

# Les événements en VBA *par Geo*

Microsoft WORD  
<http://faqword.com>

- Introduction
- Les événements prédéfinis dans *ThisDocument*
- Les événements déclarés dans un module de classe
- Interférence des traitement des événements
- Quels événements ?
- Quelques exemples simples
- Attention
- Tableau récapitulatif des événements disponibles



- 
- ▶ Un événement permet d'enclencher un traitement quand une action particulière a été faite par l'opérateur. Cela concerne en général des actions portant sur un objet.
  - ▶ L'exemple le plus simple : le clic sur un bouton produit une action, sinon ce bouton ne sert à rien. L'événement est "le clic sur le bouton", la procédure associée est une procédure Sub de nom MonBouton\_Click, cette procédure événementielle traite l'événement concerné.
  - ▶ On peut classer les événements en fonction de l'objet concerné et de la manière de les traiter :
  - ▶ Ceux qui sont déclenchés par une action sur un contrôle placé sur un document Word, en général des formulaires
  - ▶ Ceux qui sont déclenchés par une action sur un contrôle situé sur une feuille (UserForm) ou une action sur la feuille elle-même.
  - ▶ Ceux qui sont prédéfinis dans ThisDocument.
  - ▶ Ceux qui sont déclarés par le programmeur dans un module de classe.
  - ▶ Pour être exhaustif, il faudrait citer les événements déclenchés par des actions sur le ruban et ceux qui sont déclenchés par des contrôles placés sur des documents. Les premiers sont des procédures qui sont référencées dans des modules écrits en Xml et donc sortent de notre sujet. Les contrôles placés sur des feuilles (UserForm) sont décrits dans le didacticiel sur ce sujet. Ceux qui sont placés sur un document sont similaires. Aussi seuls événements du ThisDocument et des modules de classe sont étudiés ici.

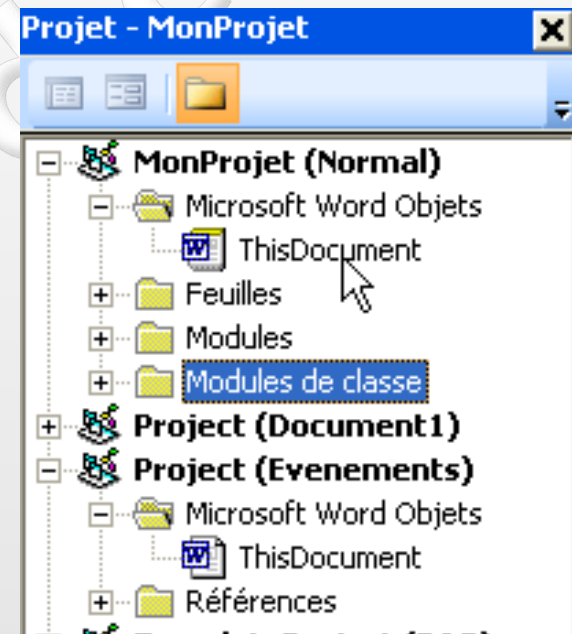


# **Les événements prédéfinis dans ThisDocument**



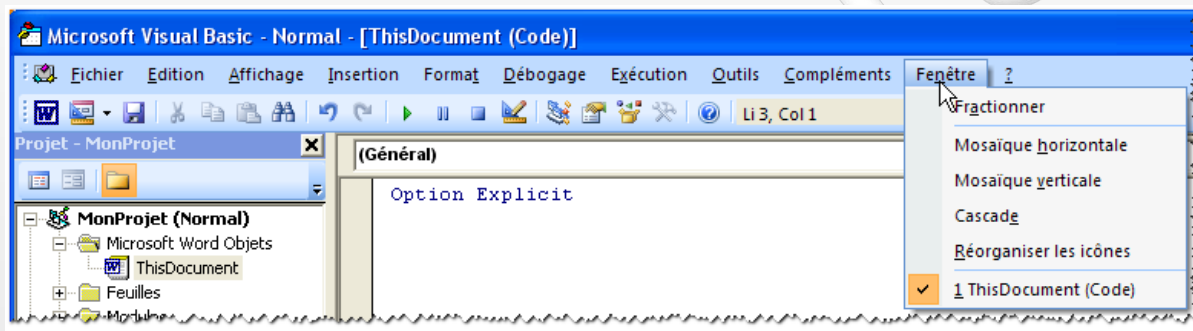
# Les événements prédéfinis dans *ThisDocument*

- Dans la fenêtre projet de VBE, développez le "Microsoft Word Objets" du modèle ou du document et faites un double clic sur "ThisDocument" (C'est le nom par défaut que vous pouvez modifier ; dans la suite de ce document c'est toujours la dénomination *ThisDocument* qui sera utilisée).



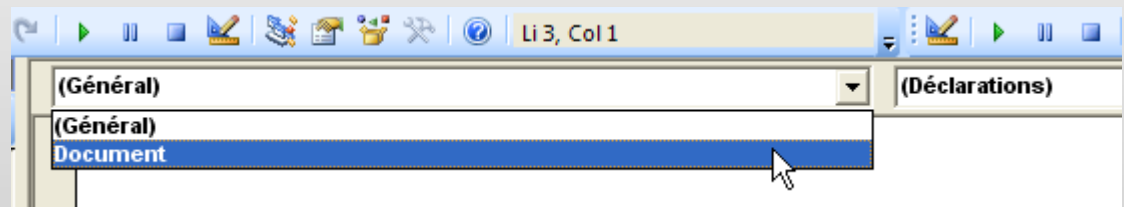
# Les événements prédéfinis dans *ThisDocument*

- ▶ Une nouvelle fenêtre de code s'ouvre, elle porte le nom ThisDocument :



