

# Les variables de document

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

Date de publication :

Dernière mise à jour :

Dans un document Word, il est possible de stocker des valeurs à un autre endroit que dans le texte. Il existe plusieurs possibilités. Nous allons parcourir quelques solutions.

1 - Introduction.....	3
2 - DocumentProperties.....	4
3 - CustomProperties.....	11
4 - Variables.....	14
5 - Les propriétés des documents avec DSO.....	15
5-A - SummaryProperties.....	15
5-B - CustomProperties.....	15
6 - Remerciements.....	16

## 1 - Introduction

**Word** offre la possibilité de stocker des informations qui peuvent être visibles ou cachées. Nous allons parcourir trois moyens de stocker des données à un autre emplacement que dans le texte. Mais si ces données sont cachées, il est tout à fait possible de les afficher dans le document.

Les trois moyens que nous allons parcourir sont :

- Les DocumentProperties
- Les CustomProperties
- Les Variables

Les illustrations sont issues de la version 2007 de Word, mais les méthodes décrites fonctionnent également pour les versions antérieures.

Si vous décidez de faire une gestion numérotée en Word, les propriétés et variables vont vous aider.




*Si vous voulez faire une gestion numérotée plus efficace, utilisez Access, vous obtiendrez de meilleurs résultats.*

## 2 - DocumentProperties

La première solution est l'utilisation des propriétés de document. Le but principal de ces propriétés est de fournir des renseignements sur l'auteur, la taille du document, le titre du document, ...

Certaines de ces propriétés peuvent être directement utilisées dans l'**Explorateur Windows**. En modifiant l'affichage, on peut obtenir l'affichage de propriétés accessibles en **VBA**. Nous pouvons utiliser ces propriétés pour faciliter la recherche des documents ou le tri, mais nous pouvons les détourner de leur but initial pour y stocker des données.

 *Les données que vous pouvez stocker dans les propriétés des documents ne peuvent pas être sensibles puisqu'elles sont facilement accessibles.*

On accède aux propriétés de document à l'aide de la collection **BuiltInDocumentProperties**. Cette collection appartient à l'objet **Document**

```
ActiveDocument.BuiltInDocumentProperties
```

Ce code n'est pas suffisant, pour l'utiliser correctement, il faut lui donner un argument qui correspond à la propriété désirée.

