

```
Rem Attribute VBA_ModuleType=VBFormModule
Option VBASupport 1
Option Explicit 'impose que tous les types soient déclarés (pour vérifier)

'variables temporaires utilisées par la fenêtre d'appel de Minimi
Public MG_UserForm As Object 'pour le dialogue d'appel
Public MG_nPage As Integer 'pour les pages du dialogue d'appel
Public MG_loaded As Boolean 'normalement initialisé à "False"
Public MG_TadrFCN As String 'contenus temporaires des contrôles du dialogue...
Public MG_TadrPar As String
Public MG_TadrDPar As String
Public MG_TParN As Integer 'non Boolean car une CheckBox pourrait avoir trois états
Public MG_TadrParN As String
Public MG_Tstep As String 'on utilise String pour les nombres car le format n'est pas encore testé
Public MG_Tepsi As String
Public MG_Timpr As String
Public MG_Terr As Integer 'non Boolean car une CheckBox pourrait avoir trois états
Public MG_Tresult As String 'il faut aussi mémoriser le résultat si on n'annule pas

Sub Minimiseur()
    'macro pour appeler le dialogue de lancement du minimiseur
    Dim MinimiDialog As Object
    Dim myCtrl As Object

    DialogLibraries.LoadLibrary("VBAProject")
    MinimiDialog = DialogLibraries.VBAProject.UserFormMinimi
    MG_UserForm = CreateUnoDialog(MinimiDialog)
    MG_nPage = 1
    If Not MG_loaded Then
        'ainsi, pas de chargement à l'initialisation
        MG_loaded = "True"
    Else
        'rechargement des valeurs temporaires
        myCtrl = MG_UserForm.getControl("TextField1")
        myCtrl.Text = MG_TadrFCN
        myCtrl = MG_UserForm.getControl("TextField2")
        myCtrl.Text = MG_TadrPar
        myCtrl = MG_UserForm.getControl("TextField3")
        myCtrl.Text = MG_TadrDPar
        myCtrl = MG_UserForm.getControl("CheckBox1")
        myCtrl.State = MG_TParN 'Booléen géré par un entier
        myCtrl = MG_UserForm.getControl("TextField4")
        myCtrl.Text = MG_TadrParN
        myCtrl.Model.enableVisible = MG_TParN
    End If
End Sub
```

```

    myCtrl = MG_UserForm.getControl("CommandButton7")
    myCtrl.Model.enableVisible = MG_TParN
    myCtrl = MG_UserForm.getControl("TextField5")
    myCtrl.Text = MG_Tstep
    myCtrl = MG_UserForm.getControl("TextField6")
    myCtrl.Text = MG_Tepsi
    myCtrl = MG_UserForm.getControl("TextField7")
    myCtrl.Text = MG_Timpr
    myCtrl = MG_UserForm.getControl("CheckBox2")
    myCtrl.State = MG_Terr 'Booléen géré par un entier
    myCtrl = MG_UserForm.getControl("TextField9")
myCtrl.Text = MG_Tresult
IF MG_minimisation Then 'utile si on n'a pas annulé
    myCtrl = MG_UserForm.getControl("CommandButton2")
    myCtrl.Label = "annuler"
    myCtrl = MG_UserForm.getControl("CommandButton3")
myCtrl.Model.enableVisible = MG_minimisation
    End If
End If
MG_UserForm.Execute()
End Sub

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()
    Dim CBx1 As Object
    Dim TF4 As Object
    Dim CBn7 As Object
    CBx1 = MG_UserForm.getControl("CheckBox1")
    TF4 = MG_UserForm.getControl("TextField4")
    If CBx1.State = 0 Then
        TF4.Text = "" 'on vide si ce n'est pas censé être rempli
    
```

```

End If
TF4.Model.enableVisible = CBx1.State
'TF4.Visible = CBx1.State ne fonctionne pas si TF4.Visible est initialisé à
False
'car l'initialisation impose enableVisible (il y a DEUX "interrupteurs" en série)
CBn7 = MG_UserForm.getControl("CommandButton7")
CBn7.Model.enableVisible = CBx1.State 'idem...
