

# VBA EXCEL La MsgBox d'écriture

Par Fred (fring) 

Date de publication : 12 avril 2008

Dernière mise à jour : 25 juin 2008

La Message Box (MsgBox) est la méthode la plus rapide, la plus simple et la plus utilisée pour afficher un message.  
Cette méthode étant probablement connue par la plupart d'entre vous, nous allons malgré tout passer en revue les différentes propriétés que l'on peut lui appliquer et qu'il est parfois intéressant d'utiliser.

---

1 - La MsgBox dans sa plus simple expression.....	3
2 - La MsgBox multiligne.....	4
3 - Les combinaisons de boutons.....	5
3.1 - Combinaisons de 2 boutons.....	5
3.2 - Combinaisons de 3 boutons.....	6
4 - Les icônes.....	8
5 - Les propriétés particulières.....	9
5.1 - Bouton supplémentaire "Aide".....	9
5.2 - Définir le bouton qui aura le focus par défaut.....	9
5.3 - Alignement du texte à droite.....	10
5.4 - Mode de Pause.....	10
5.5 - Langues arabes.....	10
6 - Une MsgBox temporaire.....	11
7 - Une MsgBox étendue et entièrement paramétrable.....	12
8 - Remerciements.....	13
9 - Téléchargements.....	14

## 1 - La MsgBox dans sa plus simple expression

Afin d'afficher rapidement un message, il n'est pas nécessaire de préciser les paramètres, par défaut la MsgBox sera constituée de votre message et du bouton "Ok"

*Exemples d'utilisation :*

### VBA - MsgBox

```
Sub mess_01a()  
'texte directement introduit dans le code  
MsgBox "Ici votre message"  
End Sub
```

### VBA - MsgBox

```
Sub mess_01b()  
'texte provenant d'une cellule  
MsgBox Sheets("Feuil1").Range("A1")  
End Sub
```

### VBA - MsgBox

```
Sub mess_01c()  
'résultat d'une variable  
Dim Var As String  
Var = "Texte à afficher"  
MsgBox Var  
End Sub
```

*Aperçu :*



## 2 - La MsgBox multiligne

Pour afficher un texte sur plusieurs lignes, le passage à la ligne s'obtient en indiquant soit **Chr(10)**, soit **vbLf**, entouré du signe de concaténation **&**

Exemples d'utilisation :

### VBA - MsgBox

```
Sub mess_02a()
'texte directement introduit dans le code
MsgBox "Ici votre 1ère ligne" & Chr(10) & _
    "2ème ligne..." & Chr(10) & _
    "3ème ligne..." & Chr(10) & _
    "4ème ligne..." & Chr(10) & _
    "5ème ligne..." & Chr(10) & _
    "etc..."
End Sub
```

### VBA - MsgBox

```
Sub mess_02b()
'texte provenant de cellules
With Sheets("Feuill1")
MsgBox .Range("A1") & Chr(10) & _
    .Range("A2") & Chr(10) & _
    .Range("A3") & Chr(10) & _
    "etc..."
End With
End Sub
```

### VBA - MsgBox

```
Sub mess_02c()
'texte provenant de variables
Dim Var1 As String, Var2 As String, Var3 As String
Var1 = "Valeur de la variable 1"
Var2 = "Valeur de la variable 2"
Var3 = "Valeur de la variable 3"

MsgBox Var1 & vbLf & _
    Var2 & vbLf & _
    Var3 & vbLf & _
    "etc..."
End Sub
```

Aperçu :



## 3 - Les combinaisons de boutons

Les combinaisons de plusieurs boutons permettent de poser une question à l'utilisateur et de modifier le déroulement de la procédure selon son choix.