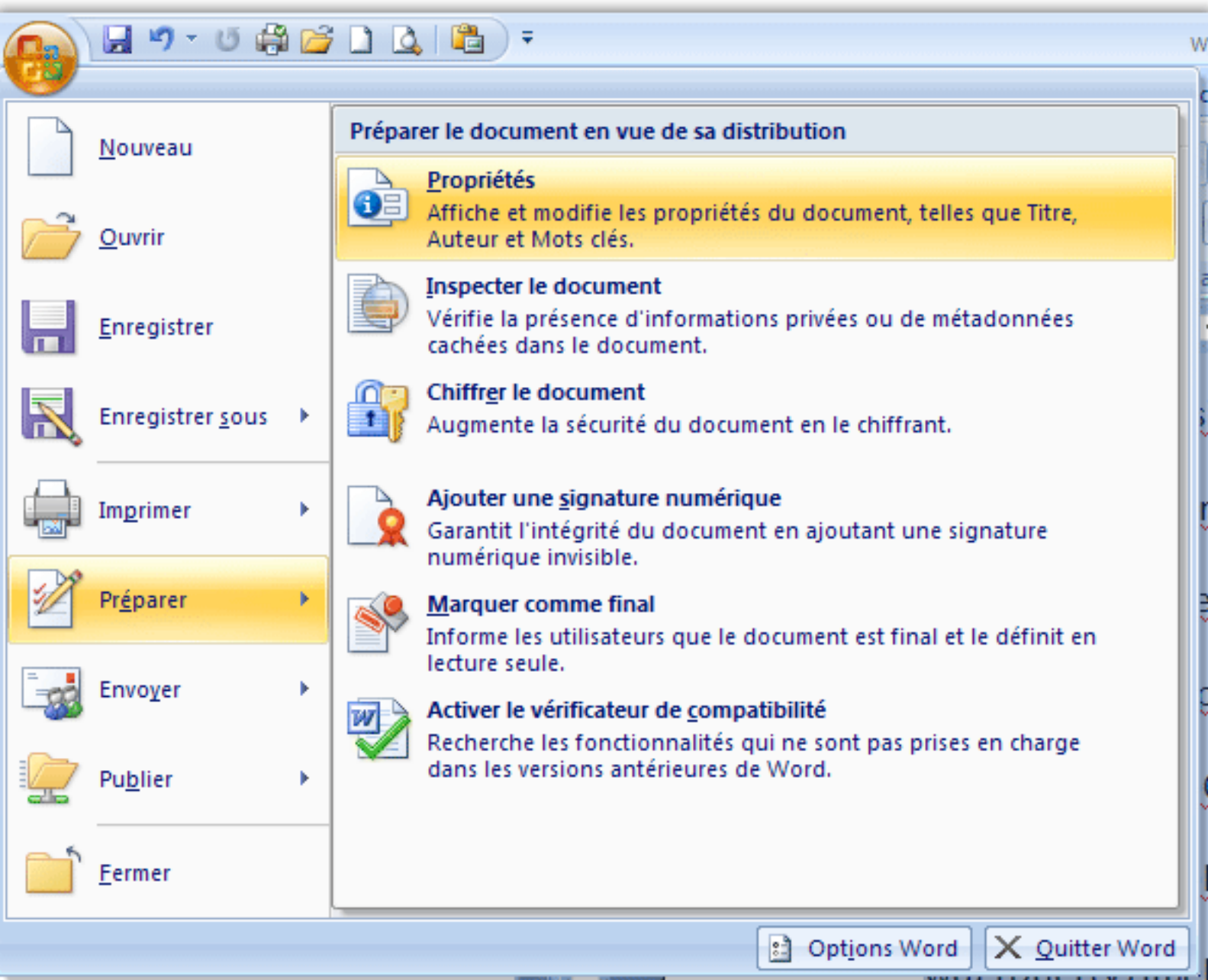
Propriété	Description
wdPropertyAppName	Nom d'application.
wdPropertyAuthor	Auteur.
wdPropertyBytes	Nombre d'octets.
wdPropertyCategory	Catégorie.
wdPropertyCharacters	Nombre de caractères.
wdPropertyCharsWSpaces	Nombre de caractères, espaces compris.
wdPropertyComments	Commentaires.
wdPropertyCompany	Société.
wdPropertyKeywords	Mots clé.
wdPropertyLastAuthor	Dernier auteur.
wdPropertyLines	Nombre de lignes.
wdPropertyManager	Gestionnaire.
wdPropertyNotes	Notes.
wdPropertyPages	Nombre de pages.
wdPropertyParas	Nombre de paragraphes.
wdPropertyRevision	Nombre de révisions.
wdPropertySecurity	Paramètre de sécurité.
wdPropertySubject	Objet du document.
wdPropertyTemplate	Nom du modèle.
wdPropertyTimeCreated	Heure de création.
wdPropertyTimeLastPrinted	Heure de la dernière impression.
wdPropertyTimeLastSaved	Heure du dernier enregistrement.
wdPropertyTitle	Titre.
wdPropertyVBATotalEdit	Nombre de modifications au projet VBA.
wdPropertyWords	Nombre de mots.

Cette liste exhaustive nous apprend que certaines propriétés ne peuvent être utilisées pour stocker des données, Word les utilise déjà.

