

Astuces et code VBA pour Exce

Rubriques

- [Classeurs \(Workbooks\)](#)
- [Contrôles et Formulaires](#)
- [Feuilles \(Worksheets\)](#)
- [Fichiers \(XL ou autres\)](#)
- [Fonctions](#)
- [Formules](#)
- [Graphiques](#)
- [Menus](#)
- [Sélection](#)
- [OLE Automation](#)
- [Excel <--> Access \(DAO\)](#)
- [Commandes diverses](#)
- [Programmes divers ...](#)
- [Téléchargement](#) sur site <http://ericrenaud.free.fr/>

Les classeurs Excel (Workbooks)

Page mise à jour le : 28/03/2001

Vous trouverez dans cette rubrique des exemples de programmes se rapportant aux classeurs.

- Ajouter des feuilles dans un classeur et les renommer.
- Enregistrer une copie du classeur (sans modifier le classeur ouvert).
- Fermer tous les classeurs ouverts sauf le classeur actif.
- Renommer les onglets d'un classeur (Ex. Mois1, Mois2, Mois3, etc...).
- Trier les onglets d'un classeur.

Ajouter des feuilles dans un classeur et les renommer

```
Sub AjouterRenommerFeuilles()  
Dim cpt As Integer  
cpt = 1  
Do While cpt < 4 ' Ajoute 3 feuilles  
' Ajout d'une feuille  
Application.Sheets.Add After:=Sheets.Item(Sheets.Count), Type:=xlWorksheet  
' Renomme la feuille  
Application.ActiveSheet.Name = "Semaine " & CStr(cpt)  
cpt = cpt + 1  
Loop  
End Sub
```

Enregistrer une copie du classeur

```
Sub SaveCopyAs()  
ActiveWorkbook.SaveCopyAs "C:\excel\MonDouble.xls"  
End Sub
```

Fermer tous les classeurs ouverts (sauf le classeur actif)

```
Sub FermeClasseurs()  
For Each Wk In Workbooks  
If Wk.Name <> ThisWorkbook.Name Then  
Wk.Close savechanges:=True  
End If  
Next Wk
```

```
End Sub
```

Renommer les onglets d'un classeur

```
Sub RenommeOnglets()  
' Renomme les onglets CL1, CL2, CL3, etc ...  
Dim I As Integer  
Application.ScreenUpdating = False  
For I = 1 To 3  
    Worksheets(I).Name = "CL" & I  
Next I  
End Sub
```

Trier les onglets d'un classeur

```
Sub TriNomsOnglets()  
Dim I As Integer, J As Integer  
For I = 1 To Sheets.Count  
    For J = 1 To I - 1  
        If UCCase(Sheets(I).Name) < UCCase(Sheets(J).Name) Then  
            Sheets(I).Move Before:=Sheets(J)  
        End If  
    Next J  
Next I  
End Sub
```

Contrôles divers en VBA

Page mise à jour le : 14/05/2002

Vous trouverez dans cette rubrique des exemples sur les différents contrôles comme TextBox, ComboBox, ListBox, Bouton d'option, Case à cocher, calendrier, indicateur de progression, etc...

Ces contrôles s'utilisent sur un Userform (ou formulaire)

Quelques notions de base ...

Comment créer un Userform ?

Depuis une feuille Excel, tapez **ALT+F11** ou par le menu **Outils, Macro, Visual Basic Editor**.

Vous affichez l'environnement Visual Basic Editor.

Insérer un Userform à l'aide du menu Insertion, Userform.

La boîte à outils doit s'afficher, sinon utilisez le menu **Affichage, Boîte à outils**.

Pour placer un contrôle sur un Userform :

- 1 - Sélectionner le contrôle désiré dans la fenêtre Boîte à outils
- 2 - Cliquer sur le contrôle choisi avec le bouton gauche de la souris et sans relâcher celui-ci, déplacer le contrôle vers le Userform puis relâcher le bouton de la souris.

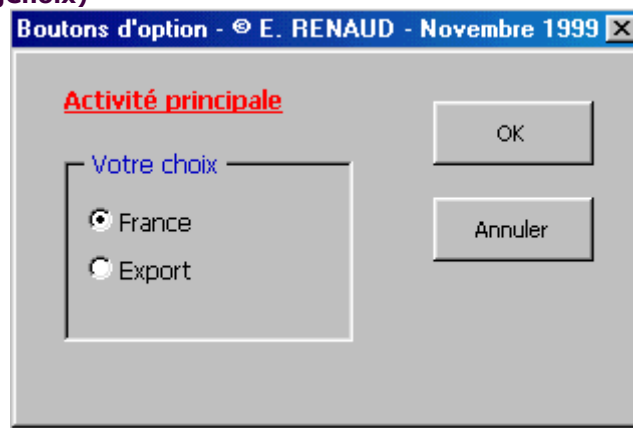
Voici les principaux contrôles de la boîte à outils (ci-dessous).



Les boîtes de dialogue

- Boite de dialogue avec bouton d'option.

Le Userform (DlgChoix)



Le code

[Code lançant l'ouverture de la boîte de dialogue](#)

```
Sub AfficheBoutonsOption()
    DlgChoix.Show
End Sub
```

[Code d'ouverture du formulaire](#)

```
Private Sub UserForm_Activate()
    DlgChoix.ChoixPaire.Value = True
End Sub
Nota : A l'ouverture du formulaire la case France est cochée par défaut.
```

[Code des boutons](#)

1 / Bouton OK

```
Private Sub OK_Click()
    DlgChoix.Hide
    If DlgChoix.ChoixPaire Then
        Msg = "Vous avez choisi l'activité France."
        Style = vbOKOnly + vbInformation
        Title = "Activité Commerciale - © E. RENAUD - Novembre 1999"
        Response = MsgBox(Msg, Style, Title)
    Else
        Msg = "Vous avez choisi l'activité Export."
        Style = vbOKOnly + vbInformation
```

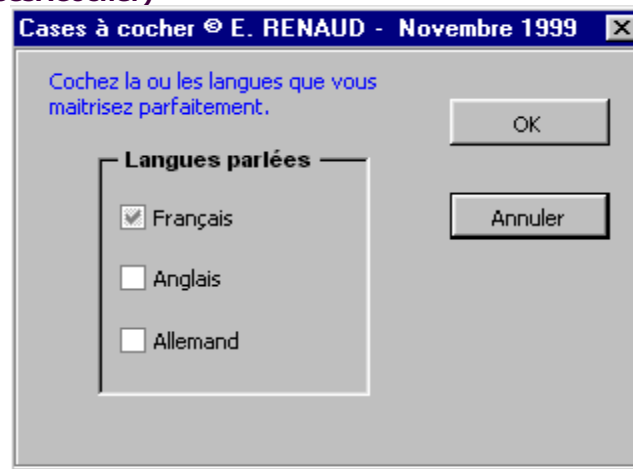
```
Title = "Activité Commerciale - © E. RENAUD - Novembre 1999"  
Response = MsgBox(Msg, Style, Title)  
End If  
End Sub
```

2 / Bouton Annuler

```
Private Sub Annuler_Click()  
DlgChoix.Hide  
End Sub
```

● Boite de dialogue avec case à cocher.

Le Userform (CasesACocher)



Le code

[Code lançant l'ouverture de la boîte de dialogue](#)

```
Sub AfficheCaseACocher()  
Range("A3:C3").Select  
Selection.ClearContents  
CasesACocher.Show  
End Sub
```

[Code d'ouverture du formulaire](#)

```
Private Sub UserForm_Activate()  
' Remplace la propriété TripleState mise à True(CheckBox1)  
CheckBox1.Value = Null  
CheckBox2.Value = False  
CheckBox3.Value = False  
End Sub
```

Nota : A l'ouverture du formulaire la case Français est coché mais en grisée.

[Code des boutons](#)

1 / Bouton OK

```
Private Sub OK_Click()  
Dim Langue1 As String  
Dim Langue2 As String  
Dim Langue3 As String  
If CheckBox1.Value = False Then  
Langue1 = "Non"  
Else
```

```

    Langue1 = "Oui"
End If
If CheckBox2.Value = False Then
    Langue2 = "Non"
Else
    Langue2 = "Oui"
End If
If CheckBox3 = False Then
    Langue3 = "Non"
Else
    Langue3 = "Oui"
End If
CasesACocher.Hide
Range("A3").Value = Langue1
Range("B3").Value = Langue2
Range("C3").Value = Langue3
Range("A1").Select
End Sub

```

2 / Bouton Annuler

```

Private Sub Annuler_Click()
    CasesACocher.Hide
End Sub

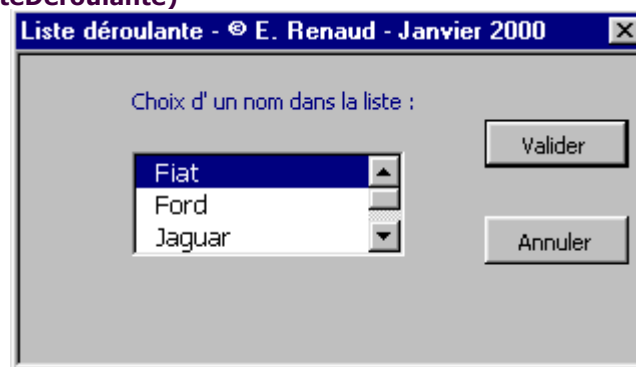
```

Le résultat

	A	B	C
1	Langues parlées		
2	Français	Anglais	Allemand
3	Oui	Non	Non

● Boite de dialogue avec liste déroulante.

Le Userform (ListeDeroulante)



Le code

Code lançant l'ouverture de la liste déroulante

```

Sub AfficheListeDeroulante()
    ListeDeroulante.Show
End Sub

```

Code d'ouverture du formulaire

```

Private Sub UserForm_Activate()
    DerniereMarque = Range("A1").End(xlDown).Address
    ' Plage de données pour afficher dans liste déroulante

```

```
Marque.RowSource = "A1:" & DerniereMarque
' Afficher la première marque de la liste (0 = Fiat)
Marque.ListIndex = 0
End Sub
```

Code des boutons

1 / Bouton Valider

```
Private Sub Valider_Click()
ListeDeroulante.Hide
' Marque est le nom donné au contrôle ListBox
Index = Marque.ListIndex
ChoixMarque = Marque.List(Index)
' Stockage du choix effectué dans la liste déroulante en D2
Range("D2").Value = ChoixMarque
End Sub
```

2 / Bouton Annuler

```
Private Sub Annuler_Click()
ListeDeroulante.Hide
End Sub
```

Les données utilisées pour remplir la liste déroulante

	A
1	Fiat
2	Ford
3	Jaguar
4	Lada
5	Maserati
6	Mercedes
7	Nissan
8	Peugeot
9	Porsche
10	Renault
11	Volkwasgen

- Boite de dialogue avec liste déroulante modifiable.

Le Userform (ListeDeroulanteModifiable)

Choisir ou Ajouter une marque dans la liste :

Fiat

Valider choix

Annuler

Les contrôles

Nom de la ComboBox = Constructeur
Nom du CommandButton "Valider Choix" = OK
Nom du CommandButton "Annuler" = Annuler

Le code

Code lançant l'ouverture de la boîte de dialogue

```
Sub AfficheListeDeroulanteModifiable()  
    ListeDeroulanteModifiable.Show  
End Sub
```

Code d'ouverture du formulaire

```
Private Sub UserForm_Activate()  
    ' Utilisation de l'option AddItem  
    Dim Arr() As String  
    Dim I As Integer, NbSheets As Integer  
    Dim Liste As Object  
    Set Liste = Worksheets(1).Cells(1, 1).Resize(Worksheets("Feuil1"). _  
        .Cells(1, 1).CurrentRegion.Rows.Count - 1, 1)  
    NbConstructeurs = Application.CountA(Range("A1").EntireColumn)  
    ListeDeroulanteModifiable.Constructeur.Clear  
    ReDim Arr(1 To NbConstructeurs)  
    For I = 1 To NbConstructeurs  
        Arr(I) = Liste(I).Value  
        ListeDeroulanteModifiable.Constructeur.AddItem Arr(I)  
    Next  
    Constructeur.ListIndex = 0  
End Sub
```

Code des boutons

1 / Bouton OK

```
Private Sub OK_Click()  
    ListeDeroulanteModifiable.Hide  
    Application.ScreenUpdating = False  
    ' Constructeur est le nom donné au contrôle ComboBox  
    Marque = Constructeur.Value  
    ' Stockage du choix effectué dans la liste déroulante en D2  
    Range("D2").Value = Marque  
    Columns("A:A").Select  
    On Error GoTo Ajoute  
    Selection.Find(What:=Marque, After:=ActiveCell, LookIn:=xlFormulas, _  
        LookAt:=xlPart, SearchOrder:=xlByRows, SearchDirection:=xlNext, _  
        MatchCase:=False).Activate  
    Range("A1").Select  
    Exit Sub  
Ajoute:  
    ' Une nouvelle marque étant ajoutée, ajout de celle-ci à la fin de la colonne A  
    Range("A1").End(xlDown).Offset(1, 0).Value = Marque  
    ' Tri des marques en ordre croissant  
    Selection.Sort Key1:=Range("A1"), Order1:=xlAscending, Header:=xlGuess, _  
        OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom  
    ' Sélectionne la plage et recolore la plage des marques après ajout nouvelle marque  
    With Worksheets(1)  
        .Range(.A1, .A1).End(xlDown).Select  
    End With  
    With Selection.Interior  
        .ColorIndex = 20  
        .PatternColorIndex = xlAutomatic  
    End With  
    Range("A1").Select  
End Sub
```

2 / Bouton Annuler

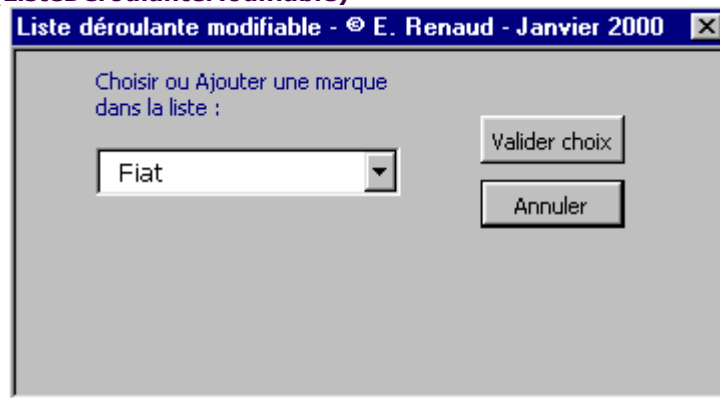
```
Private Sub Annuler_Click()  
    ListeDeroulanteModifiable.Hide  
End Sub
```

Les données utilisées pour remplir la liste déroulante

	A
1	Fiat
2	Ford
3	Jaguar
4	Lada
5	Maserati
6	Mercedes
7	Nissan
8	Peugeot
9	Porsche
10	Renault
11	Volkswagen

- Boite de dialogue avec liste déroulante modifiable.

Le Userform (ListeDeroulanteModifiable)



Les contrôles

Nom de la ComboBox = Constructeur
Nom du CommandButton "Valider Choix" = OK
Nom du CommandButton "Annuler" = Annuler

Le code

Code lançant l'ouverture de la boîte de dialogue

```
Sub AfficheListeDeroulanteModifiable()  
    ListeDeroulanteModifiable.Show  
End Sub
```

Code d'initialisation du formulaire

```
Private Sub UserForm_Initialize()  
    Me.Constructeur.RowSource = "Feuil1!A1:A" & Sheets("Feuil1").Cells(1, 1).End(xlDown).Row  
End Sub
```

Code d'activation du formulaire

```
Private Sub UserForm_Activate()  
    Me.Constructeur.ListIndex = -1  
End Sub
```


Code des boutons

1 / Bouton OK

```
Private Sub OK_Click()  
    Me.Hide  
    If Me.Constructeur.ListIndex = -1 Then  
        Sheets("Feuil1").Cells(1, 1).End(xlDown).Offset(1, 0).Value = Me.Constructeur.Value  
    ' Tri de la plage de données suite à l'ajout d'un élément  
    Sheets("Feuil1").Range([A1], [A1].End(xlDown)).Select  
    Selection.Sort Key1:=Range("A1"), Order1:=xlAscending, Header:=xlGuess, _  
        OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom  
    Else  
        End If  
    ' Mise en place de la valeur choisie en D2  
    Sheets("Feuil1").Cells(2, 4).Value = Me.Constructeur.Value  
End Sub
```

2 / Bouton Annuler

```
Private Sub Annuler_Click()  
    ListeDeroulanteModifiable.Hide  
End Sub
```

Les données utilisées pour remplir la liste déroulante

	A
1	Fiat
2	Ford
3	Jaguar
4	Lada
5	Maserati
6	Mercedes
7	Nissan
8	Peugeot
9	Porsche
10	Renault
11	Volkwasgen

- Boite de dialogue avec sélection multiple dans liste déroulante.

Le Userform (DlgChoixMultiple)



Les contrôles

Nom de la ListBox = ListeClients
 Nom du CommandButton "OK" = OK
 Nom du CommandButton "Annuler" = Annuler

Le code

Code lançant l'ouverture de la boîte de dialogue

```
Sub AfficheListeDeroulanteMultiSelect()
  DlgChoixMultiple.Show
End Sub
```

Code d'initialisation du formulaire

```
Private Sub UserForm_Initialize()
  ' Les données proviennent de la feuille Feuil1 et sont dans les colonnes A et B
  With Sheets("Feuil1").Range("A1")
    Me.ListeClients.RowSource = Range(.Cells, .End(xlDown)(1, 2)).Address(External:=True)
  End With
End Sub
```

Code d'activation du formulaire

```
Private Sub UserForm_Activate()
  ' Effacement des données de la plage qui reçoit les items sélectionnés (colonnes D et E)
  Range("D1:E1", Cells(ListeClients.ListCount, 4)).Clear
  ' Propriété Multiselect du contrôle ListeClients mis sur fmMultiSelectExtended
  ListeClients.MultiSelect = fmMultiSelectExtended
End Sub
```

Code des boutons

1 / Bouton OK

```
Private Sub OK_Click()
  DlgChoixMultiple.Hide
  j = 1
  With ListeClients
    For i = 0 To .ListCount - 1
      If .Selected(i) = True Then
        ' Le résultat est placé dans les colonnes D et E de la feuille Feuil1
        Cells(j, 4) = .List(i, 0): Cells(j, 5) = .List(i, 1)
        .Selected(i) = False
        j = j + 1
      End If
    Next
  End With
End Sub
```

2 / Bouton Annuler

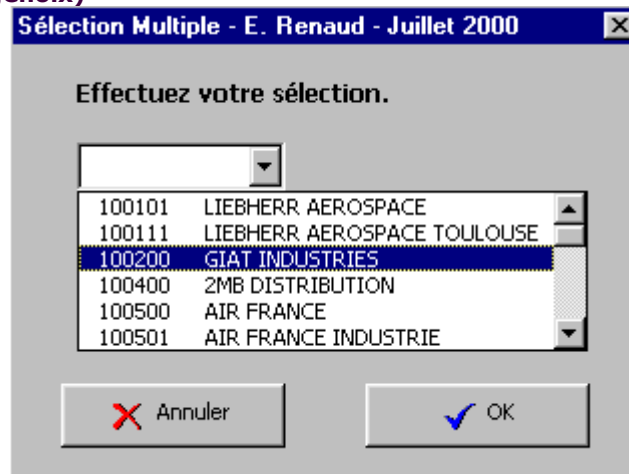
```
Private Sub Annuler_Click()
  DlgChoixMultiple.Hide
End Sub
```

Le résultat par rapport à la sélection dans le formulaire (voir image du Userform au début de la page)

D	E
100101	LIEBHERR AEROSPACE
100200	GIAT INDUSTRIES
100500	AIR FRANCE

- Boite de dialogue avec liste déroulante à 2 colonnes.

Le Userform (DlgChoix)



Les contrôles

Nom de la ComboBox = ListeClients
 Nom du CommandButton "OK" = OK
 Nom du CommandButton "Annuler" = Annuler

Quelques propriétés du contrôle ListeClients

ColumnCount = 2 (Nbre de colonne à afficher dans une zone de liste)
 ColumnWidths = 40 (Largeur de chaque colonne d'une zone de liste à plusieurs colonnes)
 Height = 18 (Hauteur en points d'un objet)
 ListWidth = 190 (Largeur de la liste d'un contrôle ComboBox)
 Width = 78 (largeur en points d'un objet)

Le code

[Code lançant l'ouverture de la boîte de dialogue](#)

```
Sub AfficheListeDeroulante()  
    DlgChoix.Show  
End Sub
```

[Code d'initialisation du formulaire](#)

```
Private Sub UserForm_Initialize()  
    ' Les données proviennent de la feuille Feuil1 et sont dans les colonnes A et B  
    With Sheets("Feuil1").Range("A1")  
        Me.ListeClients.RowSource = Range(.Cells, .End(xlDown)(1, 2)).Address(External:=True)  
    End With  
End Sub
```

[Code d'activation du formulaire](#)

```
Private Sub UserForm_Activate()  
    ' Effacement des données de la plage qui reçoit les items sélectionnés (colonnes D et E)  
    Range("D1:E1", Cells(ListeClients.ListCount, 4)).Clear  
End Sub
```

[Code des boutons](#)

1 / Bouton OK

```

Private Sub OK_Click()
    DlgChoix.Hide
    If ListeClients.ListIndex <> -1 Then
        ' Récupération du code et du nom du client
        Sheets("Feuil1").Cells(1, 4).Value = ListeClients.Column(0, ListeClients.ListIndex)
        Sheets("Feuil1").Cells(1, 5) = ListeClients.Column(1, ListeClients.ListIndex)
    End If
End Sub

```

2 / Bouton Annuler

```

Private Sub Annuler_Click()
    DlgChoix.Hide
End Sub

```

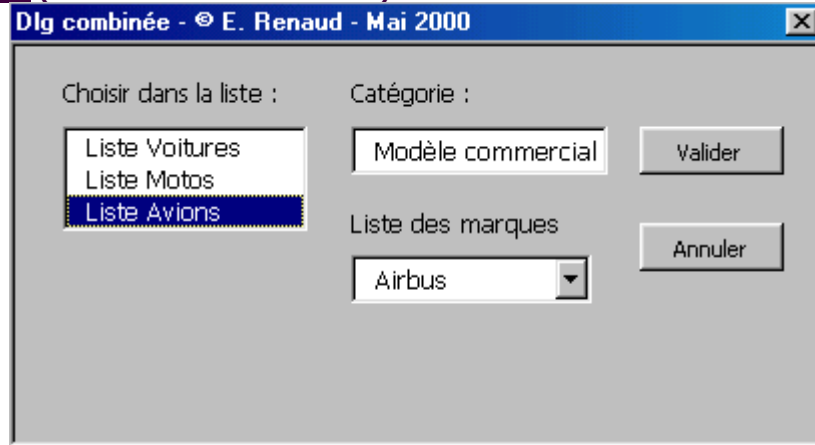
[Le résultat par rapport à la sélection dans le formulaire \(voir image du Userform au début de la page\)](#)

D	E
100200	GIAT INDUSTRIES

● Boite de dialogue combinée.