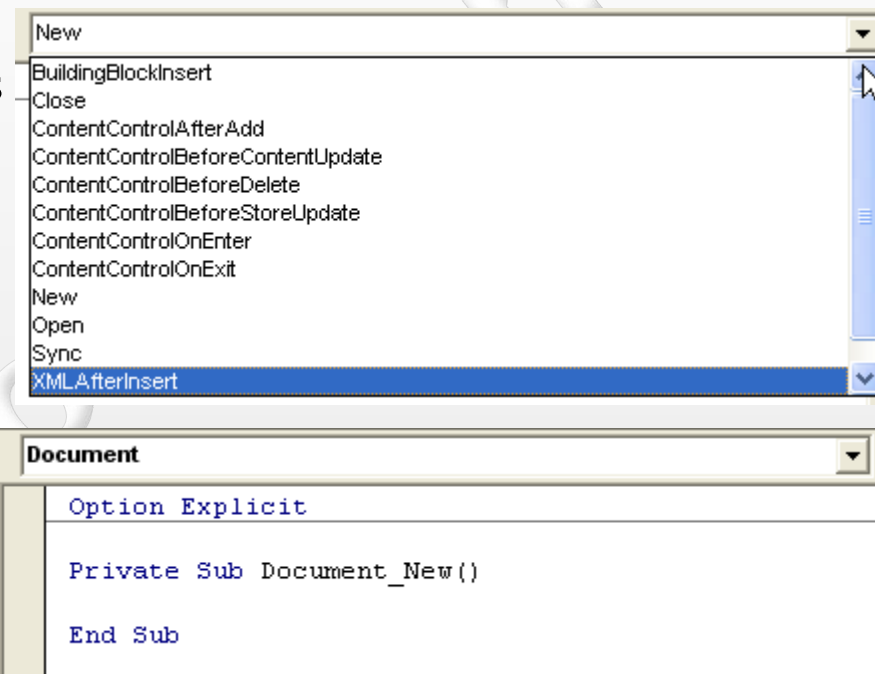
- ▶ Elle contient deux listes déroulantes, celle de gauche vous propose Général ou Document. Si vous avez mis des contrôles sur votre document, leur nom apparaît aussi dans cette liste.

Choisissez *Document*



# Les événements prédéfinis dans *ThisDocument*

- ▶ La liste de droite vous propose alors les 13 événements disponibles pour l'objet Document.



- ▶ Si vous choisissez New, la procédure suivante est créée dans la fenêtre de code.
- ▶ Il ne vous reste plus qu'à y indiquer le code que vous voulez exécuter lors de la création d'un nouveau document. Dans le cas du *New*, cette procédure a le même rôle qu'une macro automatique nommée *AutoNew*. Elle n'a de sens que dans un modèle, car, placée dans un document existant, elle ne se déclenchera jamais.

# Les événements prédéfinis dans ThisDocument

---

- ▶ Des procédures traitant les événements Open ou Close peuvent être placées dans le *ThisDocument* d'un document pour ne se déclencher que dans ce seul document. Si elles ont placées dans le *ThisDocument* d'un modèle elles sont déclenchées pour tous les documents basés sur ce modèle. Si elles sont dans le modèle *Normal*, elles se déclenchent pour tous les documents, sauf s'il y a un autre modèle associé au document.
- ▶ N'oubliez pas qu'en Word 2007, les documents qui contiennent des macros ont comme extension *docm* et les modèles *dotm*.
- ▶ Comme vous le voyez, ces événements concernent un document, vous les trouverez donc dans l'aide concernant l'objet Document".
- ▶ Voir la liste page suivante.



# Liste des événements concernant un document



Événement	Nouveau dans Word 2007	Nom de procédure dans ThisDocument
Nouveau document.		New
Insertion d'un bloc de construction dans un document.	Oui	BuildingBlockInsert
Se produit lors de la fermeture d'un document.		Close
Ajout d'un contrôle du contenu dans un document.	Oui	ContentControlAfterAdd
Avant la mise à jour d'un contrôle de contenu.	Oui	ContentControlBeforeContentUpdate
Suppression d'un contrôle du contenu.	Oui	ContentControlBeforeDelete
Avant mis à jour de la valeur d'un contrôle du magasin de données XML	Oui	ContentControlBeforeStoreUpdate
Nouveau contrôle du contenu.	Oui	ContentControlOnEnter
Sortie d'un contrôle du contenu.	Oui	ContentControlOnExit
Ouverture d'un document.		Open
Synchronisation de la copie locale d'un document avec la copie sur le serveur.		Sync
Ajoute d'un nouvel élément XML à un document.		XMLAfterInsert
Effacement d'un élément XML d'un document.		XMLBeforeDelete





# Les événements prédéfinis dans *ThisDocument*

---

- ▶ Pour les trois opérations de base faites sur un document : New, Open, Close, il existe aussi trois macros automatiques. Il semble que la tendance soit de faire disparaître les macros automatiques au profit d'événements. Cette orientation est nette dans d'autres applications Office 2007 et se confirme ici par l'apparition de nombreux événements liés à des nouveautés de la version 2007.
- ▶ **Si l'événement que vous souhaitez traiter se trouve dans ce tableau, programmez-le dans le *ThisDocument*. C'est simple et ça marche toujours.**



# **Les événements déclarés dans un module de classe**



# Les événements déclarés dans un module de classe

---

- ▶ Les possibilités offertes par les événements déclarés dans un module de classe sont largement plus vastes que les précédentes, mais leur gestion est plus délicate. Le titre lui-même, inquiète un peu. Rassurez-vous c'est à votre portée.
- ▶ Qu'est-ce qu'un module de classe ?
  - ▶ Vous savez ce qu'est un module en VBA, c'est un ensemble de lignes de code regroupées sous un même nom. Un module de classe est un ensemble de lignes de code dont l'objet est de définir une classe.
  - ▶ Qu'est ce classe ? Voici la définition de l'aide : "Définition formelle d'un objet. La classe est un modèle à partir duquel une occurrence d'un objet est créée au moment de l'exécution. La classe définit les propriétés de l'objet et les méthodes utilisées pour déterminer le comportement de ce dernier." Pour simplifier : une classe définit un nouveau type d'objet qui contient non seulement des données, mais aussi des traitements. **Le nom de la classe est le nom du module.**

# Les événements déclarés dans un module de classe

---

- ▶ La gestion des événements est un cas, parmi d'autres, de l'utilisation des classes. Prenons comme exemple une variable de type *Integer*. Avec le vocabulaire ci-dessus, on dira que *Integer* est une classe et que la déclaration :

Dim i As Integer

est une instantiation de i à partir de la classe *Integer*, ou bien : i est une occurrence de la classe *Integer*. Les macroteurs qui ne s'embarrassent pas avec des mots savants, disent que : i est un *Integer*.

