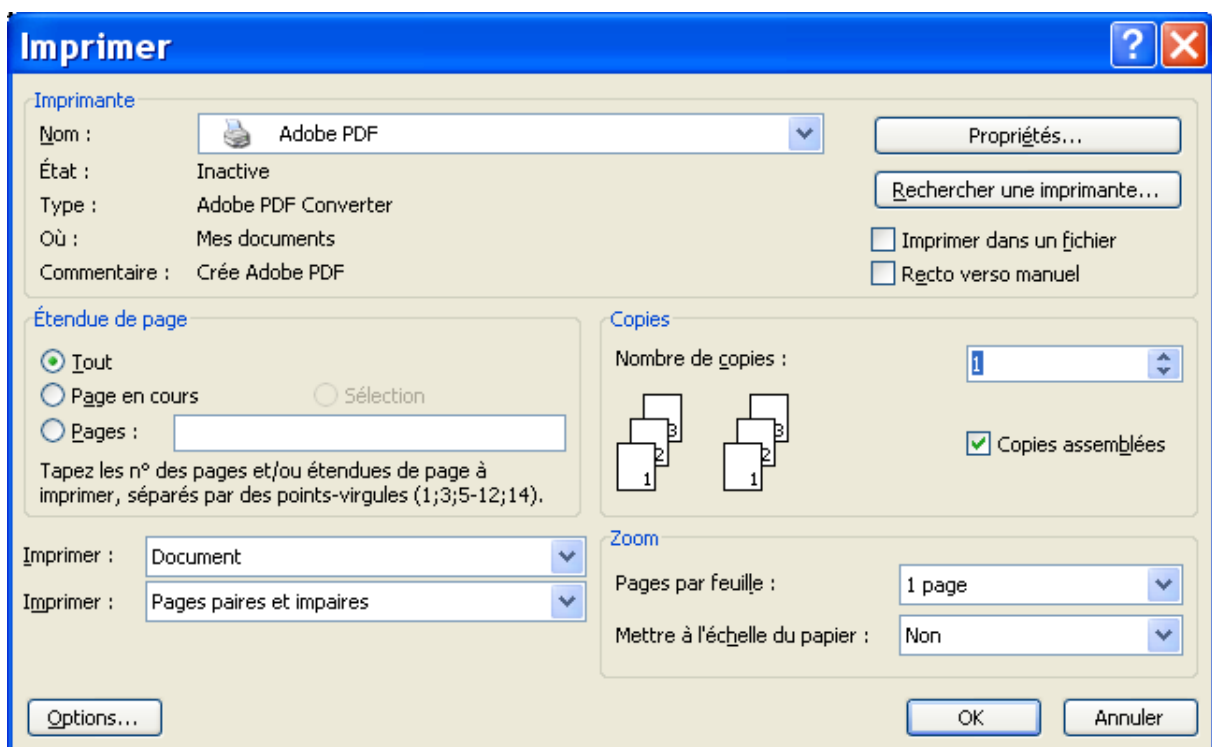
LE CODE DE LA USERFORM

- 1 AFFICHER ET FERMER LA USERFORM
- 2 INITIALISER LA USERFORM
- 3 LE CODE DES CONTROLES
- 4 LE CODE DE L'APPLICATION

LE CODE VBA FINAL DE L'EXEMPLE

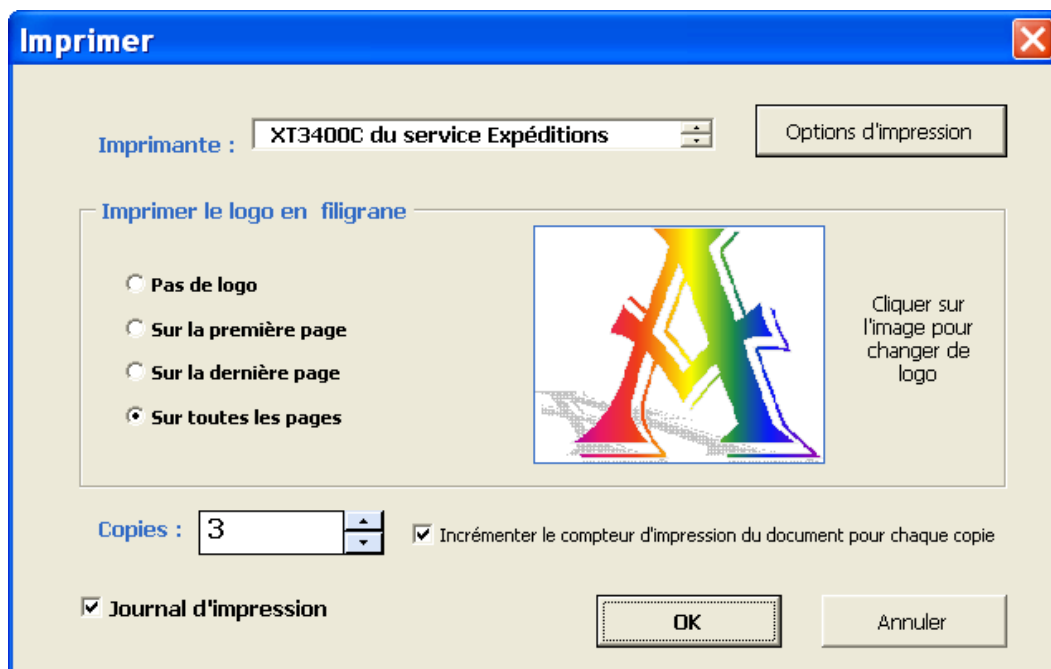
QU'EST-CE QU'UNE USERFORM ?

Une UserForm est une fenêtre ou une boîte de dialogue destinée à servir d'interface utilisateur personnalisée dans une application. Dans l'interface utilisateur Word, il existe déjà de nombreuses boîtes de dialogue. Voici par exemple la boîte de dialogue Fichier/Imprimer de Word 2003 :



On y trouve de nombreux boutons, des listes déroulantes, des cases à cocher, une zone de texte à remplir, du texte et des images d'explication... Tous ces objets s'appellent des *contrôles*. Les contrôles sont disposés sur la feuille qui sert de fenêtre à la boîte de dialogue.

Voici maintenant une autre boîte de dialogue pouvant s'ouvrir dans Word :



Inutile de chercher celle-ci dans votre Word, c'est une UserForm ! Si elle ressemble un peu à la précédente avec des contrôles similaires, elle commande à l'évidence des fonctions inhabituelles pour Word : ajouter à l'impression un filigrane non présent dans le document, gérer un compteur des copies imprimées ou tenir un journal des impressions.

Cet exemple permet de comprendre qu'une UserForm est un objet complexe qui comporte toujours les trois éléments suivants :

