

Option Explicit 'impose que tous les types soient déclarés (pour vérifier)

```
Public Function Remplace(theString As String, oldStr As String, newStr As String)
As String
```

```
    'remplace dans "theString" toutes les occurrences de "oldStr" par "newStr"
    '(pour compatibilité : la fonction "Replace" n'existe pas dans les anciennes
versions de vba pour Excel)
```

```
    Dim position As Integer
    Dim myString As String
    myString = theString
    position = InStr(myString, oldStr)
    While position > 0
        myString = Left(myString, position - 1) & newStr & Right(myString,
Len(myString) - (position - 1) - Len(oldStr))
        position = InStr(position + Len(newStr), myString, oldStr)
    Wend
    Remplace = myString
End Function
```

```
Private Sub CheckBox1_Click()
    RefEdit4.Visible = CheckBox1.Value
End Sub
```

```
Private Sub restaurePar(restaure As Boolean)
    Dim iarg As Integer
    If restaure Then 'appelé en cas d'annulation
        For iarg = 1 To MG_nPar
            Range(MG_adrPar(iarg)).Value = MG_Par(iarg) 'on restaure les
paramètres normalement stockés au début
        Next iarg
        Calculate 'puis il faut forcer (au cas où...) le recalcul
        MG_minimisation = False 'on ne restaure pas deux fois
        UserFormMinimi.TextBox4.Value = "" 'purge du résultat
        CommandButton3.Visible = False 'après une annulation, il n'y a plus rien à
accepter
    End If
End Sub
```

```
Private Sub CommandButton1_Click()
    If getAdrFCN Then
        If getAdrPar Then
            If getAdrDPar Then
                If getAdrParN Then
                    If getOptions Then 'après avoir tout vérifié, on peut appeler minimi
                        MG_minimisation = True
                        UserFormMinimi.TextBox4.Value = minimi(MG_nPar, MG_Par(),
MG_ParN(), MG_DPar(), MG_step, MG_epsilon, MG_impr, MG_err)
```

```
UserFormMinimi.MultiPage1.Value = 3 'pour afficher la page de
résultats
```

```
CommandButton3.Visible = True 'après une minimisation, il faut
demander à l'utilisateur s'il accepte le résultat
```

```
End If
```

```
End If
```

```
End If
```

```
End If
```

```
End Sub
```

```
Private Function getAdrFCN() As Boolean
```

```
Dim adrFCN As String
```

```
Dim Nadr As Integer
```

```
Dim c As Range
```

```
Dim bidon As String
```

```
adrFCN = UserFormMinimi.RefEdit1.Value
```

```
adrFCN = Remplace(adrFCN, ";", ",") 'la boîte de sélection met des ";" là où il
faut des ","
```

```
On Error GoTo ERROR_Range 'on prépare un test en cas d'adresse invalide
```

```
adrFCN = Range(adrFCN).Address 'le test s'effectue ici
```

```
Nadr = 0
```

```
For Each c In Range(adrFCN) 'on teste l'unicité de l'adresse
```

```
Nadr = Nadr + 1
```

```
If Nadr > 1 Then
```

```
bidon = MsgBox("la cellule à minimiser ne peut être multiple :)" & vbCrLf &
adrFCN, vbOKOnly)
```

```
getAdrFCN = False
```

```
Exit Function
```

```
End If
```

```
Next c
```

```
MG_adrFCN = adrFCN 'après vérification, on enregistre
```

```
getAdrFCN = True
```

```
Exit Function
```

```
ERROR_Range:
```

```
bidon = MsgBox("la cellule à minimiser est indéfinie :)" & vbCrLf & adrFCN,
vbOKOnly)
```

```
getAdrFCN = False
```

```
End Function
```

```
Private Function getAdrPar() As Boolean
```

```
Dim adrPar As String
```

```
Dim Nadr As Integer
```

```
Dim c As Range
```

```
Dim bidon As String
```

```
Dim ii As Integer
```

```
adrPar = UserFormMinimi.RefEdit2.Value
```

```
    adrPar = Remplace(adrPar, ";", ",") 'la boite de sélection met des ";" là où il faut
des ","
    On Error GoTo ERROR_Range 'on prépare un test en cas d'adresse invalide
    adrPar = Range(adrPar).Address 'le test s'effectue ici
    Nadr = 0
    For Each c In Range(adrPar).Cells 'on teste si le nombre de valeurs est
acceptable
        Nadr = Nadr + 1
        If Nadr > 30 Then
            bidon = MsgBox("les paramètres à ajuster sont limités à 30 :" & vbCrLf &
adrPar, vbOKOnly)
            getAdrPar = False
            Exit Function