- ▶ Mais comme il faut bien utiliser, au moins partiellement, ce vocabulaire pour la gestion des événements, disons qu'un module de classe regroupe du code permettant de définir une classe. Il est nécessaire de déclarer (=instancier) un ou des objets (occurrences) à partir de cette classe afin de l'utiliser, tout comme vous avez déclaré une variable de type *Integer* pour vous en servir. Si vous ne déclarez pas de variable de type *Integer*, ce type existe, mais ne vous est d'aucune utilité. Il ne faut pas perdre de vue ces deux aspects : définition et utilisation.

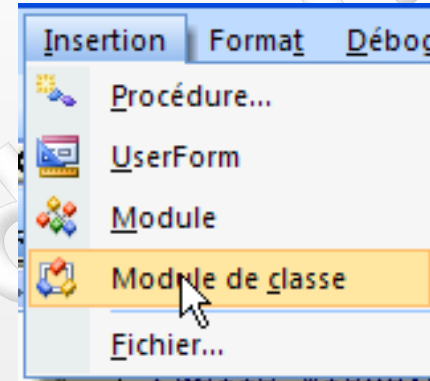
# Les événements déclarés dans un module de classe

---

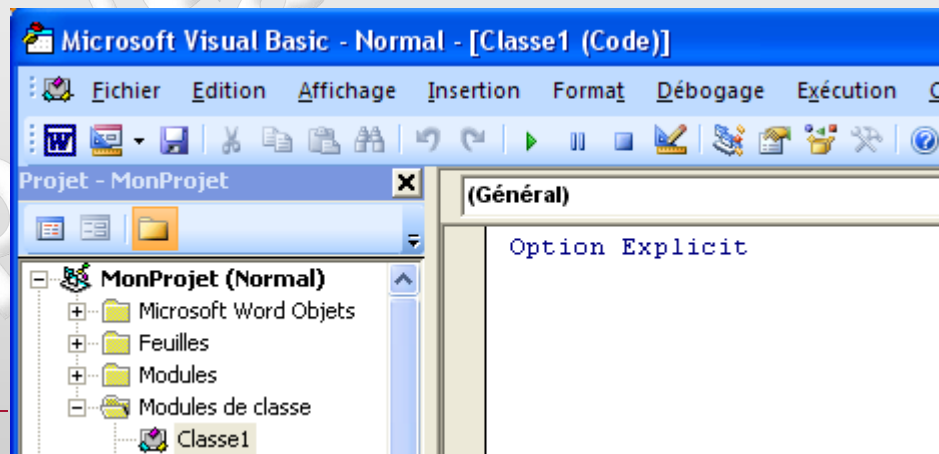
- ▶ Pour gérer un événement, il est nécessaire de :
  - ▶ Écrire un module de classe dans lequel vous devez
    - ▶ Définir une ou plusieurs membres de la classe pour répondre aux événements.
    - ▶ Rédiger le code de traitement des événements.
  - ▶ Affecter des valeurs aux membres d'une variable du type de cette classe. Ceci doit se faire dans un module exécutable, un module de classe est uniquement déclaratif.

# Les événements déclarés dans un module de classe

- Comment ajouter un module de classe à vos projets ? Dans VBE, utilisez le menu ou cliquez avec le bouton droit dans le projet concerné : Insertion et choisissez Module de classe.



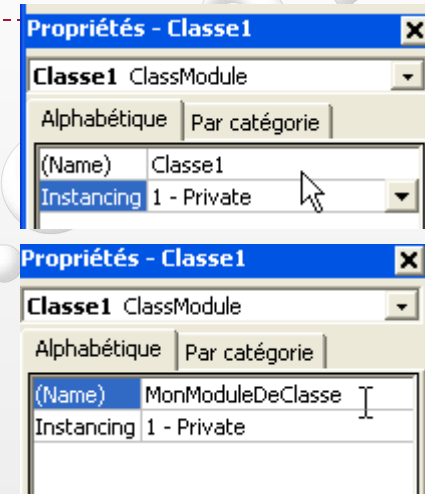
- La fenêtre de code de votre module de classe s'initialise :



# Les événements déclarés dans un module de classe

- ▶ Vous pouvez changer le nom du module dans la fenêtre propriétés.

Le nom du module étant le nom de la classe, un nom plus explicite serait certainement plus judicieux.



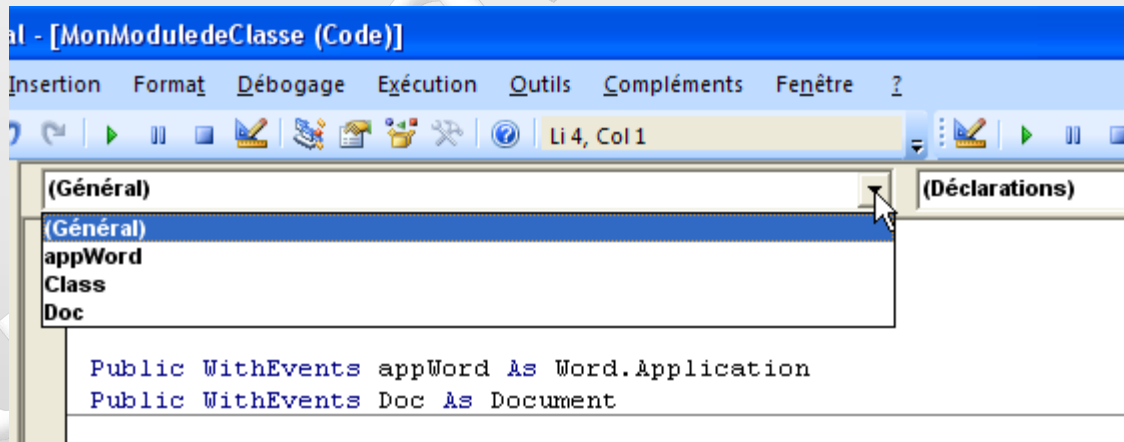
- ▶ La définition du membre dans le module de classe, est du type :

```
Public WithEvents Nom_du_Membre As Type_d_Objeto
```

Le type d'objet indiqué doit bien évidemment avoir des événements comme membres, sinon cette déclaration ne servira à rien.