Cette boîte de dialogue permet de mettre à jour les deux contrôles de droite (TextBox et ComboBox) suivant la sélection effectuée dans le contrôle de gauche (ListBox) et cela de manière dynamique.

Le Userform (ListeDeroulanteCombinee)



Le code

[Code lançant l'ouverture du Userform \(ListeDeroulanteCombinee\)](#)

```

Sub AfficheListeDeroulanteCombinee()
    Sheets("Accueil").Activate
    Range("B2").Value = ""
    Range("C2").Value = ""
    Range("D2").Value = ""
    Application.ScreenUpdating = False
    ListeDeroulanteCombinee.Show
End Sub

```

[Code d'ouverture du formulaire par l'événement Activate](#)

```

Private Sub UserForm_Activate()
Dim Arr() As String
Dim I As Integer, NbSheets As Integer
Dim Liste As Object
' Suppression des entrées de la liste si celle-ci en contient
If Onglets.ListCount >= 1 Then
Dim ElementListe As Integer
Dim NbreIt As Integer
NbreIt = Onglets.ListCount - 1
For ElementListe = NbreIt To 0 Step -1
Onglets.RemoveItem (ElementListe)
Next ElementListe
End If
' Ajout de toutes les entrées de la liste
NbSheets = Worksheets.Count
ReDim Arr(1 To NbSheets)
For I = 2 To NbSheets
Arr(I) = Worksheets(I).Name
ListeDeroulanteCombinee.Onglets.AddItem Arr(I)
Next I
Onglets.ListIndex = 0
End Sub

```

Code sur l'événement Change de la liste déroulante Onglets (Contrôle situé à gauche)

```

Private Sub Onglets_Change()
Dim OngletSelect As Integer
' Déterminer la ligne sélectionnée dans la liste déroulante
OngletSelect = ListeDeroulanteCombinee.Onglets.ListIndex + 2
' Mise à jour TextBox Catégorie (contrôle en haut et à droite)
ListeDeroulanteCombinee.Categorie = Sheets(OngletSelect).Range("C1").Value
' Mise à jour ComboBox Marque (contrôle en bas et à droite)
Dim DerniereMarque As String
Sheets(OngletSelect).Activate
DerniereMarque = Range("A1").End(xlDown).Address
Marque.RowSource = "A1:" & DerniereMarque
Marque.ListIndex = 0
End Sub

```

Code des boutons

1 / Bouton Valider

```

Private Sub Valider_Click()
ListeDeroulanteCombinee.Hide
' Récupération des index en fonction de la sélection
IndexOnglet = Onglets.ListIndex
IndexModele = Marque.ListIndex
' Récupération des valeurs en fonction des index
Onglet = Onglets.List(IndexOnglet)
Modele = Marque.List(IndexModele)
Rubrique = Categorie.Value
' Mise en place des valeurs dans la feuille de calcul
Sheets("Accueil").Activate
Range("B2").Value = Onglet
Range("C2").Value = Rubrique
Range("D2").Value = Modele
End Sub

```

2 / Bouton Annuler

```

Private Sub Annuler_Click()
ListeDeroulanteCombinee.Hide
End Sub

```

Les données utilisées pour remplir la liste déroulante Onglet Liste Voitures

	A	B	C	D
1	Fiat		Modèle tourisme	
2	Ford			
3	Jaguar			
4	Lada			
5	Maserati			
6	Mercedes			
7	Nissan			
8	Peugeot			
9	Porsche			
10	Renault			
11	Volkswagen			

Onglet Liste Motos

	A	B	C	D
1	BMW		Modèle route	
2	Ducati			
3	Harley Davidson			
4	Honda			
5	Kawasaki			
6	KTM			
7	Yamaha			

Onglet Liste Avions

	A	B	C	D
1	Airbus		Modèle commercial	
2	Antonov			
3	Boeing			
4	Dassault			
5	Learjet			
6	Lookeed			

Un exemple de sélection avec le résultat inscrit dans la feuille de calcul

Dlg combinée - © E. Renaud - Mai 2000 Tfc: 7.18 / 00:25:40 / 4...

Choisir dans la liste :

- Liste Voitures
- Liste Motos
- Liste Avions

Catégorie :

Modèle route

Valider

Liste des marques

Ducati

BMW

Ducati

Harley Davidson

Honda

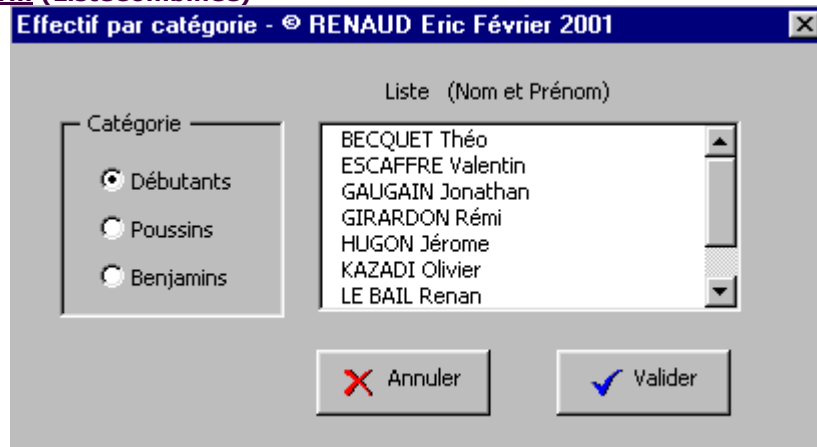
Annuler

	B	C	D
1	Onglet	Rubrique	Modèle
2	Liste Motos	Modèle route	Ducati

● Boite de dialogue combinée. ◀

Cette boîte de dialogue permet en cochant une des options du contrôle de gauche (Catégorie), d'afficher la liste correspondante dans le contrôle de droite (Liste avec Noms et Prénoms)

Le Userform (ListeCombinée)



Le code

[Code lançant l'ouverture du Userform \(ListeCombinée\)](#)

```
Sub AfficheListeCombinée()  
    ListeCombinée.Show  
End Sub
```

[Code d'ouverture du formulaire par l'événement Activate](#)

```
Private Sub UserForm_Activate()  
    ' Permet de cocher le bouton radio 1 à l'ouverture du formulaire  
    Me.OptionButton1.Value = True  
End Sub
```

[Code sur l'événement Click des boutons d'option de la zone Catégorie \(Contrôle situé à gauche\)](#)

```
Private Sub OptionButton1_Click()  
    LastInputRow = Cells(1, 1).End(xlDown).Row  
    ListBox1.RowSource = "Effectif!A1: A" & LastInputRow  
    MyCategorie = "Débutants"  
    Me.ListBox1.ListIndex = 0  
    ' Mise à jour libellé liste déroulante  
    With ListeCombinée  
        .Label1.Caption = "Liste des " & MyCategorie & " (Nom et Prénom)"  
    End With  
End Sub
```

```
Private Sub OptionButton2_Click()  
    LastInputRow = Cells(1, 2).End(xlDown).Row  
    ListBox1.RowSource = "Effectif!B1: B" & LastInputRow  
    MyCategorie = "Poussins"  
    Me.ListBox1.ListIndex = 0  
    ' Mise à jour libellé liste déroulante  
    With ListeCombinée  
        .Label1.Caption = "Liste des " & MyCategorie & " (Nom et Prénom)"  
    End With  
End Sub
```

```
Private Sub OptionButton3_Click()  
    LastInputRow = Cells(1, 3).End(xlDown).Row  
    ListBox1.RowSource = "Effectif!C1: C" & LastInputRow  
    MyCategorie = "Benjamins"  
    Me.ListBox1.ListIndex = 0
```

```
' Mise à jour libellé liste déroulante
  With ListeCombinee
    .Label1.Caption = "Liste des " & MyCategorie & " (Nom et Prénom)"
  End With
End Sub
```

Code des boutons

1 / Bouton Valider

```
Private Sub CmdValider_Click()
  Me.Hide
  MySelection = Me.ListBox1.List(ListBox1.ListIndex)
  MsgBox "Vous avez choisi le joueur " & MySelection & "." & vbCrLf & _
    "Il appartient à la catégorie des " & MyCategorie & "."
  Unload Me
End Sub
```

2 / Bouton Annuler

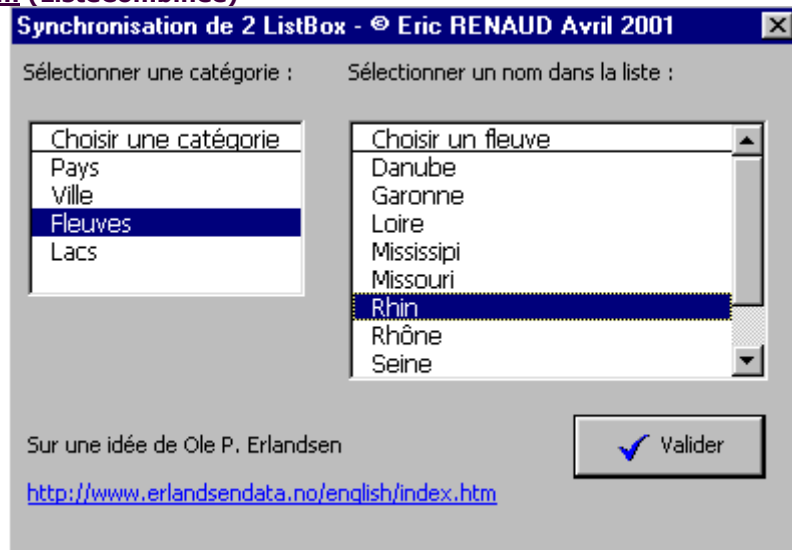
```
Private Sub Annuler_Click()
  Me.Hide
  Unload Me
End Sub
```

● Boite de dialogue avec 2 ListBox synchronisées.

Sur une idée de [Ole P. Erlandsen](#).

Cette boîte de dialogue permet en sélectionnant un élément dans la liste de gauche, d'afficher les éléments correspondants dans la liste de droite.

Le Userform (ListeCombinee)



Le code

Code lançant l'ouverture du Userform (DoubleListBox)

```
Sub AfficheListBox()
  Load DoubleListBox
  DoubleListBox.Show
  Unload DoubleListBox
End Sub
```


Code d'ouverture du formulaire par l'événement Initialize

```
Private Sub UserForm_Initialize()  
' La mise à jour des items dans ListBox1 ...  
' mettra à jour les items dans la ListBox2  
    With Me  
        UpdateListBox .ListBox1, -1  
    End With  
End Sub
```

Code sur l'événement Change de la ListBox1 (Contrôle situé à gauche)

```
Private Sub ListBox1_Change()  
' Mise à jour des items dans la ListBox2  
    UpdateListBox Me.ListBox2, Me.ListBox1.ListIndex  
End Sub
```

```
Private Sub UpdateListBox(Parametres As MSForms.ListBox, IndexValue As Integer)  
Dim LastInputRow As Integer, ColumnIndex As Integer, InputRange As Range  
' Les données commencent à la ligne 3  
Const FirstInputRow As Integer = 3  
' Détermine depuis quelle colonne on prend la liste des items  
    ColumnIndex = IndexValue + 2  
' Détermine la dernière ligne de la colonne sélectionnée et la plage correspondante  
    LastInputRow = Cells(FirstInputRow, ColumnIndex).End(xlDown).Row  
    Set InputRange = ActiveSheet.Range(Cells(FirstInputRow, ColumnIndex), Cells(LastInputRow, ColumnIndex))  
    With Parametres  
        .ColumnHeads = True ' Affiche les en-têtes de colonne  
        .RowSource = InputRange.Address ' Spécifie la source de données  
        .ListIndex = 0 ' Sélectionne le premier item  
    End With  
    Set InputRange = Nothing  
End Sub
```

Code du bouton valider

1 / Bouton Valider

```
Private Sub CmdValider_Click()  
    Me.Hide  
    MsgBox "Dans la catégorie : " & Me.ListBox1.List(Me.ListBox1.ListIndex) & Chr(13) & Chr(13) & _  
        "Vous avez choisi : " & Me.ListBox2.List(Me.ListBox2.ListIndex), vbInformation, "Résultat de votre choix :"  
End Sub
```

La feuille de données

	A	B	C	D	E
1	Catégories (ListBox1)	Pays (ListBox2)	Villes (ListBox2)	Fleuves (ListBox2)	Lacs (ListBox2)
2	Choisir une catégorie	Choisir un pays	Choisir une ville	Choisir un fleuve	Choisir un lac
3	Pays	Allemagne	Amsterdam	Danube	Genève
4	Ville	Angleterre	Athènes	Garonne	Bourget
5	Fleuves	Belgique	Bruxelles	Loire	Chambon
6	Lacs	Ecosse	Dublin	Mississippi	Erié
7		Espagne	Genève	Missouri	Uron
8		France	Istanbul	Rhin	Michigan
9		Grèce	Lisbonne	Rhône	Ontario
10		Irlande	Londres	Seine	Supérieur
11		Italie	Madrid	Volga	
12		Luxembourg	Munich	Zambèze	
13		Pays-Bas	Paris		
14		Portugal	Rome		
15		Sicile			
16		Suisse			
17		Turquie			

● Boîte de dialogue avec un contrôle TabStrip (contrôle Onglet). ◀

[Quelques infos à propos du Contrôle TabStrip](#)

Un contrôle TabStrip est un contrôle contenant une collection d'un ou de plusieurs onglets.

Dans les contrôles TabStrip, ceux-ci sont indépendants des pages. En réalité un clic sur un onglet n'affiche pas de nouvelle page.

Pour les contrôles TabStrip, la propriété Value permet de déterminer ou de définir quel onglet est actif.

Cet exemple permet en cliquant sur l'un des onglets, d'avoir instantanément les chiffres correspondants à l'activité indiquée sur l'onglet.

Le Userform (Avec 3 onglets)

	1er trimestre	2e trimestre	3e trimestre	4e trimestre	Total
	12500	14000	11250	13250	51000

Le code

Code lançant l'ouverture de la boîte de dialogue

```
Sub AfficheTabStrip()  
    UserForm1.Show  
End Sub
```

Code d'ouverture du formulaire

```
Private Sub UserForm_Activate()  
    Me.TabStrip1.Value = 0  
    For I = 1 To 5  
        Me.Controls("TextBox" & I).Value = Sheets("Résultats").Cells(I + 1, 2).Value  
    Next I  
End Sub
```

Code sur l'événement Change

```
Private Sub TabStrip1_Change()  
    Select Case TabStrip1.Value  
        Case 0  
            For I = 1 To 5  
                Me.Controls("TextBox" & I).Value = Sheets("Résultats").Cells(I + 1, 2).Value  
            Next I  
        Case 1  
            For I = 1 To 5  
                Me.Controls("TextBox" & I).Value = Sheets("Résultats").Cells(I + 1, 3).Value  
            Next I  
        Case 2  
            For I = 1 To 5  
                Me.Controls("TextBox" & I).Value = Sheets("Résultats").Cells(I + 1, 4).Value  
            Next I  
    End Select  
End Sub
```

Code du bouton Fermer

Bouton Fermer


```
Private Sub CmdFermer_Click()  
    Me.Hide  
End Sub
```

Les données de la feuille Résultats utilisées pour remplir le UserForm.

	A	B	C	D
1		Matériel	Logiciels	Formation
2	1er trimestre	12500	25000	35000
3	2e trimestre	14000	18540	27500
4	3e trimestre	11250	13200	26000
5	4e trimestre	13250	17800	24600
6	Total	51000	74540	113100

Les contrôles courants

Exemples d'utilisation de InputBox

 Page mise à jour le : 19/03/2003

InputBox peut être utilisée soit comme fonction, soit comme méthode.

La **fonction Inputbox** permet d'afficher une boîte de dialogue contenant une zone de texte légendée, invitant l'utilisateur à y saisir l'information attendue. L'information saisie est renvoyée sous forme de chaîne de caractères et stockée dans une variable afin de pouvoir être ensuite exploitée par le programme.

L'objet **Application** d'Excel possède une méthode InputBox que l'on peut substituer à la fonction InputBox de Visual Basic.

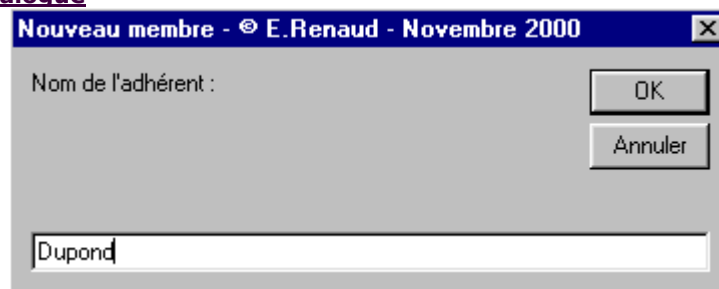
La **méthode Inputbox** se présente comme la fonction InputBox mais offre une possibilité très intéressante : elle permet de **spécifier le type de données** qui sera renvoyée (numérique, plage).

- Saisie d'information via une InputBox (fonction).
- Saisie d'information (valeur numérique) via une InputBox (méthode).
- Saisie d'information (plage de cellules) via une InputBox (méthode).

● Saisie d'information via une InputBox (fonction)

Cette boîte de dialogue avec une zone de texte légendée, invite l'utilisateur à y saisir l'information attendue. L'information saisie est renvoyée sous forme de chaîne de caractères et stockée dans une variable afin de pouvoir être ensuite exploitée par le programme.

La boîte de dialogue



Le code

[Code lançant l'ouverture de la boîte de dialogue InputBox](#)

```
Sub UtilisationDeInputboxFonction()  
    Dim Inscription As String  
    Inscription = InputBox("Nom de l'adhérent :", "Nouveau membre")  
    ' Si l'utilisateur n'a saisie aucune donnée et qu'il clique sur OK,  
    ' ou si l'utilisateur clique sur Annuler, on quitte la procédure.  
    If Inscription = "" Then  
        MsgBox "Aucune donnée n'a été saisie"  
        Exit Sub  
    ' Sinon la donnée saisie est affichée  
    Else  
        MsgBox Inscription  
    End If
```

End Sub

● Saisie d'information (valeur numérique) via une InputBox (méthode).

Cette boîte de dialogue avec une zone de texte légendée, invite l'utilisateur à y saisir l'information attendue sous forme numérique (**type:=1**). L'information saisie est renvoyée sous forme de nombre et stockée dans une variable afin de pouvoir être ensuite exploitée par le programme.

La boîte de dialogue



Le code

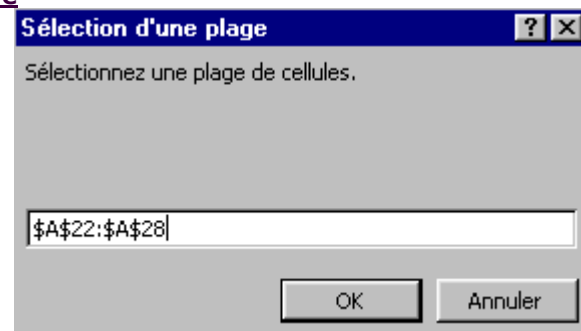
[Code lançant l'ouverture de la boîte de dialogue InputBox](#)

```
Sub UtilisationDelInputboxMethode()  
Dim Recette As Integer  
Recette = Application.InputBox("Recette effectuée :", "Saisie de la recette", Type:=1)  
' Si l'utilisateur a cliqué sur Annuler ...  
If Recette = False Then Exit Sub  
' sinon la donnée saisie est affichée  
MsgBox "La recette est de : " & Recette  
End Sub
```

● Saisie d'information (plage de cellules) via une InputBox (méthode).

Cette boîte de dialogue avec une zone de texte légendée, invite l'utilisateur à y saisir l'information attendue sous forme de sélection de plage (**type:=8**). L'information saisie est renvoyée sous forme de plage et stockée dans une variable afin de pouvoir être ensuite exploitée par le programme.

La boîte de dialogue



Le code


[Code lançant l'ouverture de la boîte de dialogue InputBox](#)

```
Sub UtilisationDelInputboxMethode()  
Dim MaPlage As Range  
On Error Resume Next  
Set MaPlage = Application.InputBox(prompt:="Sélectionnez une plage de cellules.", _  
Title:="Sélection d'une plage", Left:=5, Top:=5, Type:=8)  
' Si l'utilisateur a cliqué sur le bouton Annuler ...  
If Err.Number = 424 Then  
MsgBox "Vous avez choisi d'annuler"
```

```
End
Else
' sinon la plage sélectionnée va être affichée
MsgBox "La plage sélectionnée est : " & MaPlage.Address
End If
End Sub
```

Pour plus d'informations, consultez l'aide sur [InputBox](#).

Exemples de la fonction MsgBox

 Page mise à jour le : 02/11/2000

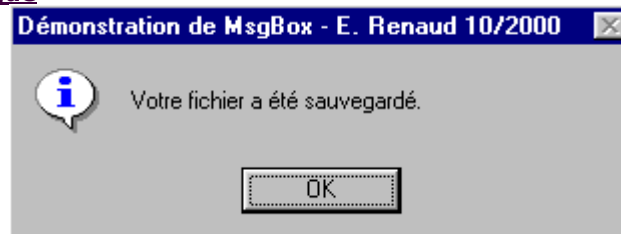
Vous trouverez dans cette rubrique des exemples de la fonction MsgBox. La fonction MsgBox permet d'afficher une boîte de dialogue présentant un message et des boutons de commande, afin d'afficher une information à l'attention de l'utilisateur. ou d'obtenir une réponse à une question qui orientera l'exécution du programme.

- Message d'information via une MsgBox.
- Message d'information via une MsgBox avec choix de l'action à effectuer (2 boutons).
- Message d'information via une MsgBox avec choix de l'action à effectuer (3 boutons).

● Message d'information via une MsgBox.

Cette boîte de dialogue affiche un message, attend que l'utilisateur clique sur le bouton pour valider la lecture de ce message.

La boîte de dialogue



Le code

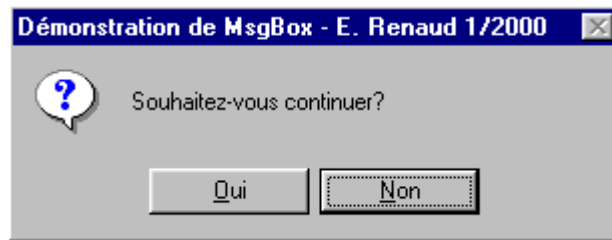
[Code lançant l'ouverture de la boîte de dialogue MsgBox](#)

```
Sub MessageInformation()
' Affiche un message d'information
Msg = "Votre fichier a été sauvegardé."
Title = "Démonstration de MsgBox - E. Renaud 10/2000"
Style = vbOKOnly + vbInformation
Reponse = MsgBox(Msg, Style, Title)
End Sub
```

● Message d'information via une MsgBox avec choix de l'action à effectuer.

Cette boîte de dialogue affiche un message, attend que l'utilisateur clique sur un bouton, puis renvoie une valeur de type Integer qui indique le bouton choisi par l'utilisateur.

