

La Mise en Forme Conditionnelle Illimitée sous Access

Par Philippe JOCHMANS 

Date de publication : 9 février 2008

Dernière mise à jour : 9 février 2008

Cet article a pour but de vous expliquer comment réaliser une mise en forme conditionnelle illimitée dans Access.

I - INTRODUCTION.....	3
II - PREREQUIS.....	3
III - PREPARATION DU FORMULAIRE.....	3
III-A - Insertion du composant dans le formulaire.....	3
III-B - Paramétrer l'aspect de notre feuille Excel.....	5
IV - LA MISE EN FORME CONDITIONNELLE SANS LIMITE.....	6
IV-A - Les données.....	6
IV-B - Préparer la feuille.....	7
IV-C - Affectation des données.....	8
IV-D - Réalisation de la mise en forme conditionnelle.....	9
IV-E - Rajouter un total.....	11
V - Conclusion.....	12
VI - TELECHARGEMENT.....	12
VII - REMERCIEMENTS.....	12

I - INTRODUCTION

Lors de la lecture de cet article de Silkyroad **Utilisation de l'objet OWC PivotTable** j'ai découvert que l'on pouvait faire des choses intéressantes avec le complément **Microsoft Office Web Components**.

Si vous ne possédez le composant Microsoft Office Web Components, vous pouvez le télécharger en fonction de votre version d'Access.

Complément d'Office 2003 : composants Web Office OWC11

Office XP Tool: Web Components OWC10

Nous allons voir dans cet article comment utiliser les propriétés d'une feuille Excel pour réaliser des choses qui ne sont pas possibles avec les composants classiques fournis avec Access.



*Tous les exemples que vous trouverez ont été réalisés sous Access 2002.
Vous trouverez une base Access en téléchargement à la fin de cet article.*

II - PREREQUIS

Une bonne connaissance du VBA Excel est nécessaire pour suivre cet article.

Vous trouverez toutes les informations nécessaires dans **la FAQ Excel** et **le Forum Excel de developpez.com**

III - PREPARATION DU FORMULAIRE

Nous allons voir dans cette partie comment insérer ce composant dans un formulaire, et les différentes options de paramètres manuels.

III-A - Insertion du composant dans le formulaire

Une fois l'installation faite, le composant se trouve dans la liste des ActiveX. Voici donc la procédure à réaliser pour mettre le composant dans un formulaire.

Ouvrez votre formulaire en mode création.

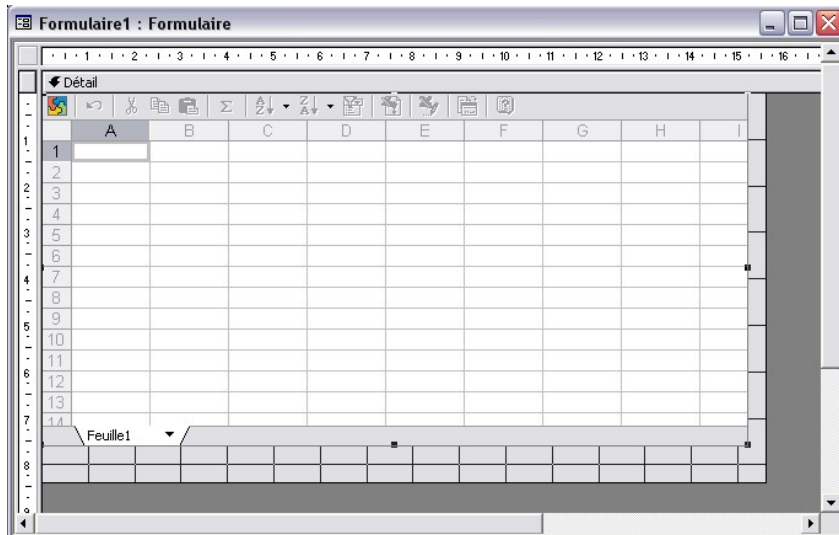
Ouvrez le menu pour récupérer la liste des ActiveX.



Sélectionnez le composant : Microsoft Office Spreadsheet 10.0.



Votre composant apparaît maintenant sur votre formulaire.



Comme vous le voyez nous avons maintenant une feuille Excel incorporée à notre formulaire.

i Lorsque vous avez incorporé votre composant, Access référence automatiquement la référence suivante Microsoft Office XP Web Components.