End Sub

```

```

Private Sub restaurePar(restaure As Boolean)
Dim iarg As Integer
Dim myCtrl As Object
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
Call CommandButton8_Click() 'pour afficher la page d'accueil
myCtrl = MG_UserForm.getControl("TextField9")
myCtrl.Text = "" 'purge du résultat
myCtrl = MG_UserForm.getControl("CommandButton2")
myCtrl.Label = "fermer" 'après une annulation, il n'y a plus rien à annuler
myCtrl = MG_UserForm.getControl("CommandButton3")
myCtrl.Model.enableVisible = False 'après une annulation, il n'y a plus rien à
accepter
End If
End Sub

```

```

Private Sub CommandButton1_Click()
Dim myCtrl As Object
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
myCtrl = MG_UserForm.getControl("TextField9")
myCtrl.Text = minimi(MG_nPar, MG_Par(),
MG_ParN(), MG_DPar(), MG_step, MG_epsilon, MG_impr, MG_err)
Call CommandButton11_Click() 'pour afficher la
page de résultats
myCtrl =
MG_UserForm.getControl("CommandButton2")
myCtrl.Label = "annuler" 'après une minimisation, il faut demander
à l'utilisateur s'il souhaite annuler

```

```

                myCtrl =
MG_UserForm.getControl("CommandButton3")
                myCtrl.Model.enableVisible = 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 myCtrl As Object
    Dim adrFCN As String
    Dim Nadr As Integer
    Dim c As Range
    Dim bidon As String
    myCtrl = MG_UserForm.getControl("TextField1")
    adrFCN = myCtrl.Text 'faute de mieux, il faut se débrouiller à la main
    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 myCtrl As Object
    Dim adrPar As String
    Dim Nadr As Integer
    Dim c As Range

```

```
Dim bidon As String
Dim ii As Integer
    myCtrl = MG_UserForm.getControl("TextField2")
    adrPar = myCtrl.Text 'faute de mieux, il faut se débrouiller à la main
    adrPar = Remplace(adrPar, ";", ",") '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
    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 myCtrl As Object
    Dim adrDPar As String
```

```

Dim Nadr As Integer
Dim c As Range
Dim bidon As String
Dim ii As Integer
    myCtrl = MG_UserForm.getControl("TextField3")
    adrDPar = myCtrl.Text 'faute de mieux, il faut se débrouiller à la main
    adrDPar = Remplace(adrDPar, ";", ",") '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
    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 myCtrl As Object
    Dim adrParN As String
    Dim Nadr As Integer
    Dim c As Range
    Dim bidon As String
    Dim ii As Integer
    myCtrl = MG_UserForm.getControl("CheckBox1")
    If myCtrl.State <> 0 Then 'l'utilisateur propose des noms
        myCtrl = MG_UserForm.getControl("TextField4")
        adrParN = myCtrl.Text 'faute de mieux, il faut se débrouiller à la main
        adrParN = Remplace(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()
    Dim myCtrl As Object
    getOptions = False
    If getStep Then
        If getEpsi Then
            If getImpr Then
                myCtrl = MG_UserForm.getControl("CheckBox2")
                MG_err = myCtrl.State
                getOptions = True
            End If
