mDF MFCmultiples à la loupe
Catégorie : VBA par l'exemple
Publié par myDearFriend! le 24-03-2008

Mise en Forme Conditionnelle avec critères illimités ?

Apparue avec Excel 97, la Mise en Forme Conditionnelle (MFC) permet l'application d'un format dynamique aux cellules de feuille de calcul. Pour les versions Excel antérieures à 2007, cet outil fort pratique souffre toutefois d'un inconvénient majeur : la Mise en Forme Conditionnelle est limitée à 3 critères maximum. Ainsi, en comptant sa couleur d'origine, une cellule peut recevoir par exemple, jusqu'à 4 couleurs de fonds suivant sa valeur contenue.

Aussi, une des questions les plus fréquemment rencontrée dans les forums de discussions est la suivante :

Comment contourner cette limite de 3 critères pour la Mise en Forme Conditionnelle Excel ?

De nombreuses solutions sont ainsi proposées...

- Il est possible notamment de gérer jusqu'à 7 conditions grâce à une astucieuse combinaison de MFC classique et de format de nombres spécifique (Alain Vallon l'illustre parfaitement dans un classeur exemple très explicite que vous trouverez sur Excelabo : le fichier se nomme av-7couleursconditionnelles.

- Et pour aller au delà, de nombreuses solutions à base de langage VBA font généralement le bonheur des demandeurs. Ces solutions souffrent par contre d'un problème important : elles requièrent un code en accord avec chaque situation exposée et demandent donc une certaine connaissance (voire une maitrise) du langage de programmation pour pouvoir adapter la proposition au projet réel et en assurer les modifications futures...

L'utilisation de VBA est donc le meilleur moyen de simuler une vraie MFC en outrepassant les limites imposées !

Historique de mDF MFCmultiples

L'idéal serait de proposer une procédure VBA :

- facile à insérer dans un projet Excel, même pour un utilisateur inexpérimenté VBA !
- paramétrable et permettant de gérer de nombreux cas de figures sans avoir à retourner dans l'éditeur VBA pour en modifier le code.
- qui représente une solution fiable et simple à mettre en oeuvre pour l'utilisateur final.
- qui ne remet pas en cause la pratique habituelle d'une Mise en Forme Conditionnelle classique.

C'est donc en réfléchissant sur ces quatre points que j'ai proposé la toute première version de mDF MFCmultiples sur le forum Excel-Downloads en Septembre 2004, dans le fil de discussions : Couleur cellule selon valeur .
J'expliquais dans ce fil les principes de fonctionnement retenus, qui restent d'ailleurs aujourd'hui toujours valables... L'objectif d'alors : détourner, en toute transparence (enfin... presque !), la vraie MFC pour en prendre le contrôle et assurer le traitement par VBA.

C'est un an plus tard que je présentais une deuxième version améliorée, toujours dans le forum d'XLD, dans le fil : Planning couleurs selon saisie.

Et c'est aujourd'hui (seulement) que je vous présente la dernière mouture de cette macro, complètement revue, améliorée et optimisée.

Cette nouvelle version mDF MFCmultiples v5.0 permet :