# Les événements déclarés dans un module de classe

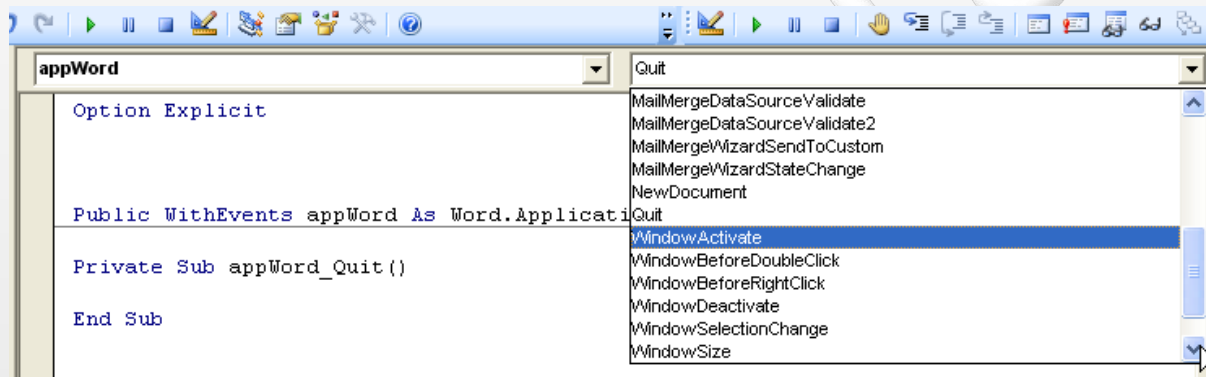
- ▶ Prenons comme exemples les plus fréquemment utilisés:  
`Public WithEvents appWord As Application`  
`Public WithEvents Doc As Document`
- ▶ Les variables de type *MonModuleDeClasse* auront donc deux membres : *appWord* et *Doc*.
- ▶ Une fois ces déclarations effectuées, vous constatez que vos membres apparaissent dans la liste déroulante de gauche de la fenêtre de code.





# Les événements déclarés dans un module de classe

- ▶ Lorsque vous sélectionnez un membre, la liste de droite affiche les événements disponibles :



- ▶ Lorsque vous choisissez un événement, dans cette liste de droite, une procédure vide est insérée automatiquement dans le code. Il ne vous reste plus qu'à la compléter.

# Les événements déclarés dans un module de classe

---

- Dans l'exemple qui suit, trois événements sont prévus :

```
Public WithEvents appWord As Word.Application
```

```
Public WithEvents Doc As Document
```

```
Private Sub appWord_Quit()
```

```
    MsgBox "Au revoir !"
```

```
End Sub
```

```
Private Sub appWord_DocumentBeforePrint(ByVal Doc As Document, Cancel As Boolean)
```

```
    MsgBox "Impression de " & Doc.name & " interdite !"
```

```
    Cancel = True
```

```
End Sub
```

```
Private Sub Doc_Close()
```

```
    MsgBox "Fermeture de " & ActiveDocument.name
```

```
End Sub
```



# Les événements déclarés dans un module de classe

---

- ▶ Vous remarquez que l'événement *DocumentBeforePrint* est un événement de l'application alors que *Close* est bien un événement du document, cela veut dire que le premier se déclenchera, ou non, pour tous les documents selon la valeur affectée au membre *Application*. Le *Close* ne s'appliquera qu'au document concerné par le membre *Doc*. C'est une différence de comportement importante avec la procédure de *Close* définie dans *ThisDocument* qui va se déclencher pour tous les documents. Retenez bien cette différence : **les événements définis dans une classe ne dérouleront le traitement qu'en fonction de la valeur du membre à cet instant.**
- ▶ La procédure *DocumentBeforePrint* a deux paramètres, le document et un paramètre de nom *Cancel*. Ce deuxième paramètre permet **d'interrompre l'opération** en cours, ici l'impression du document. C'est le rôle de l'instruction *Cancel = True*, qui bloque l'impression des documents. Cette possibilité n'existe ni dans *ThisDocument*, ni dans les macros automatiques.

# Les événements déclarés dans un module de classe

---

- ▶ Une fois le module de classe terminé, il faut déclarer une variable du type que vous venez de définir par le module de classe, puis donner une valeur aux membres concernés afin que les événements se déclenchent. Le mot **New** est obligatoire dans ces déclarations.

Exemple :

```
Dim X As New MonModuledeClasse
Sub ActiveEvenements()
    Set X.appWord = Word.Application
    Set X.Doc = ActiveDocument
End Sub
```

- ▶ Vous pouvez naturellement choisir les objets dont vous voulez armer les événements en ne mettant que l'instruction **Set** nécessaire. Si vous n'en mettez aucune, il n'y aura pas d'événements traités, bien que **X** ait été déclarée..

# Les événements déclarés dans un module de classe

---

- ▶ Les déclarations et affectations peuvent se faire dans n'importe quelle macro, le plus simple étant de le faire dans une macro qui sera exécutée systématiquement (si, bien sûr, vous souhaitez que cet événement soit "armé" en permanence), par exemple dans une macro *AutoExec* pour l'Application et dans l'*AutoOpen* et *AutoNew* pour un document. Une autre possibilité est de mettre ce code dans les procédures *Document\_Open* et *Document\_New* du *ThisDocument* du modèle *Normal*, l'une des deux au moins sera lancée ; avec une exception toutefois pour le document initial qui s'affiche au démarrage de Word.

Et voici le résultat :



# Les événements déclarés dans un module de classe

---

- Pour qu'un ensemble d'événements ne se déclenche plus, il suffit d'affecter **Nothing** aux membres concernés :

```
Sub NeutraliseEvenements()  
    Set X.appWord = Nothing  
    Set X.Doc = Nothing  
End Sub
```

# **Interférence des traitement des événements**



# Interférence des traitements des événements

---

- ▶ Vous avez remarqué que l'événement "fermeture de document" existe dans *ThisDocument* et vous l'avez aussi prévu dans votre module de classe. Vont-ils se déclencher tous les deux et dans quel ordre ?
- ▶ Et il est possible d'utiliser simultanément *ThisDocument* dans les modèles et le document, ainsi que des modules de classe dans chacun d'eux. Va-t-il se produire une cascade de traitements ? Lesquels ? Dans quel ordre ?
- ▶ A titre d'exemple, voici la séquence des traitements lors de la fermeture d'un document qui a été modifié et non sauvegardé. Sont présents : une macro *AutoClose*, un module *ThisDocument* dans le modèle Normal, dans un modèle complémentaire attaché et dans le document lui-même. Un module de classe est défini dans chacun de ces trois éléments et tous les événements activés.