        End If
    End If
End Function

Private Function getStep()
    Dim myCtrl As Object
    Dim theText As String

```



```
Dim bidon As String
    myCtrl = MG_UserForm.getControl("TextField5")
theText = myCtrl.Text
If Not IsNumeric(theText) Then
    bidon = MsgBox("le pas global doit être numérique :." & vbCrLf & theText,
vbOKOnly)
    getStep = False
    Exit Function
End If
MG_step = CDbI(theText)
getStep = True
End Function
```

```
Private Function getEpsi()
    Dim myCtrl As Object
    Dim theText As String
    Dim bidon As String
    myCtrl = MG_UserForm.getControl("TextField6")
theText = myCtrl.Text
If Not IsNumeric(theText) Then
    bidon = MsgBox("la sensibilité doit être numérique :." & vbCrLf & theText,
vbOKOnly)
    getEpsi = False
    Exit Function
End If
MG_epsi = Abs(CDbI(theText))
getEpsi = True
End Function
```

```
Private Function getImpr()
    Dim myCtrl As Object
    Dim theText As String
    Dim bidon As String
    myCtrl = MG_UserForm.getControl("TextField7")
theText = myCtrl.Text
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()
    Dim myCtrl As Object
```

```
'fermeture par le bouton d'annulation
Call restaurePar(MG_minimisation) 'on ne restaure que si une minimisation
vient d'être faite
'sinon simple fermeture, mais de toute façon on mémorise l'état des contrôles
Call CommandButton3_Click()'même sans purge, le dialogue fermé ne
mémorise pas son contenu)
End Sub
```

```
Private Sub CommandButton3_Click()
'fermeture par le bouton d'acceptation
Dim myCtrl As Object
'simple fermeture, mais de toute façon on mémorise l'état des contrôles
myCtrl = MG_UserForm.getControl("TextField1")
MG_TadrFCN = myCtrl.Text
myCtrl = MG_UserForm.getControl("TextField2")
MG_TadrPar = myCtrl.Text
myCtrl = MG_UserForm.getControl("TextField3")
MG_TadrDPar = myCtrl.Text
myCtrl = MG_UserForm.getControl("CheckBox1")
MG_TParN = myCtrl.State
myCtrl = MG_UserForm.getControl("TextField4")
MG_TadrParN = myCtrl.Text
myCtrl = MG_UserForm.getControl("TextField5")
MG_Tstep = myCtrl.Text
myCtrl = MG_UserForm.getControl("TextField6")
MG_Tepsi = myCtrl.Text
myCtrl = MG_UserForm.getControl("TextField7")
MG_Timpr = myCtrl.Text
myCtrl = MG_UserForm.getControl("CheckBox2")
MG_Terr = myCtrl.State
myCtrl = MG_UserForm.getControl("TextField9")
MG_Tresult = myCtrl.Text

MG_UserForm.endExecute '(même sans purge, le dialogue fermé ne mémorise
pas son contenu)
End Sub
```

```
Private Sub CommandButton4_Click()
'pour récupérer l'adresse
Dim myCtrl As Object
Dim myChamp As Range
myCtrl = MG_UserForm.getControl("TextField1")
If TypeName(Selection) = "Range" Then
Set myChamp = Selection
myCtrl.Text = myChamp.Address
Else
MsgBox "La sélection n'est pas un champ"
```

```
End If  
End Sub
```

```
Private Sub CommandButton5_Click()  
    'pour récupérer l'adresse  
    Dim myCtrl As Object  
    Dim myChamp As Range  
    myCtrl = MG_UserForm.getControl("TextField2")  
    If TypeName(Selection) = "Range" Then  
        Set myChamp = Selection  
        myCtrl.Text = myChamp.Address  
    Else  
        MsgBox "La sélection n'est pas un champ"  