III-B - Paramétrer l'aspect de notre feuille Excel

Nous allons maintenant voir les différents paramétrages de notre feuille.

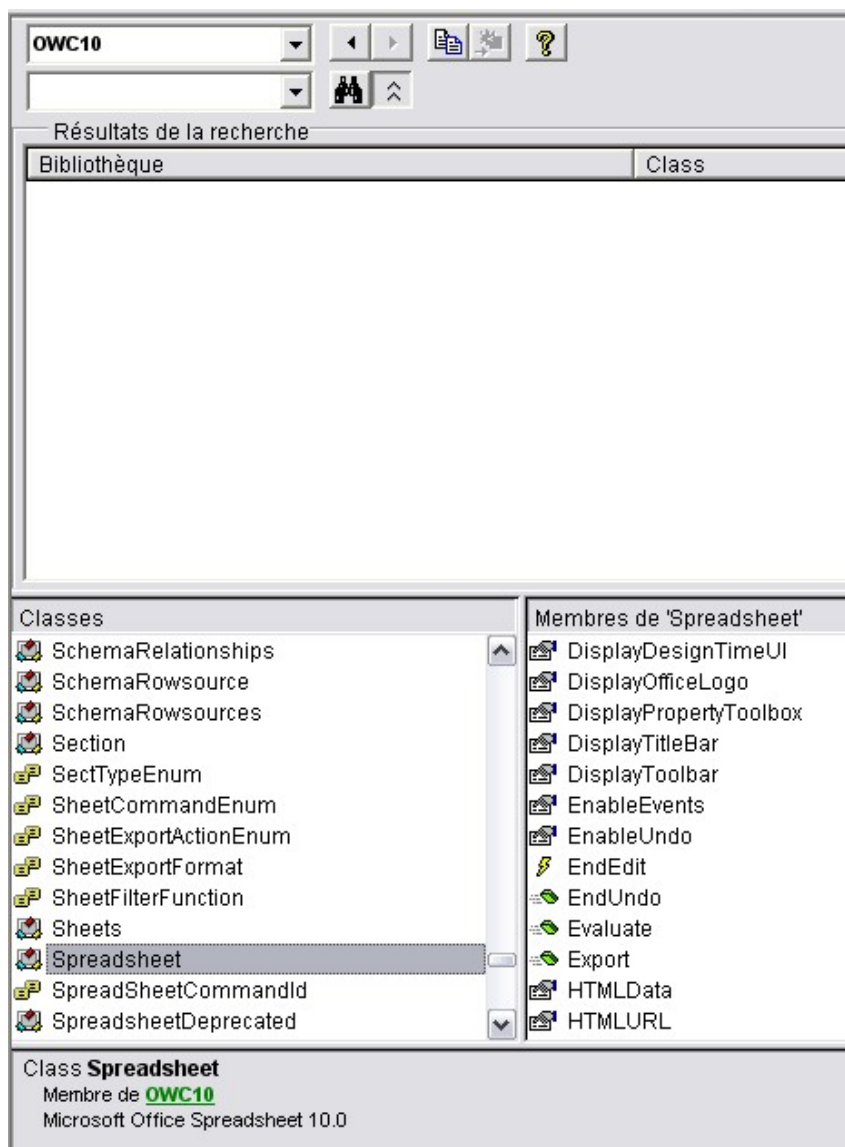
Pour cela faites un clic droit sur votre composant (toujours en mode création) et sélectionnez dans le menu contextuel **Object Microsoft Office SpreadSheet 10.0** et le sous-menu **Commands and Options ...**

Vous verrez alors une fenêtre s'ouvrir avec différents onglets.



Ces différents onglets vont vous permettre de paramétrer le comportement de votre feuille.

Toutes ces propriétés peuvent être modifiées par programmation, vous trouverez toutes les propriétés et méthodes dans l'explorateur d'objets (F2) de l'éditeur VB, en sélectionnant la bibliothèque OWC10.



IV - LA MISE EN FORME CONDITIONNELLE SANS LIMITE

Nous allons voir ici comment réaliser une mise en forme conditionnelle sans limite.

Lorsque l'on veut réaliser une mise en forme conditionnelle sous Access, on utilise en général un sous-formulaire en mode feuille de données. Hors la mise en forme conditionnelle ne peut comporter que trois conditions.