# Interférence des traitements des événements

Pas moins de 9 événements de fermeture sont déclenchés :

Emplacements dans l'ordre de déclenchement		Procédure
1	Classe modèle Normal	App_DocumentBeforeClose
2	Classe modèle complémentaire	App_DocumentBeforeClose
3	Classe Document	App_DocumentBeforeClose
4	Macro	Auto Close
5	ThisDocument Modèle complémentaire	Document_Close
6	ThisDocument Document	Document_Close
7	Classe modèle complémentaire	Doc_Close
8	Classe Document	Doc_Close
9	Classe modèle Normal	Doc_Close
	Classe modèle Normal	App_DocumentBeforeSave
	Classe modèle complémentaire	App_DocumentBeforeSave
	Classe Document	App_DocumentBeforeSave

# Interférence des traitements des événements

---

- ▶ Dans le tableau, les événements *DocumentBeforeSave* ont été séparés car ils n'appartiennent pas à proprement parler à la séquence, mais il est intéressant de les y faire figurer.
- ▶ Ce sont les événements *DocumentBeforeClose* qui se déclenchent très logiquement en premier, il est possible alors **d'interrompre le processus de fermeture**. La macro se déclenche ensuite, puis le code de deux *ThisDocument*, toujours pas celui du modèle Normal. Les procédures des modules de classe qui sont définies au niveau document arrivent en dernier. Elles sont suivies du traitement des *DocumentBeforeSave* qui peuvent aussi **interrompre la sauvegarde** mais plus la fermeture. Vous remarquez que les sous-séquences sont différentes par rapport à l'ouverture.
- ▶ La procédure *DocumentBeforeSave* a trois paramètres : le nom du document, le mode d'enregistrement et l'ordre d'interruption de l'opération (comme pour le *BeforePrint*)
- ▶ Le mode d'enregistrement est une variable booléenne qui a la valeur *Vrai* si l'enregistrement est fait par "Enregistrer sous". Cette procédure pourrait donc être utilisée pour bloquer une tentative de copier le document ailleurs ou sous un autre nom.  
[Voir l'exemple.](#)

**Quels événements ?**



# Quels événements ?

---

- ▶ Nous n'avons parlé jusqu'ici que des événements au niveau de l'application Word ou de documents Word, il en existe d'autres, même si relativement peu d'objets sont concernés :
- ▶ *Application, Document, CustomTaskPane, CustomXMLPart, CustomXMLParts, MsoEnvelope, CommandBars, CommandBarButton, CommandBarComboBox.*
- ▶ Depuis la version 2007, les barres de menu ont disparu au profit du ruban, les trois derniers n'ont donc d'intérêt que dans les versions précédentes. Vous trouverez la liste des événements pour chaque type d'objet dans l'aide VBA.
- ▶ Pour le ruban c'est dans un module Xml que les événements sont décrits avec les noms des procédures qui leur sont associées. Les procédures elles-mêmes sont dans des modules VBA.





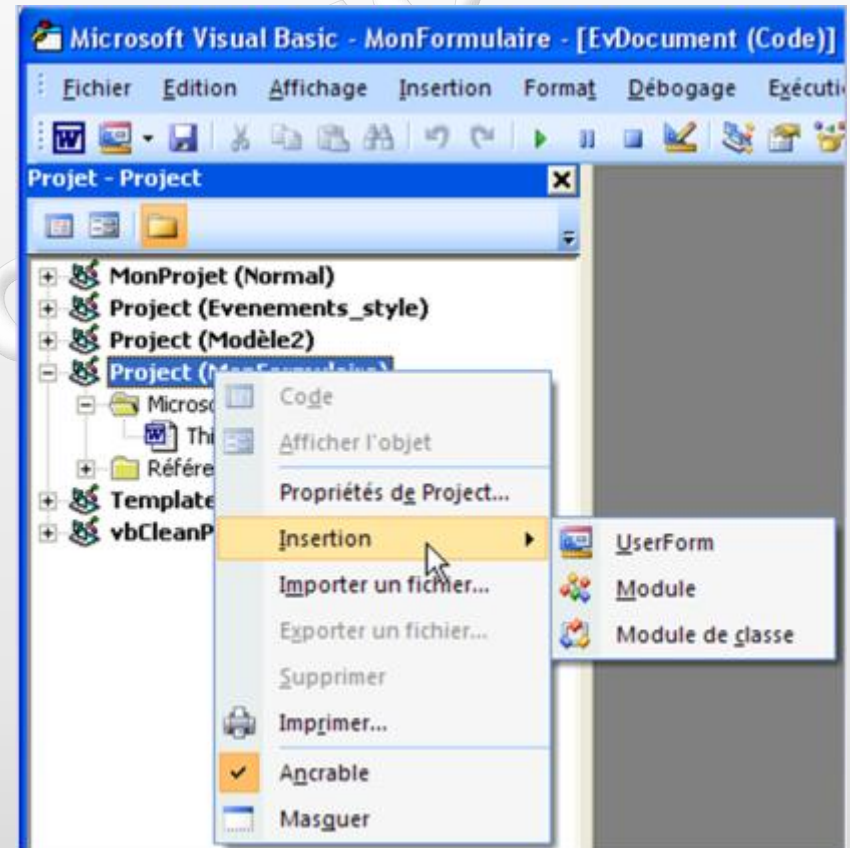
## Quelques exemples simples

Permettre d'imprimer un document sous condition  
N'autoriser l'enregistrement d'un document que sur lui-même



# Permettre d'imprimer un document sous condition

- ▶ Supposons le cas d'un formulaire dont on n'autorise l'impression que si tel champ est rempli.
- ▶ Il est évidemment nécessaire d'ouvrir le document, puis passer dans l'éditeur VBE (Alt+F11). Si la fenêtre des projets n'est pas affichée, faites-le avec CTL+R.
- ▶ Positionnez vous sur votre document, faites un clic avec le bouton droit de la souris et choisissez Module de classe.