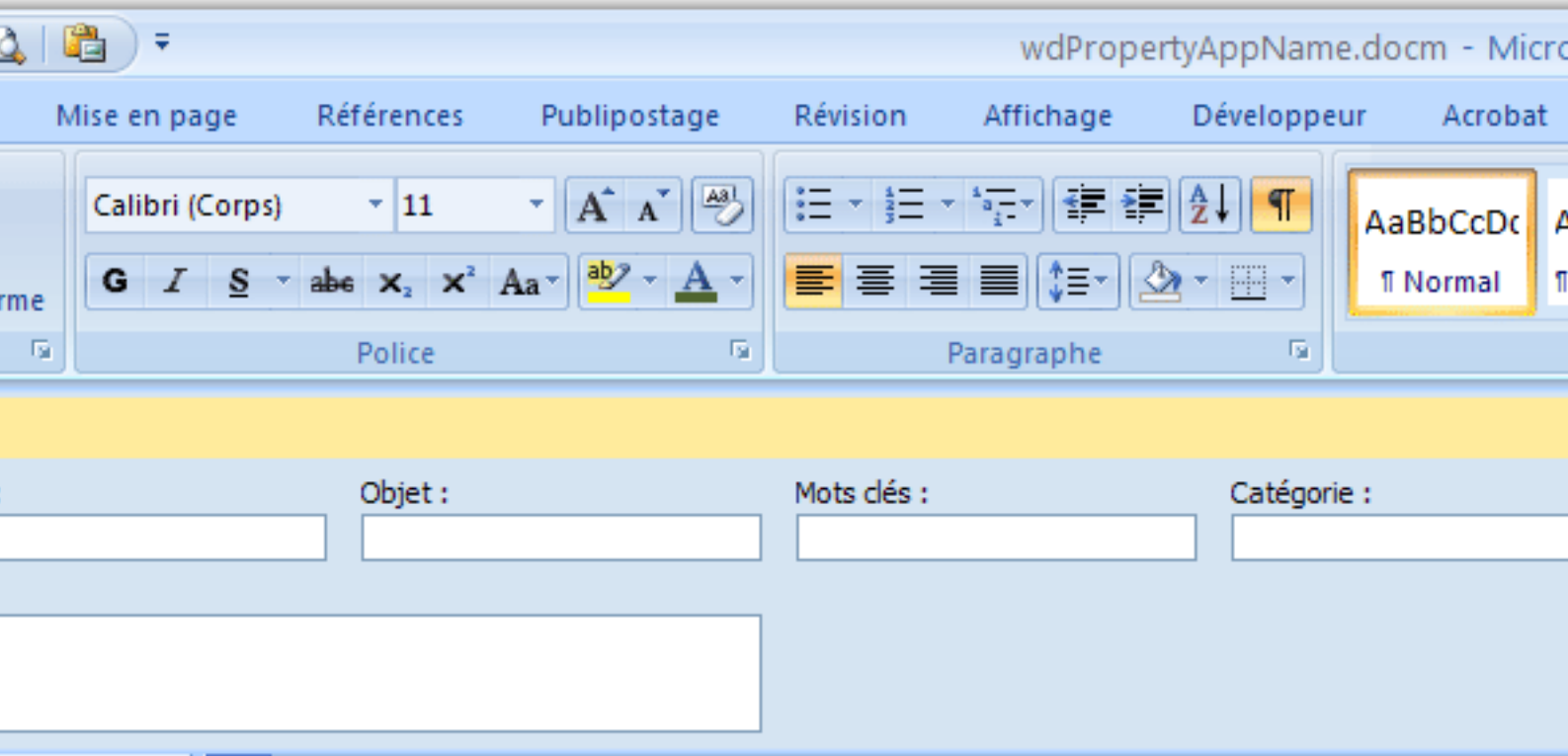
Les propriétés qui me semblent utilisables sont :

- wdPropertyAuthor
- wdPropertyComments
- wdPropertyCompany
- wdPropertyLastAuthor
- wdPropertyNotes
- wdPropertySubject
- wdPropertyTitle

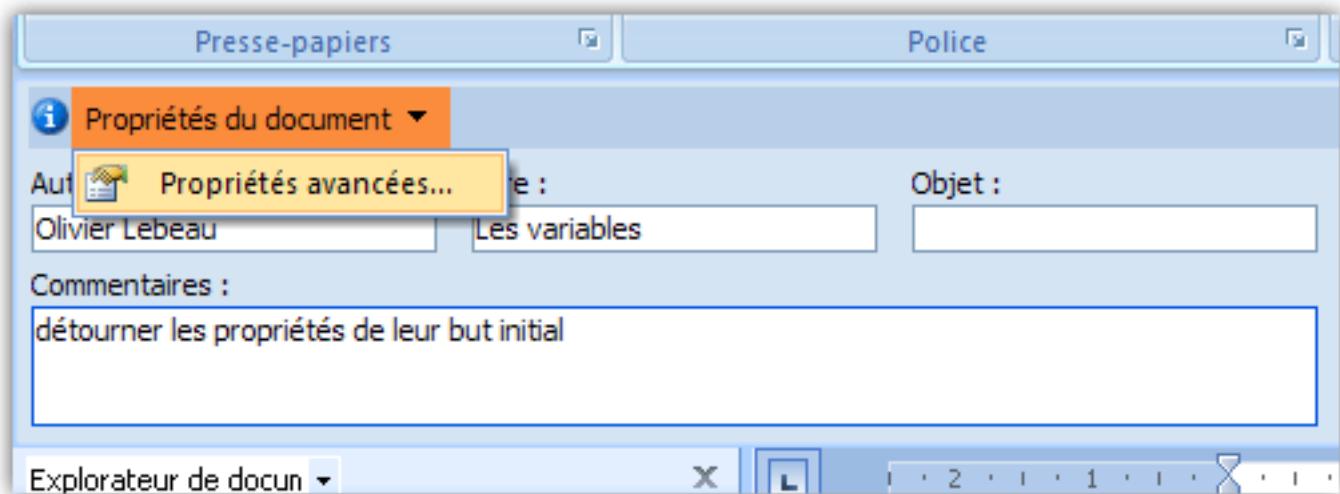
Où trouver dans l'interface graphique ces propriétés.



La modification des options du document se fait à l'aide de **Textbox**



Le simple fait d'écrire dans les **Textbox** modifie la propriété lors de la sauvegarde du document.



Il existe une boîte de dialogue avancée pour modifier les propriétés du document.

**Propriétés de wdPropertyAppName.docm**

Général **Résumé** Statistiques Contenu Personnalisation

Titre : Les variables

Sujet :

Auteur : Olivier Lebeau

Responsable :

Société : ...