La boîte de dialogue



Le code

[Code lançant l'ouverture de la boîte de dialogue MsgBox](#)

```

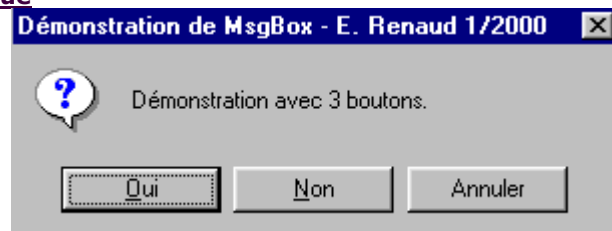
Sub RecupInfoMessage()
Dim Msg, Style, Title, Response
' Définit le message.
Msg = "Souhaitez-vous continuer?"
' Définit les boutons.
Style = vbYesNo + vbQuestion
' Définit le titre.
Title = "Démonstration de MsgBox - E. Renaud 1/2000"
' Affiche le message.
Response = MsgBox(Msg, Style, Title)
If Response = vbYes Then ' L'utilisateur a choisi Oui.
    MyString = "Oui" ' Effectue une action.
Else ' L'utilisateur a choisi Non.
    MyString = "Non" ' Effectue une action.
End If
End Sub

```

● Message d'information via une MsgBox avec choix de l'action à effectuer.

Cette boîte de dialogue affiche un message, attend que l'utilisateur clique sur un bouton, puis renvoie une valeur de type Integer qui indique le bouton choisi par l'utilisateur.

La boîte de dialogue



Le code

[Code lançant l'ouverture de la boîte de dialogue MsgBox](#)

```

Sub MessageAvecReponseOuAnnulation()
' Affiche un message d'information avec test sur le bouton sélectionné
Msg = "Démonstration avec 3 boutons."
Title = "Exemple de boîte de message"
Style = vbYesNoCancel + vbQuestion
Reponse = MsgBox(Msg, Style, Title)
' Si le bouton Annuler a été choisi ... on quitte la procédure
If Reponse = vbCancel Then Exit Sub
' sinon on teste lequel du bouton Oui ou Non a été choisi
If Reponse = vbYes Then
    MsgBox "Vous avez sélectionné Oui"
Else
    MsgBox "Vous avez sélectionné Annuler"
End If
End Sub

```

Pour plus d'informations, consultez l'aide sur MsgBox.

Contrôles avancés

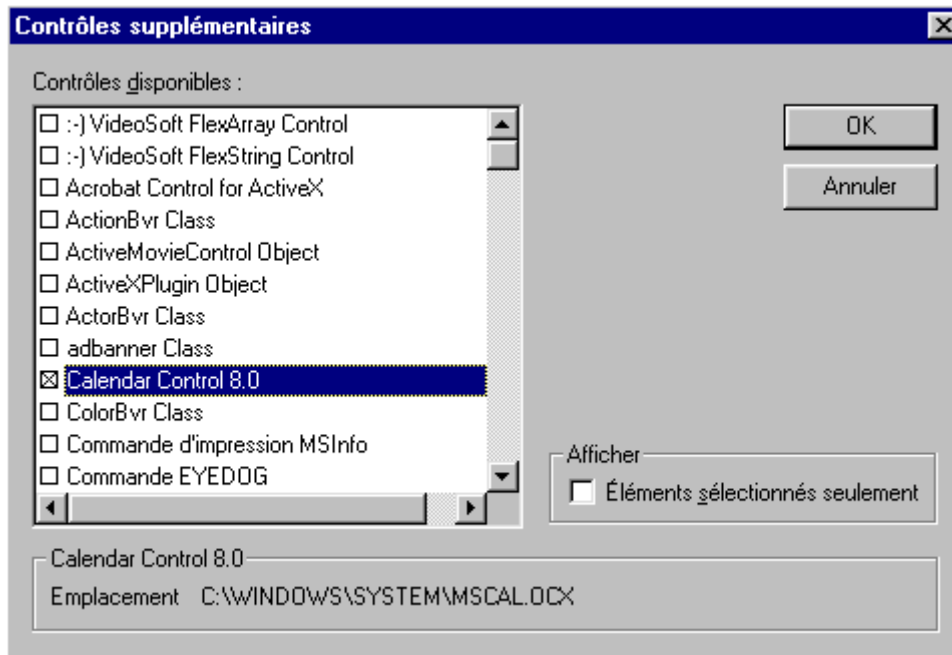
● Le contrôle Calendar (calendrier).

Ce contrôle permet grâce à un calendrier de saisir des dates de début et fin pour les utiliser dans votre application.

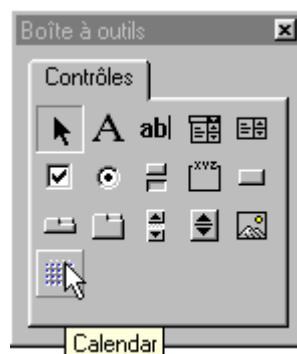
Exemple trouvé dans livre Excel 2000 & Visual Basic pour Applications 6 disponible chez Amazon.fr

Ajout du contrôle Calendar

- 1 - Activer un UserForm dans le Visual Basic Editor.
- 2 - Faites un clic droit sur la boîte à outils et sélectionnez Contrôles supplémentaires.
- 3 - Dans la liste déroulante, sélectionnez Calendar Control 8.0.
- 4 - Cliquez sur OK pour fermer la boîte de dialogue et ajoutez le contrôle dans la boîte à outils.



La boîte à outils



Le Userform (fmContratDates)

Dates de début et fin [X]

Indiquez la date à valider et sélectionnez-la sur le calendrier.

Date de début :

Date de fin :

Date à valider :

January 2000

Mon	Tue	Wed	Thu	Fri	Sat	Sun
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
31	1	2	3	4	5	6

Le code

[Code lançant l'ouverture de la boîte de dialogue](#)

```
Sub AfficheCalendar()
    fmContratDates.Show
End Sub
```

[Code d'ouverture du formulaire](#)

```
Private Sub UserForm_Initialize()
    cboDateAValider.AddItem ("Date de début")
    cboDateAValider.AddItem ("Date de fin")
    cboDateAValider.ListIndex = 0
    Calendrier.Value = Date
    Calendrier.SetFocus
End Sub
```

[Code lié au choix des dates sur le contrôle Calendar](#)

```
Private Sub Calendrier_Click()
    ' Choix d'une date sur le calendrier
    If cboDateAValider.ListIndex = 0 Then
        DateDebut.Value = Calendrier.Value
        cboDateAValider.ListIndex = 1
    Else
        DateFin.Value = Calendrier.Value
    End If
End Sub
```

[Code des boutons](#)

1 / Bouton Valider (cmdValider)

```
Private Sub CmdValider_Click()
    ' Vérification de la validité des informations
    If DateDebut.Value = "" Then
```

```

MsgBox "Vous devez spécifier une date de début.", vbOKOnly + vbCritical, "Contrat d'auteur"
Exit Sub
Elseif DateFin.Value = "" Then
    MsgBox "Vous devez spécifier une date de fin.", vbOKOnly + vbCritical, "Contrat d'auteur"
Exit Sub
Elseif DateValue(DateFin) - DateValue(DateDebut) <= 0 Then
    MsgBox "Vous devez spécifier une date de début antérieure à la date de fin.", _
vbOKOnly + vbCritical, "Contrat d'auteur"
Exit Sub
Elseif DateValue(DateFin) - DateValue(DateDebut) < 40 Then
Dim Continuer As Integer
Continuer = MsgBox("Vous avez indiqué une date de début à seulement " _
& DateValue(DateFin) - DateValue(DateDebut) & " jours de la date de fin. _
Valider cette date ?", vbYesNo + vbQuestion, "Valider la date de début ?")
If Continuer = vbNo Then Exit Sub
End If
fmContratDates.Hide
' Affichage des dates sélectionnées
MsgBox (" Les dates de Début et Fin sont : " & DateDebut.Value & " et " & DateFin.Value)
' Mise à jour des paramètres
DateDebut.Value = ""
DateFin.Value = ""
cboDateAValider.ListIndex = 0
Calendrier.Value = Date
Calendrier.SetFocus
End Sub

```

2 / Bouton Annuler (cmdAnnuler)

```

Private Sub CmdAnnuler_Click()
Dim Rep As Byte
Rep = MsgBox("Êtes-vous sûr de vouloir fermer l'application en cours ?", _
vbYesNo + vbQuestion, "Annuler l'application en cours ?")
If Rep = vbNo Then
Exit Sub
End If
Me.Hide
End
End Sub

```

3 / Bouton Retour (cmdRetour)

```

Private Sub CmdRetour_Click()
MsgBox "Vous allez fermer l'application"
Me.Hide
End Sub

```

● Le contrôle ComboBox.

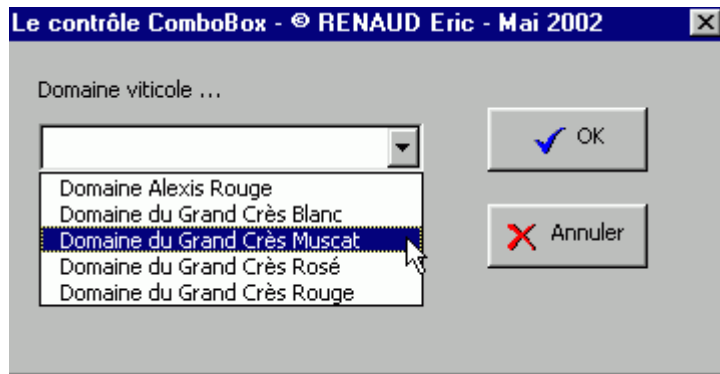
Allie les caractéristiques d'un contrôle **ListBox** et d'un contrôle **TextBox**.
L'utilisateur peut entrer une [nouvelle valeur](#), comme dans un contrôle **TextBox**
ou bien sélectionner une valeur existante comme dans un contrôle **ListBox**.
La liste d'un contrôle **ComboBox** se compose de lignes de données.
La propriété par défaut d'un contrôle **ComboBox** est la propriété **Value**.

Le principe de fonctionnement de ce contrôle, se décompose comme suit :

- 1 - Remplir le contrôle **ComboBox** avec des données.
- 2 - Choisir une de ces données en la sélectionnant.
- 3 - Récupérer la donnée sélectionnée à la fermeture du formulaire.

Remplir le contrôle Combobox avec des données

Exemple : Afficher une liste de Domaines viticoles.



Je vous donne 4 méthodes pour remplir cette ComboBox (en fonction de votre besoin)

La propriété (name) du contrôle ComboBox est `cboComboBox`

Méthode 1 : Les données sont dans une feuille de calcul (Feuil1).

```
Private Sub UserForm_Initialize()
    Me.cboComboBox.RowSource = "Feuil1!A1:A" & Sheets("Feuil1").Cells(1, 1).End(xlDown).Row
End Sub
```

Méthode 2 : Les données sont figées et vous les ajoutez grâce à la méthode AddItem.

```
Private Sub UserForm_Initialize()
    With Me.cboComboBox
        cboComboBox.AddItem "Domaine Alexis Rouge"
        cboComboBox.AddItem "Domaine du Grand Crès Blanc"
        cboComboBox.AddItem "Domaine du Grand Crès Muscat"
        cboComboBox.AddItem "Domaine du Grand Crès Rosé"
        cboComboBox.AddItem "Domaine du Grand Crès Rouge"
    End With
End Sub
```

Méthode 3 : Les données correspondent à la liste des mois (Janvier, Février, ..., Décembre).

On va utiliser la méthode AddItem avec une variable (i).

```
Private Sub UserForm_Initialize()
    Dim Mois(1 To 12) As String
    Dim i As Integer
    'Création d'un tableau des noms de mois
    For i = 1 To 12
        Mois(i) = Format(DateSerial(1, i, 1), "mmm")
        Me.cboComboBox.AddItem Mois(i)
    Next i
End Sub
```

Méthode 4 : Les données sont figées et vous les ajouter grâce _
à la fonction VBA.Array(arglist).

```
Private Sub UserForm_Initialize()
    Dim vaDepartement As Variant
    Dim stDeptList() As String
    Dim i As Integer
    vaDepartement = VBA.Array("Ain", "Aisne", "Allier", "Alpes-de-Hte-Provence", _
        "Alpes-Hautes", "Alpes-Maritimes", "Ardennes")
    ReDim stDeptList (0 To UBound(vaDepartement))
    For i = 0 To UBound(vaDepartement)
        stDeptList(i) = vaDepartement(i)
    Next i
    cboCombobox.List = stDeptList
End Sub
```

Sélectionner une données dans la liste

Il suffit simplement de sélectionner avec la souris la donnée désirée.

Récupérer l'item sélectionné

```
Private Sub cmdOK_Click()  
    MsgBox "L'item sélectionné est : " & cboComboBox.Value, vbInformation  
    Unload Me  
End Sub
```

Récupérer une nouvelle valeur saisie

Il est possible dans le cas où les données sont issues d'une feuille Excel, d'inscrire cette nouvelle valeur dans la feuille Excel.
Lorsque l'on ajoute une nouvelle donnée, celle-ci se trouve en position **ListIndex = -1**.
On utilise pour cela un test sur la propriété **ListIndex**.
En règle générale, cette propriété ListIndex à la **valeur -1** à l'ouverture du formulaire.
Si on sélectionne la première donnée de la liste, la valeur ListIndex est égale à 0, et ainsi de suite.

```
Private Sub OK_Click()  
    Me.Hide  
    If Me.cboComboBox.ListIndex = -1 Then _  
        Sheets("Feuil1").Cells(1, 1).End(xlDown).Offset(1, 0).Value = Me.cboCombobox.Value  
End Sub
```

● Le contrôle RefEdit.

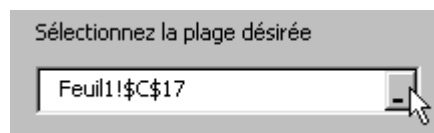
Ce contrôle placé dans un formulaire affiche l'adresse d'une plage de cellules que vous avez entrée ou sélectionnée dans une feuille de calcul.
Pour sélectionner une plage, cliquez sur le bouton dans le contrôle pour réduire le formulaire utilisateur, sélectionnez la plage, puis cliquez de nouveau sur le bouton dans le contrôle pour développer le formulaire utilisateur.

Exemple d'utilisation du contrôle RefEdit

Effectuer une opération arithmétique (+, -, * ou x) sur une plage de cellules.
(Ex. Conversion Frs/Euros, majorer un prix, appliquer une remise).

Principe d'utilisation du contrôle RefEdit

Etape 1 : Cliquez sur le trait horizontal du contrôle.



Etape 2 : Affichage d'une boîte de sélection.



Etape 3 : Sélectionnez avec la souris la plage désirée.



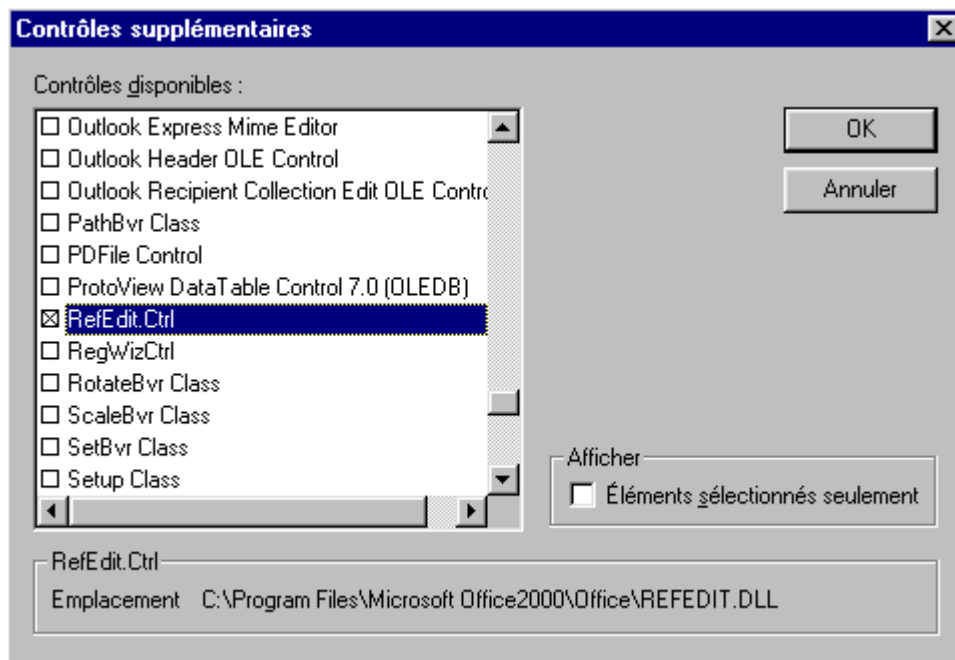
Etape 4 : Cliquez sur la flèche rouge du contrôle pour valider votre sélection.



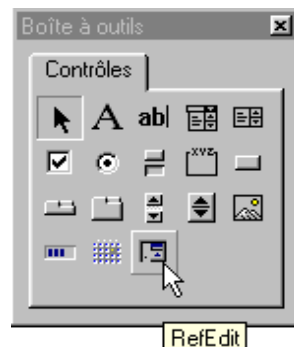
Mise en oeuvre du contrôle RefEdit

- 1 - Activer un UserForm dans le Visual Basic Editor.
- 2 - Faites un clic droit sur la boîte à outils et sélectionnez Contrôles supplémentaires.
- 3 - Dans la liste déroulante, sélectionnez RefEdit.Ctrl.
- 4 - Cliquez sur OK pour fermer la boîte de dialogue et ajoutez le contrôle dans la boîte à outils.

Ajout du contrôle RefEdit



La boîte à outils



Le Userform (SelectOperation)



Le code

Code lançant l'ouverture de la boîte de dialogue

```
Sub AfficheRefEdit()  
    SelectOperation.Show  
End Sub
```

Code d'ouverture du formulaire

```
Private Sub UserForm_Initialize()  
    ' Affiche la sélection en cours  
    RefEdit1.Text = ActiveWindow.RangeSelection.Address  
    ' Rend le bouton Addition actif par défaut  
    Frame1.OptionAddition.Value = True  
End Sub
```

Code des boutons

1 / Bouton Valider (CmdValider)

```
Private Sub CmdValider_Click()  
    ' Teste si une valeur a été saisie dans la TextBox (Opérande)  
    On Error Resume Next  
    If TextBox1.Value = "" Then  
        MsgBox "Vous devez saisir une valeur d'opérande"  
        RefEdit1.SetFocus  
        On Error GoTo 0  
        Exit Sub  
    End If  
    Set PlageSelect = Range(RefEdit1.Text)  
    MyOperande = TextBox1.Value  
    ' Teste la sélection effectuée est correct  
    If Err <> 0 Then  
        MsgBox "La plage sélectionnée est invalide"  
        RefEdit1.SetFocus  
        On Error GoTo 0  
        Exit Sub  
    End If  
    ' Effectue l'opération choisie sur la plage de cellules sélectionnées  
    For Each Cell In PlageSelect  
        If OptionAddition.Value = True Then Cell.Value = Cell.Value + MyOperande  
        If OptionSoustraction.Value = True Then Cell.Value = Cell.Value - MyOperande  
        If OptionMultiplication.Value = True Then Cell.Value = Cell.Value * MyOperande  
        If OptionDivision.Value = True Then Cell.Value = Cell.Value / MyOperande  
    Next Cell  
    SelectOperation.Hide  
    Unload SelectOperation  
End Sub
```

2 / Bouton Annuler (CmdAnnuler)

```
Private Sub CmdAnnuler_Click()  
    SelectOperation.Hide  
    Unload SelectOperation
```

● Le contrôle TextBox (Zone de texte).

Ce contrôle placé dans un Userform, permet la saisie de données.
Ces données saisies peuvent parfois être erronées (erreur de frappe, texte à la place de nombre, date non conforme, etc ...).
Pour cela il peut être intéressant de contrôler soit lors de la validation par OK ou lors de la sortie du contrôle si les données saisies sont conformes par rapport à la fonction du contrôle.

Convention d'écriture

Le Userform est nommé **UserForm1**

Le contrôle TextBox est nommé **TextBox1**

Le bouton OK est nommé **CmdOK**

Exemples de tests sur une TextBox

- S'assurer qu'il y a eu une saisie dans la TextBox
- S'assurer que la saisie dans la TextBox est de type numérique (nombre)

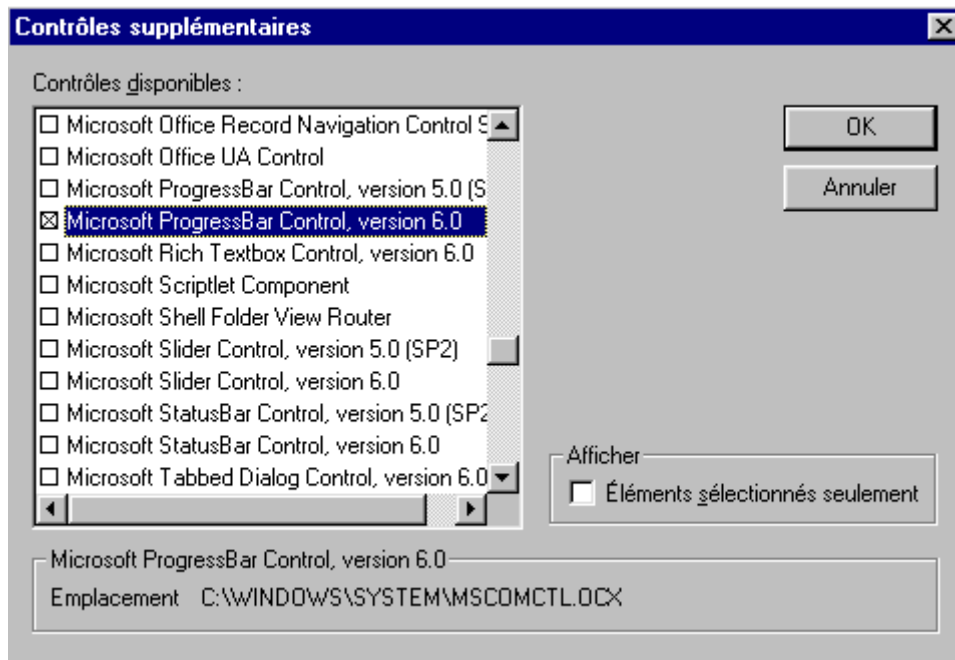
● Le contrôle ProgressBar, version 6.0.

Ce contrôle permet d'afficher par exemple, la progression d'une mise à jour de cellules (ex. multiplier une plage par un coefficient).

Exemple trouvé sur le site de [Ole P. Erlandsen](#)

- 1 - Activer un UserForm dans le Visual Basic Editor.
- 2 - Faites un clic droit sur la boîte à outils et sélectionnez Contrôles supplémentaires.
- 3 - Dans la liste déroulante, sélectionnez Microsoft ProgressBar Control, version 6.0.
- 4 - Cliquez sur OK pour fermer la boîte de dialogue et ajoutez le contrôle dans la boîte à outils.