    End If  
End Sub
```

```
Private Sub CommandButton6_Click()  
    'pour récupérer l'adresse  
    Dim myCtrl As Object  
    Dim myChamp As Range  
    myCtrl = MG_UserForm.getControl("TextField3")  
    If TypeName(Selection) = "Range" Then  
        Set myChamp = Selection  
        myCtrl.Text = myChamp.Address  
    Else  
        MsgBox "La sélection n'est pas un champ"  
    End If  
End Sub
```

```
Private Sub CommandButton7_Click()  
    'pour récupérer l'adresse  
    Dim myCtrl As Object  
    Dim myChamp As Range  
    myCtrl = MG_UserForm.getControl("TextField4")  
    If TypeName(Selection) = "Range" Then  
        Set myChamp = Selection  
        myCtrl.Text = myChamp.Address  
    Else  
        MsgBox "La sélection n'est pas un champ"  
    End If  
End Sub
```

```
Private Sub RollCToF1(aCtrl As String,aDX As Integer,aDY As Integer)  
    'la superposition de pages n'est pas si pratique ; on utilise un mécanisme  
    différent  
    Dim myCtrl As Object  
    Dim myX,myY As Integer
```

```
myCtrl = MG_UserForm.getControl(aCtrl)
myX = myCtrl.Model.PositionX Mod aDX
myCtrl.Model.PositionX = myX
myY = (myCtrl.Model.PositionY - 60) Mod aDY
myCtrl.Model.PositionY = 60 + myY
End Sub
```

```
Private Sub RollCToF2(aCtrl As String,aDX As Integer,aDY As Integer)
    'la superposition de pages n'est pas si pratique ; on utilise un mécanisme
différent
    Dim myCtrl As Object
    Dim myX,myY As Integer
    myCtrl = MG_UserForm.getControl(aCtrl)
    myX = myCtrl.Model.PositionX Mod aDX
    myCtrl.Model.PositionX = myX + aDX
    myY = (myCtrl.Model.PositionY - 60) Mod aDY
    myCtrl.Model.PositionY = 60 + myY
End Sub
```

```
Private Sub RollCToF3(aCtrl As String,aDX As Integer,aDY As Integer)
    'la superposition de pages n'est pas si pratique ; on utilise un mécanisme
différent
    Dim myCtrl As Object
    Dim myX,myY As Integer
    myCtrl = MG_UserForm.getControl(aCtrl)
    myX = myCtrl.Model.PositionX Mod aDX
    myCtrl.Model.PositionX = myX + aDX
    myY = (myCtrl.Model.PositionY - 60) Mod aDY
    myCtrl.Model.PositionY = 60 + myY + aDY
End Sub
```

```
Private Sub RollCToF4(aCtrl As String,aDX As Integer,aDY As Integer)
    'la superposition de pages n'est pas si pratique ; on utilise un mécanisme
différent
    Dim myCtrl As Object
    Dim myX,myY As Integer
    myCtrl = MG_UserForm.getControl(aCtrl)
    myX = myCtrl.Model.PositionX Mod aDX
    myCtrl.Model.PositionX = myX
    myY = (myCtrl.Model.PositionY - 60) Mod aDY
    myCtrl.Model.PositionY = 60 + myY + aDY
End Sub
```

```
Private Sub CommandButton8_Click()
    'pour afficher la première page (et elle seule)
    Dim myCtrl As Object
    Dim myX,myDX,myY,myDY As Integer
```

```
if MG_nPage <> 1 then
MG_nPage = 1
'on adapte l'en-tête
myCtrl = MG_UserForm.getControl("CommandButton8")
myCtrl.Label = "MODÈLE"
myCtrl = MG_UserForm.getControl("CommandButton9")
myCtrl.Label = "options..."
myCtrl = MG_UserForm.getControl("CommandButton10")
myCtrl.Label = "contraintes..."
myCtrl = MG_UserForm.getControl("CommandButton11")
myCtrl.Label = "résultats..."