Catégorie :

Mots clés :

Commentaires : détourner les propriétés de leur but initial

Répertoire Web :


Modèle : Normal.dotm


☐ Enregistrer l'image de l'aperçu


OK Annuler

Le résultat obtenu dans l'**Explorateur Windows**.

Rechercher

 Dossiers



 Synchronisation des dossiers

Mes documents\Word

	Taille	Type	Date de modification	Auteur	Commentaires	Mots-clés
AppName.docm	11 Ko	Docume...	18/01/2008 20:29	Olivier Lebeau	détourner les prop...	
	14 Ko	Docume...	29/12/2007 20:30	...		
...ple.rar	11 Ko	WinRAR...	2/01/2008 12:42			
...ple.docm.docx	10 Ko	Docume...	2/01/2008 12:37	...		

Le but poursuivi n'est pas l'utilisation de l'interface graphique, mais l'accès aux propriétés du document en VBA.

Pour accéder aux différentes propriétés de document en VBA, vous devez les passer en argument à la propriété.

```
ActiveDocument.BuiltInDocumentProperties(wdPropertyAuthor) = "Mes Données"
```

Ces propriétés n'acceptent que des données simples, il n'est pas possible d'utiliser des tableaux. Heureusement, VBA recèle des ressources qui nous permettent de transformer du texte simple en tableau.

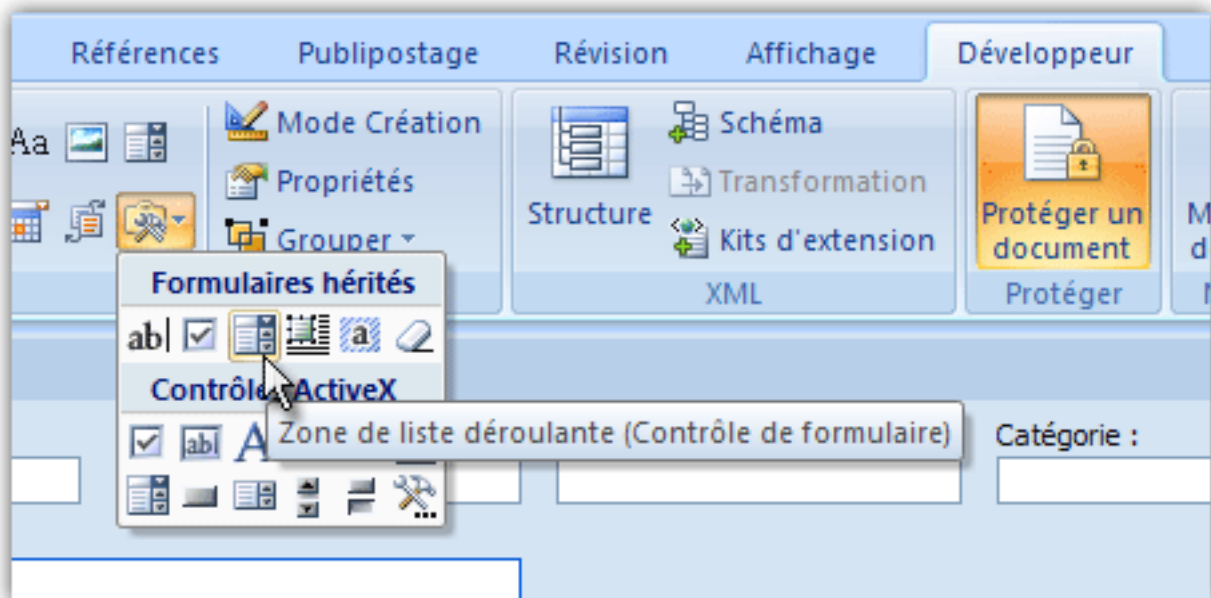
Certains objets utilisés par Word nécessitent l'utilisation de tableau, les listes déroulantes en font partie.

Nous allons utiliser une fonction VBA pour transformer notre ligne en tableau.

La fonction VBA qui permet la transformation de texte en tableau est **Split**. Si nous utilisons un caractère précis pour séparer les données, nous aurons la possibilité d'utiliser ce caractère en paramètre dans la fonction **Split**.