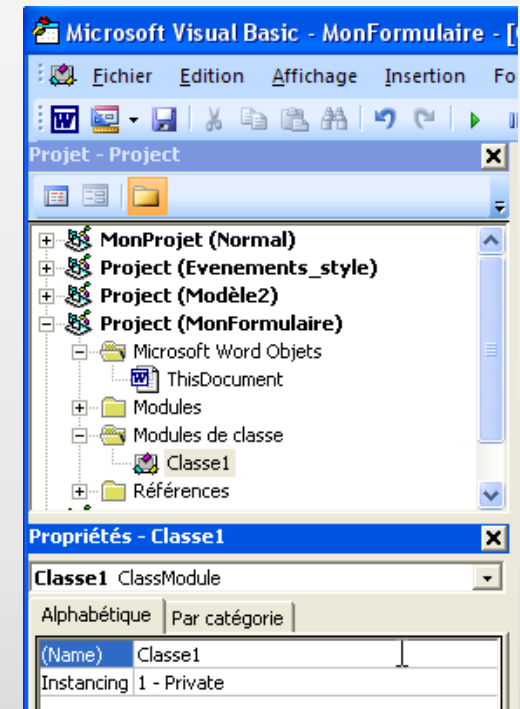


# Permettre d'imprimer un document sous condition

- Affichez la fenêtre des propriétés (F4) pour changer le nom de la classe *Classe1* en *FiltreImpression*. Dans la fenêtre de code qui s'est ouverte, placez le code :

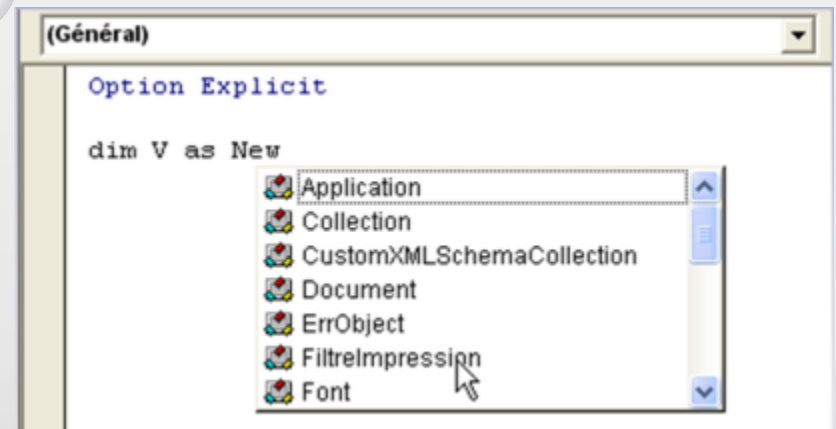
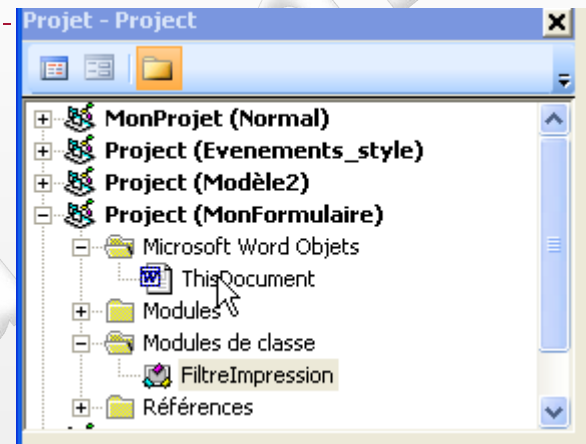
```
Option Explicit
Public WithEvents appWord As Word.Application
Dim Imprimer As Boolean

Private Sub appWord_DocumentBeforePrint(ByVal Doc As Document,
    Cancel As Boolean)
    Imprimer = False
    If ActiveDocument.Bookmarks.Exists("Signature") Then
        If Not ActiveDocument.Bookmarks("Signature").Empty Then
            Imprimer = True
        End If
    End If
    ' autres conditions
    If Not Imprimer Then
        MsgBox "Impression de " & Doc.Name & " interdite !", _
            vbCritical, "Pas de signature sur le formulaire"
        Cancel = True
    Else
        Cancel = False
    End If
End Sub
```



# Permettre d'imprimer un document sous condition

- ▶ Dans l'explorateur de projets, positionnez vous sur *ThisDocument* :
- ▶ Et faites un double clic, dans la fenêtre de code qui s'ouvre, vous déclarez une variable de type *FiltreImpression*.

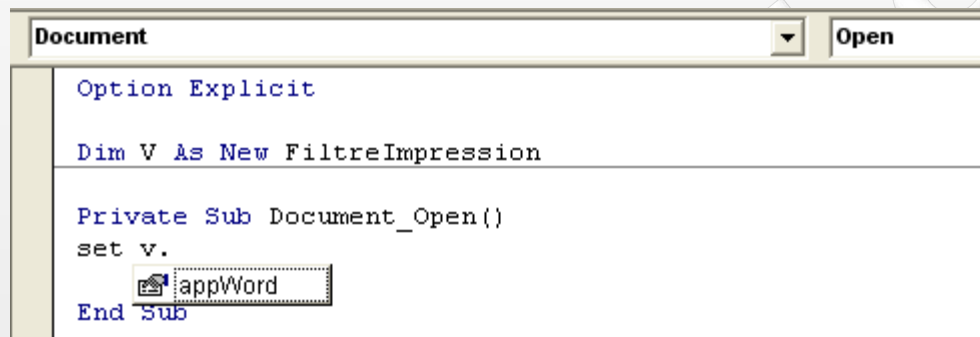


Vous voyez que votre nouveau type de données est disponible.



# Permettre d'imprimer un document sous condition

- Dans l'événement Open vous affectez le membre correspondant, l'intellisense vous y aide :



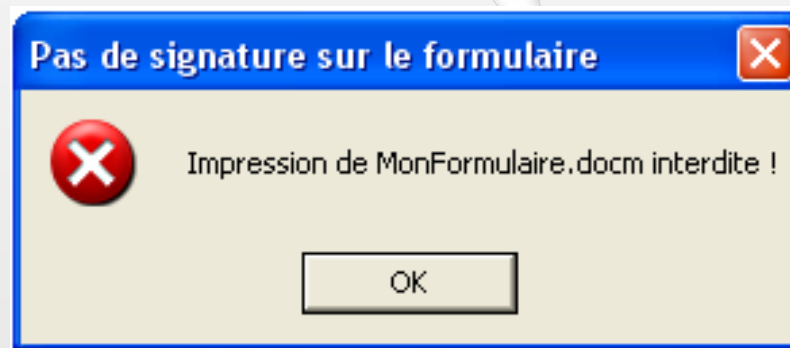
Voici le contenu de *ThisDocument* :

```
Option Explicit  
  
Dim V As New FiltreImpression  
  
Private Sub Document_Open()  
    Set V.appWord = Word.Application  
End Sub
```

