

L'utilisation d'un formulaire Word avec Outlook et Access

par Olivier Lebeau ([Heureux-oli sur DVP](#))

Date de publication : 01-03-2008

Dernière mise à jour :

Cet article va vous expliquer les mécanismes à utiliser pour l'envoi, la réception et le traitement de formulaires Word envoyés par messagerie Outlook.

- 1 - Introduction
- 2 - Le formulaire Word
 - 2-A - Création de votre formulaire
 - 2-B - L'utilisation du Formulaire
- 3 - Le traitement avec Outlook
- 4 - Le traitement des données
- 5 - Remerciements

1 - Introduction

Niveau : Débutant

Suite à de nombreuses questions sur le forum, j'ai pris l'initiative de me lancer dans un tutoriel décrivant les mécanismes à mettre en place pour l'envoi, la réception et le traitement de formulaires.

Ces formulaires seront de simples formulaires Word. Ils seront envoyés avec Outlook à une série de destinataires et seront traités à leur retour pour alimenter une base de données Access.

Le tout avec le moins d'interventions manuelles possible.

Le tutoriel va se composer de trois grandes parties :

- Le formulaire Word
- L'envoi et le traitement des réponses avec Outlook
- Le traitement des données


Les illustrations de ce tutoriel sont issues de la suite Office 2007, mais le tutoriel est entièrement utilisable sur les versions antérieures de la suite Office.

2 - Le formulaire Word

Le formulaire Word est un formulaire et non pas un UserForm. Ces deux objets sont souvent confondus à tort. Le premier est simplement un document contenant des fonctionnalités particulières, alors que le second est un objet généré par programmation.

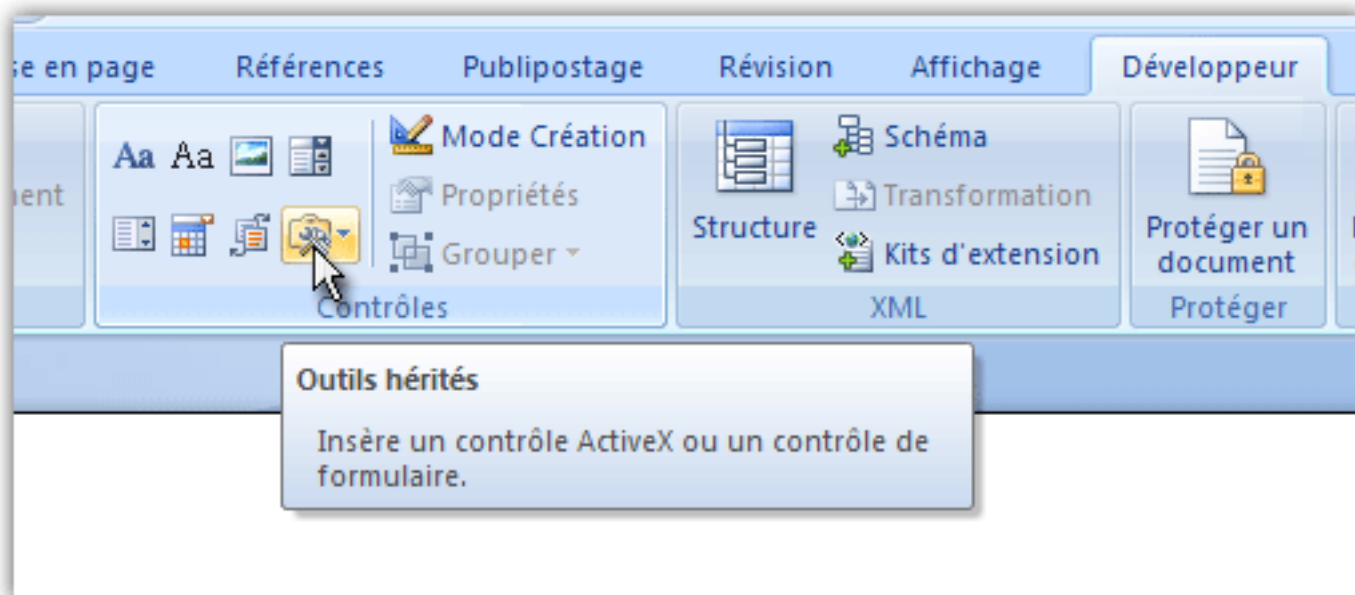
2-A - Création de votre formulaire

Pour pouvoir créer un formulaire, vous devez activer l'onglet développeur.

 Si cet onglet n'est pas disponible, vous pouvez l'activer de la manière suivante :

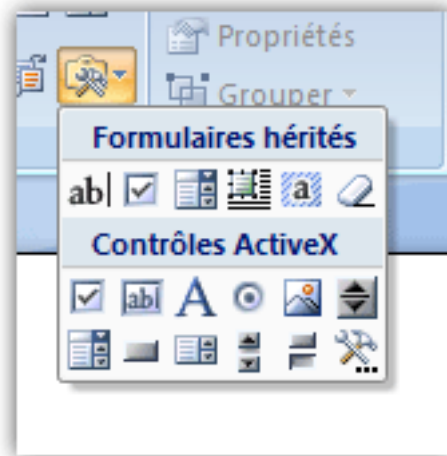
Bouton Office => Options Word => Afficher l'onglet développeur dans le Ruban.