La syntaxe est la suivante : **MsgBox("Votre message", combinaison de boutons, "Titre de la MsgBox")**

Différentes combinaisons de boutons sont possible :

### 3.1 - Combinaisons de 2 boutons

3 combinaisons possibles :

- Boutons Ok + Annuler (constante : **vbOkCancel**)
- Boutons Oui + Non (constante : **vbYesNo**)
- Boutons Réessayer + Annuler (constante : **vbRetryCancel**)

*Exemples d'utilisation :*

#### VBA - MsgBox

```
Sub mess_03a()
'Msgbox Ok + Annuler
Select Case MsgBox("Votre message ici", vbOkCancel, "Titre de la MsgBox")
Case vbOK
'procédure si click sur Ok
Case vbCancel
'procédure si click sur Annuler
End Select
End Sub
```

#### VBA - MsgBox

```
Sub mess_03b()
'MsgBox Oui + Non
Select Case MsgBox("Votre message ici", vbYesNo, "Titre de la MsgBox")
Case vbYes
'procédure si click sur Oui
Case vbNo
'procédure si click sur Non
End Select
End Sub
```

#### VBA - MsgBox

```
Sub mess_03c()
'MsgBox Réessayer + Annuler
Select Case MsgBox("Votre message ici", vbRetryCancel, "Titre de la MsgBox")
Case vbRetry
'procédure si click sur Réessayer
Case vbCancel
'procédure si click sur Annuler
End Select
End Sub
```

*Aperçus :*



### 3.2 - Combinaisons de 3 boutons

2 combinaisons possibles :

- Boutons Oui + Non + Annuler (constante : **vbYesNoCancel**)
- Boutons Abandonner + Réessayer + Ignorer (constante : **vbAbortRetryIgnore**)

Exemples d'utilisation :

#### VBA - MsgBox

```
Sub mess_04a()
'MsgBox Oui + Non + Abandonner
Select Case MsgBox("Votre message ici", vbYesNoCancel, "Titre de la MsgBox")
Case vbYes
'procédure si click sur Oui
Case vbNo
'procédure si click sur Non
Case vbCancel
'procédure si click sur Annuler
End Select
End Sub
```

#### VBA - MsgBox

```
Sub mess_04b()
'MsgBox Abandonner + Réessayer + Ignorer
Select Case MsgBox("Votre message ici", vbAbortRetryIgnore, "Titre de la MsgBox")
Case vbAbort
'procédure si click sur Abandonner
Case vbRetry
'procédure si click sur Réessayer
Case vbIgnore
'procédure si click sur Ignorer
End Select
End Sub
```

Aperçus :







## 4 - Les icônes

En plus des combinaisons de boutons ci-dessus, vous pouvez ajouter une icône à votre MsgBox afin d'indiquer à l'utilisateur le type du message.

La syntaxe est la suivante : **MsgBox("Votre message", combinaison de boutons + icône, "Titre de la MsgBox")**

4 possibilités :

Question	Information	Exclamation	Arrêt critique
			
constante : <b>vbQuestion</b>	constante : <b>vbInformation</b>	constante : <b>vbExclamation</b>	constante : <b>vbCritical</b>

*Exemple d'utilisation :*

### VBA - MsgBox

```

Sub mess_05 ()
'MsgBox Oui + Non + Abandonner + icône Question
Select Case MsgBox("Votre message ici", vbYesNoCancel + vbQuestion, "Titre de la MsgBox")
Case vbYes
'procédure si click sur Oui
Case vbNo
'procédure si click sur Non
Case vbCancel
'procédure si click sur Annuler
End Select
End Sub
    
```

*Aperçu :*



## 5 - Les propriétés particulières

En plus de toutes les propriétés décrites ci-dessus, vous avez encore la possibilité de configurer quelques propriétés particulières.

### 5.1 - Bouton supplémentaire "Aide"

Via ce bouton **Aide**, vous pouvez afficher un fichier de type \*.hlp qui permettrait par exemple d'expliquer à l'utilisateur les différentes étapes de votre procédure.