La feuille : c'est la fenêtre qui s'affiche.

Les contrôles : ce sont tous les objets présents dans la feuille et pouvant interagir avec l'utilisateur.

Le code : c'est ce qui décide des actions déclenchées par l'utilisateur.

La feuille est désignée par le nom de l'objet UserForm lui-même. Les contrôles sont des objets incorporés dans la UserForm : ils font partie de ses *propriétés*. Le code enfin, sans lequel une UserForm n'aurait aucune action est contenu également dans la UserForm au même titre qu'un module dans un projet.

Un mot encore sur le code. L'essentiel en est la gestion des *événements*. Une fois ouverte et affichée, la UserForm doit réagir à un certains nombres d'actions de l'utilisateur. Ainsi cliquer sur le bouton OK est un événement. Par cette action on s'attend que le code valide toutes les données entrées et réalise le propos de la boîte de dialogue comme ici lancer une impression au moment de refermer la UserForm elle-même.

EXEMPLE DE USERFORM POUR COMPLETER UN DOCUMENT

Nous choisirons un exemple simple de Userform utilisé comme formulaire pour compléter un document. Supposons un modèle de document ainsi constitué :

«Titre»,

Nous avons pris bonne note de votre question portant sur «Question»

Votre question a été transmise à «Répondeur» qui devrait pouvoir vous communiquer une réponse pour le «Date» prochain.