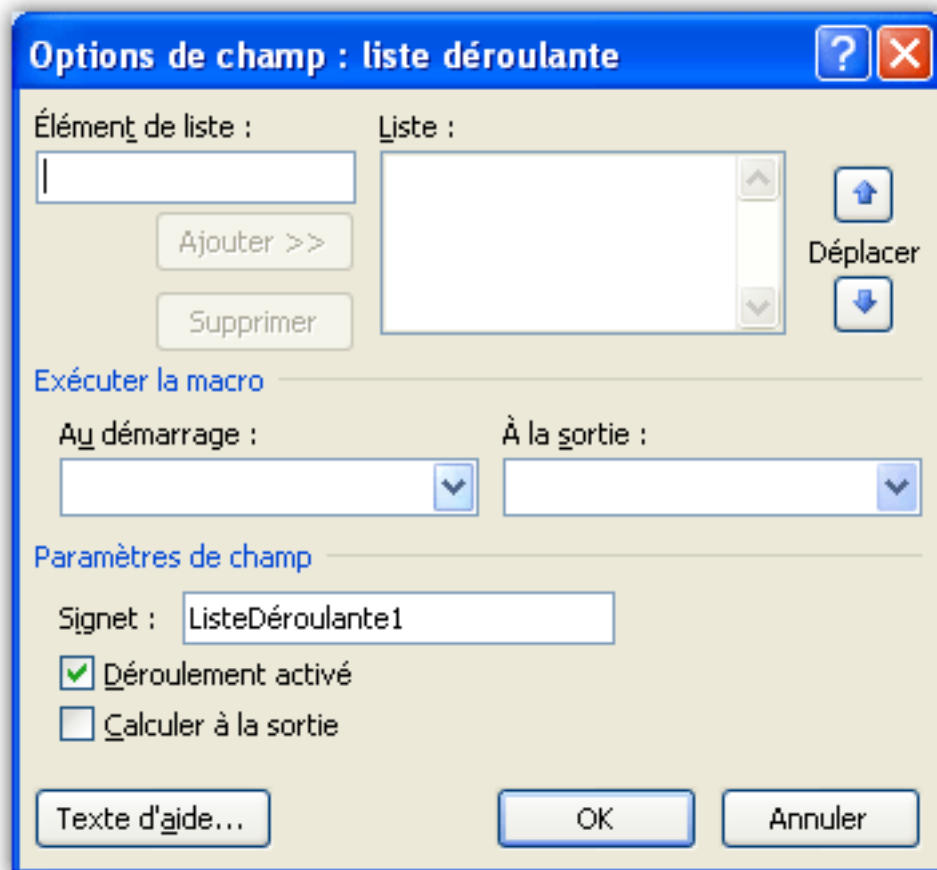
Le caractère que je vous conseille d'utiliser est le ";". Il est en règle générale utilisé pour séparer les éléments d'une liste.

Nous allons dans un document insérer un contrôle de formulaire "ListeDéroulante".



Si vous double-cliquez sur le contrôle, vous aurez accès à ses propriétés.





Pour notre test, nous allons travailler en deux phases, la première va servir à remplir la propriété avec un texte de test, la seconde à remplir le contrôle avec les valeurs contenues dans la propriété.

```
Sub RemplirProp()  
Dim stTemp As String  
  
stTemp = "valeur1;valeur2;valeur3;valeur4"  
ActiveDocument.BuiltInDocumentProperties(wdPropertyComments) = stTemp  
  
End Sub
```

Ensuite, nous allons remplir la liste avec nos valeurs issues de notre propriété et transformées en tableau.

```
Sub RemplirListe()  
Dim stTab() As String  
Dim i As Integer  
stTab = Split(ActiveDocument.BuiltInDocumentProperties(wdPropertyComments), ";")  
With ActiveDocument.FormFields(1).DropDown  
For i = 0 To UBound(stTab)  
.ListEntries.Add stTab(i)  
Next i  
End With  
  
End Sub
```