Lorsque vous avez activé l'onglet développeur, vous avez alors accès à un groupe d'outils vous permettant d'utiliser les contrôles de formulaire. Les contrôles à utiliser sont les **Outils Hérités**. Ces outils vous assureront une plus grande compatibilité avec les versions antérieures de Word.



Les outils hérités

Nous allons utiliser les Outils Formulaires Hérités.



Formulaires hérités

Le nombre d'outils est assez restreint, mais quand même suffisant pour un formulaire de demande de renseignements.

Questionnaire vous est parvenu pour nous permettre de mieux vous connaître

vous venir en aide en répondant à quelques questions et en nous retournant

compl. ¶

¶

¶

¶

¶

Marié ou assimilé: ¶

Veuf ou séparé: ¶

Célibataire: ¶

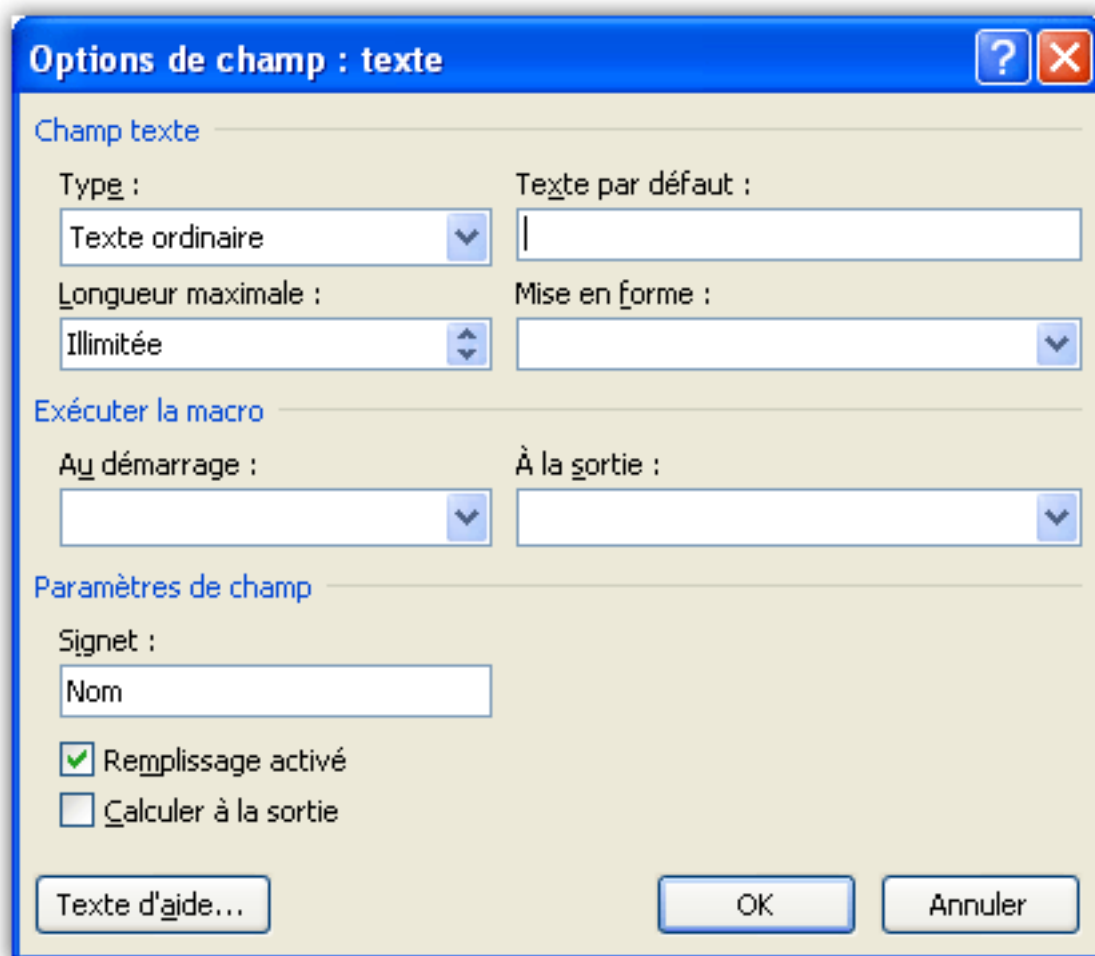
Des enfants: Oui Non ¶

Des enfants combien? ¶

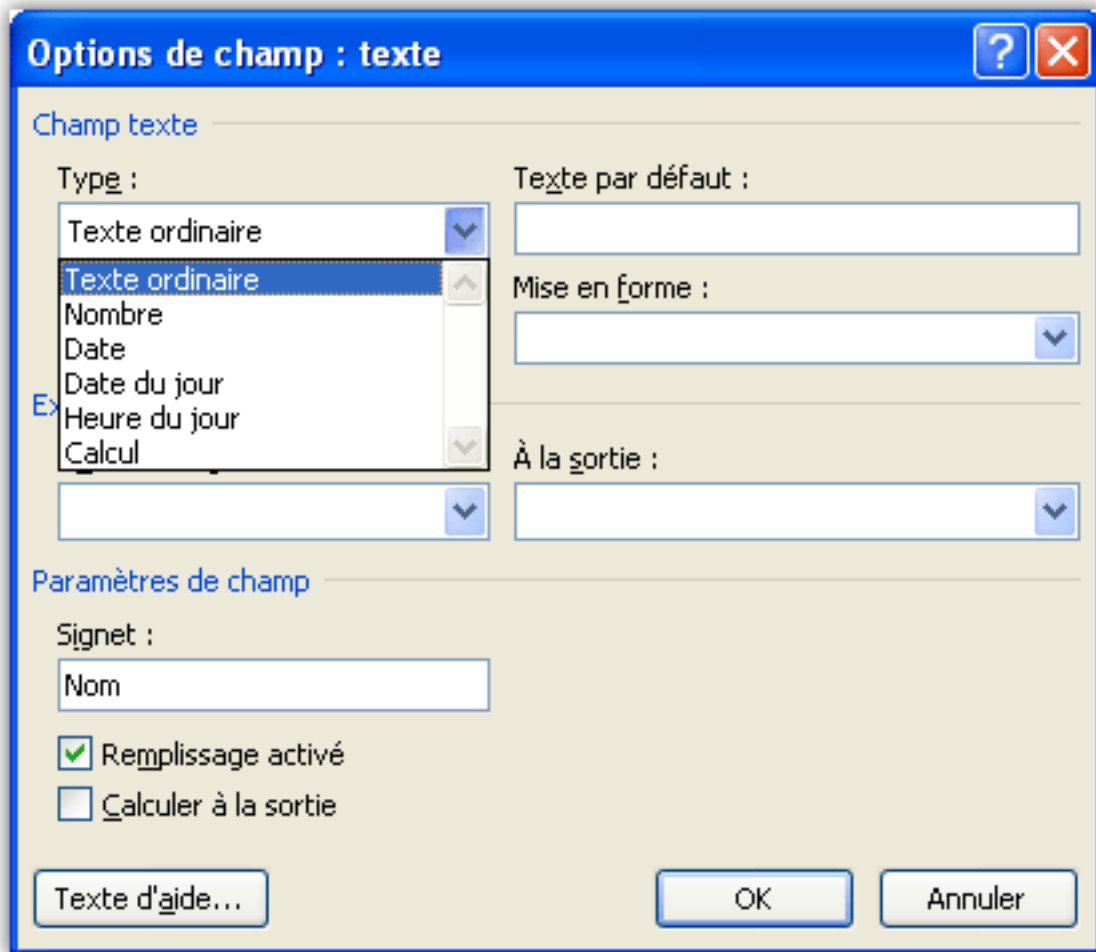
Un exemple de formulaire

Pour ce formulaire, j'ai utilisé les trois types de champs qui me semblent être les plus utiles : les zones d'édition, les cases à cocher et les zones de liste déroulante. Word offre la possibilité de personnaliser ces contrôles, on peut en modifier le nom et ajouter une macro sur un des événements associés à ce contrôle.