Recevez, «Titre», l'expression de nos sentiments dévoués

L'équipe bénévole du forum Word

Nous voulons qu'à la création d'un nouveau document basé sur ce modèle s'ouvre un formulaire facilitant la saisie rapide du texte placé entre guillemets « Titre », « Question », « Répondeur », « Date », lesquels sont en réalité des *signets* dans le document.

CREER UNE NOUVELLE USERFORM

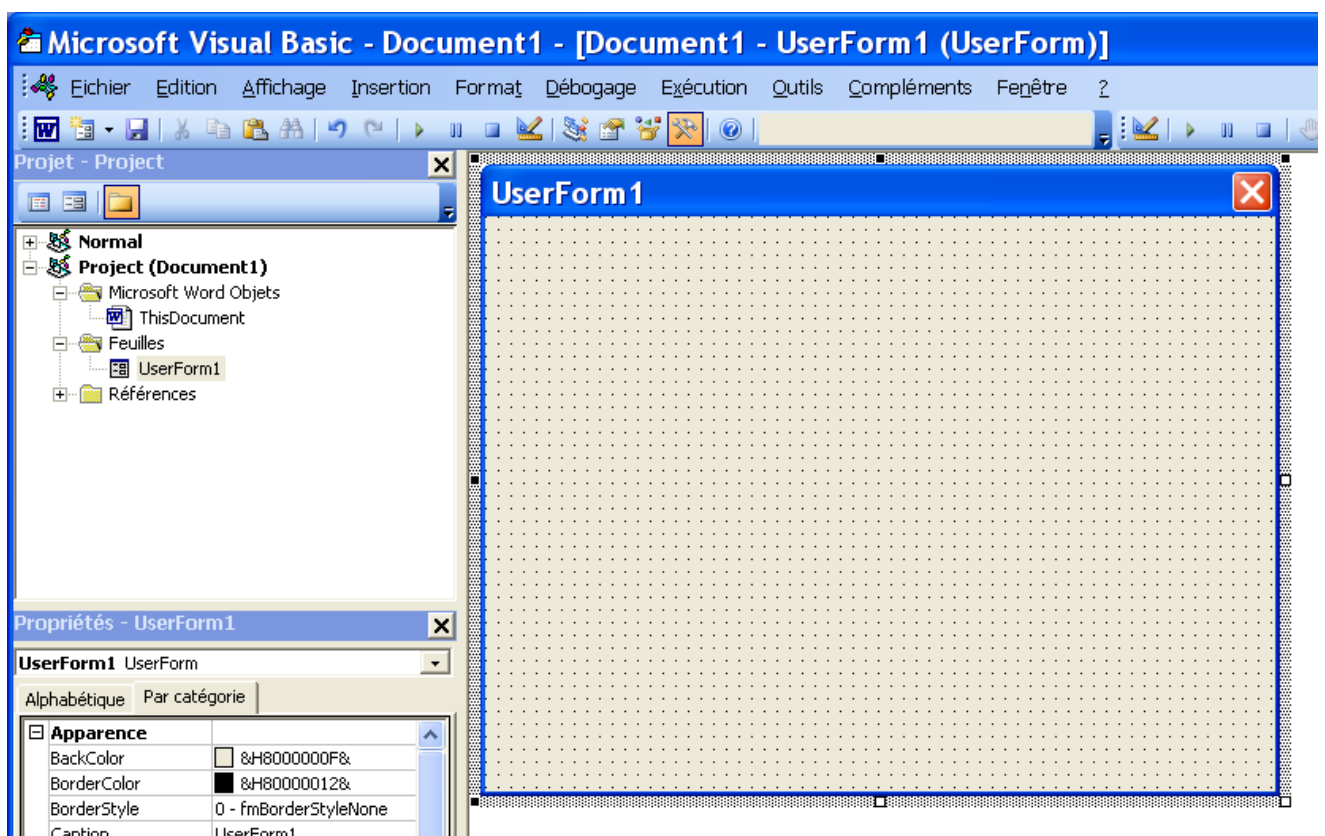
La création d'une UserForm est très simple dans VBE l'éditeur visual-basic :