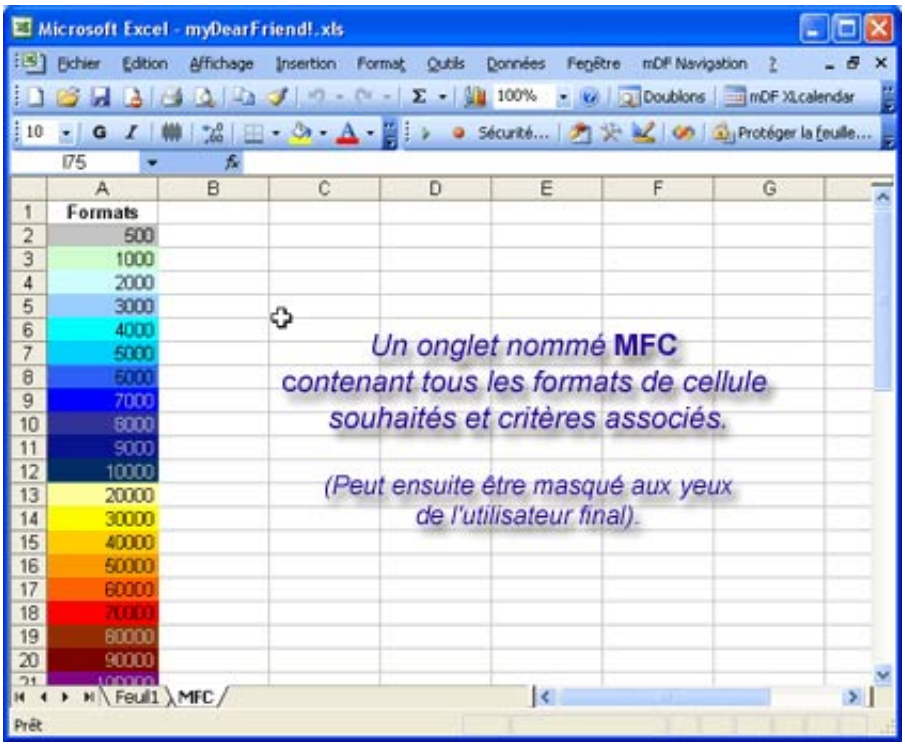
1. de simuler une Mise en Forme Conditionnelle avec nombre de critères et formats illimités.
2. de prendre en compte pour traitement, les cellules contenant des constantes, mais également les cellules contenant des formules.
3. d'appliquer individuellement des formats distincts sur des cellules isolées.
4. d'appliquer, à l'aide d'une seule condition, un même format sur une plage entière de cellules en ligne ou en colonne, ou sur une partie seulement de la ligne ou de la colonne, ou même sur plusieurs plages distinctes sur une même ligne ou colonne.
5. d'utiliser le comparateur « égale à » pour tout type de valeur (numérique et alphabétique)
6. d'utiliser les comparateurs numériques suivants : « inférieur à », « inférieur ou égale à », «supérieur à » ou « supérieur ou égale à ».

Téléchargez le classeur exemple de la dernière version : Téléchargement

Mise en oeuvre

Préparatifs (2 étapes) :

1. Il convient de créer un nouvel onglet nommé MFC dans le classeur cible. Cet onglet contiendra en colonne A, tous les critères et formats associés que l'utilisateur ou le concepteur du document pourra modifier à sa guise. Les critères saisis seront automatiquement triés par la macro dès l'activation d'un autre onglet. Bien sûr, cette feuille « MFC » peut ensuite être masqué pour utilisation finale.



2. Copiez-collez ensuite l'ensemble du code VBA suivant, dans le module de code de l'objet ThisWorkbook de votre classeur :