*Pour de plus amples explications quant à la création d'un fichier d'aide, jetez un oeil [ICI](#).*

La constante de ce bouton est : **vbMsgBoxHelpButton**

Cette constante fait appel à 2 paramètres supplémentaires :

- **[HelpFile]** = le chemin complet et le nom du fichier \*.hlp
- **[Context]** = permet d'ouvrir le fichier d'aide à un chapitre prédéfini (*ce paramètre est facultatif*)

La syntaxe est la suivante : **MsgBox("Votre message", combinaison de boutons + icône + vbMsgBoxHelpButton, "Titre de la MsgBox", HelpFile, Context)**

*Exemple d'utilisation :*

```
VBA - MsgBox

Sub mess_06()
'MsgBox Ok + Help
Dim Fichier_Aide As String
Fichier_Aide = "C:\WINDOWS\Help\calc.hlp"
Select Case MsgBox("Votre message ici", vbInformation + vbMsgBoxHelpButton, "Titre de la MsgBox",
    Fichier_Aide, 1)
    Case vbOK
        'procédure si click sur Ok
End Select
End Sub
```

*Aperçu :*



### 5.2 - Définir le bouton qui aura le focus par défaut

Sans préciser ce paramètre, par défaut c'est le premier bouton de votre MsgBox qui recevra le focus.

Pour les MsgBox comportant plusieurs boutons (max.4 : combinaison de 3 boutons + bouton Aide), il peut être intéressant de définir à l'avance le bouton qui recevra le focus,

ce qui permet par exemple de définir l'action si l'utilisateur appuie sur la touche "Enter" sans avoir sélectionné un bouton.

Les constantes sont : **vbDefaultButton1** ou **vbDefaultButton2** ou **vbDefaultButton3** ou **vbDefaultButton4**.

Exemple de syntaxe : **MsgBox("Message", vbYesNoCancel + vbQuestion + vbMsgBoxHelpButton + vbDefaultButton3, "Titre", HelpFile, Context)**

*Dans cet exemple, c'est le bouton Annuler qui recevra le focus.*

## 5.3 - Alignement du texte à droite

Si vous souhaitez afficher le texte de votre MsgBox avec un alignement à droite, la constante à ajouter est **vbMsgBoxRight**.

Exemple de syntaxe : **MsgBox("Message", vbYesNoCancel + vbQuestion + vbMsgBoxRight, "Titre")**

Aperçu :



## 5.4 - Mode de Pause

Par défaut, lorsque la MsgBox s'affiche, uniquement l'application active (dans notre cas Excel) se met en pause tant que l'utilisateur n'aura pas précisé son choix.

Par défaut, la constante est **vbApplicationModal**, mais il n'est pas besoin de la spécifier

Vous pouvez également mettre en pause tout le système en ajoutant la constante **vbSystemModal**

## 5.5 - Langues arabes

Ce dernier paramètre permet de spécifier que le texte doit s'afficher de droite à gauche pour les systèmes en langues arabes.

La constante est : **vbMsgBoxRtlReading**

*Je n'ai pas encore eu l'occasion de tester ce paramètre afin d'en vérifier le fonctionnement.*

## 6 - Une MsgBox temporaire

Issu de la **FAQ Excel** (par **SilkyRoad**).

Il est possible d'afficher une MsgBox durant un temps prédéfini et de la supprimer automatiquement.

Exemple : la MsgBox s'affichera durant 3 secondes

### Code VBA

```
CreateObject("Wscript.shell").Popup "Le Message", 3, "Le Titre", vbExclamation
```

## 7 - Une MsgBox étendue et entièrement paramétrable

Cet outil, réalisé par **Arkham46**, permet de créer des MsgBox étendues affichant du texte formaté (RTF). Le texte des boutons peut également être modifié.

Tutoriel : **OFFICE : Fenêtres de message étendues**

## 8 - Remerciements

*Merci à :*

- Pierre Fauconnier pour m'avoir mis le pied à l'étrier, pour son écoute, sa gentillesse, sa patience et son aide précieuse
- bbil et heureux-oli pour m'avoir aidé à démêler les noeuds du kit xml
- JauB pour la relecture
- à toute l'équipe Office pour m'avoir accordé leur confiance et me permettre de publier ce premier article

## 9 - Téléchargements

### Fichier Demo