Créer un document vierge Fichier / Nouveau

Ouvrir VBE par Outils / Macro / Visual Basic Editor ou **Alt+F11**

Dans la fenêtre de l'explorateur de projets, sélectionner le projet du document créé

Menu Insertion / Userform



C'est tout : le document contient désormais une UserForm. Son nom est UserForm1 par défaut. Affichons au besoin les propriétés de UserForm1 avec F4. Nous pouvons redimensionner UserForm1

et modifier ses propriétés. Nous pouvons changer son nom et changer son titre (caption). Gardons le nom UserForm1 mais changeons le titre en Formulaire.

Avant de commencer à remplir notre UserForm1, jetons un coup d'œil sur son code en cliquant sur le bouton 'Afficher le code' de la fenêtre Projet ou avec **F7** pour constater que ce code est vide pour l'instant. Revenons à l'objet UserForm avec **Maj+F7**.

PLACER DES CONTROLES

C'est un simple petit jeu de construction graphique ! Il suffit d'utiliser la boîte à outils des contrôles : Affichage / Boîte à outils, de choisir parmi les contrôles disponibles et de les dessiner sur la UserForm.



La boîte à outils contrôles

Affichons aussi la barre d'outils UserForm qui contient de quoi faciliter la mise en place des contrôles. Pour chaque contrôle, il convient dès sa création d'en définir les propriétés initiales en utilisant très largement la fenêtre Propriétés. Si vous hésitez à modifier une de ces propriétés, sélectionnez-là et voyez avec **F1** dans l'Aide VBA à quoi elle correspond. À tout moment il sera possible de redéfinir les propriétés d'un contrôle.

Notre exemple comportera les contrôles suivants :

- Une zone de texte

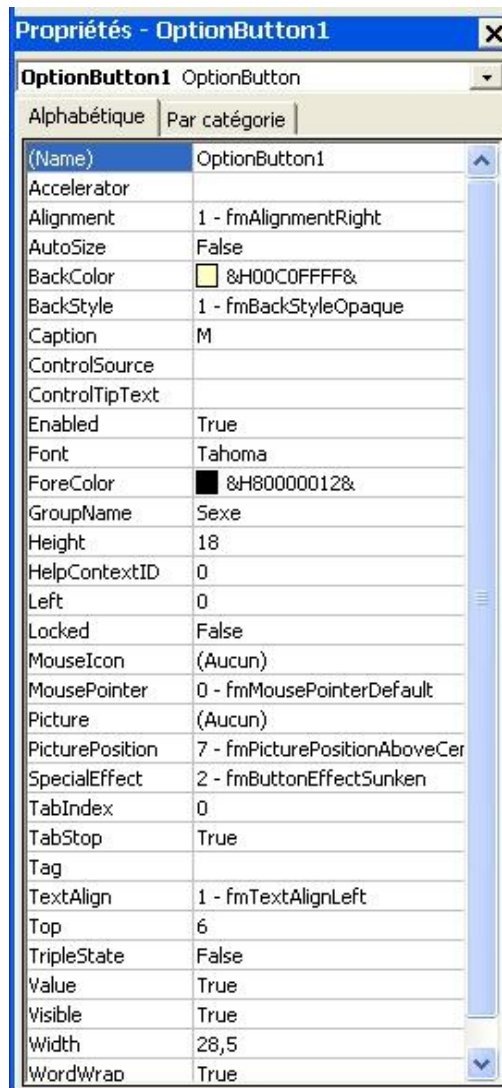
- Une liste déroulante ComboBox

- Deux radio-boutons s'excluant l'un l'autre dans un contrôle Cadre par leur propriété

- Un contrôle calendrier. Ce dernier n'est pas disponible par défaut dans la boîte à outils. Il nous faut le chercher par clic droit dans la boîte à outils / Contrôles supplémentaires / Contrôle Calendrier

- Deux boutons : l'un pour annuler l'autre pour valider (OK)

- Enfin trois Intitulés ou labels, simples contrôles passifs pour la lisibilité du formulaire.