DANS LE MODULE DE CODE DE L'OBJET THISWORKBOOK

```
Option Explicit
'-------------------------------------------------------------------------------
----------
' Auteur    : Didier FOURGEOT (myDearFriend!) - www.mdf-xlpages.com
' Date      : 21/03/2008
' Sujet     : mDF MFCmultiples v5.0
'-------------------------------------------------------------------------------
----------
Private Sub Workbook_SheetChange(ByVal Sh As Object, ByVal Target As Range)
Dim FCible As Range, RCible As Range, Cible As Range, Plage As
 Range, T As Range, _
Tplage As Range, PlageFC As Range
Dim Adr As String
Dim N As Boolean, B As Boolean, P As Boolean, A As Boolean, VFC As Boolean
    On Error Resume Next
    Set PlageFC = Sh.Cells.SpecialCells(xlCellTypeAllFormatConditions)
    If PlageFC Is Nothing Then Exit Sub
    'Définition de la Plage cible
    Set Plage = Target
    Set Tplage = Plage.Dependents
    Set Plage = Application.Union(Plage, Tplage)
```

```
    On Error GoTo 0
    Set Plage = Application.Intersect(Plage, PlageFC)
    If Plage Is Nothing Then Exit Sub
    Application.ScreenUpdating = False
    Set Tplage = Nothing
    For Each T In Plage
        VFC = VerifFCond(T)
        If VFC Then
            If Tplage Is Nothing Then
                Set Tplage = T
            Else
                Set Tplage = Union(Tplage, T)
            End If
        End If
    Next T
    'Traitement de la plage Cible
    If Not Tplage Is Nothing Then
        With ActiveWorkbook.Styles("Normal")
            N = .IncludeNumber
            B = .IncludeBorder
            P = .IncludeProtection
            A = .IncludeAlignment
            .IncludeNumber = False
            .IncludeBorder = False
            .IncludeProtection = False
            .IncludeAlignment = False
        End With
        For Each Cible In Tplage
            Set FCible = FormatCible(Cible)
            Set RCible = Nothing
            On Error Resume Next
            With Cible
                Adr = Mid(.ID, 3)
                Select Case Adr
                Case "Cel"
                    Set RCible = Cible
                Case "Lig"
                    Set
RCible = Application.Intersect(.EntireRow, ActiveSheet.UsedRange)
                Case Else
                    Adr = Replace(Adr, ";", ",")
                    If Val(Replace(Adr, "$", "")) > 0 Then
                        Set
RCible = Application.Intersect(.EntireColumn, Range(Adr))
                    Else
                        Set
RCible = Application.Intersect(.EntireRow, Range(Adr))
                    End If
                End Select
            End With
            On Error GoTo 0
            If Not RCible Is Nothing Then
                With RCible
                    If FCible.Row = 65536 Then
                        'Format standard
```

```
                        .Style = "Normal"
                    Else
                        'Format MFC
                        With .Font
                            .Bold = FCible.Font.Bold
                            .Color = FCible.Font.Color
                            .Italic = FCible.Font.Italic
                            .Name = FCible.Font.Name
                            .Size = FCible.Font.Size
                            .Strikethrough = FCible.Font.Strikethrough
                            .Subscript = FCible.Font.Subscript
                            .Superscript = FCible.Font.Superscript
                            .Underline = FCible.Font.Underline
                        End With
                        With .Interior
                            .Color = FCible.Interior.Color
                            .Pattern = FCible.Interior.Pattern
                            .PatternColor = FCible.Interior.PatternColor
                        End With
                    End If
                End With
            End If
        Next Cible
        With ActiveWorkbook.Styles("Normal")
            .IncludeNumber = N
            .IncludeBorder = B
            .IncludeProtection = P
            .IncludeAlignment = A
        End With
    End If
    Application.ScreenUpdating = True
End Sub

Private Function VerifFCond(C As Range) As Boolean
Dim FCF As String, Op As String
    On Error Resume Next
    With C.FormatConditions(1)
        FCF = .Formula1
        Op = CStr(.Operator)
    End With
    On Error GoTo 0

    Select Case Val(Op)
    Case 3, 5 To 8
        Op = Op & "|"
    Case Else
        Exit Function
    End Select

    VerifFCond = True
    Select Case Left(FCF, 5)
    Case "=mDF"
        C.ID = Op & "Cel"
    Case "=mDF("
        If FCF = "=mDF()" Then
```

```
            C.ID = Op & "Lig"
        Else
            C.ID = Op & Replace(Replace(FCF, ")", ""), "=mDF(", "")
        End If
    Case Else
        C.ID = ""
        VerifFCond = False
    End Select
End Function

Private Function FormatCible(Cible As Range) As Range
Dim C As Range
Dim L As Variant, Veg As Variant, Veginf As Variant
    With Sheets("MFC")
        If Not IsEmpty(Cible) Then
            If Not (Val(Cible.ID) > 3 And Not IsNumeric(Cible.Value)) Then
                Veg = Application.Match(Cible.Value, .Columns(1), 0)
                Veginf = Application.Match(Cible.Value, .Columns(1), 1)
                Select Case Val(Cible.ID)
                Case 3  '=
                    L = IIf(IsError(Veg), 0, Veg)
                Case 5  '>
                    L = IIf(IsError(Veginf), 0, Veginf) - 1
                Case 6  '<
                    L = Application.Max(IIf(IsError(Veginf), 0, Veginf) + 1, 2
)
                Case 7  '>=
                    L = IIf(IsError(Veg), 0, Veg)
                    If L = 0 Then
                        L = IIf(IsError(Veginf), 0, Veginf)
                    End If
                Case 8  '<=
                    L = IIf(IsError(Veg), 0, Veg)
                    If L = 0 Then
                        L = Application.Max(IIf(IsError(Veginf), 0, Veginf) +
1, 2)
                    End If
                End Select
                If L > 1 Then
                    Set C = .Cells(L, 1)
                End If
            End If
        End If
        If C Is Nothing Then Set C = .Cells(65536, 1)
    End With
    Set FormatCible = C
End Function

Private Sub Workbook_SheetDeactivate(ByVal Sh As Object)
'Trie automatiquement les critères de l'onglet MFC
    If Sh.Name = "MFC" Then
        Application.ScreenUpdating = False
        With Sh
            .Columns(1).Sort Key1:=.Cells(2, 1), Order1:=xlAscending, Header:=
xlGuess, _
```

```
                OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom
        End With
        Application.ScreenUpdating = True
    End If
End Sub
```

Comment faire ?

C'est tout !
Votre classeur est maintenant prêt pour tirer bénéfice de cette macro.