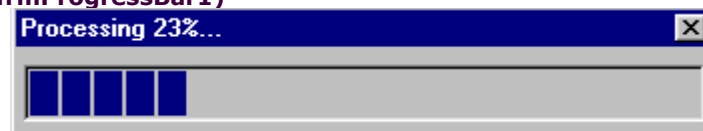
Ajout du contrôle ProgressBar Control



La boîte à outils



Le Userform (frmProgressBar1)



Le code

[Code lançant l'indicateur de progression](#)

```

Sub StartProcessing1()
' Displays a progress bar while a macro runs, requires a reference to MSCOMCTLOCX
Dim lngTotal As Long, lngI As Long
' Initiate ProgressBar
Load frmProgressBar
With frmProgressBar
    .ProgressBar.Scrolling = ccScrollingStandard ' or ccScrollingSmooth
    .Show ' set the UserForms ShowModal property to false before running
    ' or .Show False
End With
UpdateProgressBar 0, "Processing..." ' set initial progress status
' start the process
lngTotal = 2000
For lngI = 1 To lngTotal
    If lngI Mod 50 = 0 Then ' Update the ProgressBar for every 50th loop
        UpdateProgressBar lngI / lngTotal * 100, "Processing " & Format _
            (lngI / lngTotal, "0%") & "..."
    End If
    ' Do something, place your code here
    Range("D1").Formula = Format(Time, "hh:mm:ss")
Next lngI
Range("D1").ClearContents

```



```
' Clean up
frmProgressBar.Hide
Unload frmProgressBar
End Sub
```

Code lié à la macro UpdateProgressBar

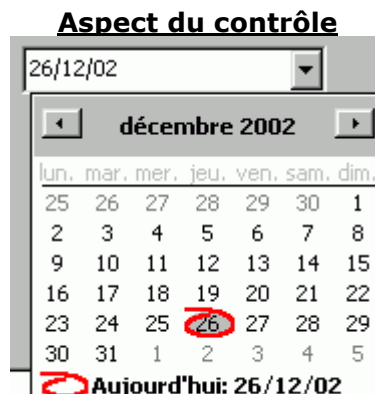
```
Private Sub UpdateProgressBar(NewValue As Single, Optional NewCaption As String)
' Updates the progressbar dialog
With frmProgressBar
    If Not IsMissing(NewCaption) Then .Caption = NewCaption
    .ProgressBar.Value = NewValue
    If NewValue = 0 Then .Repaint
End With
End Sub
```

● Le contrôle DTPicker (calendrier).

Ce contrôle permet grâce à un calendrier de saisir des dates de début et des dates de fin

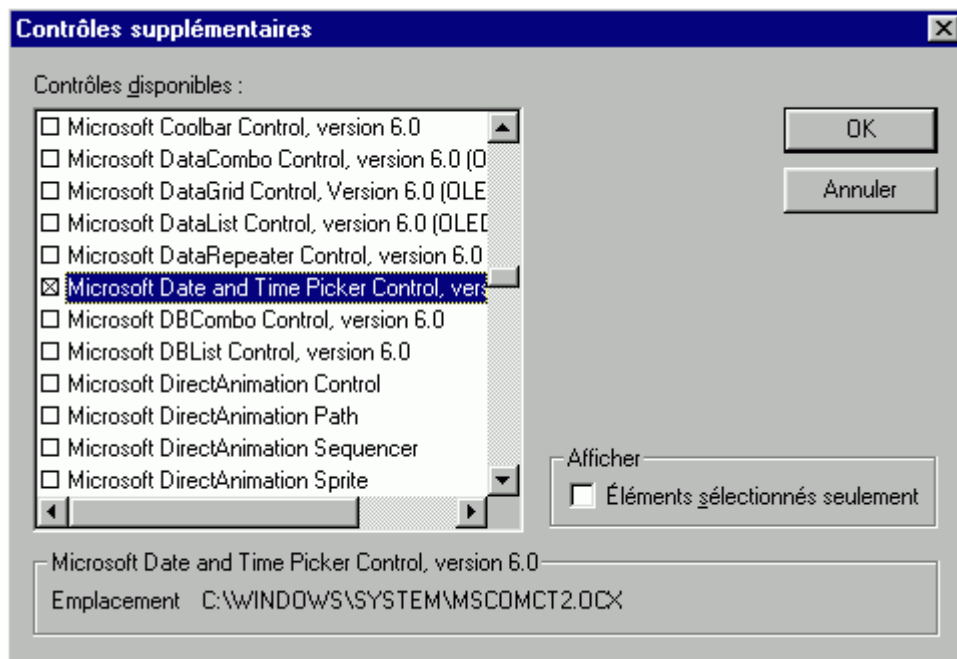
de façon simple et sans erreur possible, pour les utiliser ensuite dans votre application.

Cet exemple a été réalisé sous Excel 2000.



Ajout du contrôle DTPicker

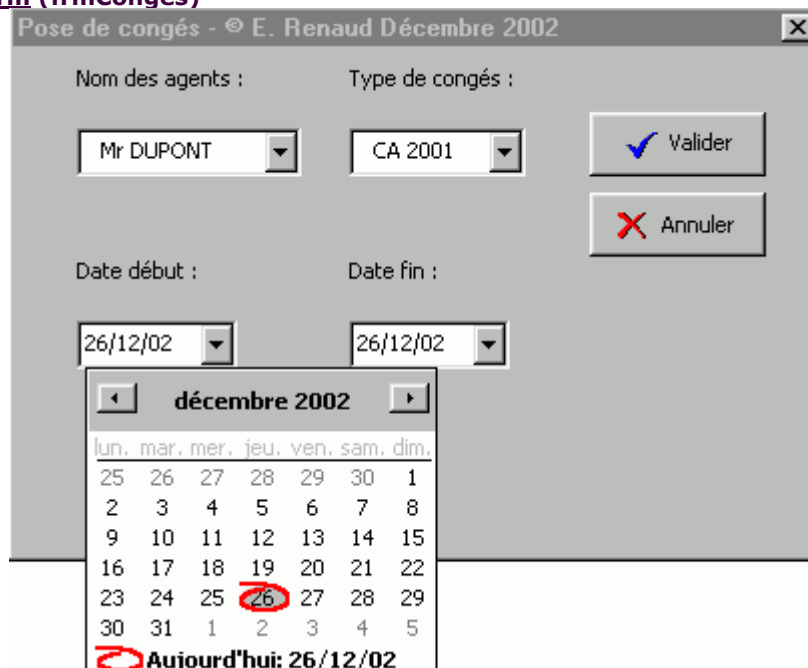
- 1 - Activer un UserForm dans le Visual Basic Editor.
- 2 - Faites un clic droit sur la boîte à outils et sélectionnez Contrôles supplémentaires.
- 3 - Dans la liste déroulante, sélectionnez Microsoft Date and Time Picker Control, version 6.0
- 4 - Cliquez sur OK pour fermer la boîte de dialogue et ajoutez le contrôle dans la boîte à outils.



La boîte à outils



Le Userform (frmConges)



Le code

Code lançant l'ouverture du formulaire

```
Sub AfficheConges()  
    frmConges.Show  
End Sub
```

Codes d'ouverture du formulaire

```
Private Sub UserForm_Initialize()  
    ' Définir butée mini pour contrôle Date début (DTPicker1)  
    Me.DTPicker1.MinDate = Now - 60  
    ' Définir butée maxi pour contrôle Date fin (DTPicker2)  
    Me.DTPicker2.MaxDate = Now + 365  
End Sub  
  
Private Sub UserForm_Activate()  
    ' Remplissage des 2 listes déroulantes (Nom agent et type de congés)  
    Me.cboAgents.RowSource = "Congés!B7:B21"  
    Me.cboConges.RowSource = "Congés!B24:B31"  
    ' Activer le premier enregistrement de chaque liste déroulante  
    Me.cboAgents.ListIndex = 0  
    Me.cboConges.ListIndex = 0  
    ' Définir la date affichée par défaut dans chaque contrôle DTPicker  
    Me.DTPicker1.Value = Now  
    Me.DTPicker2.Value = Now  
End Sub
```

Code lié à l'événement Change sur le contrôle DTPicker2

```
Private Sub DTPicker2_Change()  
    ' Récupération des dates sélectionnées  
    MyDateDebut = DTPicker1.Value  
    MyDateFin = DTPicker2.Value  
    If MyDateFin <= MyDateDebut Then  
        MsgBox "La date de fin ne peut être inférieure à la date de début"  
        DTPicker2.SetFocus  
    End If  
End Sub
```

Code des boutons

1 / Bouton Valider (cmdValider)

```
Private Sub CmdValider_Click()  
    MyDateDebut = DTPicker1.Value  
    MyDateFin = DTPicker2.Value  
    ' Dans le cas où le choix est effectué sur le DTPicker1  
    If MyDateFin <= MyDateDebut Then  
        MsgBox "La date de fin ne peut être inférieure à la date de début"  
        DTPicker1.SetFocus  
    Else  
        Unload Me  
        ' Mise en place des données dans la feuille BaseCongés  
        Sheets("BaseCongés").Cells(65536, 1).End(xlUp).Offset(1, 0).Value = Me.cboAgents.List _  
            (Me.cboAgents.ListIndex)  
        Sheets("BaseCongés").Cells(65536, 2).End(xlUp).Offset(1, 0).Value = Me.cboConges.List _  
            (Me.cboConges.ListIndex)  
        Sheets("BaseCongés").Cells(65536, 3).End(xlUp).Offset(1, 0).Value = Format(MyDateDebut, "mm/dd/yyyy")  
        Sheets("BaseCongés").Cells(65536, 4).End(xlUp).Offset(1, 0).Value = Format(MyDateFin, "mm/dd/yyyy")  
    End If  
End Sub
```

2 / Bouton Annuler (cmdAnnuler)

```
Private Sub CmdAnnuler_Click()  
    Me.Hide  
End Sub
```

Quelques infos sur les UserForms

Dans cette rubrique vous trouverez des infos pour la gestion des UserForms.

Retour au menu Contrôles divers

● Empêcher le déplacement d'un Userform.

● Mettre hors fonction la croix (X) de fermeture du UserForm.

● Minimiser un formulaire (Excel 2000 uniquement).

● Ajout d'une image sur un bouton de formulaire.



Empêcher le déplacement d'un Userform, et non affichage de la croix de fermeture.

[Exemple proposé par [Laurent Longre](#)]

Le code qui suit devra être dans le module de code du UserForm.

```
Private Declare Function FindWindowA Lib "User32" _  
    (ByVal lpClassName As String, ByVal lpWindowName As String) As Long
```

```
Private Declare Function GetWindowLongA Lib "User32" _  
    (ByVal hWnd As Long, ByVal nIndex As Long) As Long
```

```
Private Declare Function SetWindowLongA Lib "User32" _  
    (ByVal hWnd As Long, ByVal nIndex As Long, _  
    ByVal dwNewLong As Long) As Long
```

```
Private Sub UserForm_Activate()  
    Dim hWnd As Long, exLong As Long  
    hWnd = FindWindowA(vbNullString, Me.Caption)  
    exLong = GetWindowLongA(hWnd, -16)  
    If exLong And &H880000 Then  
        SetWindowLongA hWnd, -16, exLong And &HFF77FFFF  
        Me.Hide: Me.Show  
    End If  
End Sub
```

Mettre hors fonction la croix (X) de fermeture du UserForm.

```
Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)  
    If CloseMode = vbFormControlMenu Then  
        MsgBox "Vous ne pouvez pas utiliser ce bouton de fermeture."  
        Cancel = True  
    End If  
End Sub
```

Minimiser un UserForm.

[Exemple proposé par [Laurent Longre](#)]

Mettre la propriété **ShowModal** du UserForm sur **False** (cette propriété n'existe pas sous Excel 97)

Définitions :

- 1 - Lorsqu'un objet **UserForm** est modal (ShowModal=True), l'utilisateur doit fournir des informations ou fermer l'objet UserForm pour pouvoir utiliser toute autre partie de l'application.
- 2 - Lorsqu'un objet **UserForm** n'est pas modal (ShowModal=False), l'utilisateur peut afficher d'autres feuilles

ou fenêtres sans fermer cet objet.

Le code qui suit devra être dans le module de code du UserForm.

```
Private Declare Function FindWindowA Lib "User32" _  
    (ByVal lpClassName As String, ByVal lpWindowName As String) As Long
```

```
Private Declare Function GetWindowLongA Lib "User32" _  
    (ByVal hWnd As Long, ByVal nIndex As Long) As Long
```

```
Private Declare Function SetWindowLongA Lib "User32" _  
    (ByVal hWnd As Long, ByVal nIndex As Long, _  
    ByVal dwNewLong As Long) As Long
```

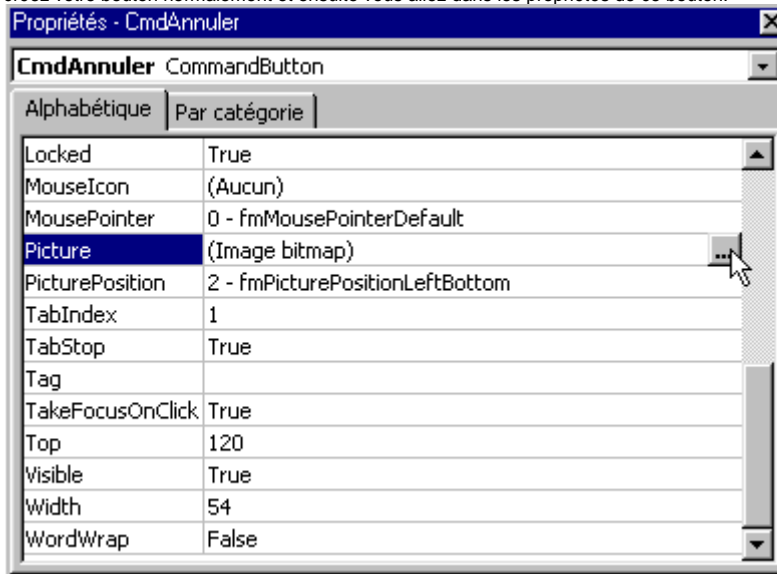
```
Private Sub UserForm_Activate()  
    Dim hWnd As Long, exLong As Long  
    hWnd = FindWindowA(vbNullString, Me.Caption)  
    exLong = GetWindowLongA(hWnd, -16)  
    If (exLong And &H20000) = 0 Then  
        SetWindowLongA hWnd, -16, exLong Or &H20000  
        Me.Hide: Me.Show  
    End If  
End Sub
```

Ajouter une image à un bouton de formulaire.

Pour ajouter une image à un bouton de formulaire, il faut posséder une image de type Gif par exemple avec une taille d'environ 26x20 comme celles ci-dessous.



Vous créez votre bouton normalement et ensuite vous allez dans les propriétés de ce bouton.



- 1 - Sélectionner la propriété **Picture** et cliquez sur le bouton avec 3 points. Vous sélectionnez votre image.
- 2 - Sélectionner la propriété **PicturePosition** et choisissez l'option **2 - fmPicturePositionLeftBottom**. (C'est celle qui me paraît être la mieux adaptée).

Remplir une combobox ou listbox

Depuis une sheet

```
' *****  
' Remplir une ComboBox ou ListBox depuis  
' une feuille de calcul  
' *****
```

```
Private Sub UserForm_Initialize()  
    With Sheets( "Feuil1" ).Range( "A1" )  
        Me.ComboBox1.RowSource = "Feuil1!A1:A" & Sheets( "Feuil1" ).Cells( 1 , 1 ).End(xlDown).Row  
    End With  
End Sub
```

Depuis une sheet (2 colonnes)

```
Private Sub UserForm_Initialize()  
    With Sheets( "Feuil1" ).Range( "A1" )  
        Me.ComboBox1.RowSource = "Feuil1!A1:B" & Sheets( "Feuil1" ).Cells( 1 , 1  
    ).End(xlDown).Row  
    End With  
End Sub  
  
' *****  
' Avec une autre syntaxe  
' *****  
Private Sub UserForm1_Initialize()  
    With Sheets( "Feuil1" ).Range( "A1" )  
        Me.ComboBox1.RowSource = Range(.Cells, .End(xlDown))( 1 , 2 ).Address(External:= True )  
    End With  
End Sub
```

Avec la méthode Additem

```
' *****  
' Ajout d'item avec la méthode AddItem pour  
' remplir une ComboBox ou ListBox.  
' *****  
Private Sub UserForm_Initialize()  
    With Me.ComboBox1  
        ComboBox1.AddItem "Domaine Alexis Rouge"  
        ComboBox1.AddItem "Domaine du Grand Crès Blanc"  
        ComboBox1.AddItem "Domaine du Grand Crès Muscat"  
        ComboBox1.AddItem "Domaine du Grand Crès Rosé"  
        ComboBox1.AddItem "Domaine du Grand Crès Rouge"  
    End With  
End Sub
```

Avec la liste des onglets

```
' *****  
' Ajout de la liste des onglets du classeur pour  
' remplir une ComboBox ou ListBox.  
' *****  
Public Sub UserForm_Initialize()  
    For Each c In ActiveWorkbook.Sheets  
        ComboBox1.AddItem c.Name  
    Next  
End Sub
```

Avec les noms des mois

```
' *****  
' Ajout d'items correspondant à des mois pour  
' remplir une ComboBox ou ListBox.  
' *****  
Private Sub UserForm_Initialize()  
    Dim Mois( 1 To 12 ) As String  
    Dim i As Integer  
    ' Creation d'un tableau des noms de mois  
    For i = 1 To 12  
        Mois(i) = Format ( DateSerial ( 1 , i , 1 ), "mmm" )  
        Me.ComboBox1.AddItem Mois(i)  
    Next i  
End Sub
```

Avec une plage nommée

```
' *****  
' Utiliser une plage nommée pour  
' remplir une ComboBox ou ListBox.  
' *****
```

' La liste est définie dans le Userform par son nom entré à l'aide
' de la commande Insertion/Nom/Définir avec la formule suivante :
' NomClient=DECALER(Codes!\$A\$1;;;NBVAL(Codes!\$A:\$A);)
' ou Codes correspond au nom de l'onglet du classeur et
' Nomclient étant le nom de la plage de données.

```
Private Sub UserForm_Initialize()  
    ' Activation du classeur Workbook.xls  
    Workbooks( "Workbook.xls" ).Activate  
    ComboBox1.RowSource = "Codes!NomClient"  
End Sub
```

Avec une boucle et additem

' *****
' Ajout d'items correspondant à des années pour
' remplir une ComboBox ou Listbox.
' *****

```
Private Sub UserForm_Initialize()  
    ReDim Annee( 1 To 12 )  
    For i = 1 To 12  
        Annee(i) = 1998 + (i)  
        Me.ComboBox1.AddItem Annee(i)  
    Next  
End Sub
```

Avec un fichier texte

' *****
' Ajout d'item depuis un fichier texte pour
' remplir une ComboBox ou Listbox.
' *****

```
Private Sub UserForm_Initialize()  
    ' Ouvre le fichier en lecture.  
    Open "C:\Excel\Donnees.txt" For Input As # 1  
    Do While Not EOF ( 1 )  
        ' Lit les données  
        Input # 1 , NomClient  
        ' Ajoute les données dans la ComboBox ou Listbox  
        Me.ComboBox1.AddItem NomClient  
    Loop  
    Close # 1  
End Sub
```

Avec la fonction array (tableau)

```
Private Sub UserForm_Initialize()  
    Dim vaDepartment As Variant  
    Dim vaDeptCode As Variant  
    Dim stDeptList() As String  
    Dim i As Integer  
    ' Liste des départements  
    vaDepartment = VBA.Array( "Ain" , "Aisne" , "Allier" , "Alpes (Hte Prov.)" , "Alpes (Hautes)" ,  
"Alpes (Maritimes)" , _  
    "Ardèche" , "Ardennes" )  
    ' Code des départements  
    vaDeptCode = VBA.Array( "01" , "02" , "03" , "04" , "05" , "06" , "07" , "08" )  
    ReDim stDeptList( 0 To UBound( vaDepartment), 0 To 1 )  
    ' Boucle pour mise en place des valeurs  
    For i = 0 To UBound( vaDepartment)  
        stDeptList(i, 0) = vaDeptCode(i)  
        stDeptList(i, 1) = vaDepartment(i)  
    Next i  
    ComboBox1.List = stDeptList  
End Sub
```

' *****
' Autre variante possible
' *****

```
Private Sub UserForm1_Initialize()  
    Dim TabMonnaies( 4 , 2 )  
    ComboBox1.ColumnCount = 2  
    ' Nom des monnaies proposées  
    TabMonnaies( 0 , 0 ) = "Franc"
```

```

TabMonnaies( 1 , 0 ) = "Euro"
TabMonnaies( 2 , 0 ) = "Dollar"
TabMonnaies( 3 , 0 ) = "Livre"
'Symboles des monnaies
TabMonnaies( 0 , 1 ) = " F"
TabMonnaies( 1 , 1 ) = " €"
TabMonnaies( 2 , 1 ) = " $"
TabMonnaies( 3 , 1 ) = " £"
ComboBox1.List() = TabMonnaies
    
```

End Sub

Avec les fichiers XLS d'un répertoire

```

' *****
' Lister les fichiers d'un répertoire donné
' et les afficher dans une ComboBox ou Listbox.
' Seul le nom du fichier sera affiché (Fonction ShortFilename)
' *****
    
```

```

Private Sub UserForm_Initialize()
    Dim F
    Dim LongFilename As String
    With Application.FileSearch
        .NewSearch
        .LookIn = "C:\Excel\Traitement des déchets"
        .Execute
        On Error Resume Next
        For Each F In .FoundFiles
            LongFilename = F
            Namefile = ShortFilename(LongFilename)
            Me.ComboBox1.AddItem Namefile
        Next F
    End With
End Sub
    
```

Boucle sur des contrôles

Efface le contenu des textbox

```

' *****
' Met à blanc les différentes TextBox
' contenues dans le formulaire
' *****
    
```

```

Private Sub cmdOK_Click()
    Dim i As Integer
    For Each Ctrl In Me.Controls
        If TypeOf Ctrl Is MSForms.TextBox Then
            i = i + 1
            Ctrl.Value = ""
        End If
    Next Ctrl
End Sub
    
```

Exporter les données des textbox

```

' *****
' Exporte les valeurs des différentes TextBox
' dans les cellules A1, A2, A3, etc..., lors
' de la validation par le bouton OK.
' *****
    
```

```

Private Sub cmdOK_Click()
    Dim i As Integer
    For Each Ctrl In Me.Controls
        If TypeOf Ctrl Is MSForms.TextBox Then
            i = i + 1
        End If
    Next Ctrl
End Sub
    
```



```

Cells(i, 1 ) = Ctrl
End If
Next Ctrl
End Sub

```

Remplir les textbox à l'ouverture

```

' *****
' Remplit les contrôles TextBox du formulaire
' avec les valeurs contenues dans les cellules A1, A2, etc...,
' lors de l'ouverture du formulaire.
' *****
Private Sub UserForm_Initialize()
    Dim i As Integer
    For Each Ctrl In Me.Controls
        If TypeOf Ctrl Is MSForms.TextBox Then
            i = i + 1
            Controls( "TextBox" & i).Value = Cells(i, 1 ).Value
        End If
    Next Ctrl
End Sub

```

Initialisation de checkbox

```

' *****
' Permet via un bouton placé sur le formulaire de
' réinitialiser les contrôles Checkbox (Mise à False)
' *****
Private Sub CmdInitCheckBox_Click()
    Dim Ctrl As MSForms.Control
    ' Remise à False des cases à cocher
    For Each Ctrl In UserForm1.Controls
        If TypeOf Ctrl Is MSForms.CheckBox Then Ctrl.Value = False
    Next Ctrl
End Sub

```

Les feuilles Excel (Worksheets)

Page mise à jour le : 28/03/2001

Vous trouverez dans cette rubrique des exemples concernant des programmes s'appliquant aux feuilles de calcul.