'on repère les décalages d'après le Frame2
myCtrl = MG_UserForm.getControl("FrameControl2")
myDX = myCtrl.Model.Width + 8
myDY = myCtrl.Model.Height + 8

'on déplace les contrôles de la page 1
Call RollCToF1("Label1",myDX,myDY)
Call RollCToF1("TextField1",myDX,myDY)
Call RollCToF1("CommandButton4",myDX,myDY)
Call RollCToF1("Label2",myDX,myDY)
Call RollCToF1("TextField2",myDX,myDY)
Call RollCToF1("CommandButton5",myDX,myDY)
Call RollCToF1("Label3",myDX,myDY)
Call RollCToF1("TextField3",myDX,myDY)
Call RollCToF1("CommandButton6",myDX,myDY)
Call RollCToF1("CheckBox1",myDX,myDY)
Call RollCToF1("TextField4",myDX,myDY)
Call RollCToF1("CommandButton7",myDX,myDY)

'on déplace les contrôles de la page 2
Call RollCToF2("Label4",myDX,myDY)
Call RollCToF2("TextField5",myDX,myDY)
Call RollCToF2("Label5",myDX,myDY)
Call RollCToF2("TextField6",myDX,myDY)
Call RollCToF2("Label6",myDX,myDY)
Call RollCToF2("TextField7",myDX,myDY)
Call RollCToF2("CheckBox2",myDX,myDY)

'on déplace les contrôles de la page 3
Call RollCToF3("TextField8",myDX,myDY)

'on déplace les contrôles de la page 4
Call RollCToF4("TextField9",myDX,myDY)
End if
End Sub
```

```
Private Sub CommandButton9_Click()
    'pour afficher la deuxième page (et elle seule)
    Dim myCtrl As Object
    Dim myX,myDX,myY,myDY As Integer
    if MG_nPage <> 2 then
        MG_nPage = 2
        'on adapte l'en-tête
        myCtrl = MG_UserForm.getControl("CommandButton8")
        myCtrl.Label = "modèle..."
        myCtrl = MG_UserForm.getControl("CommandButton9")
        myCtrl.Label = "OPTIONS"
        myCtrl = MG_UserForm.getControl("CommandButton10")
        myCtrl.Label = "contraintes..."
        myCtrl = MG_UserForm.getControl("CommandButton11")
        myCtrl.Label = "résultats..."
        'on repère les décalages d'après le Frame2
        myCtrl = MG_UserForm.getControl("FrameControl2")
        myDX = myCtrl.Model.Width + 8
        myDY = myCtrl.Model.Height + 8

        'on déplace les contrôles de la page 1
        Call RollCToF4("Label1",myDX,myDY)
        Call RollCToF4("TextField1",myDX,myDY)
        Call RollCToF4("CommandButton4",myDX,myDY)
        Call RollCToF4("Label2",myDX,myDY)
        Call RollCToF4("TextField2",myDX,myDY)
        Call RollCToF4("CommandButton5",myDX,myDY)
        Call RollCToF4("Label3",myDX,myDY)
        Call RollCToF4("TextField3",myDX,myDY)
        Call RollCToF4("CommandButton6",myDX,myDY)
        Call RollCToF4("CheckBox1",myDX,myDY)
        Call RollCToF4("TextField4",myDX,myDY)
        Call RollCToF4("CommandButton7",myDX,myDY)

        'on déplace les contrôles de la page 2
        Call RollCToF1("Label4",myDX,myDY)
        Call RollCToF1("TextField5",myDX,myDY)
        Call RollCToF1("Label5",myDX,myDY)
        Call RollCToF1("TextField6",myDX,myDY)
        Call RollCToF1("Label6",myDX,myDY)
        Call RollCToF1("TextField7",myDX,myDY)
        Call RollCToF1("CheckBox2",myDX,myDY)

        'on déplace les contrôles de la page 3
        Call RollCToF2("TextField8",myDX,myDY)

        'on déplace les contrôles de la page 4
```

```
    Call RollCToF3("TextField9",myDX,myDY)
End if
End Sub
```

```
Private Sub CommandButton10_Click()
'pour afficher la troisième page (et elle seule)
Dim myCtrl As Object
    Dim myX,myDX,myY,myDY As Integer
    if MG_nPage <> 3 then
        MG_nPage = 3
        'on adapte l'en-tête
        myCtrl = MG_UserForm.getControl("CommandButton8")
        myCtrl.Label = "modèle..."
        myCtrl = MG_UserForm.getControl("CommandButton9")
        myCtrl.Label = "options..."
        myCtrl = MG_UserForm.getControl("CommandButton10")
        myCtrl.Label = "CONTRAINTES"
        myCtrl = MG_UserForm.getControl("CommandButton11")
        myCtrl.Label = "résultats..."