Utilisation :

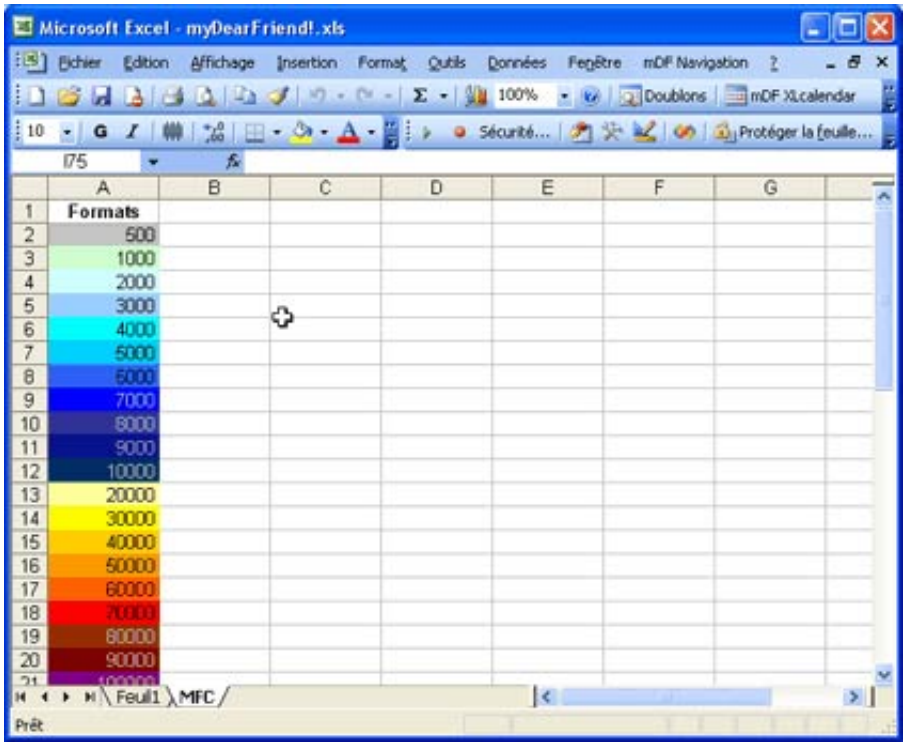
Vous allez maintenant décider des cellules qui devront être gérées par cette macro n'importe où dans votre classeur. L'opération est simple, elle consiste à leur appliquer un format conditionnel spécial. Procédez comme suit :

- Sélectionnez la (ou les) cellule(s) cible(s)
- Faites menu Format / Mise en forme conditionnelle...
- Dans Condition 1 : choisissez « La valeur de la cellule est »
- Dans la liste déroulante, sélectionnez un des comparateurs supportés par cette macro, soit :
    - « égale à »
    - « inférieur à »
    - « inférieur ou égale à »
    - « supérieur à »
    - « supérieur ou égale à »
- Dans la zone de critère, saisissez le critère spécial comme suit :
    - Pour agir sur une cellule isolée : =mDF
    - Pour agir sur une ligne entière de données : =mDF()
    - Pour agir sur une partie de la ligne de données : =mDF($C:$H)
    - Pour agir sur plusieurs parties de la ligne de données : =mDF($C:$H;$K:$K;$P:$T)
    - Pour agir sur la colonne de données : =mDF($1:$45)
    - Pour agir sur plusieurs parties de la colonne de données : =mDF($5:$18;$22:$25)

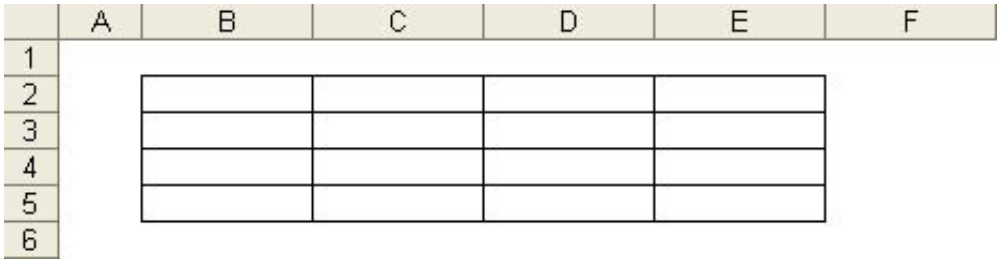
Nb : utilisez impérativement des références de plages (lignes/colonnes) absolues (utilisation du signe $).
Les différentes plages doivent être séparées par un point-virgule (;).