Pour la mise en forme conditionnelle classique, je vous conseille la lecture de cet article : **La mise en forme conditionnelle sous Access 2000/XP/2003**

IV-A - Les données

Pour réaliser cet exemple nous allons utiliser le personnel d'une société. Dans une table nous allons regrouper les structures hiérarchiques de celle-ci (il ne s'agit que d'un exemple, car en réalité les données seront réparties dans plusieurs tables, et seront reliées entre elles dans une requête).

tbl_Societe : Table					
	strNom	strPrenom	strService	strFonction	sngSalaireMensuel
▶	DURAND	Paul	Administratif	Directeur	5000
	DUPOND	Chantal	Administratif	Assistante de Direction	2000
	LEFEVRE	Christine	Administratif	Secrétaire	1500
	MARTIN	Christophe	Production	Responsable	3000
	DUMAS	Didier	Production	Chef d'équipe	2000
	DURAND	Denis	Production	Chef d'équipe	2000
	ROGER	Alain	Logistique	Responsable	3000
	PINEAU	Jean-Louis	Logistique	Cariste	1500
	CARTON	Gérard	Logistique	Magasinier	1700
	COUVRAT	Christine	Production	Opérateur	1400
	FERNAND	Didier	Production	Opérateur	1400
*					0

IV-B - Préparer la feuille

Nous allons préparer la feuille pour que visuellement elle ressemble à un sous-formulaire.

```

VBA

Public Sub PrepaMFC()
' =====
' Auteur      : Starec - Philippe JOCHMANS - http://starec.developpez.com
' Description  : Cette routine va permettre de préparer la feuille pour qu'elle
'               ressemble à une grille d'un sous-formulaire en mode feuille de donnée
' =====

' ===== déclaration =====
Dim wks As OWC10.Spreadsheet

' ===== affectation =====
Set wks = Me.SpreadMFC.Object

' ===== préparation de l'aspect =====
With wks
    .DisplayToolbar = False           ' on désactive la barre d'outils
    With .Windows(1)
        .DisplayHorizontalScrollBar = False
' on désactive la barre de défilement horizontale
        .DisplayWorkbookTabs = False ' on désactive la visualisation des onglets
        .DisplayColumnHeadings = False ' on désactive les entêtes de colonnes
        .DisplayRowHeadings = False  ' on désactive les entêtes de lignes
    End With
End With

End Sub

```

Quelques explications s'imposent sur ce code.

```

VBA

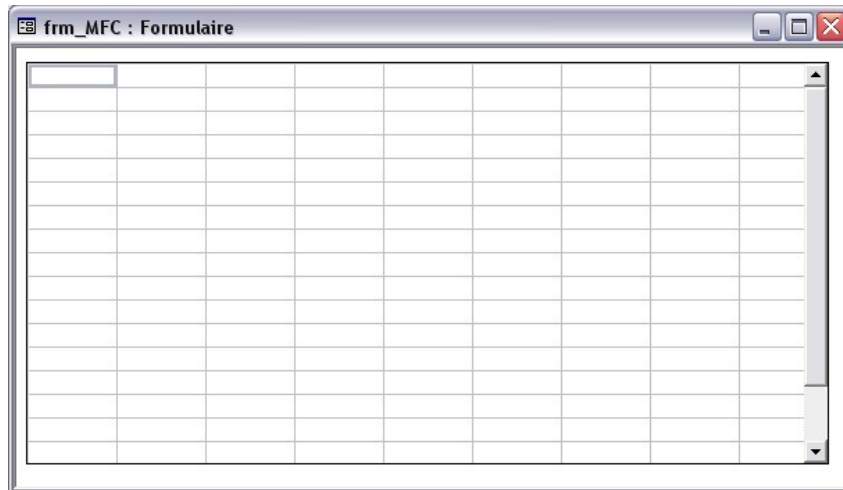
' ===== déclaration =====
Dim wks As OWC10.Spreadsheet

' ===== affectation =====
Set wks = Me.SpreadMFC.Object

```

Cette déclaration va nous permettre d'utiliser l'intellisense pour avoir les propriétés de notre feuille.

Le résultat commence à être significatif.