        End If
        If Not IsNumeric(c.Value) Then
            bidon = MsgBox("les valeurs initiales des paramètres doivent être
numériques :" & vbCrLf & c.Value, vbOKOnly)
            getAdrPar = False
            Exit Function
        End If
    Next c
    For ii = 1 To 30 'on réinitialise les adresses des valeurs et les valeurs
        MG_adrPar(ii) = ""
        MG_Par(ii) = 0#
    Next ii
    Nadr = 0
    For Each c In Range(adrPar) 'après vérification, on enregistre
        Nadr = Nadr + 1
        MG_adrPar(Nadr) = c.Address
        MG_Par(Nadr) = c.Value
    Next c
    MG_nPar = Nadr
    getAdrPar = True
    Exit Function
ERROR_Range:
    bidon = MsgBox("les paramètres à ajuster sont indéfinis :" & vbCrLf & adrPar,
vbOKOnly)
    getAdrPar = False
End Function

Private Function getAdrDPar()
    Dim adrDPar As String
    Dim Nadr As Integer
    Dim c As Range
    Dim bidon As String
    Dim ii As Integer
    adrDPar = UserFormMinimi.RefEdit3.Value
```

```

    adrDPar = Remplace(adrDPar, ";", ",") 'la boite de sélection met des ";" là où il
faut des ","
    On Error GoTo ERROR_Range 'on prépare un test en cas d'adresse invalide
    adrDPar = Range(adrDPar).Address 'le test s'effectue ici
    Nadr = 0
    For Each c In Range(adrDPar) 'on teste si le nombre de valeurs est acceptable
        Nadr = Nadr + 1
        If Nadr > MG_nPar Then
            bidon = MsgBox("il n'y a que " & MG_nPar & " paramètres à ajuster ;" &
vbCrLf & "il ne peut y avoir plus de pas initiaux : " & vbCrLf & adrDPar, vbOKOnly)
            getAdrDPar = False
            Exit Function
        End If
        If Not IsNumeric(c.Value) Then
            bidon = MsgBox("les pas initiaux doivent être numériques :" & vbCrLf &
c.Value, vbOKOnly)
            getAdrDPar = False
            Exit Function
        End If
    Next c
    If Nadr < MG_nPar Then
        bidon = MsgBox("il y a " & MG_nPar & " paramètres à ajuster ;" & vbCrLf & "il
ne peut y avoir seulement " & Nadr & " pas initiaux : " & vbCrLf & adrDPar,
vbOKOnly)
        getAdrDPar = False
        Exit Function
    End If
    For ii = 1 To 30 'on initialise les adresses des valeurs et les valeurs
        MG_adrDPar(ii) = ""
        MG_DPar(ii) = 0#
    Next ii
    Nadr = 0
    For Each c In Range(adrDPar) 'après vérification, on enregistre
        Nadr = Nadr + 1
        MG_adrDPar(Nadr) = c.Address
        MG_DPar(Nadr) = c.Value
    Next c
    getAdrDPar = True
    Exit Function
ERROR_Range:
    bidon = MsgBox("les pas initiaux d'ajustement sont indéfinis :" & vbCrLf &
adrDPar, vbOKOnly)
    getAdrDPar = False
End Function

Private Function getAdrParN()
    Dim adrParN As String

```

```

Dim Nadr As Integer
Dim c As Range
Dim bidon As String
Dim ii As Integer
If CheckBox1.Value Then 'l'utilisateur propose des noms
    adrParN = UserFormMinimi.RefEdit4.Value
    adrParN = Replace(adrParN, ";", ",") 'la boîte de sélection met des ";" là où il
faut des ","
    On Error GoTo ERROR_Range 'on prépare un test en cas d'adresse invalide
    adrParN = Range(adrParN).Address 'le test s'effectue ici
    Nadr = 0
    For Each c In Range(adrParN) 'on teste si le nombre de valeurs est
acceptable
        Nadr = Nadr + 1
        If Nadr > MG_nPar Then
            bidon = MsgBox("nommer les paramètres n'est pas obligatoire..." &
vbCrLf & "mais il n'y a que " & MG_nPar & " paramètres à ajuster ;" & vbCrLf & "il
ne peut y avoir plus de noms : " & vbCrLf & adrParN, vbOKOnly)
            getAdrParN = False
            Exit Function
        End If
        If (TypeName(c.Value) <> "String") Or Not (Left(c.Value, 1) Like "[a-z;A-Z]")
Then
            bidon = MsgBox("nommer les paramètres n'est pas obligatoire..." &
vbCrLf & "mais les noms doivent être de type String" & vbCrLf & "et commencer
par une lettre :" & vbCrLf & c.Value, vbOKOnly)
            getAdrParN = False
            Exit Function
        End If
    Next c
    If Nadr < MG_nPar Then
        bidon = MsgBox("préciser les noms des paramètres n'est pas obligatoire..."