# Permettre d'imprimer un document sous condition

---

- ▶ Fermez votre formulaire et ouvre-le, ne remplissez pas le champ signature, cliquez sur l'icône d'impression rapide et vous verrez ceci :



# N'autoriser l'enregistrement d'un document que sur lui-même

---

- ▶ Pour les manipulations de base, reportez-vous à l'exemple précédent. Cette procédure s'ajoute à la précédente.
- ▶ Nous utiliserons cette fois la procédure *DocumentBeforeSave* qui a trois paramètres : le nom du document, le mode d'enregistrement et l'ordre d'interruption de l'opération (comme *BeforePrint*)
- ▶ Le mode d'enregistrement est une variable booléenne qui a la valeur *Vrai* si l'enregistrement est fait par "Enregistrer sous". Cette procédure pourrait être utilisée pour bloquer une tentative de copier le document ailleurs ou sous un autre nom.
- ▶ Une seule ligne suffit donc, le reste est généré automatiquement :

```
Private Sub appWord_DocumentBeforeSave(ByVal Doc As  
    Document, _  
    SaveAsUI As Boolean, Cancel As Boolean)  
    Cancel = SaveAsUI  
End Sub
```

## Attention...

Déclaration de plusieurs variables pour la classe

Déclaration des variables de type modules de classe

Bien placer les instructions d'affectation des membres

Le *BeforeSave* se déclenche pour les sauvegardes automatiques



# Déclaration de plusieurs variables pour la classe

- Puisque qu'une classe est un type de données, il doit être possible de déclarer plusieurs variables de ce type, comment va se comporter Word dans ce cas ?

Soit :

```
Public X As New MonModuledeClasse
Public Y As New MonModuledeClasse
Public Z As New MonModuledeClasse
Sub ActiveEvenements()
    Set X.appWord = Word.Application
    Set X.Doc = ActiveDocument
    Set Y.Doc = Documents("Document1")
    Set Z.Doc = Documents("Document2")
End Sub
```

- Très logiquement les procédures de traitement associé se dérouleront pour les événements de l'application et pour chacun des trois documents référencés dans les trois variables.

# Déclaration de plusieurs variables pour la classe

---

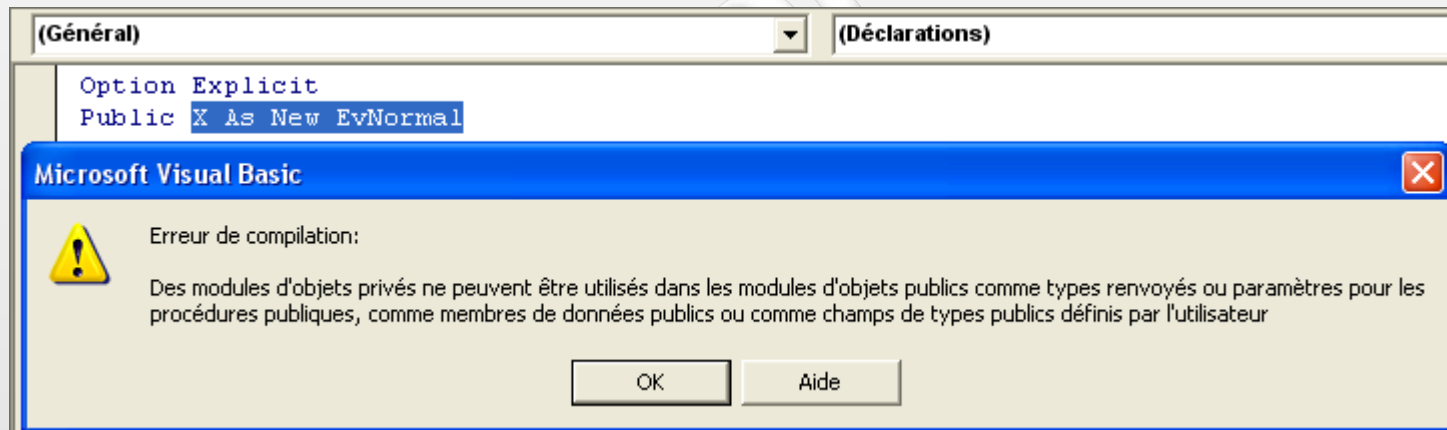
- Si le code est complété ainsi (ce qui est tentant, n'est-ce pas ?) :

```
Public X As New MonModuledeClasse
Public Y As New MonModuledeClasse
Public Z As New MonModuledeClasse
Sub ActiveEvenements()
    Set X.appWord = Word.Application
    Set Y.appWord = Word.Application
    Set Z.appWord = Word.Application
    Set X.Doc = ActiveDocument
    Set Y.Doc = Documents("Document1")
    Set Z.Doc = Documents("Document2")
End Sub
```

Les procédures d'événements de l'application se dérouleront **trois fois** !  
Vous recevrez trois boîtes de dialogue du message d'adieu prévu dans *Application\_Quit*.

# Déclaration des variables de type modules de classe

- ▶ Les variables publiques de type de votre module de classe ne peuvent pas être déclarées dans le module *ThisDocument*, voici le message d'erreur sur lequel je vous laisse méditer :



- ▶ Pour déclarer une variable publique du type de la classe définie, il faut le faire dans un module de VBA.

# Bien placer les instructions d'affectation des membres

---

- ▶ Ne pas faire ceci dans le module de classe :

```
Private Sub Doc_Open()  
    Set Na.Doc = ActiveDocument  
End Sub
```

- ▶ C'est le risque d'une boucle plus ou moins infinie.





# Le BeforeSave se déclenche pour les sauvegardes automatiques

- Si vous utilisez les sauvegardes automatiques :



- L'événement *BeforeSave* se déclenche à chaque enregistrement des informations de récupération, ça peut surprendre, même si c'est logique !