Exemples

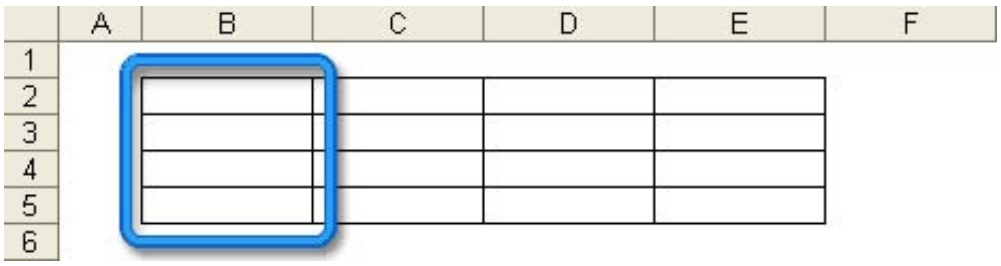
Considérons l'onglet MFC suivant :



Et le tableau ci-dessous dans une feuille du même classeur :



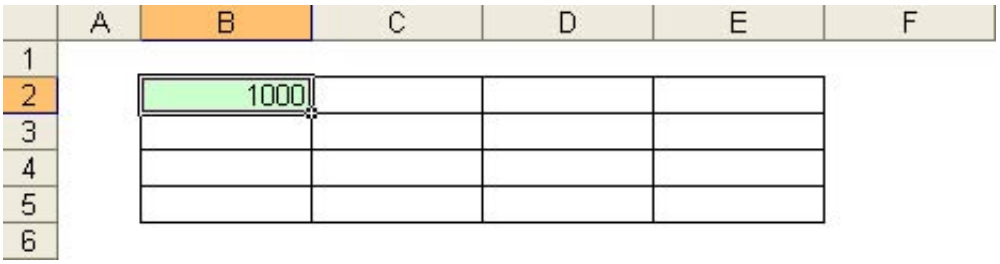
J'ai appliqué des Mises en Forme Conditionnelles spéciales dans chaque cellule de la colonne B :



- En cellule B2, j'ai appliqué la MFC suivante :

Comparateur	Critère
égale à	=mDF

Si je saisis la valeur 1000 dans cette cellule, j'obtiens ce résultat :

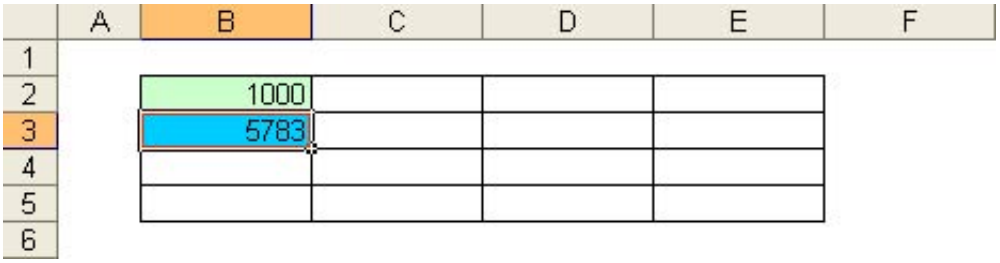


L'effet aurait été identique si la valeur 1000 était issue du résultat d'une formule.

- En cellule B3, j'ai appliqué la MFC suivante :

Comparateur	Critère
supérieure ou égale à	=mDF

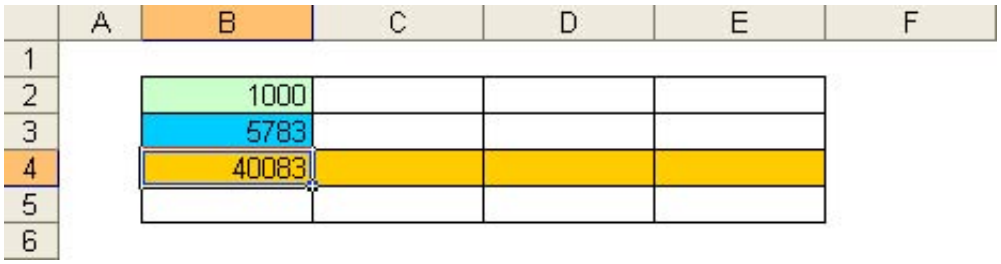
Si je saisis la valeur 5783 dans cette cellule, j'obtiens ce résultat :