La fenêtre des Propriétés.

Notez ici la propriété GroupName commune aux deux boutons d'option. En définissant le même nom de groupe, les deux options (M/F) s'excluent mutuellement sans avoir besoin de code.

Au total donc 11 contrôles sur notre UserForm une fois construite :

Questionneur

M F

Question

Pied de page en mode paysage

Répondeur

Circé

avr. 2006

avr. 2006

lun.	mar.	mer.	jeu.	ven.	sam.	dim.
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Date de la réponse.

Annuler OK

Il est important de bien prévoir et disposer ses contrôles avant de passer à l'étape d'écriture du code.

LE CODE DE LA USERFORM

Reste maintenant à écrire tout le code VBA pour notre UserForm ! Plus la UserForm disposera de contrôles plus le code sera compliqué. À l'aide de notre petit exemple nous en distinguerons 4 parties classiques :

AFFICHER ET FERMER LA USERFORM

La Userform doit d'abord être chargé en mémoire avec tous ses objets. C'est le rôle de la commande **Load**. Par la suite notre objet UserForm désigné par son nom pourra être affichée par sa méthode **Show** et refermée par sa méthode **Hide**. VBA accepte d'exécuter implicitement la commande **Load** si on l'oublie avant l'exécution d'un **Show**.

Dans notre exemple le chargement et l'affichage de notre UserForm est exécuté dans une macro automatique à chaque création d'un nouveau document à partir de notre modèle. Nous y ajouterons la réouverture de la UserForm pour éventuelle correction à l'aide d'un bouton-macro. Ce bouton-macro exécute la macro **Formulaire** qui se contente d'exécuter à nouveau **Show**.

INITIALISER LA USERFORM

Il s'agit dans cette phase de renseigner les diverses propriétés de la UserForm et de ses contrôles avant qu'elle ne s'ouvre. Dans notre exemple il faut notamment renseigner la liste des répondeurs dans la combobox et ajuster le contrôle calendrier à la date du jour.

Le code d'initialisation d'une UserForm peut être très complexe surtout s'il faut rechercher des données dans le document lui-même ou dans des fichiers externes. On aurait pu imaginer par

exemple que la liste des répondeurs de la combobox soit à chercher dans une feuille excel, laquelle contiendrait également pour chacun le délai moyen de réponse à appliquer au calendrier.

Toute cette phase d'initialisation peut être placée dans le code d'appel entre **Load** et **Show** ou mieux dans le code de l'événement **Initialize** de la UserForm elle-même. Cet événement est lancé à chaque Load. Notez que **Show** ne provoque pas un événement **Initialize**.

LE CODE DES CONTROLES

C'est tout le code événementiel dictant comment doit réagir chaque contrôle aux actions de l'utilisateur. Les événements possibles sont très nombreux pour chaque contrôle mais il suffit de coder uniquement ceux qui doivent déclencher une réponse. Un bon nombre de contrôles n'a pas besoin de code. Ce sont surtout les clics sur les boutons qu'il faut coder.

On accède facilement au code d'un contrôle par clic droit / Code : la fenêtre de code affiche en haut le contrôle et les divers événements qu'il est possible de coder.

Le bouton Annuler referme la UserForm par un **Hide** sans rien faire d'autre.

C'est le bouton de validation qui est le plus important. L'événement **Click** sur le bouton OK doit en effet 1) collecter les données de la UserForm 2) vérifier leur cohérence en refusant au besoin de refermer la UserForm 3) exécuter le propos de la UserForm et enfin 4) fermer la UserForm.

L'étape de vérification des données peut aussi s'effectuer au niveau d'un contrôle par exemple avec son événement **Change** chaque fois que sa valeur est modifiée.

Dans notre exemple après vérification de la zone de texte et du calendrier, la validation de la saisie doit provoquer l'insertion des données dans le document.

LE CODE DE L'APPLICATION

Ici on range le code spécialisé, les macros, les procédures et fonctions du projet pour lequel on a généralement développé précisément la UserForm. Ce code est généralement placé dans un ou plusieurs modules de façon à ne pas surcharger le code événementiel de la Userform qui y fera appel.