- Comparer deux colonnes.
- Effectuer un calcul sur une page (Ex. Multiplier par 2 les cellules de la page).
- Effectuer une recopie incrémentée.

- Insérer une ligne vide toutes les 2 lignes (ou plus).
 - Introduire une variable dans une formule SOMME (Ex. Un N° de ligne).
 - Introduire une variable dans une formule SOMME (Ex. l'adresse d'une cellule).
 - Mettre en évidence les cellules répondant à un critère.
 - Recherche du nombre d'occurrence.
 - Rechercher la dernière cellule d'une plage.
 - Recopier une plage variable vers une autre feuille du classeur.
 - Récupérer la colonne active sous forme littérale (A, AB, AZ, etc...).
 - Récupérer la somme d'une colonne dans une variable.
 - Récupérer le nombre de colonnes.
 - Récupérer le nombre de lignes.
 - Récupérer la somme du produit de 2 colonnes (Somme des N (A*B)).
 - Répertoire et nom de fichier dans pied de page.
 - Supprimer les doublons dans une colonne.
 - Supprimer les lignes vides d'une plage.
 - Supprimer un nom dans une feuille de calcul.
 - Supprimer tous les noms dans une feuille de calcul.
 - Supprimer les noms avec un joker dans une feuille de calcul.
 - Supprimer les lignes contenant du texte.
 - Utiliser des macro-commandes dans une feuille protégée.
-

Comparer deux colonnes.

Permet d'ajouter dans la colonne 1 les valeurs qui seraient dans la colonne 2 et pas dans la colonne 1.

```
Sub CompareTwoColonnes()
Dim Cell As Range, Plage As Range, I As Long
Set Plage = Range("A1", [A1].End(xlDown))
I = Plage.Count
Application.ScreenUpdating = False
For Each Cell In Range("B1", [B1].End(xlDown))
    If Plage.Find(Cell, Plage(1), xlValues, xlWhole) Is Nothing Then
        I = I + 1
        Cells(I, 1) = Cell
    End If
Next Cell
End Sub
```

Effectuer un calcul sur une plage.

Permet de multiplier l'ensemble d'une plage par une valeur (2 dans l'exemple)

```
Sub MultiplieParDeux()
' Utilisation d'un tableau intermédiaire
Dim Tabl
Dim I As Integer, J As Integer
Tabl = Range("A1:C1000").Value
For I = 1 To 1000
    For J = 1 To 3
        Tabl(I, J) = Tabl(I, J) * 2
    Next J
Next I
Application.ScreenUpdating = False
Range("A1:C1000").Value = Tabl
End Sub
```

Effectuer une recopie incrémentée.

Permet d'effectuer une recopie incrémentée à partir de A1 et A2

```
Sub RecopieIncrementee()
Set PlageSource = Worksheets("Feuil1").Range("A1:A2")
' Effectue une recopie incrémentée de A1 à A20
Set PlageARemplir = Worksheets("Feuil1").Range("A1:A20")
```

```
PlageSource.AutoFill Destination:=PlageARemplir
End Sub
```

Insérer une ligne vide toutes les 2 lignes (ou plus...)

```
Permet d'insérer dans un tableau une ligne vide toutes les deux lignes (ou plus)
Sub MacroInsertUneLigneSurDeux()
Dim Line As Integer
Range("A2").Select
Line = 1
Recommence:
Line = Line + 2
Rows(Line).Select
Selection.Insert Shift:=xlDown
If Line < ActiveSheet.UsedRange.Rows.Count Then
    GoTo Recommence
End If
End Sub
```

Introduire une variable dans une formule SOMME. (Ex. Un N° de ligne)

```
Sub VariableSomme()
' La colonne A est la colonne où l'on doit effectuer une somme
' Se placer sur la première ligne vide de la colonne pour poser la formule Somme
Range("A1").End(xlDown).Offset(1, 0).Select
' Récupération du nombre de lignes à comptabiliser dans la formule
MonNoDeLigne = -(ActiveCell.Row) + 1
ActiveCell.FormulaR1C1 = "=SUM(R[" & MonNoDeLigne & "]:R[-1]C)"
End Sub
```

Introduire une variable dans une formule SOMME. (Ex. L'adresse d'une cellule)

```
Sub VariableAdressDansSomme()
Range("A1").End(xlDown).Offset(1, 0).Value = "=SUM(A1:" & Range("A1").End _
(xlDown).Address & ")"
End Sub
```

Mettre en évidence les cellules répondant à un critère.

```
Ce programme va sélectionner toutes les cellules correspondant
à la valeur contenue dans D2.
Sub SelectCellulesValeurDeterminee()
LaValeur = Range("D2").Value
Range("A1").Select
For Each cll In ActiveCell.CurrentRegion
    If cll.Value = LaValeur Then Plg = Plg & cll.Address() & ","
Next cll
If Len(Plg) > 0 Then Range(Left(Plg, Len(Plg) - 1)).Select
End Sub
```

Pour sélectionner les lignes à la place des cellules :

```
Remplacer la ligne de code
If cll.Value = LaValeur Then Plg = Plg & cll.Address() & ","
par
If cll.Value = LaValeur Then Plg = Plg & cll.Row() & ":" & cll.Row() & ","
```

Recherche du nombre d'occurrence.

```
Ce programme va compter le nombre de fois qu'il rencontre la valeur ValeurAChercher
Sub RechercheNbOccurrence()
Résultat = Application.CountIf(Range("A:A"), "ValeurAChercher")
MsgBox ("Le texte ValeurAChercher est présent : " & Résultat & " fois.")
End Sub
```

Rechercher la dernière cellule d'une plage.

```
Ce programme permet de positionner le curseur sur la dernière cellule d'une plage
correspondant à la cellule en cours.
Sub AllerADernierecellule()
'Sélectionne la plage en cours
ActiveSheet.UsedRange.Select
'Sélectionne la dernière cellule de la plage en cours
```

```
ActiveSheet.UsedRange.SpecialCells(xlCellTypeLastCell).Offset(0, 0).Select
End Sub
```

Recopier une plage variable vers une autre feuille du classeur.

Ce programme permet de recopier une plage variable A8 à Gx vers une autre feuille (Feuil2) en A1. correspondant à la cellule en cours.

```
Sub SelectRecopie()
' Redéfinir la plage en fonction du nombre de lignes
' Sélectionner cette plage
' Copier les valeurs contenues dans cette plage sur la feuille 2 à la cellule "A1"
With Worksheets(1)
.Range(.[A1], .[G1].End(xlDown)).Copy Worksheets(2).[A1]
End With
End Sub
```

Récupérer la colonne active sous forme littérale (A, AB, AZ, etc...).

Ce programme permet de récupérer la lettre de la colonne de la cellule active.

```
Sub LetCol()
Dim Let_Col As String
Let_Col = Left(Mid(ActiveCell.Address, 2), _
Len(Mid(ActiveCell.Address, 2)) - (Len(Mid(Mid(ActiveCell.Address, 2), _
Application.Search("$", Mid(ActiveCell.Address, 2), 1) + 1)) + 1))
MsgBox Let_Col
End Sub
Autre variante avec utilisation des valeurs VRAI(-1) et FAUX (0)
Sub LettreColonne()
' Utilisation des valeurs VRAI (-1) et FAUX (0)
Lettre_Col = Left(ActiveCell.Address(0, 0), (ActiveCell.Column < 27) + 2)
MsgBox Lettre_Col
End Sub
```

Récupérer la somme d'une colonne dans une variable.

Ce programme effectue la somme de la colonne A et la met dans une variable LaSomme.

```
Sub FaireSommeColonne()
LaSomme = Application.Sum(Range("A1").EntireColumn)
MsgBox LaSomme
End Sub
```

Récupérer le nombre de colonnes.

1 - Nombre de colonnes comprises entre la première cellule et la dernière cellule de la plage sélectionnée (y compris éventuellement les colonnes vides)

```
Sub NombreDeColonnes()
NbColonnes=ActiveSheet.UsedRange.Columns.Count
MsgBox NbColonnes
End Sub
```

2 - Nombre de colonnes comprises entre la colonne A (vide ou pas) et la dernière cellule utilisée (cette sélection peut contenir des colonnes vides).

```
Sub NombreDeColonnes()
NbColonnes = Cells.SpecialCells(xlCellTypeLastCell).Column
MsgBox NbColonnes
End Sub
```

Récupérer le nombre de lignes.

1 - Nombre de lignes comprises entre la première cellule et la dernière cellule de la plage sélectionnée (y compris éventuellement les lignes vides)

```
Sub NombreDeColonnes()
NbLignes=ActiveSheet.UsedRange.Rows.Count
MsgBox NbLignes
End Sub
```

2 - Nombre de lignes comprises entre la colonne A (vide ou pas) et la dernière cellule utilisée (cette sélection peut contenir des lignes vides).

```
Sub NombreDeLignes()
NbLignes = Cells.SpecialCells(xlCellTypeLastCell).Column
MsgBox NbLignes
End Sub
```

Récupérer la somme du produit de 2 colonnes. Somme des N valeurs (A1*B1 + A2*B2 ...)

Ce programme effectue le produit des valeurs de la colonne A par celles de la colonne B et additionne ces résultats.

```
Sub SommeProduit()
Cells(1, "A").Select
NbLignes = Cells(Range("A:A").Count, ActiveCell.Column).End(xlUp).Row
NbLignes = Cells(16, 1).End(xlUp).Row
TotalIntermediaire = 0
```

```

For Lgn = 2 To NbLignes 'Etiquettes en ligne 1
  If Not Rows(NbLignes).Hidden Then
    TotalIntermediaire = TotalIntermediaire + Cells(Lgn, "A") * Cells(Lgn, "B")
  End If
Next Lgn
MsgBox TotalIntermediaire
End Sub

```

Répertoire et nom de fichier dans pied de page.

Procédure à mettre dans This Workbook

La mise à jour se fait dès que l'on active un onglet.

```

Private Sub Workbook_SheetActivate(ByVal Sh As Object)
  Dim ZtPath As String
  ZtPath = Application.ActiveWorkbook.Path
  ActiveSheet.PageSetup.LeftFooter = ZtPath & "\" & "&F"
End Sub

```

Supprimer les doublons dans une colonne.

Ce programme efface les doublons dans la colonne sans supprimer les lignes.

```

Sub EffaceDoublonsColonnnes()
  Const Cell_Départ As String = "A1"
  Dim Fin As Range, I As Long, J As Long, Col As Integer
  Dim ModeCalcul As Long
  With Application
    ModeCalcul = .Calculation
    .Calculation = xlCalculationManual
    .ScreenUpdating = False
  End With
  Col = Range(Cell_Départ).Column
  Set Fin = Range(Cell_Départ).End(xlDown)(2)
  On Error Resume Next
  Do
    I = J + 1
    J = Range(Cells(I, 1), Fin).ColumnDifferences(Cells(I, 1))(0).Row
    If J > I Then Range(Cells(I + 1, 1), Cells(J, 1)).ClearContents
  Loop Until Err
  Application.Calculation = ModeCalcul
End Sub

```

Supprimer les lignes vides d'une plage.

Ce programme supprime les lignes vides dans une plage.

```

Sub DetruireLigne()
  DernièreLigne = ActiveSheet.UsedRange.Rows.Count
  Application.ScreenUpdating = False
  For R = DernièreLigne To 1 Step -1
    If Application.CountA(Rows(R)) = 0 Then Rows(R).Delete
  Next R
End Sub

```

Supprimer un nom dans une feuille de calcul.

Ce programme supprime le nom "Base" dans toutes les feuilles du classeur.

```

Sub SupprimeUnNom()
  For Each Nm In ActiveWorkbook.Names
    If Nm.Name Like "Base" Then
      Nm.Delete
    End If
  Next Nm
End Sub

```

Supprimer tous les noms dans une feuille de calcul.

Ce programme supprime tous les noms dans le classeur.

```

Sub SupprimeTouslesNoms()
  For Each Nm In ActiveWorkbook.Names
    Nm.Delete
  Next Nm
End Sub

```

Supprimer tous les noms (avec un joker) dans une feuille de calcul.

Ce programme supprime tous les noms dans le classeur.

```

Sub SupprimeNomsAvecJoker()
  For Each Nm In ActiveWorkbook.Names
    If Nm.Name Like "*Base" Then
      Nm.Delete
    End If
  Next Nm
End Sub

```

Supprimer les lignes contenant du texte

Ce programme recherche dans la colonne A les cellules contenant du texte et supprime les lignes associées à ces cellules.

```
Sub SupprimeLignesAvecTexte()  
NbRw = Application.CountA(Columns("A:A"))  
For Rw = NbRw To 1 Step -1  
    If Application.IsText(Cells(Rw, 1)) Then Rows(Rw).Delete  
Next Rw  
End Sub
```

Utiliser des macro-commandes dans une feuille protégée.

Ce programme permet d'utiliser des macro-commandes dans une feuille protégée par Outils, Protection.

Worksheets("Feuil1").Protect UserInterfaceOnly:=True permet de manipuler par macro une feuille protégée, tout en laissant la protection active par rapport aux manipulations de l'utilisateur.

```
Private Sub Workbook_Open()  
    Worksheets("Feuil1").Protect UserInterfaceOnly:=True  
End Sub
```

Instruction à mettre dans Workbook_Open, cette propriété n'étant pas enregistrée avec le classeur.

Les fichiers (Excel ou autres)

Page mise à jour le : 10/02/2002

Vous trouverez dans cette rubrique des exemples de programmes se rapportant aux traitements sur des fichiers Excel ou autres (txt, ini, log).

- Ecrire dans un fichier de type texte (Date, heure) à l'ouverture ou la fermeture d'Excel.
- Lire et incrémenter un fichier INI.
- Lire les enregistrements dans un fichier texte.
- Lister des fichiers texte et les ouvrir avec GetOpenFileName.
- Lister tous les fichiers XLS dans une feuille de calcul.
- Réaliser une copie du classeur actif sous un autre nom.
- Récupérer le nom d'un fichier sélectionné par GetOpenFileName.

Ecrire dans un fichier de type texte (date et heure).

Ce programme utilise les événements **WorkBook_Open** et **Workbook_BeforeClose** pour inscrire des informations dans un fichier de type texte lors de l'ouverture et de la fermeture d'Excel.
Dans l'exemple, le fichier texte se nomme **activite.log**

```
Private Sub Workbook_Open()  
    Dim LogFile As String  
    LogFile = "C:\Excel\activite.log"  
    ChDir "C:\Excel"  
    Donnees = Now()
```

```

Open LogFile For Append Shared As #1
Print #1, "Ouverture d'Excel a " & Donnees
Close #1
End Sub

```

```

Private Sub Workbook_BeforeClose(Cancel As Boolean)
Dim LogFile As String
LogFile = "C:\Excel\activite.log"
ChDir "C:\Excel"
Donnees = Now()
Open LogFile For Append Shared As #1
Print #1, "Fermeture d'Excel a " & Donnees
Print #1, "-----"
Close #1
End Sub

```

Exemple de fichier généré

```

Ouverture d'Excel a 29/04/01 18:17:15
Fermeture d'Excel a 29/04/01 18:17:41
-----
Ouverture d'Excel a 29/04/01 18:17:45
Fermeture d'Excel a 29/04/01 18:18:23
-----
Ouverture d'Excel a 29/04/01 18:18:27

```

Lire et incrémenter un fichier INI

Ce programme permet d'incrémenter un fichier `incred.ini` et de récupérer la valeur contenue dans ce fichier. Arrivé à 1000, le compteur est réinitialisé à 1.

Structure du fichier `incred.ini`

```

[Numero]
NUMERO=4

```

Ne pas oublier de copier les deux lignes qui suivent en tête de votre module.

```

Declare Function GetPrivateProfileStringA Lib "Kernel32" (ByVal lpAppName As _
String, ByVal lpKeyName As String, ByVal lpDefault As String, ByVal lpReturnedString _
As String, ByVal nSize As Long, ByVal lpFileName As String) As Long

```

```

Declare Function WritePrivateProfileStringA Lib "Kernel32" (ByVal lpAppName _
As String, ByVal lpKeyName As String, ByVal lpString As String, _
ByVal lpFileName As String) As Long

```

```

Sub IncrémenterIni()
Dim Compteur As String * 10
GetPrivateProfileStringA "Numero", "NUMERO", 1, Compteur, 10, "C:\Windows\Incred.ini"
WritePrivateProfileStringA "Numero", "NUMERO", CStr(CLng(Compteur) + 1), _
"C:\Windows\Incred.ini"
MsgBox "Le compteur est incrémenté à : " & Compteur & "."
If Compteur = 1000 Then
MsgBox (" La valeur de 1000 est atteinte. Remise à 1 du compteur.")
Compteur = 1
WritePrivateProfileStringA "Numero", "NUMERO", CStr(CLng(Compteur)), _
"C:\Windows\Incred.ini"
End If
End Sub

```

Lire les enregistrements dans un fichier texte

Ce programme lit les différents enregistrements dans un fichier texte et les inscrit dans une feuille de calcul.

Structure du fichier `Listing.txt`

```

Jordan,Durand,15
Eric,Bataille,52
Marcel,Dupond,35

```

```

Sub LireFichierTexte()
Dim Prenom, Nom, Age
'Ouvre le fichier en lecture

```

```

Open "C:\Excel\Listing.txt" For Input As #1
' Effectue la boucle jusqu'à la fin du fichier
Do While Not EOF(1)
' Lit les données dans trois variables
Input #1, Prenom, Nom, Age
' Ecrit les données dans la feuille de calcul à partir de la ligne 2
Range("A65536").End(xlUp)(2).Value = Prenom
Range("B65536").End(xlUp)(2).Value = Nom
Range("C65536").End(xlUp)(2).Value = Age
Loop
' Ferme le fichier
Close #1
End Sub

```

Le résultat

	A	B	C
1	Prénom	Nom	Age
2	Jordan	Durand	15
3	Eric	Bataille	52
4	Marcel	Dupond	35

Lister des fichiers texte et les ouvrir avec GetOpenFileName

GetOpenFileName:

Affiche la boîte de dialogue standard Ouvrir et lit un nom de fichier tapé ou sélectionné par l'utilisateur sans réellement ouvrir les fichiers.

```

Sub ChoixFichierTexteAOuvrir()
  ChDir "C:\\"
  ChDir "c:\Excel"
  CeFichier = Application.GetOpenFilename("Text Files (*.txt), *.txt")
  If VarType(CeFichier) = vbBoolean Then
    Exit Sub
  Else
    Workbooks.OpenText Filename:=CeFichier, Origin:=xlWindows, _
      StartRow:=1, DataType:=xlDelimited, TextQualifier:=xlDoubleQuote, _
      ConsecutiveDelimiter:=False, Tab:=True, Semicolon:=False, Comma:=False, _
      Space:=False, Other:=False, FieldInfo:=Array(1, 1)
  End If
End Sub

```

Lister tous les fichiers XLS dans une feuille de calcul

Ce programme permet de rechercher tous les fichiers Excel du répertoire Excel, de les placer dans un tableau et de copier ce tableau dans une feuille de calcul.

```

Sub RechercheClasseursSurDisque()
  Dim Classeurs() As String, I As Long
  With Application.FileSearch
    .NewSearch
    .FileType = msoFileTypeExcelWorkbooks
    .LookIn = "C:\Excel\"
    .SearchSubFolders = True
    .Execute
  End With
  With .FoundFiles
    ReDim Classeurs(1 To .Count, 1 To 1)
    For I = 1 To .Count
      Classeurs(I, 1) = .Item(I)
    Next I
  End With
  Application.ScreenUpdating = False
  With Range("A1").Resize(.Count)
    .Value = Classeurs
    .Sort [A1]
  End With
End Sub

```

Réaliser une copie du classeur actif sous un autre nom

Ce programme enregistre le classeur actif sous un autre nom (une copie) sans pour autant modifier le nom du classeur actif.

```

Sub SaveCopyAs()
  ActiveWorkbook.SaveCopyAs "C:\excel\Double.xls"
End Sub

```


Récupérer le nom d'un fichier sélectionné par GetOpenFileName

Cas d'utilisation

Par exemple si vous faites une lecture/écriture (via Open FileName For Input As #1) d'un fichier sélectionné par GetOpenFileName pour inscrire les enregistrements dans un nouveau classeur (via Workbooks.Add template:=xlWorksheet), vous n'avez à aucun moment réellement ouvert ce fichier mais vous voulez en connaître le nom pour le donner à votre classeur actif.

```
Public NameSansExtension As String
Sub SelectionFichier()
    Dim LongFilename As String
    LongFilename = Application.GetOpenFilename("Text Files (*.txt), *.txt")
    ShortFilename (LongFilename)
    MsgBox "Le nom sans extension du fichier est : " & NameSansExtension
End Sub
```

```
Function ShortFilename(LongFilename As String) As String
    For i = Len(LongFilename) To 1 Step -1
        If Mid(LongFilename, i, 1) = "\" Then Exit For
    Next
    ShortFilename = Mid(LongFilename, i + 1, Len(LongFilename))
    NameSansExtension = Mid(ShortFilename, 1, Len(ShortFilename) - 4)
End Function
```

Les fonctions personnalisées

Page mise à jour le : 28/03/2001

- Insérer le chemin du classeur dans une feuille de calcul.
- Somme des cellules avec un motif rouge par exemple.

● Insérer le chemin d'un classeur dans une feuille de calcul.

Cette fonction insère le chemin du classeur dans la feuille de calcul.

Le code

Code de la fonction InsereCheminClasseur

```
Function InsereCheminClasseur() As String
    Application.Volatile
    InsereCheminClasseur = Application.Caller.Parent.Parent.FullName
End Function
```

● Somme des cellules avec un motif rouge.

Cette fonction calcule la somme d'une plage en fonction de la couleur du motif des cellules.

Le code

Code de la fonction SumByColor

```
Function SumByColor(PlageEntree As Range, CouleurPlage As Range) As Double
Dim Cell As Range, TempSum As Double, ColorIndex As Integer
ColorIndex = CouleurPlage.Cells(1, 1).Interior.ColorIndex
TempSum = 0
On Error Resume Next
For Each Cell In PlageEntree.Cells
    If Cell.Formula < > "" Then
        If Cell.Interior.ColorIndex = ColorIndex Then TempSum = TempSum + _
            Cell.Value
    End If
Next Cell
On Error GoTo 0
Set Cell = Nothing
SumByColor = TempSum
End Function
```

Les paramètres de la fonction

PlageEntree : Permet de sélectionner la plage à calculer.

CouleurPlage : Sélectionne la cellule servant de référence pour la couleur du motif.

SumByColor

PlageEntree A1:A12 = {5;10;12;16;19.5;2}

CouleurPlage D1 = 0

= 40

Pour obtenir de l'aide sur cette fonction et ses arguments, consultez l'Aide.

PlageEntree

Résultat = 40

OK Annuler

Les formules Excel

Page mise à jour le : 10/02/2002

Vous trouverez dans cette rubrique un recueil de formules pouvant vous rendre de précieux services.

- Calcul sur 12 mois glissants.
- Calcul sur les heures.
- Calcul de l'ancienneté en Mois et Années.
- Calculer une somme entre deux bornes.
- Exemples de formats personnalisés.
- Moyenne ne tenant pas compte des valeurs zéro.
- Réaliser un filtre élaboré (Champ calculé).
- Réaliser un filtre élaboré (Champ avec condition ET).
- Récupérer la valeur d'une cellule résultant d'une intersection Ligne/Colonne.
- Somme des cellules (A1 par ex.) de plusieurs feuilles.
- Somme d'une colonne avec un nombre de lignes variable.
- Trouver le quantième du jour.
- Trouver les dates mobiles (jours fériés).