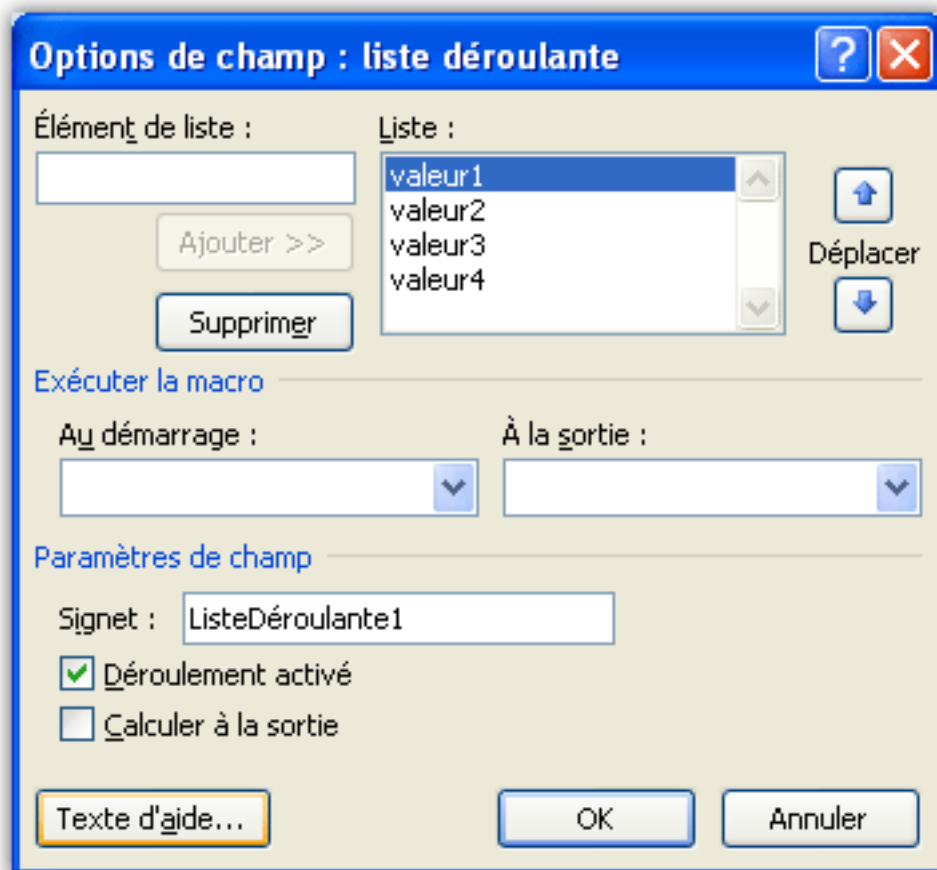
## Détail du code

- Déclaration de nos variables : Dim StTab() As String et Dim i As Integer

- Récupération des données contenues dans la propriété et séparée pour en faire un tableau :  
Split(ActiveDocument.BuiltInProperties(wdPropertyComment),";")
- Remplissage de la liste déroulante : ActiveDocument.FormFields(1).DropDown.ListEntries.Add et pour chaque élément du tableau dans une boucle.

### Tutoriel : Utiliser les variables tableaux en VBA Excel

Le résultat :



*Les propriétés avec les éléments de la liste*

Il existe une méthode simple pour visualiser dans un document les données contenues dans une propriété de document.

Cette méthode passe par l'utilisation de champs.

```
{ COMMENTS \* MERGEFORMAT }
```

### Tutoriel : Les champs dans Word

L'inconvénient majeur à l'utilisation des propriétés du document pour assurer la persistance des données est la possibilité d'accéder aux données très facilement. N'importe quel utilisateur peut modifier les données.


Comme vous l'avez vu plus haut, l'utilisateur peut accéder aux propriétés par l'interface graphique et les modifier sans aucune difficulté.


### 3 - CustomProperties

En plus des propriétés du document incluses, Word vous offre la possibilité d'utiliser des propriétés personnelles. Ces propriétés sont contenues dans la collection **CustomDocumentProperties**. La taille des données autorisée pour ces propriétés est comme pour les propriétés prédéfinies de 255 caractères.

L'utilisation de ces propriétés se fait de la même manière que pour les propriétés incluses avec comme seule différence le nom de la propriété qui est libre.

```
Sub AjoutPropPerso()  
ActiveDocument.CustomDocumentProperties.Add Name:="MonNom", LinkToContent:=False, Value:="Olivier",  
Type:=msoPropertyTypeString  
End Sub
```

 *Les propriétés personnelles doivent être gérées avec rigueur, Word n'accepte pas la création d'une propriété existante. Si vous exécutez le code ci-dessus deux fois de suite, une erreur est levée.*

 *Tous les arguments doivent être spécifiés. Si vous omettez un argument, vous lèverez une erreur spécifiant que le nombre d'arguments n'est pas correct. Curieusement, il n'y a pas d'aide à la saisie.*

On peut gérer cette erreur de la façon suivante :

```
Sub TestExistProp()  
On Error GoTo Ajout  
If Not (IsNull(ActiveDocument.CustomDocumentProperties("MonNom").Value)) Then Exit Sub  
Ajout:  
If Err.Number = "5" Then  
ActiveDocument.CustomDocumentProperties.Add Name:="MonNom", LinkToContent:=False, Value:="Olivier",  
Type:=msoPropertyTypeString  
Else  
MsgBox "Erreur non gérée"  
End If  
End Sub
```

Ce code gère l'erreur mais n'est pas très performant. Si l'on désire avoir une gestion plus efficace des propriétés personnelles, il est préférable de le faire à l'aide d'une routine séparée.

```
Sub TestExistProp(MaVar As String)  
On Error GoTo Ajout  
If Not (IsNull(ActiveDocument.CustomDocumentProperties("MonNom").Value)) Then  
ActiveDocument.CustomDocumentProperties("MonNom").Value = MaVar  
Ajout:  
Select Case Err.Number  
Case "5"  
ActiveDocument.CustomDocumentProperties.Add Name:="MonNom", LinkToContent:=False, Value:=MaVar,  
Type:=msoPropertyTypeString  
Case "0"  
Exit Sub  
Case Else  
MsgBox "Erreur non gérée" & vbCrLf & Err.Number & vbCrLf & Err.Description  
End Select  
End Sub
```