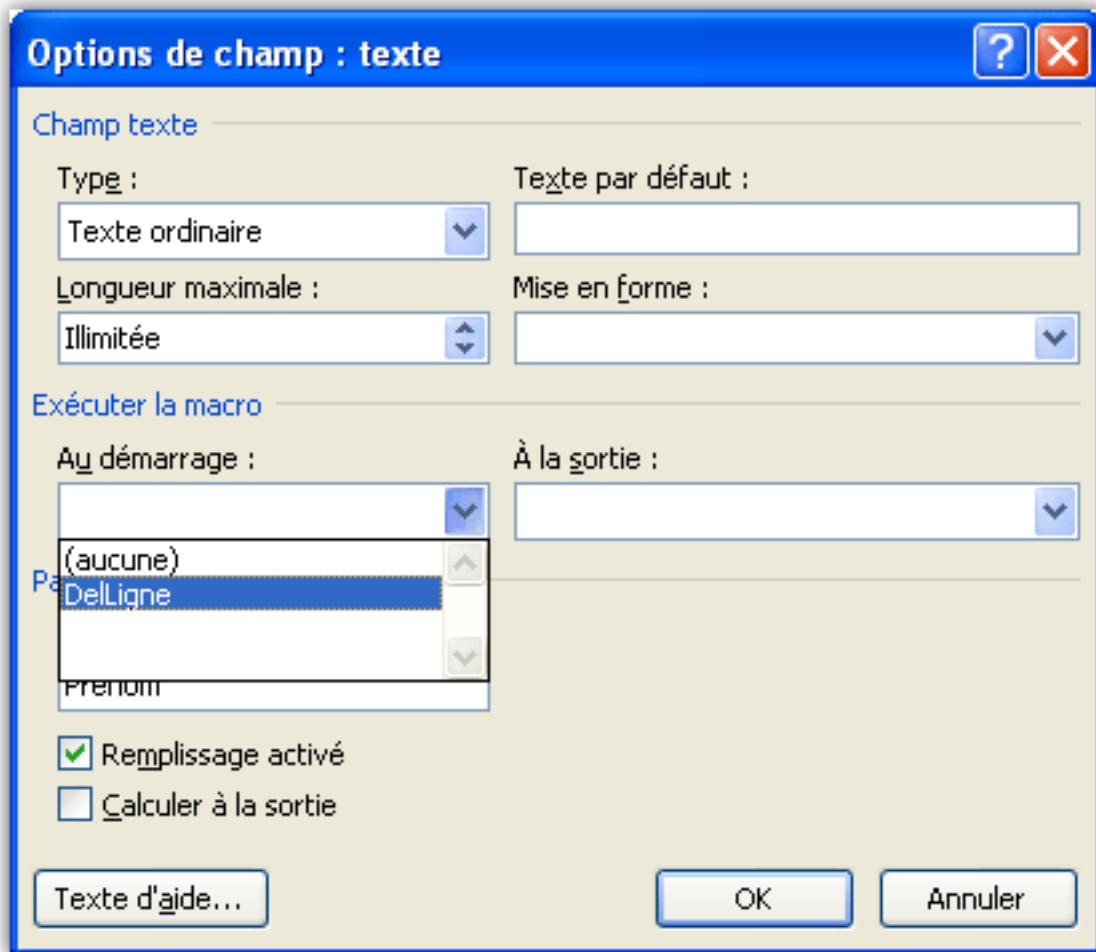
Si vous double-cliquez sur le contrôle, vous ouvrirez une boîte de dialogue contenant certaines propriétés. Je vous conseille de changer le nom du contrôle par un nom qui devrait mieux convenir à votre usage.



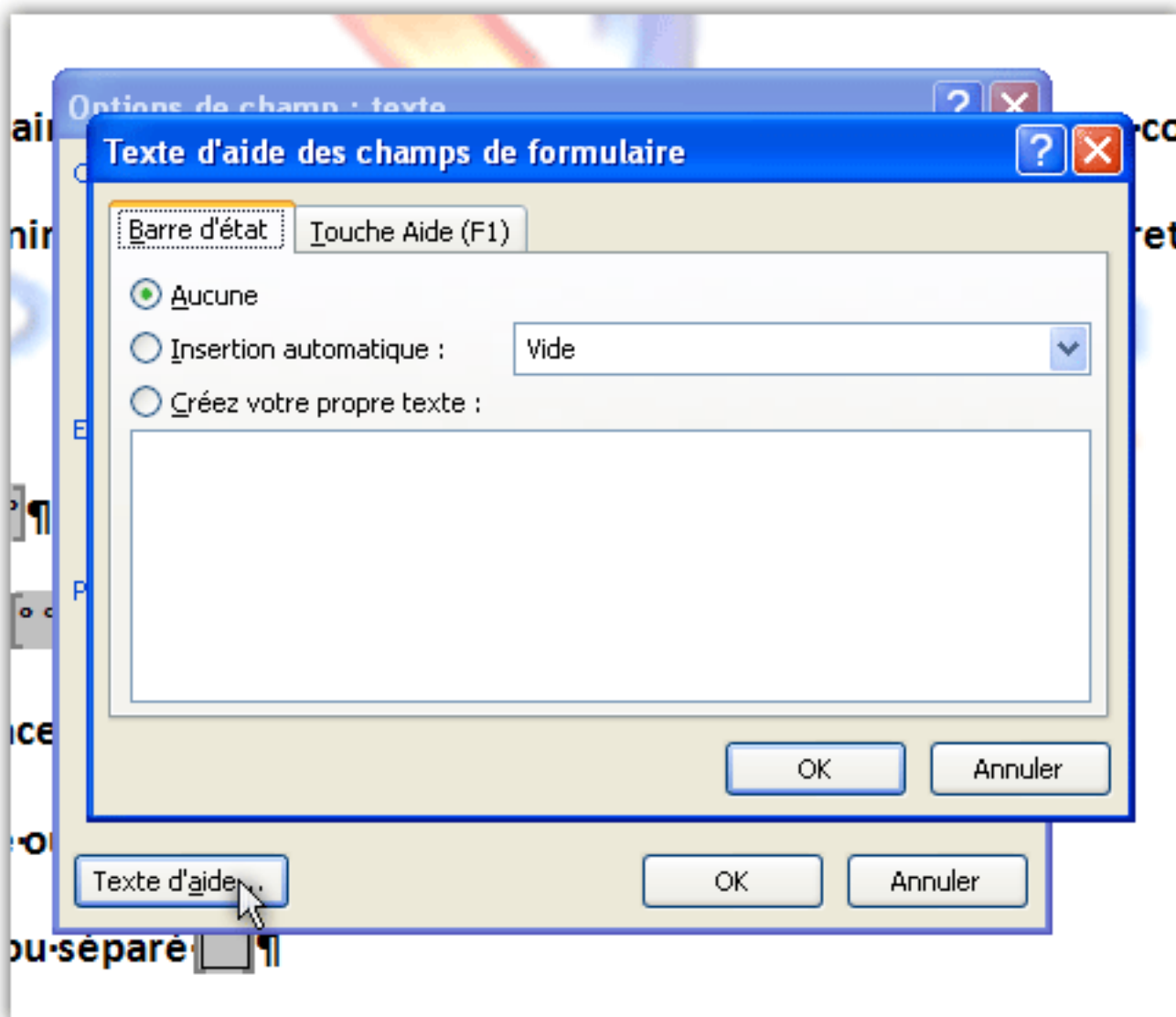
Vous pouvez intervenir sur certaines propriétés. Si vous voulez formater votre contrôle pour une date, une liste déroulante vous permet de faire ce choix.