- En cellule B4, j'ai appliqué la MFC suivante :

Comparateur	Critère
supérieure ou égale à	=mDF()

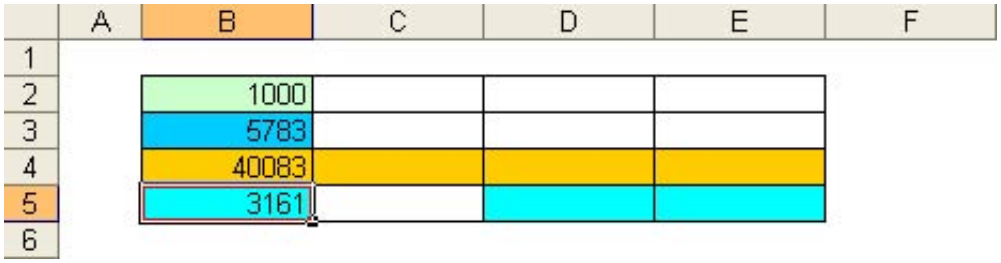
Si je saisis la valeur 40083 dans cette cellule, j'obtiens ce résultat :



- En cellule B5, j'ai appliqué la MFC suivante :

Comparateur	Critère
inférieure à	=mDF($B:$B;$D:$E)

Si je saisis la valeur 3161 dans cette cellule, j'obtiens ce résultat :



Autre exemple :

Basé sur le même onglet MFC ci-dessus, dans le tableau suivant, chaque cellule de la colonne C (à

partir de C3) s'est vue appliquer la Mise en Forme Conditionnelle suivante :

Comparateur	Critère
supérieure ou égale à	=mDF($B:$C)

	A	B	C	D
1				
2		Pays	Superficie (km2)	Population
3		Luxembourg	2586	468571
4		Chypre	9250	780133
5		Monténégro	13812	716029
6		Slovénie	20273	2011070
7		Macédoine	25333	2045262
8		Belgique	30528	10364388
9		Moldavie	33843	4455421
10		Suisse	41290	7489370
11		Pays Bas	41526	16407491
12		Danemark	43094	5432335
13		Estonie	45226	1332893
14		Slovaquie	48845	5431363
15		Bosnie	51129	4430494
16		Croatie	56542	4495904
17		Lettonie	64589	2290237
18		Lituanie	65200	3596617
19		Irlande	70280	4015676
20		Tchéquie	78866	10241138
21		Autriche	83870	8184691
22		Serbie	88538	10113146
23		Portugal	92391	10566212
24		Hongrie	93030	10006835
25		Islande	103000	296737
26		Bulgarie	110910	7450349
27		Grèce	131940	10668354
28		Bélarus	207600	10300483
		[illegible]	237500	22329977

Démonstration

Vous trouverez et pourrez tester dans le fichier Exemple en téléchargement un panel des possibilités de cette macro.

Limites de cette macro et points de vigilance

- Cette macro est compatible Excel PC pour versions 2000 à 2007 (mais présente sans doute peu d'intérêt sur Excel 2007).
- La MFC spéciale doit impérativement être affectée en Condition 1 de la boîte de dialogue Mise en Forme Conditionnelle.

- Si vous attribuez de « vraies » MFC en conditions 2 ou 3 de cette boîte de dialogue, ces MFC classiques primeront sur la MFC spéciale en cas de conflit.
- Le traitement d'une cellule contenant une formule est effectif si, et seulement si, les antécédents de cette formule sont physiquement sur la même feuille de calcul. Aussi, prenez note qu'une formule faisant référence à une autre classeur, une autre feuille ou un autre onglet ne sera pas gérée par cette MFC spéciale.
- L'abus de formules traitées par cette MFC spéciale risque fort d'être source de lenteur pour votre projet.
- Seul le comparateur « égale à » est géré lors de la comparaison des chaines de caractères (valeurs non numériques).
- Il est impératif d'utiliser des références absolues (utilisation du signe $) lorsque vous spécifiez des plages Lignes ou Colonnes en critère.
- Pourquoi ne pas en faire une .XLA (macro complémentaire) ? C'est bien évidemment l'objectif final de cette macro (voir mon post du 05/09/2004, 20h34 dans le fil de discussions Couleur cellule selon valeur). Un complément permettrait notamment à l'utilisateur de faire abstraction de la phase intégration VBA lors des préparatifs du classeur. A ce jour, j'estime que le code de cette macro n'est pas suffisamment mature pour permettre cette métamorphose... ça viendra peut-être.