        'on repère les décalages d'après le Frame2
        myCtrl = MG_UserForm.getControl("FrameControl2")
        myDX = myCtrl.Model.Width + 8
        myDY = myCtrl.Model.Height + 8

        'on déplace les contrôles de la page 1
        Call RollCToF3("Label1",myDX,myDY)
        Call RollCToF3("TextField1",myDX,myDY)
        Call RollCToF3("CommandButton4",myDX,myDY)
        Call RollCToF3("Label2",myDX,myDY)
        Call RollCToF3("TextField2",myDX,myDY)
        Call RollCToF3("CommandButton5",myDX,myDY)
        Call RollCToF3("Label3",myDX,myDY)
        Call RollCToF3("TextField3",myDX,myDY)
        Call RollCToF3("CommandButton6",myDX,myDY)
        Call RollCToF3("CheckBox1",myDX,myDY)
        Call RollCToF3("TextField4",myDX,myDY)
        Call RollCToF3("CommandButton7",myDX,myDY)

        'on déplace les contrôles de la page 2
        Call RollCToF4("Label4",myDX,myDY)
        Call RollCToF4("TextField5",myDX,myDY)
        Call RollCToF4("Label5",myDX,myDY)
        Call RollCToF4("TextField6",myDX,myDY)
        Call RollCToF4("Label6",myDX,myDY)
        Call RollCToF4("TextField7",myDX,myDY)
        Call RollCToF4("CheckBox2",myDX,myDY)
    end if
End Sub
```

```
'on déplace les contrôles de la page 3
Call RollCToF1("TextField8",myDX,myDY)

'ou déplace les contrôles de la page 4
Call RollCToF2("TextField9",myDX,myDY)
End if
End Sub

Private Sub CommandButton11_Click()
'pour afficher la quatrième page (et elle seule)
Dim myCtrl As Object
Dim myX,myDX,myY,myDY As Integer
if MG_nPage <> 4 then
MG_nPage = 4
'on adapte l'en-tête
myCtrl = MG_UserForm.getControl("CommandButton8")
myCtrl.Label = "modèle..."
myCtrl = MG_UserForm.getControl("CommandButton9")
myCtrl.Label = "options..."
myCtrl = MG_UserForm.getControl("CommandButton10")
myCtrl.Label = "contraintes..."
myCtrl = MG_UserForm.getControl("CommandButton11")
myCtrl.Label = "RÉSULTATS"
'on repère les décalages d'après le Frame2
myCtrl = MG_UserForm.getControl("FrameControl2")
myDX = myCtrl.Model.Width + 8
myDY = myCtrl.Model.Height + 8

'on déplace les contrôles de la page 1
Call RollCToF2("Label1",myDX,myDY)
Call RollCToF2("TextField1",myDX,myDY)
Call RollCToF2("CommandButton4",myDX,myDY)
Call RollCToF2("Label2",myDX,myDY)
Call RollCToF2("TextField2",myDX,myDY)
Call RollCToF2("CommandButton5",myDX,myDY)
Call RollCToF2("Label3",myDX,myDY)
Call RollCToF2("TextField3",myDX,myDY)
Call RollCToF2("CommandButton6",myDX,myDY)
Call RollCToF2("CheckBox1",myDX,myDY)
Call RollCToF2("TextField4",myDX,myDY)
Call RollCToF2("CommandButton7",myDX,myDY)

'on déplace les contrôles de la page 2
Call RollCToF3("Label4",myDX,myDY)
Call RollCToF3("TextField5",myDX,myDY)
Call RollCToF3("Label5",myDX,myDY)
Call RollCToF3("TextField6",myDX,myDY)
```



```
Call RollCToF3("Label6",myDX,myDY)
Call RollCToF3("TextField7",myDX,myDY)
Call RollCToF3("CheckBox2",myDX,myDY)
```

```
'on déplace les contrôles de la page 3
Call RollCToF4("TextField8",myDX,myDY)
```

```
'on déplace les contrôles de la page 4
Call RollCToF1("TextField9",myDX,myDY)
End if
```

```
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
```