# **Tableau récapitulatif des événements disponibles**



Événement	Word 2003	Word 2007	Nom de procédure dans ThisDocument	Module de classe : exécution selon la valeur de la donnée	Module de classe : Nom de la procédure d'événement	Interrompt l'opération	Macro automatique
Avant fermeture d'un document.	X	X		Application	DocumentBeforeClose	Oui	AutoClose
Avant impression d'un document.	X	X		Application	DocumentBeforePrint	Oui	
Avant l'enregistrement d'un document.	X	X		Application	DocumentBeforeSave	Oui	
Changement de document actif.	X	X		Application	DocumentChange		
Ouverture d'un document.	X	X		Application	DocumentOpen		AutoOpen
Synchronisation de la copie locale d'un document avec la copie sur le serveur.	X	X		Application	DocumentSync		
Insertion d'un affranchissement électronique.	X	X		Application	EPostageInsert		
Insertion d'un affranchissement électronique.	X	X		Application	EPostageInsertEx	Oui	
Affichage de la boîte de dialogue de l'affranchissement électronique.	X	X		Application	EPostagePropertyDialog		
Fin de fusion.	X	X		Application	MailMergeAfterMerge		



Événement	Word 2003	Word 2007	Nom de procédure dans ThisDocument	Module de classe : exécution selon la valeur de la donnée	Module de classe : Nom de la procédure d'événement	Interrompt l'opération	Macro automatique
Fin de publipostage.	X	X		Application	MailMergeAfterRecordMerge		
Début de fusion.	X	X		Application	MailMergeBeforeMerge	Oui	
Début de publipostage	X	X		Application	MailMergeBeforeRecordMerge	Oui	
Source de données chargée.	X	X		Application	MailMergeDataSourceLoad		
Validation des destinataires de publipostage par l'opérateur.	X	X		Application	MailMergeDataSourceValidate		
Validation des destinataires de publipostage par l'opérateur.		X		Application	MailMergeDataSourceValidate2		
Choix du bouton personnalisé à l'étape six de l'Assistant Fusion et publipostage.	X	X		Application	MailMergeWizardSendToCustom		
Passage d'une étape de l'Assistant Fusion et publipostage à une autre.	X	X		Application	MailMergeWizardStateChange		
Nouveau document.	X	X	New	Application	NewDocument		AutoNew
Arrêt de Word.	X	X		Application	Quit		AutoExit
Activation d'une fenêtre de document.	X	X		Application	WindowActivate		



Événement	Word 2003	Word 2007	Nom de procédure dans ThisDocument	Module de classe : exécution selon la valeur de la donnée	Module de classe : Nom de la procédure d'événement	Interrompt l'opération	Macro automatique
Avant un double clic dans la zone d'édition d'une fenêtre de document.	X	X		Application	WindowBeforeDoubleClick	Oui	
Avant le clic droit dans la zone d'édition d'une fenêtre.	X	X		Application	WindowBeforeRightClick	Oui	
Désactivation d'une fenêtre.	X	X		Application	WindowDeactivate		
Se produit lorsque la sélection change dans la fenêtre du document actif.	X	X		Application	WindowSelectionChange		
Redimensionnement ou déplacement de la fenêtre Word.	X	X		Application	WindowSize		
Le nœud XML parent de la sélection est modifié.		X		Application	XMLSelectionChange		
Erreur de validation du code Xml	X	X		Application	XMLValidationError		
Insertion d'un bloc de construction dans un document.		X	BuildingBlockInsert	Document	BuildingBlockInsert		
Se produit lors de la fermeture d'un document.	X	X	Close	Document	Close		AutoClose
Ajout d'un contrôle du contenu dans un document.		X	ContentControlAfterAdd	Document	ContentControlAfterAdd		

Événement	Word 2003	Word 2007	Nom de procédure dans ThisDocument	Module de classe : exécution selon la valeur de la donnée	Module de classe : Nom de la procédure d'événement	Interrompt l'opération	Macro automatique
Avant la mise à jour d'un contrôle de contenu.		X	ContentControlBeforeContentUpdate	Document	ContentControlBeforeContentUpdate		
Suppression d'un contrôle du contenu.		X	ContentControlBeforeDelete	Document	ContentControlBeforeDelete		
Avant mis à jour de la valeur d'un contrôle du magasin de données XML		X	ContentControlBeforeStoreUpdate	Document	ContentControlBeforeStoreUpdate		
Nouveau contrôle du contenu.		X	ContentControlOnEnter	Document	ContentControlOnEnter		
Sortie d'un contrôle du contenu.		X	ContentControlOnExit	Document	ContentControlOnExit		
Nouveau document.		X	New	Document	New		AutoNew
Ouverture d'un document.	X	X	Open	Document	Open		AutoOpen
Synchronisation de la copie locale d'un document avec la copie sur le serveur.	X	X	Sync	Document	Sync		
Ajoute d'un nouvel élément XML à un document.	X	X	XMLAfterInsert	Document	XMLAfterInsert		
Effacement d'un élément XML d'un document.	X	X	XMLBeforeDelete	Document	XMLBeforeDelete		



Événement	Word 2003	Word 2007	Nom de procédure dans ThisDocument	Module de classe : exécution selon la valeur de la donnée	Module de classe : Nom de la procédure d'événement	Interrompt l'opération	Macro automatique
Masquage de l'interface qui correspond à l'objet MsoEnvelope.	X	X		MsoEnvelope	EnvelopeHide		
Affichage de l'interface qui correspond à l'objet MsoEnvelope.		X		MsoEnvelope	EnvelopeShow		
Après l'ajout d'un objet CustomXMLPart à la collection CustomXMLParts.		X		CustomXMLParts	PartAfterAdd		
Après le chargement d'un objet CustomXMLPart.		X		CustomXMLParts	PartAfterLoad		
Après la suppression d'un objet CustomXMLPart à la collection CustomXMLParts.		X		CustomXMLParts	PartBeforeDelete		
Après la suppression d'un nœud dans un objet CustomXMLPart.		X		CustomXMLPart	NodeAfterDelete		
Après l'ajout d'un nœud dans un objet CustomXMLPart.		X		CustomXMLPart	NodeAfterInsert		
Après le remplacement d'un nœud dans un objet CustomXMLPart.		X		CustomXMLPart	NodeAfterReplace		
Modification de la visibilité du volet Office personnalisé.		X		CustomTaskPane	DockPositionStateChange		

Événement	Word 2003	Word 2007	Nom de procédure dans ThisDocument	Module de classe : exécution selon la valeur de la donnée	Module de classe : Nom de la procédure d'événement	Interrompt l'opération	Macro automatique
Démarrage de Word		X					AutoExec
Modification d'une barre de commandes.	X			CommandBars	OnUpdate		
Clic sur un objet CommandBarButton	X			CommandBarButton	Click	Oui	
Changement de sélection dans une zone de liste déroulante CommandBar	X			CommandBarComboBox	Change		