Vous pouvez également définir si l'exécution d'une macro doit être initiée sur l'entrée ou la sortie du contrôle. La liste déroulante vous donne les macros disponibles.



Et enfin vous pouvez déterminer un texte d'aide à la saisie. Ce texte peut être affiché dans la barre d'état de l'application ou sur pression de la touche F1.



Dans notre document, nous avons l'équivalent d'un groupe d'options. Dans une UserForm, il est possible de créer un groupe d'option, mais pas dans un formulaire. Pour obtenir l'équivalent, nous utiliserons une macro qui va sur chaque mise à jour mettre la valeur des autres cases à cocher sur **False**.

Par facilité, j'ai modifié les 5 CaseÀCocher du document pour leur attribuer un nom un peu plus court : CCX où X vaut de 1 à 5. Les CCX sont réparties en deux groupes, le groupe pour l'état civil et le groupe pour les enfants. Nous aurons donc besoin de deux groupes de macros.

Les macros sont à mettre dans "ThisDocument".

```
Sub CC1()  
If ActiveDocument.FormFields("CC1").Result = True Then  
    ActiveDocument.FormFields("CC2").Result = False  
    ActiveDocument.FormFields("CC3").Result = False  
End If  
End Sub
```

Ce code est à répéter pour chaque case à cocher utilisée comme **OptionButton**.

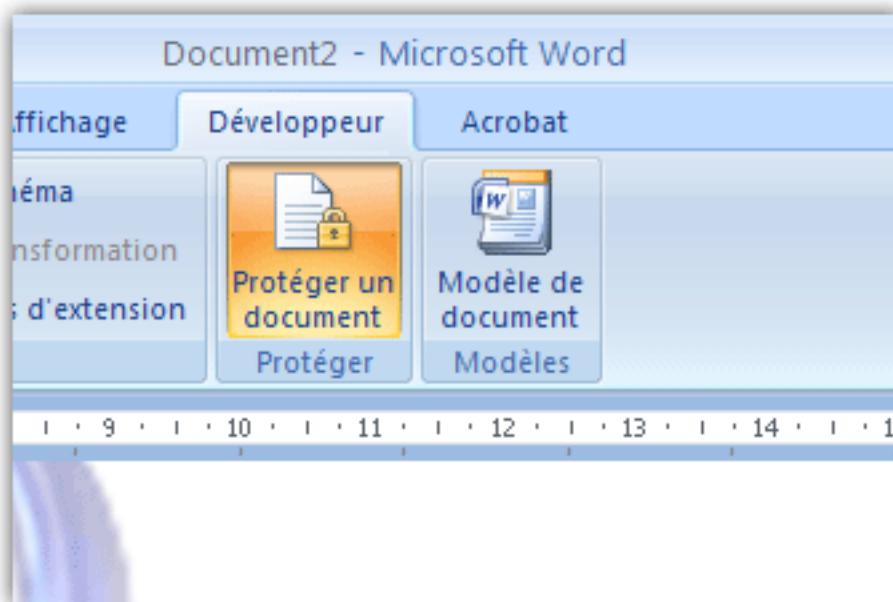
```
Sub CC2()  
If ActiveDocument.FormFields("CC2").Result = True Then  
    ActiveDocument.FormFields("CC1").Result = False  
    ActiveDocument.FormFields("CC3").Result = False  
End If  
End Sub
```

Ce sont les seules lignes de code que nous allons utiliser dans notre document.