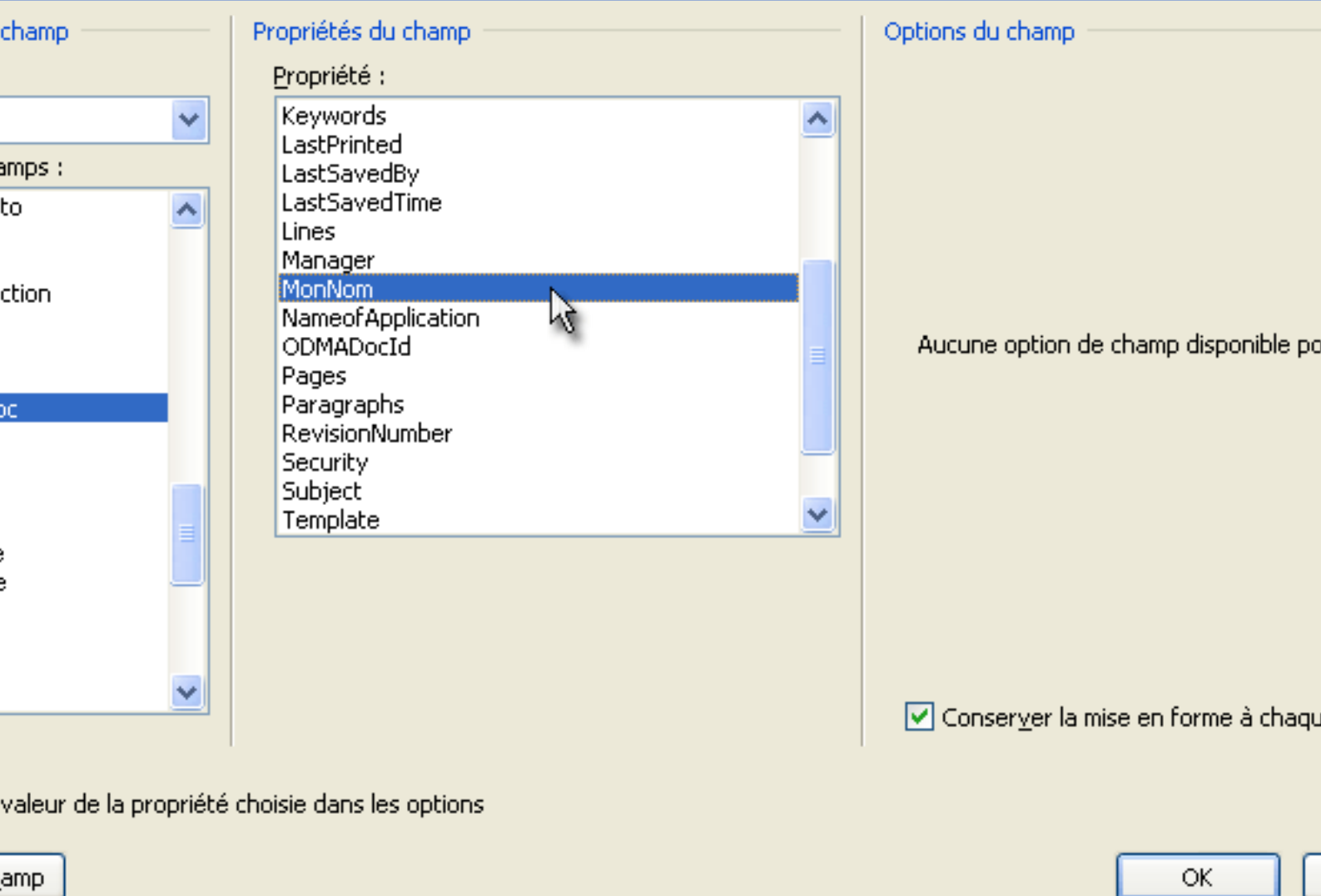
Et son appel :

```
Sub AjoutDonnee()  
TestExistProp ("Vive ma chérie")  
End Sub
```

Si vous désirez supprimer votre propriété :

```
Sub SuppressionPropriete()  
ActiveDocument.CustomDocumentProperties("MonNom").Delete  
End Sub
```

Dès son ajout, cette propriété devient disponible dans la liste des propriétés du document. Si vous désirez ajouter un champ contenant une propriété du document, votre propriété s'y trouve.



*Notre nouvelle propriété*

Lors du passage des arguments à notre méthode, nous avons la possibilité de choisir le type de données qui sera utilisé.