Calcul sur 12 mois glissants 

	A	B	C	D	E	F	G
1	Calcul avec 12 mois glissants						
2	Mois	Valeur			Avec formule pour glissement		
3	avril-98	20					
4	mai-98	5		10.17	=MOYENNE(DECALER(B3;NB(B:B)-1;;-12))		
5	juin-98	12					
6	juillet-98	10					
7	août-98	15					
8	septembre-98	12					
9	octobre-98	13					
10	novembre-98	8					
11	décembre-98	5					
12	janvier-99	10			<i>La formule prend en compte uniquement les 12 dernières valeurs</i>		
13	février-99	12					
14	mars-99	15					
15	avril-99	10					
16	mai-99	10					
17	juin-99	10					
18	juillet-99	10					
19	août-99	10					
20	septembre-99	10					
21	octobre-99	10					
22	novembre-99	10					

Calcul sur les heures 

	A	B	C	D	E	F	G
1	Calcul sur les heures						
2	1 - Comment transformer des heures (heures+centièmes) en heures+minutes et vice-versa						
3	Par exemple :						
4	<i>Convertir 1.5 heures en 1heure 30 minutes</i>						
5	Résultat						
6	Heure décimale	Heure sexagésimale	Format cellule colonne B		Heure sexagésimale	Heure décimale	Format cellule colonne F
7	1.5	1:30:00	[h]:mm:ss		01:30:00	1.5	Standard
8	Formules						
9	1.5	=A8/24	[h]:mm:ss		01:30:00	=E8*24	Standard
10							
11	2 - Additionner des heures, minutes et secondes						
12	<i>Calcul sur les heures (en sexagésimal)</i>						
13					Format	Formule	
14					[h]:mm:ss		
15	12:50:00	14:20:00	18:25:00		45:35:00	=A14+B14+C14	
16							
17							
18	Précisions sur les formats d'heure						
19	Les codes [h], [m] et [s] permettent de représenter des nombres cumulés (c'est-à-dire des durées) d'heures, de minutes ou de secondes, alors que les codes h, m et s renvoient des nombres d'heures, de minutes ou de secondes à l'intérieur d'une autre unité de temps (jour, heure ou minute).						
20	Par exemple, la formule ="23:05:00"+"10:47:03" renvoie :						
21	- 09:52:03 si la cellule est mise au format "hh:mm:ss"						
22	- 33:52:03 si la cellule est mise au format "[hh]:mm:ss"						
23	Le premier format correspond à l'heure du jour (ici 9 h 52 au lendemain de la première date-heure), alors que le deuxième fait le cumul des heures sans "reboucler" quand l'addition donne un nombre d'heures supérieur ou égal à 24.						
24	D'une manière générale, il faut éviter le format 'hh:mm:ss' dès lors que l'on fait des calculs de durée, et lui préférer soit '[hh]:mm:ss', soit par exemple 'j "jour(s)" hh:mm:ss' si l'on veut indiquer une durée en nombre de jours et d'heures.						

Calcul de l'ancienneté en Mois et Années 

	A	B	C	D	E	F	G	H
1	Existe-t-il une formule qui me donnerait l'ancienneté d'un employé en année et en mois ?							
2								
3	=DATEDIF(Date_début;Date_fin;Type)							
4								
5	L'argument Type peut prendre les valeurs suivantes :							
6								
7	"y" = différence absolue en années							
8	"m" = différence absolue en mois							
9	"d" = différence absolue en jours							
10	"ym" = différence en mois si les 2 dates étaient dans la même année							
11	"yd" = différence en jours si les 2 dates étaient dans la même année							
12	"md" = différence en jours si les 2 dates étaient dans le même mois							
13								
14					Résultat en ans	Résultat en mois	Résultat en jours	
15	Date d'entrée	--- >	08/04/69					
16	Date du jour	--- >	28/12/99		30	368	11221	
17					=DATEDIF(C18;C19;"y")			
18					=DATEDIF(C18;C19;"m")			
19					=DATEDIF(C18;C19;"d")			
20								
21	Inutile de rechercher DATEDIF dans l'assistant fonction, dans l'aide en ligne ou							
22	dans les manuels, tu ne la trouveras pas. Elle est masquée et non documentée, pour							
23	des raisons que j'ignore. Mais elle marche impeccablement.							

Calculer une somme entre deux bornes

	A	B	C	D	E	F	G	H
1	Faire la somme d'une colonne B ou d'une plage que si les montants sont dans une certaine tranche (>E7 et < F7)							
2								
3		10						
4		24						
5		30						
6		50						
7		100						
8		150						
9		210						
10								
11								
12								
13								

Fourchette de calcul	
Supérieur à	Inférieur à
25	200

Résultats	
330	=SOMME.SI(\$B:\$B;">="&\$E\$7)-SOMME.SI(\$B:\$B;">"&\$F\$7)
330	=SOMME(SOMME.SI(Plage;">=25";"<=200";"<>"))*(1;-1))
330	{=SOMME((Plage>=25)*(Plage<=200)*Plage)}

Exemples de formats personnalisés

Aspect	Format personnalisé
2 500	###0_F;-##0_F
1 260	###0_F;[Rouge]-##0_F
-1 260	###0_F;[Rouge]-##0_F
1 025.00	###0.00_F;-##0.00_F
4 521.00	###0.00_F;[Rouge]-##0.00_F
-4 521.00	###0.00_F;[Rouge]-##0.00_F
1 025 F	##0 F;-##0 F
2 560 F	##0 F;[Rouge]-##0 F
-2 560 F	##0 F;[Rouge]-##0 F
1 025.00 F	##0.00 F;-##0.00 F
1 025.00 F	##0.00 F;[Rouge]-##0.00 F
-1 025.00 F	##0.00 F;[Rouge]-##0.00 F
1 025.00 €	###0.00\ "€";-##0.00\ "€"
1 025 €	###0\ "€";-##0#\ "€"
PC 1025	PC" "0
12 ans	"0 ans"
1 ans	"0 ans"
1 an	[<1.05]0" an";0" ans"
2 ans	[<1.05]0" an";0" ans"
10.00	0,00;"(0,00)";0,00 (10)
(5.00)	0,00;"(0,00)";0,01 (-5)
Total :15 000 F	= " Total : "& TEXTE(15000;" ##0 F;- ##0 F ")
Total : 150 F	= " Total : "& TEXTE(D24;" ##0 F;- ##0 F")
Imprimé le : 28/12/1999	= "Imprimé le : "& TEXTE(MAINTENANT();"jj/mm/aaaaa")
Imprimé le : 28/12/1999 - 18:03	= "Imprimé le : "& TEXTE(MAINTENANT();"jj/mm/aaa - h:mm")

Moyenne ne tenant pas compte des valeurs zéro

	A	B	C	D	E	F	G
1	Comment faire une MOYENNE ne tenant pas compte des valeurs ZÉRO ?						
2							
3							
4							
5	20						
6	20						
7	20		20		=SOMME(Plage)/NB.SI(Plage;">0")	Ignore les lignes avec la valeur 0	
8	0		16		=MOYENNE(Plage)	Prend en compte le lignes avec 0	
9	20						
10							
11							
12		Le nom Plage est attribué à la zone colorée en bleu clair.					

Réaliser un filtre élaboré (Champ calculé)

Le champ calculé permet d'extraire les noms dont le salaire actuel est \geq au salaire de début x 2 .

	A	B	C	D	E	F	G	H	I	J	K
1	NOM	DEBUT	ACTUEL		NOM	DEBUT			NOM	DEBUT	ACTUEL
2	RENAUD	5000	7500				FAUX		DUVAL	4800	9700
3	DUPOND	5200	8900						DIRECT	4500	9500
4	DUVAL	4800	9700		Formule en G2		=ACTUEL>=DEBUT*2		JERS	6000	13000
5	DIRECT	4500	9500						PILARD	6100	14000
6	JERS	6000	13000								
7	PILARD	6100	14000				Zone de critères				
8	MASSON	5950	10500								
9	VAUX	6000	11500				Zone d'extraction				

Réaliser un filtre élaboré (Champ calculé avec condition ET) 

Extraire les noms dont la date de début est >= au 31/12/1997 **ET** la date de fin < 01/01/2001.

	A	B	C	D	E	F	G	H
1	Nom	Date		Nom			Nom	Date
2	RENAUD	01/01/98			VRAI		RENAUD	01/01/98
3	DUPOND	01/01/99					DUPOND	01/01/99
4	DUVAL	01/01/00		Formule en E2 :			DUVAL	01/01/00
5	DIRECT	01/01/01		=ET(B2>35795;B2<36892)				
6	JERS	01/01/02						
7	PILARD	01/01/03		Bornes				
8	MASSON	01/01/04		35795	31/12/97			
9	VAUX	01/01/05		36892	01/01/01			
10	DEPT	01/01/06						
11	GRAUX	01/01/07						
12								
13				Zone de critères				
14								
15				Zone d'extraction				

Récupérer la valeur d'une cellule résultant d'une intersection 

Ligne/Colonne

	A	B	C	D	E	F	G	H
1		Poids en grammes						
2	Départements	0	51	251	500		Noms de plage	
3	01	100	160	200	300		B2:E2 = Poids	
4	02	95	140	180	250		A3:A5 = Dépt	
5	03	80	120	150	200			
6								
7								
8		Département	02					
9		Poids	181					
10				Les formules en C11, C12, C13 et C14				
11			4	=2+EQUIV(D8;Dépt;0)				
12			3	=1+EQUIV(D9;Poids;1)				
13			\$C\$4	=ADRESSE(2+EQUIV(D8;Dépt;0);1+EQUIV(D9;Poids;1))				
14		Frais de port	140	=INDIRECT(ADRESSE(2+EQUIV(D8;Dépt;0);1+EQUIV(D9;Poids;1);;))				

Somme des cellules (A1 par ex.) de plusieurs feuilles 

J'ai un classeur, une floppée de feuilles...				
La première feuille "Bilan" somme les valeurs respectives, de certaines cellules de même adresse, de chaque feuille.				
Aujourd'hui, je tape dans A1 par exemple : =Janvier!A1+Février!A1+Mars!A1				
Avant				
6150	=Janvier!A1+Février!A1+Mars!A1+Avril!A1			
Après				
Une solution plus rationnelle				
6150	=SOMME(Janvier:Avril!A1)			
L'avantage de la formule ci-dessus, c'est qu'elle somme toutes les valeurs des cellules A1 comprises entre Janvier et Avril qui en fait servent de bornes.				
Si on ajoute une feuille avec un nom quelconque entre Janvier et Avril, la valeur A1 de cette nouvelle feuille sera prise en compte dans la formule.				

Somme d'une colonne avec un nombre de lignes variable

Cas 1 : Les valeurs commencent à la ligne 1

=SOMME(DECALER(\$A\$1;NB(A:A);;-NB(A:A)))

Cas 2 : Les valeurs commencent à la ligne 2

=SOMME(DECALER(\$A\$1;NB(A:A);;-NB(A:A)-1))

Trouver le quantième du jour

	A	B	C
1	Calculer le quantième du jour d'une date		
2			
3	Date	Quantième	Formule
4	31/12/96	366	=A3-DATE(ANNEE(A3);1;0)
5	31/12/97	365	
6	31/12/98	365	
7	01/01/99	1	
8	23/07/99	204	
9	31/12/99	365	
10	01/01/00	1	
11	31/12/00	366	
12			
13	Valeur retournée par DATE(ANNEE(A18);1;0)		
14			
15	22/07/99	31/12/98	=DATE(ANNEE(A15);1;0)
16			
17	Calculer le nombre de jours restants pour atteindre la fin de l'année		
18			
19	Date	Nb jours restants	Formule
20	31/12/96	0	=DATE(ANNEE(A20);12;31)-A16
21	31/12/97	0	
22	31/12/98	0	
23	01/01/99	364	
24	23/07/99	161	
25	31/12/99	0	
26	01/01/00	365	
27	31/12/00	0	
28			
29	Valeur retournée par DATE(ANNEE(A31);12;31)		
30			
31	22/07/99	31/12/99	=DATE(ANNEE(A31);12;31)

Trouver les dates mobiles (jours fériés)

Les jours fériés sont de deux sortes :

Jours à date fixe (comme le 14 juillet, le 11 novembre et le 25 décembre)

Jours mobiles, dont la date varie d'une année sur l'autre. Ces jours fériés mobiles sont calculés d'après la date du dimanche de Pâques.

Dès lors que l'on connaît la date du dimanche de Pâques, on peut en dériver les dates de tous les jours fériés mobiles par simple addition.

Le dimanche de Pâques est le premier dimanche qui suit la première pleine lune après l'équinoxe de printemps (21 mars).

La formule suivante, signée Norbert Hetterich, renvoie la date du dimanche de Pâques pour l'année A

(entre 1900 et 2078) :

=FRANC((JOUR(MINUTE(A/38)/2+55)&"/4/"&A)/7;)*7-6

Les jours fériés mobiles en France et en Belgique sont le Lundi de Pâques (Dimanche de Pâques ("DP") + 1),

le Jeudi de l'Ascension (DP + 39) et le Lundi de Pentecôte (DP + 50).

En partant de la formule de N. Hetterich, on peut donc obtenir facilement les dates de ces trois jours

fériés pour une année donnée (entre 1900 et 2078) :

Fêtes mobiles	Formules
Dimanche de Pâques	
2 avril 2000	=FRANC((JOUR(MINUTE(ANNEE(MAINTENANT())/38)/2+55)&"/4/"&2000)/7;)*7-6
Lundi de Pâques	
3 avril 2000	=FRANC((JOUR(MINUTE(ANNEE(MAINTENANT())/38)/2+55)&"/4/"&2000)/7;)*7-5
Judi de l'Ascension	
13 mai 1999	=FRANC((JOUR(MINUTE(ANNEE(MAINTENANT())/38)/2+55)&"/4/"&ANNEE(MAINTENANT())/7;)*7+33
Lundi de Pentecôte	
24 mai 1999	=FRANC((JOUR(MINUTE(ANNEE(MAINTENANT())/38)/2+55)&"/4/"&ANNEE(MAINTENANT())/7;)*7+44

Les graphiques Excel

Page mise à jour le : 28/03/2001

Vous trouverez dans cette rubrique un recueil de graphiques pouvant vous rendre de précieux services.

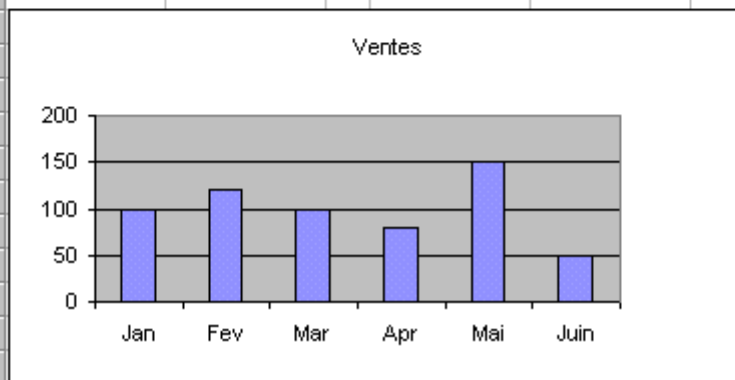
- Réaliser un graphique dynamique (Données en colonnes).
- Réaliser un graphique dynamique sur les 12 dernières valeurs (Données en colonnes).
- Réaliser un graphique dynamique (Données en lignes).
- Réaliser un graphique dynamique sur les 12 dernières valeurs (Données en lignes).

Réaliser un graphique dynamique (données en colonnes)

L'onglet s'intitule Données et le fichier GraphiqueDynamique.
Les données sont en colonnes.

	A	B	C	D	E	F	G
1	Comment créer un graphique dynamique qui prendra en compte les nouvelles données entrées.						
2							
3							
4	Date	Ventes					
5	Jan	100					
6	Fev	120					
7	Mar	100					
8	Apr	80					
9	Mai	150					
10	Juin	50					
11							
12	Comment procéder pour afficher les nouvelles valeurs saisies ?						
13	1 / Utiliser Insertion/Nom/Définir pour créer les deux noms (Ventes et Date)						
14	Date=DECALER(Données!\$A\$5;0;0;NBVAL(Données!\$A:\$A)-1)						
15	Ventes=DECALER(Données!\$B\$5;0;0;NBVAL(Données!\$B:\$B)-1)						
16	2 / Dessiner le graphique en sélectionnant A4:B10						
17	3 / Sélectionner la série de données du graphique						
18	4 / Clic droit avec la souris et sélectionner Données source						
19	5 / Sélectionner l'onglet Série						
20	5.1 - La zone de saisie intitulée Valeurs affiche :						
21	=Données!\$B\$5:\$B\$10						
22	Remplacez par : *=GraphiqueDynamique.xls!Ventes						
23	5.2 La zone de saisie intitulée Abscisses affiche :						
24	=Données!\$A\$5:\$A\$10						
25	Remplacez par : *=GraphiqueDynamique.xls!Date						
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							
36							
37							

Dans l'exemple, on peut ajouter dans la colonne Date, les mois suivants à savoir Juillet, Août, etc... Le graphique se retrace automatiquement.



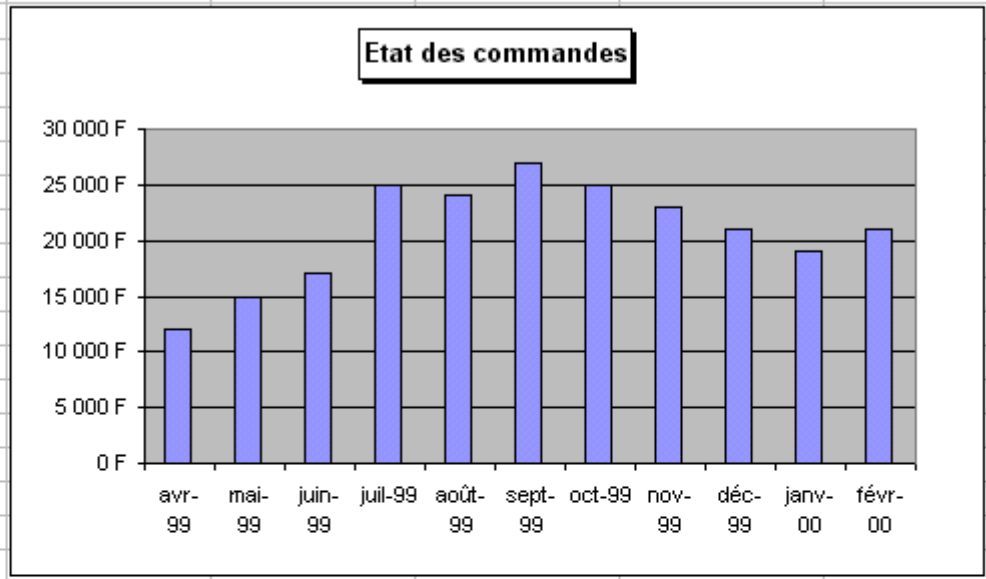
Réaliser un graphique dynamique sur les 12 dernières valeurs (en colonnes)

L'onglet s'intitule Etat Commandes et le fichier GraphiqueDynamique.

Les données sont en colonnes.

Comment créer un graphique dynamique qui prendra en compte les 12 dernières données saisies (en colonnes).

Commandes	Montant
janv-99	18 000 F
févr-99	25 000 F
mars-99	12 000 F
avr-99	15 000 F
mai-99	17 000 F
juin-99	25 000 F
juil-99	24 000 F
août-99	27 000 F
sept-99	25 000 F
oct-99	23 000 F
nov-99	21 000 F
déc-99	19 000 F
janv-00	21 000 F
févr-00	23 000 F

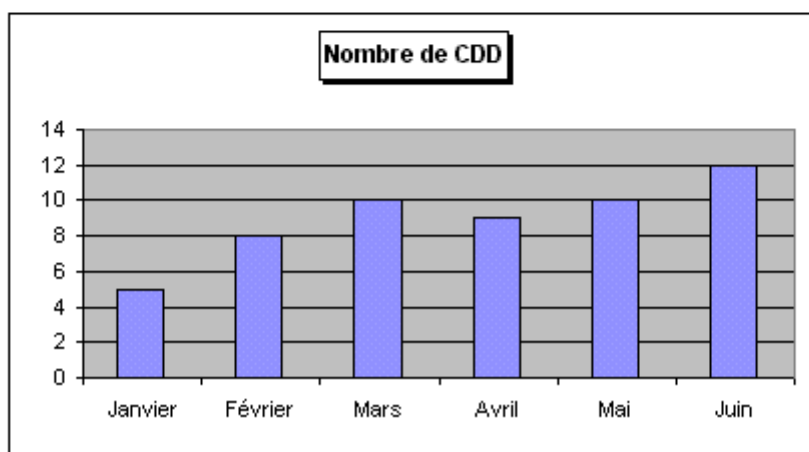


Comment procéder pour afficher les 12 derniers mois ?

- 1 / Utiliser Insertion/Nom/Définir pour créer les deux noms (Commandes et Montant)
 Commandes =DECALER(Etat Commandes!\$A\$4;NBVAL(Etat Commandes!\$A:\$A)-1;;-12)
 Montant =DECALER(Etat Commandes!\$B\$4;NBVAL(Etat Commandes!\$B:\$B)-1;;-12)
- 2 / Dessiner le graphique en sélectionnant A4:B18
- 3 / Sélectionner la série de données du graphique
- 4 / Clic droit avec la souris et sélectionner Données source
- 5 / Sélectionner l'onglet Série
 - 5.1 - La zone de saisie intitulée Valeurs affiche :
 =Etat Commandes!\$B\$5:\$B\$18
 Remplacez par : =GraphiqueDynamique.xls!Montant
 - 5.2 La zone de saisie intitulée Abcisses affiche :
 =Etat Commandes!\$A\$5:\$A\$18
 Remplacez par : =GraphiqueDynamique.xls!Commandes

Réaliser un graphique dynamique sur données en lignes L'onglet s'intitule Personnel et le fichier GraphiqueDynamique. Les données sont en lignes.