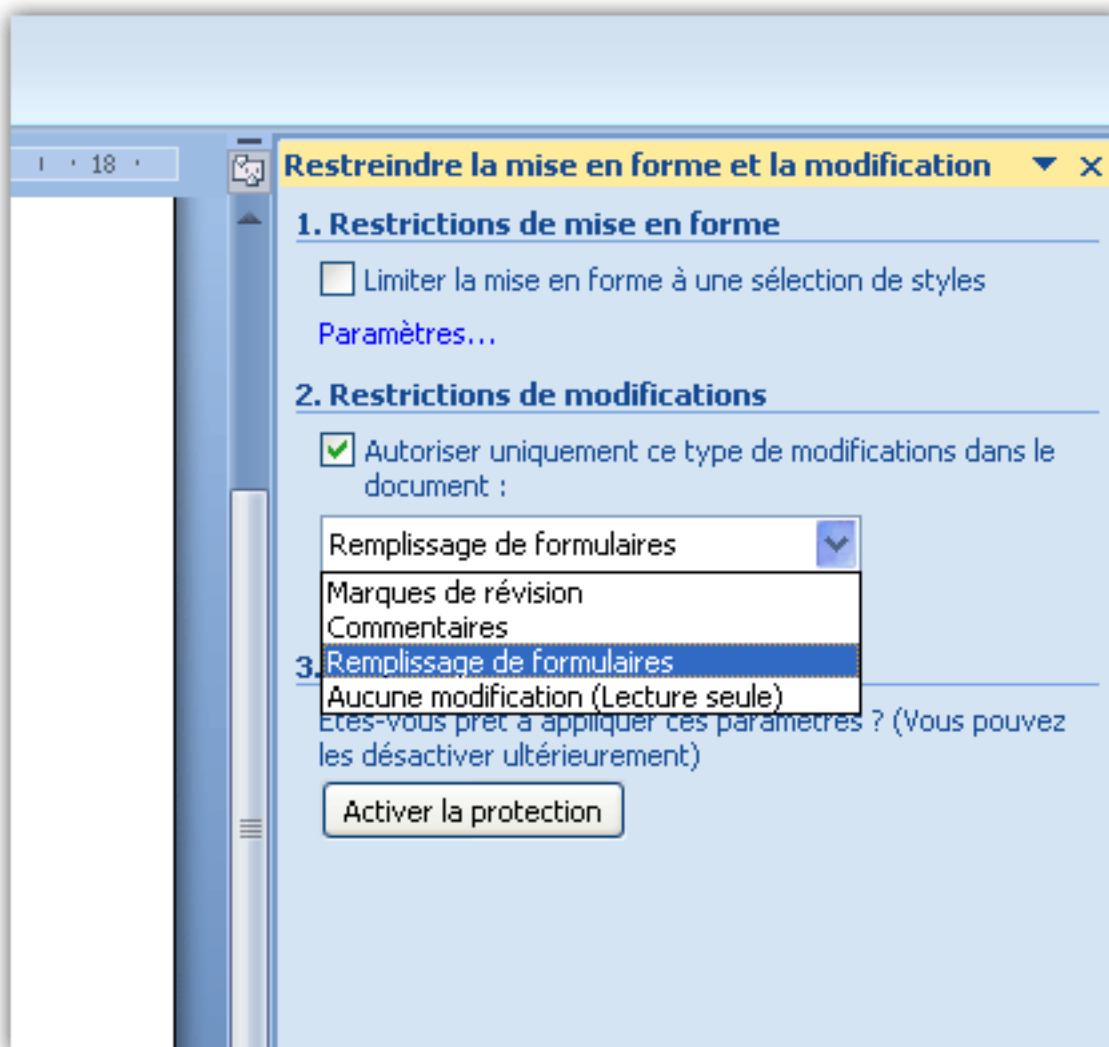
2-B - L'utilisation du Formulaire

Vous avez probablement remarqué que le formulaire ne réagit pas comme un formulaire mais comme un simple document. En effet, il n'est pas possible d'introduire des données dans les contrôles prévus à cet effet.

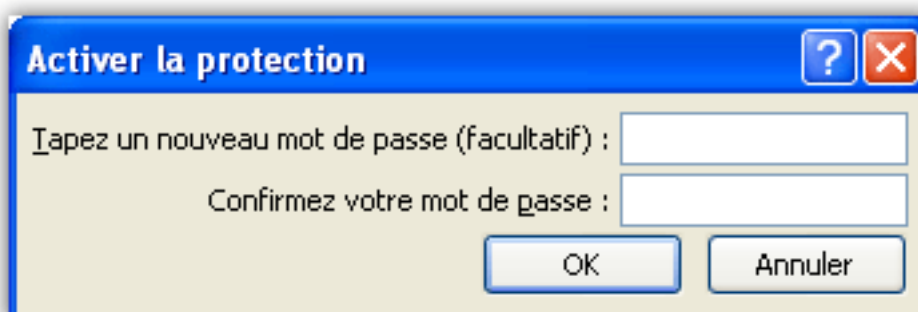
Pour pouvoir utiliser les contrôles, vous devez protéger le document. Dans le cas d'un formulaire, Word est doté d'un niveau de protection spécifique.