& vbCrLf & "mais il y a " & MG_nPar & " paramètres à ajuster ;" & vbCrLf & "il ne
peut y avoir seulement " & Nadr & " noms : " & vbCrLf & adrParN, vbOKOnly)
        getAdrParN = False
        Exit Function
    End If
    For ii = 1 To 30 'on initialise les adresses des valeurs
        MG_adrParN(ii) = ""
        MG_ParN(ii) = ""
    Next ii
    Nadr = 0
    For Each c In Range(adrParN) 'après vérification, on enregistre
        Nadr = Nadr + 1
        MG_adrParN(Nadr) = c.Address
        MG_ParN(Nadr) = c.Value
    Next c

```

```
    getAdrParN = True
    Exit Function
Else
    For ii = 1 To MG_nPar 'faute de noms proposés, on impose les noms par
défaut
        MG_adrParN(ii) = ""
        MG_ParN(ii) = "p" & Str(ii)
    Next ii
    For ii = MG_nPar + 1 To 30
        MG_adrParN(ii) = ""
        MG_ParN(ii) = ""
    Next ii
    getAdrParN = True
    Exit Function
End If
ERROR_Range:
    bidon = MsgBox("les noms des paramètres sont indéfinis :)" & vbCrLf & adrParN,
vbOKOnly)
    getAdrParN = False
End Function

Private Function getOptions()
    getOptions = False
    If getStep Then
        If getEpsi Then
            If getImpr Then
                MG_err = UserFormMinimi.CheckBox2.Value
                getOptions = True
            End If
        End If
    End If
End Function

Private Function getStep()
    Dim theText As String
    Dim bidon As String
    theText = UserFormMinimi.TextBox1.Value
    If Not IsNumeric(theText) Then
        bidon = MsgBox("le pas global doit être numérique :)" & vbCrLf & theText,
vbOKOnly)
        getStep = False
        Exit Function
    End If
    MG_step = Cdbl(theText)
    getStep = True
End Function
```

```
Private Function getEpsi()  
    Dim theText As String  
    Dim bidon As String  
    theText = UserFormMinimi.TextBox2.Value  
    If Not IsNumeric(theText) Then  
        bidon = MsgBox("la sensibilité doit être numérique :!" & vbCrLf & theText,  
vbOKOnly)  
        getEpsi = False  
        Exit Function  
    End If  
    MG_epsi = Abs(CDbl(theText))  
    getEpsi = True  
End Function
```

```
Private Function getImpr()  
    Dim theText As String  
    Dim bidon As String  
    theText = UserFormMinimi.TextBox3.Value  
    If Not IsNumeric(theText) Then  
        bidon = MsgBox("l'intervalle des bilans doit être numérique :!" & vbCrLf &  
theText, vbOKOnly)  
        getImpr = False  
        Exit Function  
    End If  
    MG_impr = Abs(CLng(theText))  
    getImpr = True  
End Function
```

```
Private Sub CommandButton2_Click()  
    'fermeture par le bouton d'annulation  
    Call restaurePar(MG_minimisation) 'on ne restaure que si une minimisation  
vient d'être faite  
    UserFormMinimi.MultiPage1.Value = 1 'Excel 2007 et 2011 se plantent en  
quittant le dialogue avec l'index 0 si on utilise des RefEdit...  
    UserFormMinimi.Hide 'sinon simple fermeture  
End Sub
```

```
Private Sub CommandButton3_Click()  
    'fermeture par le bouton d'acceptation  
    UserFormMinimi.MultiPage1.Value = 1 'Excel 2007 et 2011 se plantent en  
quittant le dialogue avec l'index 0 si on utilise des RefEdit...  
    UserFormMinimi.Hide 'sinon simple fermeture  
End Sub
```

```
Private Sub Userform_QueryClose(Cancel As Integer, CloseMode As Integer)  
    'fermeture interdite par la barre de titre (il faut décider le choix)  
    If CloseMode = vbFormControlMenu Then Cancel = True
```

End Sub