Argument	Type
msoPropertyTypeDate	Date
msoPropertyTypeBoolean	Boolean
msoPropertyTypeFloat	Floating point
msoPropertyTypeNumber	Integer
msoPropertyTypeString	String

En comparaison avec les données prédéfinies de la partie précédente, nous obtenons un avantage supplémentaire, les données ne sont plus modifiables aussi facilement par l'utilisateur.

## 4 - Variables


La dernière façon d'ajouter des données persistantes à un document que nous allons aborder est l'ajout de variables de document. Ces variables sont accessibles par le code et sont affichables par le biais des champs. Les variables de document se présentent sous la forme suivante :

```
ActiveDocument.Variables("MaVariable").Value
```

Une différence marquante existe entre une variable de document et une propriété, la variable n'est pas limitée à 255 caractères.

Pour ajouter une variable, nous utilisons la méthode Add, pour la supprimer, la méthode Delete.

```
Sub AjoutVriable()  
ActiveDocument.Variables.Add Name:="MaVariable", Value:="LaValeur"  
End Sub
```

 *Lorsqu'une variable existe et que vous essayez de l'ajouter à nouveau, vous levez une erreur.*

Si vous gérez l'erreur, il est possible d'utiliser une routine pour l'ajout des données dans une variable. Si cette variable n'existe pas, elle sera créée, si elle existe, on lui affectera une valeur.

```
Sub AddVariables(stName As String, stValue As String)  
On Error Resume Next  
ActiveDocument.Variables.Add Name:=stName, Value:=stValue  
Select Case Err.Number  
Case "5903"  
ActiveDocument.Variables(stName).Value = stValue  
Case Else  
Exit Sub  
End Select  
  
End Sub
```

L'appel de la routine :

```
Sub TestAjout()  
AddVariables "NouvelleVariable", "NouvelleValeur"  
End Sub
```

Word offre aussi la possibilité d'afficher les variables au travers des champs.

```
{ DOCVARIABLE "Test" \* MERGEFORMAT }
```

## 5 - Les propriétés des documents avec DSO

Pour lire ou modifier les propriétés des documents office, il n'est pas nécessaire de les ouvrir avec l'application concernée.

Microsoft met à notre disposition une bibliothèque permettant de lire et modifier certaines propriétés.

### Bibliothèque "DSO oleDocument Properties Reader 2.0"

Cette bibliothèque est utilisable en VBA par toutes les applications Office. Pour l'utiliser, vous devez juste la référencer dans votre projet.

Deux Classes nous intéressent dans cette bibliothèque :

**CustomProperties**

**SummaryProperties**

### 5-A - SummaryProperties

La bibliothèque DSO permet la lecture et la modification des propriétés de document. L'avantage de cette bibliothèque, c'est qu'il n'est pas nécessaire d'ouvrir les fichiers avec l'application qui y est associée.

#### Afficher l'auteur

```
Sub TestSumProp()  
Dim oDso As DSOFile.OleDocumentProperties  
Set oDso = New DSOFile.OleDocumentProperties  
oDso.Open sfilename:="C:\Documents and Settings\Olivier\Mes documents\Word\wdPropertyAppName.docm"  
Debug.Print oDso.SummaryProperties.Author  
End Sub
```

### 5-B - CustomProperties

Il est possible d'obtenir les propriétés personnelles d'un document par l'utilisation de DSO.

#### Exemple pour obtenir la liste des propriétés personnelles

```
Sub ListeCustomPropDSO()  
'Déclaration des objets  
Dim oDSO As DSOFile.OleDocumentProperties  
Dim oCustProp As DSOFile.CustomProperty  
  
'Affectation des Objets  
Set oDSO = New DSOFile.OleDocumentProperties  
oDSO.Open sfilename:="C:\Documents and Settings\Olivier\Mes documents\Word\wdPropertyAppName.docm"  
  
For Each oCustProp In oDSO.CustomProperties  
Debug.Print oCustProp.Name & " - " & oCustProp.Value  
Next oCustProp  
End Sub
```

Avec DSO, vous pouvez en plus de la lecture des propriétés personnelles, ajouter des propriétés à votre document.

Pour plus de détails sur le DSO et un apprentissage du FSO (File System Object), je vous conseille vivement les deux liens ci-dessous.

**La manipulation des fichiers en VBA Lire et modifier les propriétés des classeurs et autres fichiers**

## 6 - Remerciements

Je remercie l'équipe de la rédaction de DVP sans laquelle je n'aurais pu écrire cet article.

### Toute ma gratitude à :

SilkyRoad

Morgan Billy

RideKick



**L'espace de SilkyRoad**

**Espace Perso de Dolphy**

**Joris Crozier**