Si vous cliquez sur ce bouton, vous ouvrez un volet pour la protection de votre document. La protection peut avoir plusieurs niveaux, mais seule la protection formulaire nous intéresse.



Lors de l'activation de la protection, une boîte de dialogue vous propose de mettre un mot de passe sur votre document. Ce mot de passe n'est pas nécessaire, si vous laissez les cases vides, il ne vous sera pas demandé d'entrer un mot de passe lors de la déprotection de votre document.



Nous voilà prêts pour l'expédition de notre document.

3 - Le traitement avec Outlook

Pour le traitement, nous avons deux possibilités, la première est le contrôle du message lors de son entrée dans la boîte de réception, la seconde, parcourir tous les messages dans le dossier "Boîte de réception".


L'option la plus facile à mettre en #uvre est le contrôle des messages dans la "Boîte de réception" et de traiter les messages correspondant aux critères. Dans le cas qui suit, j'ai simplement testé le nom du fichier joint.


```
Sub SaveAttachement()  
Dim myFld As Folder  
Dim myNS As NameSpace  
Dim myItem As MailItem  
Dim oApp As Outlook.Application  
  
Set oApp = Outlook.Application  
Set myNS = oApp.GetNamespace("MAPI")  
Set myFld = myNS.GetDefaultFolder(olFolderInbox)  
  
For Each myItem In myFld.Items  
If myItem.Attachments.Count = 1 Then  
    'ces deux lignes servent simplement de contrôle dans la fenêtre exécution  
    Debug.Print myItem.Attachments.Count  
    Debug.Print myItem.Attachments.Item(1).FileName  
    'Test sur le nom de fichier joint  
If myItem.Attachments.Item(1).FileName = "sondage.docm" Then  
    myItem.Attachments.Item(1).SaveAsFile "C:\Temp\sondage\" & myItem.SenderName & ".docm"  
End If  
End If  
Next myItem  
set oApp = Nothing  
Set myNS = Nothing  
Set myFld = Nothing  
End Sub
```

Il est possible d'agir dès l'arrivée d'un nouveau message. Le code qui suit va, lors de l'arrivée d'un nouveau message vérifier si ce message contient bien un fichier joint et vérifier le nom de celui-ci, si ce nom correspond au nom recherché, le fichier joint sera sauvegardé dans le répertoire "C:\temp\Sondage\". Ce code diffère peu du précédent.

```
Private Sub Application_NewMail()  
Dim myFld As Folder  
Dim myDestFolder As Folder  
Dim myNS As NameSpace  
Dim myItem As MailItem  
Dim oApp As Outlook.Application  
  
Set oApp = Outlook.Application  
Set myNS = oApp.GetNamespace("MAPI")  
Set myFld = myNS.GetDefaultFolder(olFolderInbox)  
Set myDestFolder = myFld.Folders("Temp")  
  
For Each myItem In myFld.Items  
If myItem.Attachments.Count = 1 Then  
  
If myItem.Attachments.Item(1).FileName = "sondage.docm" Then  
    myItem.Attachments.Item(1).SaveAsFile "C:\Temp\sondage\" & myItem.SenderName & ".docm"  
    myItem.Move myDestFolder  
  
End If  
End If  
Next myItem
```

```
Set oApp = Nothing
Set myNS = Nothing
Set myFld = Nothing
Set myDestFolder = Nothing
End Sub
```

 Si l'expéditeur change le nom du fichier, l'automatisme ne pourra plus être utilisé.

 Les codes ci-dessus ont été testés avec Outlook 2007.

Petit ajout de dernière minute, le code ci-dessous fonctionne mieux avec Outlook 2003

Avec Outlook 2003

```
Private Sub Application_NewMail()
On Error Resume Next

Dim myFld As MAPIFolder
Dim myDestFolder As MAPIFolder
Dim myNS As NameSpace
Dim myItem As MailItem
Dim oApp As Outlook.Application

Set oApp = Outlook.Application
Set myNS = oApp.GetNamespace("MAPI")
Set myFld = myNS.GetDefaultFolder(olFolderInbox)
Set myDestFolder = myFld.Folders("Temp")

Debug.Print myFld.Items.Count
For Each myItem In myFld.Items
    If myItem.Attachments.Count = 1 Then
Debug.Print myItem.Attachments.Count & " " & myItem.Attachments.Item(1).FileName
        If myItem.Attachments.Item(1).FileName = "sondage.doc" Then
            myItem.Attachments.Item(1).SaveAsFile "C:\Temp\sondage\" & myItem.SenderName & ".doc"
            myItem.Move myDestFolder
        End If
    End If
Next myItem
Set oApp = Nothing
Set myNS = Nothing
Set myFld = Nothing
Set myDestFolder = Nothing
End Sub
```

4 - Le traitement des données

Après le retour des fichiers, l'étape suivante sera le traitement des fichiers. Ce traitement, consistera à l'extraction des données et le déplacement des fichiers dans un autre répertoire pour éviter qu'ils soient traités plusieurs fois de suite.