	A	B	C	D	E	F	G
1	Comment créer un graphique dynamique						
2	qui prendra en compte les nouvelles données saisies (en lignes).						
3							
4	Mois	Janvier	Février	Mars	Avril	Mai	Juin
5	CDD	5	8	10	9	10	12
6							
7	On peut ajouter des Mois et des valeurs CDD, le graphique les prendra en compte.						
8							
9	Comment procéder pour afficher les nouvelles valeurs saisies ?						
10	1 / Utiliser Insertion/Nom/Définir pour créer les deux noms (Mois et CDD)						
11	Mois =DECALER(Personnel\$B\$4;;;NBVAL(Personnel\$4:\$4)-1)						
12	CDD =DECALER(Personnel\$B\$5;;;NBVAL(Personnel\$5:\$5)-1)						
13	2 / Dessiner le graphique en sélectionnant A4:G5						
14	3 / Sélectionner la série de données du graphique						
15	4 / Clic droit avec la souris et sélectionner Données source						
16	5 / Sélectionner l'onglet Série						
17	5.1 - La zone de saisie intitulée Valeurs affiche :						
18	=Personnel\$B\$5:\$G\$5						
19	Remplacez par : =GraphiqueDynamique.xls!CDD						
20	5.2 La zone de saisie intitulée Abcisses affiche :						
21	=Personnel\$B\$4:\$G\$4						
22	Remplacez par : =GraphiqueDynamique.xls!Mois						



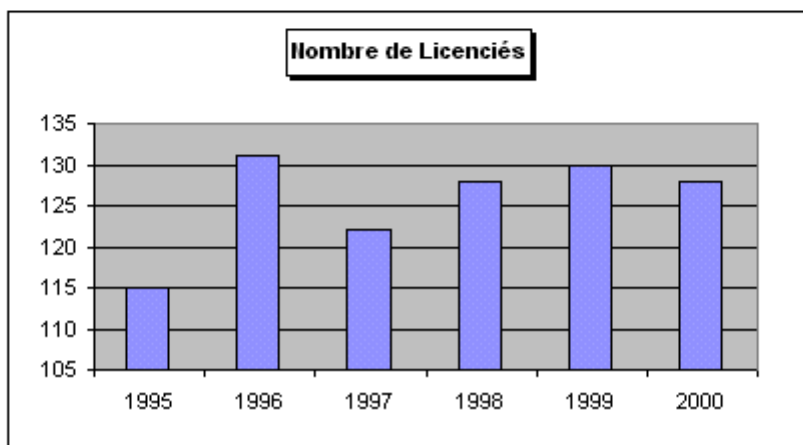
Réaliser un graphique dynamique sur les 12 dernières valeurs en lignes 
L'onglet s'intitule Effectif et le fichier GraphiqueDynamique.
Les données sont en lignes.

	A	B	C	D	E	F	G	H	I
1	Comment créer un graphique dynamique								
2	qui prendra en compte les 6 dernières données saisies (en lignes).								

	Saison	1993	1994	1995	1996	1997	1998	1999	2000
5	Licenciés	115	120	115	131	122	128	130	128

8	Comment procéder pour afficher les 6 dernières années ?								
---	--	--	--	--	--	--	--	--	--

- | | | | | | | | | | |
|----|---|--|--|--|--|--|--|--|--|
| 9 | 1 / Utiliser Insertion/Nom/Définir pour créer les deux noms (Année et Licenciés) | | | | | | | | |
| 10 | Saison =DECALER(Effectif!\$B\$4;;NBVAL(Effectif!\$4:\$4)-2;;-6) | | | | | | | | |
| 11 | Licenciés =DECALER(Effectif!\$A\$5;;NBVAL(Effectif!\$5:\$5)-1;;-6) | | | | | | | | |
| 12 | 2 / Dessiner le graphique en sélectionnant A4:H5 | | | | | | | | |
| 13 | 3 / Sélectionner la série de données du graphique | | | | | | | | |
| 14 | 4 / Clic droit avec la souris et sélectionner Données source | | | | | | | | |
| 15 | 5 / Sélectionner l'onglet Série | | | | | | | | |
| 16 | 5.1 - La zone de saisie intitulée Valeurs affiche : | | | | | | | | |
| 17 | =Effectif!\$B\$5:\$H\$5 | | | | | | | | |
| 18 | Remplacez par =GraphiqueDynamique.xls!Licenciés | | | | | | | | |
| 19 | 5.2 La zone de saisie intitulée Abcisses affiche : | | | | | | | | |
| 20 | =Effectif!\$B\$4:\$H\$4 | | | | | | | | |
| 21 | Remplacez par =GraphiqueDynamique.xls!Saison | | | | | | | | |

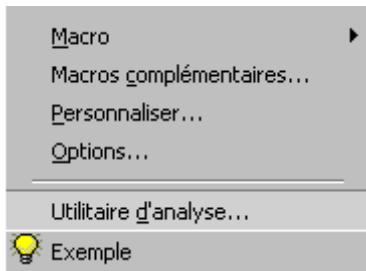


Les menus et barres d'outils

Vous trouverez dans cette rubrique des exemples pour créer, modifier ou personnaliser les menus d'Excel.

Ajout d'un item au menu Outils.

- 1 - Ajouter un item intitulé **Exemple** en dernière position du menu **Outils**.



Le code pour installer l'item dans le menu Outils

```
Sub AddItemInMenuOutils()
  With Application.CommandBars("Tools").Controls.Add(msoControlButton)
    .Caption = "Exemple"
    .OnAction = "Affiche"
    .FacelId = 343
  End With
End Sub
```

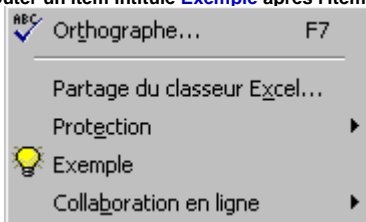
Le code pour effacer l'item du menu Outils

```
Sub EffaceItem()
  CommandBars("Tools").Controls("Exemple").Delete
End Sub
```

Le code de l'action associée à l'item Exemple

```
Sub Affiche()
  Dim Msg, Style, Title
  Msg = "Vous avez ajouté un item intitulé {Exemple} dans le menu Outils"
  Style = vbYes
  Title = "Ajout d'items dans le menu Outils - © Eric RENAUD - 11/1999"
  Response = MsgBox(Msg, Style, Title)
End Sub
```

2 - Ajouter un item intitulé Exemple après l'item Protection du menu Outils.



Le code pour installer l'item dans le menu Outils

```
Sub AddItemInMenuOutils()
  With Application.CommandBars("Tools").Controls.Add(msoControlButton, before:=7)
    .Caption = "Exemple"
    .OnAction = "Affiche"
    .FacelId = 343
  End With
End Sub
```

Le code pour effacer l'item du menu Outils

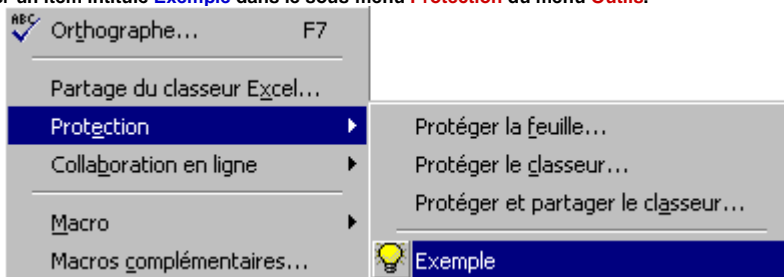
```
Sub EffaceItem()
  CommandBars("Tools").Controls("Exemple").Delete
End Sub
```

Le code de l'action associée à l'item Exemple

```
Sub Affiche()
  Dim Msg, Style, Title
  Msg = "Vous avez ajouté un item intitulé {Exemple} dans le menu Outils"
  Style = vbYes
  Title = "Ajout d'items dans le menu Outils - © Eric RENAUD - 11/1999"
  Response = MsgBox(Msg, Style, Title)
End Sub
```

Ajout d'un item au menu Outils, Protection. 

Ajouter un item intitulé Exemple dans le sous-menu Protection du menu Outils.



Le code pour installer l'item dans le menu Outils, Protection

```
Sub AddItemInMenuOutilsProtection()  
With Application.CommandBars("Protection").Controls.Add(msoControlButton)  
    .Caption = "Exemple"  
    .OnAction = "Affiche"  
    .FaceId = 343  
End With  
End Sub
```

Le code pour effacer l'item du menu Outils, Protection

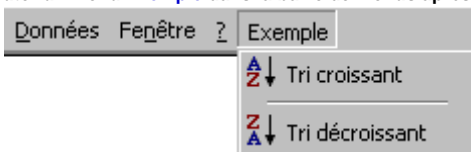
```
Sub EffaceItem()  
CommandBars("Tools").Controls("Protection").Controls(4).Delete  
End Sub
```

Le code de l'action associée à l'item Exemple

```
Sub Affiche()  
Dim Msg, Style, Title  
Msg = "Vous avez ajouté un item intitulé {Exemple} dans le menu Outils."  
Style = vbYes  
Title = "Ajout d'items dans le menu Outils, Protection - © Eric RENAUD - 11/1999"  
Response = MsgBox(Msg, Style, Title)  
End Sub
```

Ajout d'un nouveau menu.

1 - Ajouter un menu Exemple dans la barre de menus après le menu ?



Le code pour ajouter le menu Exemple.

```
Sub InserenouveauMenu()  
With CommandBars(1).Controls.Add(msoControlPopup)  
    .Caption = "Exemple"  
    With .Controls.Add(msoControlButton)  
        .Caption = "Tri croissant"  
        .FaceId = 210  
        .BeginGroup = True  
        .OnAction = "Affiche"  
    End With  
    With .Controls.Add(msoControlButton)  
        .Caption = "Tri décroissant"  
        .FaceId = 211  
        .BeginGroup = True  
        .OnAction = "Affiche"  
    End With  
End With  
End Sub
```

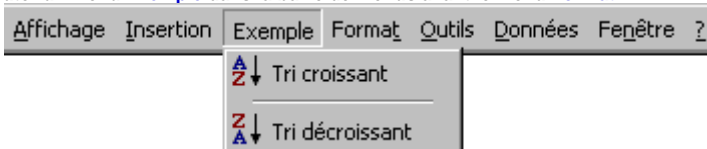
Le code pour supprimer le menu Exemple.

```
Sub SupprimeNouveauMenu()  
CommandBars(1).Controls("Exemple").Delete  
End Sub
```

Le code de l'action associée aux items Exemple

```
Sub Affiche()  
Dim Msg, Style, Title  
Msg = "Vous avez ajouté un nouveau menu {Exemple} dans la barre de menus"  
Style = vbYes  
Title = "Ajout d'un nouveau menu - © Eric RENAUD - 11/1999"  
Response = MsgBox(Msg, Style, Title)  
End Sub
```

2 - Ajouter un menu Exemple dans la barre de menus avant le menu Format.



Le code pour ajouter le menu Exemple avant le menu Format.

Ce code est identique à l'exemple précédent sauf qu'il faut ajouter `before:=5`, 5 étant le le numéro du menu avant lequel on veut insérer le menu Exemple.

La ligne de code :

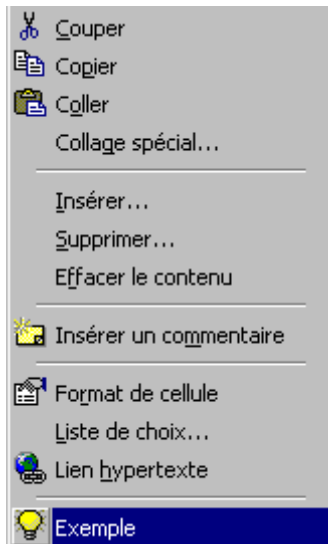
```
With CommandBars(1).Controls.Add(msoControlPopup)
```

devient :

```
With CommandBars(1).Controls.Add(msoControlPopup, before:=5)
```

Ajout d'items au menu contextuel (clic droit).

1 - Ajouter un item Exemple dans menu contextuel (clic droit souris).



Le code pour ajouter le menu contextuel.

```
Sub InsereMenuContextuel()
With Application.CommandBars("Cell").Controls.Add(msoControlButton)
    .Caption = "Exemple"
    .BeginGroup = True
    .FaceId = 343
    .OnAction = "Affiche"
End With
End Sub
```

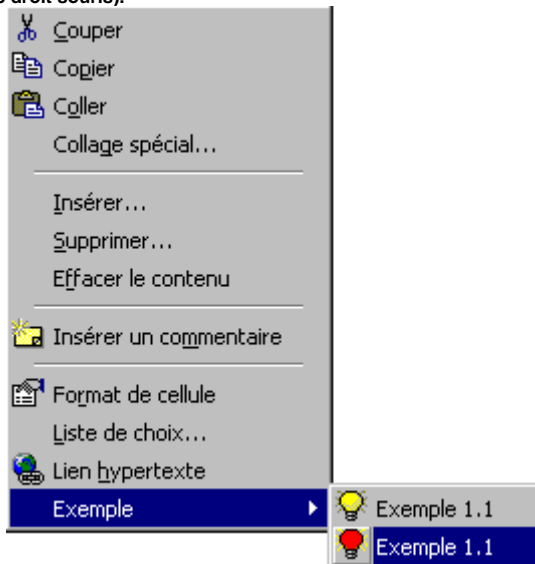
Le code pour supprimer le menu contextuel.

```
Sub SupprimeMenuContextuel()
Application.CommandBars("Cell").Reset
End Sub
```

Le code de l'action associée à l'item Exemple

```
Sub Affiche()
Dim Msg, Style, Title
Msg = "Vous avez ajouté un item intitulé {Exemple} dans le menu Outils"
Style = vbYes
Title = "Ajout d'item dans menu contextuel - © Eric RENAUD - 11/1999"
Response = MsgBox(Msg, Style, Title)
End Sub
```

2 - Ajouter un menu popup depuis l'item Exemple dans menu contextuel (clic droit souris).



Le code pour ajouter le menu contextuel.

```
Sub InsereMenuContextuePopUp()
With Application.CommandBars("Cell")
    With .Controls.Add(msoControlPopup)
        .Caption = "Exemple"
        .BeginGroup = False
        ' Sous-menu 1 (Exemple 1.1)
        .Controls.Add (msoControlButton)
        .Controls(1).Caption = "Exemple 1.1"
        With .Controls(1)

```

```

        .OnAction = "Affiche"
        .FacelId = 351
    End With
' Sous-menu 2 (Exemple 1.2)
    .Controls.Add (msoControlButton)
    .Controls(2).Caption = "Exemple 1.2"
    With .Controls(2)
        .OnAction = "Affiche"
        .FacelId = 352
    End With
End With
End Sub
Le code pour supprimer le menu contextuel.
Sub SupprimeMenuContextuel()
Application.CommandBars("Cell").Reset
End Sub
Le code de l'action associée à l'item Exemple
Sub Affiche()
Dim Msg, Style, Title
Msg = "Vous avez ajouté un item intitulé {Exemple} dans le menu Outils"
Style = vbYes
Title = "Ajout d'item dans menu contextuel - © Eric RENAUD - 11/1999"
Response = MsgBox(Msg, Style, Title)
End Sub

```

Masquer des items dans un menu (ou réafficher).

1 - Rendre inaccessible le menu **Outils**.



Le code pour rendre inaccessible l'item **Outils**.

```

Sub InhibeMenuOutils()
CommandBars(1).Controls(6).Enabled = False
End Sub

```

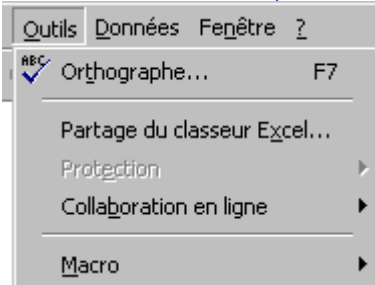
Le code pour rendre accessible l'item **Outils**.

```

Sub AfficheMenuOutils()
CommandBars(1).Controls(6).Enabled = True
End Sub

```

2 - Rendre inaccessible le menu **Outils, Protection**.



Le code pour rendre inaccessible l'item **Outils, Protection**.

```

Sub InhibeOutilsProtection()
CommandBars(1).Controls(6).Controls(6).Enabled = False
End Sub

```

Le code pour rendre accessible l'item **Outils, Protection**.

```

Sub AfficheOutilsProtection()
CommandBars(1).Controls(6).Controls(6).Enabled = True
End Sub

```

Supprimer l'affichage d'un menu (ou réafficher).

Supprimer l'intitulé **Outils** de la barre de menus.



Le code pour supprimer l'intitulé **Outils** de la barre de menus.

```

Sub SupprimeIntituleOutils()
CommandBars(1).Controls(6).Visible = False
End Sub

```

Le code pour réafficher l'intitulé **Outils** dans la barre de menus.

```

Sub ReinstalleIntituleOutils()
CommandBars(1).Controls(6).Visible = True

```

Sélection dans une feuille de calcul (à l'aide du code VBA)

Page mise à jour le : 28/03/2001

Vous trouverez dans cette rubrique des exemples de programmes VBA pour effectuer différents types de sélection de cellules.

- Sélection d'une plage avec la souris.
- Sélection d'une plage.
- Sélection de 2 plages (non contigües).
- Sélection de deux colonnes non contigües.
- Sélection de cellules caractéristiques (par ex. cellules égales à 20).
- Sélection de cellules vers le bas depuis la cellule active (Ctrl+Shift+Bas).
- Sélection de cellules vers le haut depuis la cellule active (Ctrl+Shift+Haut).
- Sélection de cellules vers la droite depuis la cellule active (Ctrl+Shift+Droit).
- Sélection de cellules vers la gauche depuis la cellule active (Ctrl+Shift+Gauche).
- Sélection de cellules dans la plage courante depuis la cellule active.
- Sélection de cellules contigües dans la colonne de la cellule active.
- Sélection de cellules contigües dans la ligne de la cellule active.
- Sélection d'une colonne entière par rapport à la cellule active.
- Sélection d'une colonne entière par rapport à la cellule active.

Sélection d'une plage avec la souris.

Permet de sélectionner une plage avec la souris (Méthode InputBox)
Sub SelectionPlageAvecSouris()
Dim Plage As Range
Set Plage = Application.InputBox("Sélectionnez une plage !", "Sélection de cellules", Type:=8)
MsgBox ("La plage que vous avez sélectionnez est : " & Plage.Address)
End Sub

Sélection d'une plage.

Permet de sélectionner la zone en cours par rapport à la cellule active.
Sub SelectionPlage()
ActiveCell.CurrentRegion.Select
MsgBox ("La plage sélectionnée est : " & Selection.Address)
End Sub

Sélection de 2 plages (non contigües).

Permet de sélectionner deux plages discontinues (Plage A1:Ax et C1:Cx)

```
Sub SelectionDiscontinue()
    Dim Z1, Z2, MaPlageMultiZone As Range
    Worksheets("Feuil1").Select
    Range("A1").Select
    ActiveCell.End(xlDown).Select
    Zone1 = ActiveCell.Address
    Selection.Offset(0, 2).Select
    Zone2 = ActiveCell.Address
    Set Z1 = Range("A1", Zone1)
    Set Z2 = Range("C1", Zone2)
    Set MaPlageMultiZone = Union(Z1, Z2)
    ZoneSelection = MaPlageMultiZone.Select
End Sub
```

A	B	C
Nom	Age	Taille
Dupond	20	1.70
Durand	25	1.75
Bidon	22	1.80
Carre	30	1.68
Venter	21	1.82
Menier	32	1.78
Marloir	40	1.76
Perrier	25	1.70
Soltard	18	1.72

Sélection de deux colonnes non contigües.

Permet de sélectionner deux colonnes A et D par exemple.

```
Sub SelectionDeuxColonnesNonContigues()
    ' Soit les colonnes A (1) et D (4) à sélectionner
    NCol1 = 1
    NCol2 = 4
    Union(Cells(1, NCol1), Cells(1, NCol2)).EntireColumn.Select
End Sub
```

Sélection de cellules caractéristiques (par ex. cellules égales à 20).

Permet de sélectionner toutes les cellules dont la valeur est 20

```
Sub CellulesValeurDeterminée()
    ' La valeur 20 est saisie en E2
    LaValeur = Range("E2").Value
    Range("A1").Select
    For Each CII In ActiveCell.CurrentRegion
        If CII.Value = LaValeur Then plg = plg & CII.Address() & ","
    Next CII
    If Len(plg) > 0 Then Range(Left(plg, Len(plg) - 1)).Select
End Sub
```

	A	B	C	D	E
1	Nom	Age	Taille		Age :
2	Dupond	20	1.70		20
3	Durand	25	1.75		
4	Bidon	22	1.80		
5	Carre	30	1.68		
6	Venter	20	1.82		
7	Menier	32	1.78		
8	Marloir	40	1.76		
9	Perrier	25	1.70		
10	Soltard	20	1.72		

Sélection de cellules vers le bas depuis la cellule active.

```
Sub SelectDown()
    Range(ActiveCell, ActiveCell.End(xlDown)).Select
End Sub
```

```
End Sub
```

Sélection de cellules vers le haut depuis la cellule active.

```
Sub SelectDown()  
Range(ActiveCell, ActiveCell.End(xlUp)).Select  
End Sub
```

Sélection de cellules vers la droite depuis la cellule active.

```
Sub SelectDown()  
Range(ActiveCell, ActiveCell.End(xlToLeft)).Select  
End Sub
```

Sélection de cellules vers la gauche depuis la cellule active.

```
Sub SelectDown()  
Range(ActiveCell, ActiveCell.End(xlToRight)).Select  
End Sub
```

Sélection de cellules dans la plage courante depuis la cellule active.

```
Sub SelectCurrentRegion()  
ActiveCell.CurrentRegion.Select  
End Sub
```

Sélection de cellules contigues dans la colonne de la cellule active.

```
Sub SelectActiveColumn()  
If IsEmpty(ActiveCell) Then Exit Sub  
' Ignore error si Activecell est dans la ligne 1  
On Error Resume Next  
If IsEmpty(ActiveCell.Offset(-1, 0)) Then Set TopCell = ActiveCell Else Set TopCell = ActiveCell.End(xlUp)  
If IsEmpty(ActiveCell.Offset(1, 0)) Then Set BottomCell = ActiveCell Else Set BottomCell = ActiveCell.End(xlDown)  
Range(TopCell, BottomCell).Select  
End Sub
```

Sélection de cellules contigues dans la ligne de la cellule active.

```
Sub SelectActiveRow()  
If IsEmpty(ActiveCell) Then Exit Sub  
' Ignore error si Activecell est dans la colonne A  
On Error Resume Next  
If IsEmpty(ActiveCell.Offset(-1, 0)) Then Set TopCell = ActiveCell Else Set TopCell = ActiveCell.End(xlToLeft)  
If IsEmpty(ActiveCell.Offset(1, 0)) Then Set BottomCell = ActiveCell Else Set BottomCell = ActiveCell.End(xlToRight)  
Range(TopCell, BottomCell).Select  
End Sub
```

Sélection d'une colonne entière par rapport à la cellule active.

```
Sub SelectEntireColumn()  
Selection.EntireColumn.Select  
End Sub End Sub
```

Sélection d'une ligne entière par rapport à la cellule active.

```
Sub SelectEntireRow()  
Selection.EntireRow.Select  
End Sub
```

Commandes, codes, fonctions, etc ... à utiliser dans vos applications

Page mise à jour le : 28/03/2001

- Adapter la largeur d'une colonne en fonction de son contenu.
`Columns("A").AutoFit`
- Afficher un message ordinaire.
`MsgBox("Vous êtes le bienvenu")`
- Afficher un message dans la barre de status.
`Application.StatusBar="Programme en cours d'exécution ..."`
- Copier les données issues d'un filtre automatique vers une autre feuille.
`Sheets("Feuil1").AutoFilter.Range.Copy Sheets("Feuil2").Range("A1")`
ou
`Sheets("Feuil1").Range("_FilterDatabase").Copy Sheets("Feuil2").Range("A1")`
- Définir l'imprimante par défaut.
`ActivePrinter="Canon LBP-465 sur LPT1"`
- Définir la ligne de titre.
`ActiveSheet.PageSetUp.PrintTitleRows="$1$1"`
- Délimiter du texte dans une cellule.
`ActiveCell.WrapText=True`
- Documenter une fonction personnalisée.
`Application.MacroOptions "InsereCheminClasseur", "Récupère le chemin du classeur."`
`InsereCheminClasseur` : Nom de la fonction
Récupère le chemin du classeur : Définition de la fonction
- Figurer le défilement de l'écran.
`Application.ScreenUpdating=False`
- Insérer un commentaire dans une cellule (non VBA).
`=SOMME(A10:A11)+N("Ceci est un commentaire")`
- Insérer la formule SOMME à une position variable.
`Range("A1").End(xlDown).Offset(1, 0).Value = "=SUM(A1:" & Range_ ("A1").End(xlDown).Address & ")"`
- Redéfinir la plage pour utiliser la combinaison de touches Ctrl+Fin.
Lorsque l'on supprime des lignes ou colonnes, Excel garde toujours la même référence de la dernière cellule utilisée bien qu'elle soit vide.
Pour remédier à cela, une ligne de code :
`ActiveSheet.UsedRange`
- Sélectionner la plage utilisée dans une feuille de calcul.
`ActiveSheet.UsedRange.Select`
- Sélectionner la plage en cours sans prendre en compte la ligne de titres.
`With Selection.CurrentRegion`
`Intersect(.Cells, .Offset(1)).Select`
`End With`
- Sélectionner la zone en cours.
Nota : La zone en cours est une plage limitée par toute combinaison de lignes et de colonnes vides.
`Range("A1").CurrentRegion.Select`
- Somme d'une plage variable (Non VBA).
Nota : La plage se situe dans la colonne A (A1 à Ax) et celle-ci possède un titre.
`=SOMME(DECALER(A2;NB(A:A)-1;;-NB(A:A)-1))`
- Trouver la première ligne vide.
Cette commande permet de trouver dans une plage, la première ligne vide.
`NoLgn = ActiveSheet.UsedRange.Row + ActiveSheet.UsedRange.Rows.Count`

Cette commande permet de trouver dans une colonne (A), la première ligne vide.
`premLiVide = Columns(1).Find("", [A65536], , , xlByRows, xlNext).Row`

Des programmes de toutes sortes

Page mise à jour le : 3/05/2002

- Récupérer le contenu d'une variable d'environnement (SET).
- Récupérer le nom de login (réseau).
- Récupérer une valeur stockée dans la base de registre.