Dans notre exemple nous plaçons un module *Signets* chargé de traiter la mise à jour des signets dans un document. En effet le simple remplacement du texte d'un signet par un autre efface le signet en vba. La procédure **RemplirSignet**(Signet, Texte) remplace le contenu d'un Signet par Texte sans détruire ce signet, que ce dernier soit initialement vide ou non. Cette procédure est utilisée par notre UserForm pour remplir le document.

Placé dans le même projet ce code est appelé par le code événementiel des contrôles de la Userform.

LE CODE VBA FINAL DE L'EXEMPLE

Tout l'exemple est téléchargeable dans le modèle ci-joint '**Exemple Userform.dot**' qui s'utilise comme n'importe quel modèle de document. En allant dans VBE avec **Alt+F11** on y trouvera la UserForm décrite plus haut ainsi que tout le code reproduit ci après :

```
Private Sub Document_New()  
' Lancement automatique à chaque nouveau document  
Load UserForm1  
UserForm1.Show
```

```
End Sub
```

```
Public Sub Formulaire()  
    ' Réaffiche la UserForm  
    UserForm1.Show  
End Sub
```

```
Option Explicit
```

```
Private Sub UserForm_Initialize()  
    ' Lors du chargement de la UserForm : opérations d'initialisations  
  
    ' 1 - Remplissage de la liste déroulante  
    With Me.ComboBox1  
        .AddItem "Circé"  
        .AddItem "Geo"  
        .AddItem "André"  
        .AddItem "Guy"  
        .AddItem "Anacolithe"  
    End With  
  
    ' 2 - Initialisation du calendrier: aujourd'hui + 2 jours  
    Me.Calendar1.Value = Date + 2  
  
    ' 3 - Initialisation du focus  
    Me.TextBox1.SetFocus  
  
End Sub
```

```
Private Sub CommandButton1_Click()  
    ' = Bouton Annuler  
    ' Lui affecter la propriété Cancel à True  
    ' pour bénéficier de la touche Echap (ESC)  
    Me.Hide  
End Sub
```

```
Private Sub CommandButton2_Click()  
    ' = Bouton OK  
  
    ' Les données de la UserForm :  
    Dim T As String ' Titre  
    Dim Q As String ' Question  
    Dim R As String ' Répondeur  
    Dim D As String ' Date
```

```
' 1 - Récupérer les données

If Me.OptionButton1 = True Then
    T = "Cher abonné"
Else: T = "Chère abonnée"
End If

Q = Me.TextBox1.Text

R = Me.ComboBox1.Value

D = Str(Me.Calendar1.Day)
If D = " 1" Then D = " 1er"
D = D & " " & Format(Me.Calendar1.Value, "MMMM")

' 2 - Vérifier la cohérence des données

If Me.TextBox1.Value = "" Then
    MsgBox "Il manque la question !", vbExclamation, "Erreur"
    Exit Sub
End If

If Me.Calendar1.Value < Date Then
    MsgBox "On ne peut répondre avant aujourd'hui !", vbExclamation,
"Erreur"
    Me.Calendar1.Value = Date + 2
    Exit Sub
End If

' 3 - Placer les données dans le document

RemplirSignet "Titre", T
RemplirSignet "Question", Q
RemplirSignet "Répondeur", R
RemplirSignet "Date", D

ActiveDocument.Fields.Update ' maj des champs pour le renvoi sur
Titre

' 4 - Fermer la Userform

Me.Hide

End Sub
```

```
' *****
' *** Utilisation des signets ***
' *****

Public Sub RemplirSignet(S As String, T As String)
' Remplit le signet S avec le texte T sans détruire S
On Error GoTo rien
Dim Place As Long
Place = ActiveDocument.Bookmarks(S).Range.Start
ActiveDocument.Bookmarks(S).Range.Text = T
ActiveDocument.Bookmarks.Add Name:=S, _
    Range:=ActiveDocument.Range(Place, Place + Len(T))
rien:
End Sub
```