Pour cette partie nous allons, pour récupérer les fichiers d'un répertoire, faire appel à la bibliothèque "**File System Object**", si vous souhaitez approfondir un peu plus cette bibliothèque, je vous conseille de consulter le tutoriel fait par Christophe Warin.

La manipulation des fichiers

Ce tutoriel n'est pas le seul que vous pouvez consulter, il y a également du même auteur un tutoriel sur le DAO. **Le DAO**

Pour récupérer les fichiers et en extraire les données, nous allons parcourir le répertoire dans lequel nous les avons copiés. Pour chaque fichier, nous allons utiliser une fonction qui va simplement ouvrir le fichier, faire une boucle sur les données et extraire les données pour les envoyer dans une table. Par souci de simplicité, nous allons faire ce traitement au départ d'Access.

Le code qui suit est à mettre dans un module.


Routine pour les fichiers du répertoire

```
Sub ReucpFichier()  
Dim oFSO As New FileSystemObject  
Dim oFil As File  
Dim oFold As Folder  
  
Set oFold = oFSO.GetFolder("C:\Temp\sondage\  
  
For Each oFil In oFold.Files  
If Right(oFil.Name, 4) = "docm" Then  
Extract (oFil.Name)  
oFil.Move "C:\temp\sondage\done\  
End If  
Next oFil  
  
Set oFSO = Nothing  
  
End Sub
```

Nous supposons que ce répertoire ne contient pas d'autres fichiers que ceux issus du sondage. Comme le nom du fichier est modifié lors de sa sauvegarde, nous ne pouvons faire de test sur le nom complet du fichier, nous ferons donc simplement un test sur l'extension du fichier et nous assurer que nous ne traiterons que les fichiers Word. Pour ne pas traiter deux fois les mêmes fichiers, après traitement, les fichiers seront déplacés dans un autre répertoire.

Test sur le type de fichier

```
Right(oFil.Name, 4) = "docm"
```

 *docm est l'extension que j'ai utilisée, elle correspond aux fichiers de la version 2007, pour les versions antérieures, il sera question de doc.*

De même nous utilisons une constante pour le répertoire, si ce répertoire devait être différent, il suffit de simplement changer ceci :

```
Set oFold = oFSO.GetFolder("C:\Temp\sondage\")
```

Pour le traitement du fichier, nous utilisons une fonction : **Extract**. Cette fonction va créer un objet application Word et ouvrir le document passé en paramètre. Ensuite, avec une boucle, nous allons parcourir chaque champ du document pour en extraire le contenu et l'injecter dans une table à l'aide du DAO.

La fonction Extract

```
Public Function Extract(oFN As String)
On Error Resume Next
Dim wApp As New Word.Application
Dim oDoc As Word.Document
Dim rs As DAO.Recordset
Dim sql As String
Dim i As Integer, j As Integer

Set oDoc = wApp.Documents.Open(FileName:="C:\temp\sondage\" & oFN)

oDoc.Unprotect


i = oDoc.FormFields.Count
Debug.Print i
Set rs = CurrentDb.OpenRecordset("tbl_sondage", dbOpenTable)
'édition du jeu d'enregistrement par ajout
rs.AddNew
For j = 1 To i
    rs.Fields(j) = oDoc.FormFields(j).Result


Next j
rs.Fields(j + 1) = oFN
rs.Update
oDoc.Close SaveChanges:=False
rs.Close
Set rs = Nothing
Set oDoc = Nothing
wApp.Quit

End Function
```

Pour ajouter les données dans la table, il est en général préférable d'utiliser le nom des champs en lieu et place de leur index. Je n'ai pas voulu passer par le nom des champs, mais par leur index, cette méthode permet de traiter plusieurs fichiers différents en même temps.

Si dans les propriétés du document ou dans une variable vous stockez une information sur le document, vous pouvez utiliser cette information pour identifier l'enregistrement et pourvoir extraire par une requête les enregistrements relatifs à un même formulaire.

 *Vérifiez que le nombre de champs de votre table soit au moins égal au nombre de champs de votre formulaire.*

 *Vous devez ajouter dans votre projet les références à Microsoft Word XX.X et à Microsoft Scripting Runtime.*

5 - Remerciements

Mes premiers remerciements vont à Christophe. Sans ses questions, je ne me serais pas penché sur ces solutions. Merci aussi à Starec et Jeannot45

Merci à toute l'équipe Office qui forme une équipe hétérogène mais soudée.