● Récupérer le contenu d'une variable d'environnement (SET).

Ce programme permet de récupérer une variable d'environnement (voir DOS pour détail sur commande SET).

Dans cet exemple, la variable AU_LOCAL_DIR me retourne un chemin (différent suivant les utilisateurs) où je trouve un fichier à traiter avec Excel. Ce chemin est du type : AU_LOCAL_DIR="f:\renaud\audros\tmp"

Le code

Code de la procédure RechercheVariableEnvironnement

Sub RechercheVariableEnvironnement()

' Déclaration des variables.

Dim EnvString, Indx, Msg, PathLen

Indx = 1 *' Initialise l'index à 1*

Do

' Extrait la variable d'environnement (AU_LOCAL_DIR)

EnvString = **Environ** (Indx)

' Vérifie l'entrée AU_LOCAL_DIR

If **Left** (EnvString, 13) = "AU_LOCAL_DIR=" **Then**

' Extrait la longueur et met l'entrée dans une variable.

PathLen = **Len** (**Environ** ("AU_LOCAL_DIR"))

Msg = **Mid** (EnvString, 14 , PathLen)

Exit Do

Else

' Pas d'entrée AU_LOCAL_DIR, donc on incrémente.

Indx = Indx + 1

End If

Loop Until EnvString = ""

If PathLen > 0 **Then**

MsgBox Msg *' Affiche le message "f:\renaud\audros\tmp"*

Else

MsgBox " Il n'existe pas de variable " & "d'environnement AU_LOCAL_DIR."

End If

End Sub

● Récupérer le nom de login (réseau).

Cette fonction permet de récupérer le nom de login (et non pas le nom d'utilisateur déclaré dans Excel).

Le code

Code de la fonction GetUserName

```
Declare Function WNetGetUser Lib "mpr.dll" Alias "WNetGetUserA" ( ByVal IpName As String , _
    ByVal IpUserName As String , IpLength As Long ) As Long
Const NoError = 0
Function GetUserName ()
    Const IpLength As Integer = 255
    Dim status As Integer
    Dim IpName, IpUserName As String
    IpUserName = Space$ (IpLength + 1 )
    status = WNetGetUser(IpName, IpUserName, IpLength)
    If status = NoError Then
        IpUserName = Left$ (IpUserName, InStr (IpUserName, Chr ( 0 )) - 1 )
    Else
        MsgBox "Impossible d'obtenir le login."
    End If
    GetUserName = IpUserName
End Function
```

Code de la procédure AfficheLogin

```
Sub AfficheLogin()
    MsgBox GetUserName
End Sub
```

● Récupérer une valeur stockée dans la base de registre.

Cette procédure permet de récupérer une valeur stockée dans la base de registre.
Cette valeur est stockée dans la clé :

HKEY_CURRENT_USER\Software\VB and VBA Program Settings\

J'utilise personnellement cette procédure pour incrémenter un numéro de facture.

Lecture/Ecriture dans la base de registre

```
Private Sub LectureRegistry()
    ' Lecture des infos dans la base de registre
    Call InfoRegistry
    ' Récupération du N° de facture
    NumFac = GetSetting (appname:= "Facture Grand Crès" , section:= "NumFacture" , key:= "Numéro" )
    ' Incrémentation du N° de facture
    NumFac = NumFac + 1
    ' Ecriture de la nouvelle valeur NumFac dans la base de registre
    SaveSetting appname:= "Facture Grand Crès" , section:= "NumFacture" , key:= "Numéro" ,
    setting:=NumFac
End Sub
```

```
Private Sub InfoRegistry()
    ' Lecture de la valeur dans la base de registre
    NumFac = GetSetting (appname:= "Facture Grand Crès" , section:= "NumFacture" , key:= "Numéro" )
    ' Si aucune valeur n'est présente dans la base de registre ...
    If NumFac = "" Then
        Numdepart = InputBox( "Indiquer le N° de votre dernière facture manuelle." , "Initialisation du
numéro de facture" )
        ' Si on clique sur le bouton Annuler, fin de procédure
        If Numdepart = "" Then End
        ' Sinon, on inscrit le numéro saisi dans la boîte de dialogue
        SaveSetting appname:= "Facture Grand Crès" , section:= "NumFacture" , key:= "Numéro" ,
        setting:=Numdepart
    End If
End Sub
```



```
End If
End Sub
```

Effacement de la clé dans la base de registre

```
Sub EffaceRegistry()
' Effacement de la clé dans la base de registre
On Error Resume Next
DeleteSetting "Facture Grand Crès"
End Sub
```

Echanges de données entre Excel et Word

Page mise à jour le : 28/03/2001

Vous trouverez dans cette rubrique des exemples de programmes permettant d'envoyer des données Excel vers Word ou récupérer des données Word dans Excel.

Ces exemples fonctionnent avec Word et Excel 2000. Ils n'ont pas été testés sous la version 97.

Pour que ces exemples fonctionnent, vous devrez créer une référence de bibliothèques d'objets de Word.

Dans Excel, choisissez la commande **Références** du menu **Outils** de Visual Basic Editor et cochez **Microsoft Word 9.0 Object Library**.

- Récupérer les données d'un document Word et les copier dans Excel.
 - Copier des données Excel dans un document Word (format tableau).
 - Copier des données Excel dans un nouveau document Word (format tabulation).
 - Copier des données Excel dans un tableau Word afin de le compléter.
-

Récupérer les données d'un document Word et les copier dans Excel

Ce programme permet d'aller chercher des données dans un document Word, de les copier, de les coller dans Excel et de refermer Word.

```
Sub DonnéesWordVersExcel()
Dim DocWord As Word.Document
Dim AppWord As Word.Application
Set AppWord = New Word.Application
AppWord.ShowMe
AppWord.Visible = True
'Ouvre le document Word (Fichier.doc) et effectue une copie des données
```

```

Set DocWord = AppWord.Documents.Open( "c:\Excel\Fichier.doc" , ReadOnly:= True )
With AppWord
    .Selection.WholeStory
    .Selection.Copy
End With
' Copie des données dans Excel
ThisWorkbook.Worksheets( "Feuil1" ).Paste
' Fermeture de Word
AppWord.Application.Quit
Application.CutCopyMode = False
End Sub

```

Les données Word

Dépôt	Dates
500	08/12/00
400	20/09/00
200	10/10/00
100	15/10/00
200	18/10/00

La copie dans Excel

	A	B
1	Dépot	Dates
2	500	12/08/00
3	400	20/09/00
4	200	10/10/00
5	100	15/10/00
6	200	18/10/00

Copier des données Excel dans un document Word (format tableau)

Ce programme permet de copier des données d'Excel, d'ouvrir un document Word et d'y coller les données sous forme de tableau.

Sub EnvoyerDonneesExcelVersWord()

```

Dim DocWord As Word.Document
Dim AppWord As Word.Application
Set AppWord = New Word.Application
Application.DisplayAlerts = True
AppWord.ShowMe
AppWord.Visible = True
'Ouvre le document Word
Set DocWord = AppWord.Documents.Open( "c:\excel\Fichier.doc" , ReadOnly:= False )
' Copie les données Excel
ThisWorkbook.Worksheets( "Feuil1" ).Range( "A1:B6" ).Copy
' Colle les données dans Word
DocWord.Range.PasteSpecial
Application.CutCopyMode = False
DocWord.Application.ActiveDocument.Save
AppWord.Application.Quit
End Sub

```

Les données Excel

	A	B
1	Dépot	Dates
2	500	12/08/00
3	400	20/09/00
4	200	10/10/00
5	100	15/10/00
6	200	18/10/00

La copie dans Word

Dépôt	Dates
500	08/12/00
400	20/09/00
200	10/10/00
100	15/10/00
200	18/10/00

Copier des données Excel dans un document Word (format tabulations)

Ce programme permet de copier des données d'Excel, d'ouvrir un document Word et d'y coller les données. Les données étant séparées par des tabulations.

Sub CopierDonneesExcelVersWord()

```

Range( "A1:B6" ).Copy
Set WW = CreateObject( "word.application" )
WW.Visible = True

```

```

WW.Documents.Add
WW.Selection.PasteSpecial Link:= False , DataType:=wdPasteText, Placement:= _
wdInLine, DisplayAsIcon:= False
End Sub

```

Les données Excel

	A	B
1	Dépot	Dates
2	500	12/08/00
3	400	20/09/00
4	200	10/10/00
5	100	15/10/00
6	200	18/10/00

La copie dans Word

```

Dépot→ Dates¶
500 → 12/08/00¶
400 → 20/09/00¶
200 → 10/10/00¶
100 → 15/10/00¶
200 → 18/10/00¶

```

Copier des données Excel dans un tableau Word afin de le compléter

Ce programme permet de copier des données d'Excel, d'ouvrir un document Word existant, d'ajouter une ligne au tableau existant dans Word et de compléter chaque cellule de la ligne créée.

Sub RemplirTableauWordDepuisDonnéesExcel()

```

Dim DocWord As Word.Document
Dim AppWord As Word.Application
Set AppWord = New Word.Application
Application.DisplayAlerts = True
AppWord.ShowMe
AppWord.Visible = True
'Ouvre le document Word
Set DocWord = AppWord.Documents.Open( "c:\excel\Fichier.doc" , _
ReadOnly:= False )
' Copie les données Excel
Contrats_ISBN = Range( "A2" ).Value
Contrats_Titre = Range( "B2" ).Value
Contrats_Nom = Range( "C2" ).Value
' Colle les données dans Word
DocWord.Tables( 1 ).Rows.Add
Derligne = DocWord.Tables( 1 ).Rows.Count
With DocWord.Tables( 1 )
    .Cell( Derligne, 1 ).Range.InsertAfter Contrats_ISBN
    .Cell( Derligne, 2 ).Range.InsertAfter Contrats_Titre
    .Cell( Derligne, 3 ).Range.InsertAfter Contrats_Nom
End With
DocWord.Application.ActiveDocument.Save
AppWord.Application.Quit
End Sub

```

Le tableau Word avant mise à jour

ISBN	Titre	Nom
0548-9*	Les animaux du monde*	Adam*
0653-6*	Les ours du Tibet*	Bidault*

Les données Excel à copier dans le tableau Word

	A	B	C
1	ISBN	Titre	Nom
2	0548-20	Les plantes décoratives	Renaud

Le tableau Word après mise à jour

ISBN	Titre	Nom
0548-9	Les animaux du monde	Adam
0653-6	Les ours du Tibet	Bidault
0548-20	Les plantes décoratives	Renaud

Echanges de données entre Excel et Access

Page mise à jour le : 1/08/2002

Vous trouverez dans cette rubrique des exemples de programmes permettant d'envoyer des données Excel vers Access, de récupérer des données d'Access dans Excel ou de mettre à jour des enregistrements dans Access depuis Excel.

Ces exemples fonctionnent avec Access et Excel 2000. Ils n'ont pas été testé sous la version 97.

Pour que ces exemples fonctionnent, vous devrez créer une référence de bibliothèques d'objets d'accès aux données de Microsoft (DAO). Dans Excel, choisissez la commande **Références** du menu **Outils** de Visual Basic Editor et cochez **Microsoft DAO 3.6 Object Library**.

Un lien intéressant (mes sources) traitant de [External Data Access](#) with DAO (Data Access Objects) and ADO (ActiveX Data Objects)

- Envoyer les données d'une feuille Excel vers une table Access.
- Récupérer les données d'une table Access dans une feuille de calcul Excel.
- Récupérer les données d'une table Access dans une feuille de calcul Excel via une requête .
- Mettre à jour un enregistrement dans Access depuis Excel.

Envoyer les données d'une feuille Excel vers une table Access

Ce programme permet de copier les données d'une feuille de calcul Excel vers une table Access.

Définition des éléments utilisés :

Nom de la base de données --> **Commandes.mdb**
 Table recevant les données Excel --> **Factures**
 Champs de la table Factures --> **NoFacture, Client, Date et Solde**
 Feuille de calcul Excel d'où l'on exporte les données --> **DAOSheet**

Sub WritingWorksheetData_DAO()

Dim Plage As Range
 Dim Array1 As Variant
 Dim x As Variant

```

Dim Db1 As Database
Dim Rs1 As Recordset
' Ouverture de la base de données Commandes.mdb
Set Db1 = DBEngine.OpenDatabase(ThisWorkbook.Path & "\Commandes.mdb" )
' Ouverture de la table Factures
' Un objet Recordset représente les enregistrements d'une table
Set Rs1 = Db1.OpenRecordset( "Factures" , dbOpenDynaset)
' Détermination de la taille de la plage à envoyer vers Access
Set Plage = Worksheets( "DAOSheet" ).Range( "A1" ).CurrentRegion.Offset( 1 , 0 )
Set Plage = Plage.Resize(Plage.Rows.Count - 1 , Plage.Columns.Count)
Plage.Select
' Lecture de la plage pour renvoyer une valeur contenant un tableau
Array1 = Plage.Value
' Ecriture des données depuis Excel vers les enregistrement de la table Factures
For x = 1 To UBound( Array1, 1 )
    With Rs1
        .AddNew
        .Fields( "NoFacture" ) = Array1(x, 1 )
        .Fields( "Client" ) = Array1(x, 2 )
        .Fields( "Date" ) = Array1(x, 3 )
        .Fields( "Solde" ) = Array1(x, 4 )
        .Update
    End With
Next
' Fermeture de la base Commandes.mdb
Db1.Close
' Effacement des données copiées vers la base (sauf les titres)
With Selection.CurrentRegion
    Intersect(.Cells, .Offset( 1 )).Select
End With
Selection.ClearContents
End Sub

```

Les données Excel

	A	B	C	D
1	N° Facture	Client	Date	Réglé
2	2002-07/41	ALBERT REICHMUTH	24/07/02	
3	2002-07/42	ALLEE DES VINS	27/07/02	
4	2002-07/43	ALLWINES HECK ET WOLFS	27/07/02	

Les données copiées dans la table Factures

Factures : Table				
	NoFacture	Client	Date	Solde
+	2002-07/41	ALBERT REICHMUTH	24/07/02	
+	2002-07/42	ALLEE DES VINS	27/07/02	
+	2002-07/43	ALLWINES HECK ET WOLFS	27/07/02	
*				

Récupérer les données d'une table Access dans une feuille de calcul Excel

Ce programme permet de récupérer les données d'une table Access et de les copier dans une feuille Excel.

Définition des éléments utilisés :

Nom de la base de données --> **Commandes.mdb**

Table possédant les données à récupérer dans Excel --> **Factures**

Champs de la table Factures --> **NoFacture, Client, Date et Solde**

Feuille de calcul Excel recevant les données de la table; --> **DonnéesDataBase**

Sub CopyFromRecordset_DAO()

Dim Db1 As Database

Dim Rs1 As Recordset

' Ouverture de la base de données

Set Db1 = DBEngine.OpenDatabase(ThisWorkbook.Path & "\Commandes.mdb")

```

' Ouverture de la table Factures
' Un objet Recordset représente les enregistrements d'une table
Set Rs1 = Db1.OpenRecordset( Name := "Factures" , Type :=dbOpenDynaset)
' Effacement des données existantes dans la WorkSheet (sauf les titres)
' et copie des enregistrements
With Worksheets( "DonnéesDataBase" ).Range( "A2" )
    With Selection.CurrentRegion
        Intersect(.Cells, .Offset( 1 )).Select
    End With
    Selection.ClearContents
    .CopyFromRecordset Rs1
End With
'Fermeture de la Base de données
Db1.Close
End Sub

```

Les données de la table Factures

Factures : Table				
	NoFacture	Client	Date	Solde
+	2002-07/41	ALBERT REICHMUTH	24/07/02	
+	2002-07/42	ALLEE DES VINS	27/07/02	
+	2002-07/43	ALLWINES HECK ET WOLFS	27/07/02	
*				

Les données récupérées dans la feuille Excel

	A	B	C	D
1	N° Facture	Client	Date	Réglé
2	2002-07/41	ALBERT REICHMUTH	24/07/02	
3	2002-07/42	ALLEE DES VINS	27/07/02	
4	2002-07/43	ALLWINES HECK ET WOLFS	27/07/02	

Récupérer les données d'une table Access dans une feuille de calcul Excel

via une requête

Ce programme permet de récupérer les données d'une table Access par l'intermédiaire d'une requête et de les copier dans une feuille Excel.

Définition des éléments utilisés :

Nom de la base de données --> **Commandes.mdb**

Table possédant les données à récupérer dans Excel --> **Factures**

Champs de la table Factures --> **NoFacture, Client, Date et Solde**

Nom de la requête --> **Factures pour un client**

Code SQL --> **SELECT Factures.NoFacture, Factures.Client, Factures.Date, Factures.Solde FROM Factures WHERE (((Factures.Client) Like "ALLEE DES VINS"));**

Feuille de calcul Excel recevant les données de la table; --> **DonnéesDataBase**

Sub CopyFromRecordset_DAO()

Dim Db1 As Database

Dim Rs1 As Recordset

' Ouverture de la base de données

Set Db1 = DBEngine.OpenDatabase(ThisWorkbook.Path & "\Commandes.mdb")

' Ouverture de la table Factures

' Un objet Recordset représente les enregistrements d'une table

Set Rs1 = Db1.OpenRecordset(Name := "Factures pour un client" , Type :=dbOpenSnapshot)

' Effacement des données existantes dans la WorkSheet (sauf les titres)

' et copie des enregistrements

With Worksheets("DonnéesDataBase").Range("A2")

With Selection.CurrentRegion

Intersect(.Cells, .Offset(1)).Select

End With

Selection.ClearContents

.CopyFromRecordset Rs1

End With

'Fermeture de la Base de données

Db1.Close
End Sub

Les données de la table Factures

Factures : Table				
	Nofacture	Client	Date	Solde
+	2002-07/41	ALBERT REICHMUTH	24/07/02	
+	2002-07/42	ALLEE DES VINS	27/07/02	
+	2002-07/43	ALLWINES HECK ET WOLFS	27/07/02	
+	2002-07/44	ALLWINES HECK ET WOLFS	28/07/02	
+	2002-07/45	ALLEE DES VINS	29/07/02	
+	2002-07/46	ALLWINES HECK ET WOLFS	29/07/02	
+	2002-07/47	ALLWINES HECK ET WOLFS	30/07/02	
+	2002-07/48	ALLEE DES VINS	31/07/02	

Les données récupérées dans la feuille Excel

	A	B	C	D
1	N° Facture	Client	Date	Réglé
2	2002-07/42	ALLEE DES VINS	27/07/02	
3	2002-07/45	ALLEE DES VINS	29/07/02	
4	2002-07/48	ALLEE DES VINS	31/07/02	

La requête Factures pour un client en mode Création

Champ :	Nofacture	Client	Date	Solde
Table :	Factures	Factures	Factures	Factures
Tri :				
Afficher :	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Critères :		Comme "ALLEE DES VINS"		
Ou :				

Mettre à jour un enregistrement dans Access depuis Excel

Ce programme permet de mettre à jour dans la table Factures, le champ (Solde) correspondant à l'enregistrement dont le N° de facture est égal à la variable N°Facture.

Définition des éléments utilisés :

Nom de la base de données --> **Commandes.mdb**
Table possédant les données à mettre à jour; --> **Factures**
Champ de la table Factures à mettre à jour --> **Solde**
Recherche à partir du --> **NoFacture**

Note : Afin de proposer les 2 possibilités de syntaxe, l'exemple se propose de tester quel est le type de la variable Nofacture car la syntaxe est différente suivant que la variable est un nombre (vbInteger) ou une chaîne (vbString).











Sub UpdateDataAccess_DAO()

```
Dim Db1 As Database
TypeVariable = VarType (NumFacture)
' Ouverture de la base de données
Set Db1 = DBEngine.OpenDatabase(ThisWorkbook.Path & "\Commandes.mdb" )
' Mise à jour de l'enregistrement
If TypeVariable = 2 Then ' La variable Numfacture est de type vbInteger (2)
    Db1.Execute "UPDATE Factures SET Solde= 'Oui' WHERE NoFacture = " & NumFacture
Else ' La variable Numfacture est de type vbString (8)
    Db1.Execute "UPDATE Factures SET Solde = 'Oui' WHERE NoFacture = ' " & NumFacture & " ' "
End If
Db1.Close
End Sub
```

Téléchargement

 Page mise à jour le : 23/04/2001

Vous trouverez sur cette page, des exemples présentés dans les différentes rubriques et disponibles en téléchargement.

Exemples présents sur le site	Fichier Archive
Boite de dialogue avec case à cocher	
Boite de dialogue avec boutons d'options	
Boite de dialogue avec liste déroulante	
Boite de dialogue avec liste déroulante modifiable	
Boite de dialogue avec liste déroulante modifiable simplifiée	
Boite de dialogue avec liste déroulante à sélection multiple	
Boite de dialogue avec liste déroulante à deux colonnes	
Boite de dialogue avec liste déroulante combinée	
Le contrôle Calendar (Calendrier)	
Le contrôle ProgressBar, version 6.0	
Supprimer les doublons dans une colonne	