IV-C - Affectation des données

Pour remplir notre feuille, nous allons utiliser la technologie **DAO**

Le code suivant va nous permettre de remplir la feuille (celui-ci est entièrement commenté).

```
VBA
Public Sub RemplirFeuille()
    ' =====
    ' Auteur      : Starec - Philippe JOCHMANS - http://starec.developpez.com
    ' Description  : Cette routine va nous permettre de remplir la feuille
    ' =====

    ' ===== déclaration =====
    Dim wks As OWC10.Spreadsheet
    Dim rst As DAO.Recordset
    Dim strSql As String
    Dim i As Integer

    ' ===== affectation =====
    Set wks = Me.SpreadMFC.Object
    strSql = "SELECT strNom, strPrenom, strService, strFonction, sngSalaireMensuel " & _
            "FROM tbl_Societe;"
    Set rst = CurrentDb.OpenRecordset(strSql)

    ' ===== on vide la feuille =====
    wks.Cells.Delete
    wks.Windows(1).FreezePanes = False

    ' ===== mettre les entêtes de colonnes =====
    With wks
        .Range("A1").Value = "Nom"
        .Range("B1").Value = "Prénom"
        .Range("C1").Value = "Service"
        .Range("D1").Value = "Fonction"
        .Range("E1").Value = "Salaire Mensuel"
    End With

    ' ===== formatage de l'entête =====
    wks.Range("A1:E1").Interior.Color = RGB(220, 200, 250)

    ' ===== on fige les volets pour avoir l'entête fixe =====
    wks.Range("A2").Select
    wks.Windows(1).FreezePanes = True

    ' ===== on remplit maintenant la feuille =====
    i = 2
    While Not rst.EOF
        With wks
```


VBA

```

        .Range("A" & i).Value = rst("strNom")
        .Range("B" & i).Value = rst("strPrenom")
        .Range("C" & i).Value = rst("strService")
        .Range("D" & i).Value = rst("strFonction")
        .Range("E" & i).Value = rst("sngSalaireMensuel")
        .Range("E" & i).NumberFormat = "# ##0.00 $"
    End With
    i = i + 1
    rst.MoveNext
Wend

' ===== formatage de la feuille =====
With wks.Range("A1:E" & wks.Range("A1").End(xlDown).Row)
    .Columns.AutoFit
    .Borders.LineStyle = xlContinuous
End With

' ===== libération =====
rst.Close
Set rst = Nothing
Set wks = Nothing
End Sub

```

Le résultat :

i Il se peut que la commande pour figer les volets ne fonctionne pas. Dans ce cas, il faut les figer manuellement par l'intermédiaire de la fenêtre de paramétrage (III-B). Il suffit de vous positionner sur la cellule A2 et de cliquer sur l'icône qui permet de figer les volets.

IV-D - Réalisation de la mise en forme conditionnelle

Nous allons maintenant réaliser une mise en forme conditionnelle, à partir du cahier des charges suivant :

- Personnel Administratif en Rouge
- Avec un dégradé par sous catégorie
- Personnel de Production en Vert
- Avec un dégradé par sous catégorie
- Personnel de Logistique en Bleu
- Avec un dégradé par sous catégorie

Voici donc le code :

VBA

```

Public Sub RealisationMFC()
' =====
' Auteur      : Starec - Philippe JOCHMANS - http://starec.developpez.com

' Description  : Cette procédure va nous permettre de réaliser notre mise en forme conditionnelle
' =====

' ===== déclaration =====
Dim wks As OWC10.Spreadsheet
Dim i As Integer

' ===== affectation =====
Set wks = Me.SpreadMFC.Object

' ===== la mise en forme =====
For i = 2 To wks.Range("A2").End(xlDown).Row
    Select Case wks.Range("C" & i).Value
        Case "Administratif"
            Select Case wks.Range("D" & i).Value
                Case "Directeur"
                    wks.Range("A" & i & ":E" & i).Interior.Color = RGB(250, 200, 200)
                Case "Assistante de Direction"
                    wks.Range("A" & i & ":E" & i).Interior.Color = RGB(230, 180, 180)
                Case "Secrétaire"
                    wks.Range("A" & i & ":E" & i).Interior.Color = RGB(210, 160, 160)
            End Select
        Case "Production"
            Select Case wks.Range("D" & i).Value
                Case "Responsable"
                    wks.Range("A" & i & ":E" & i).Interior.Color = RGB(200, 250, 200)
                Case "Chef d'équipe"
                    wks.Range("A" & i & ":E" & i).Interior.Color = RGB(180, 230, 180)
                Case "Opérateur"
                    wks.Range("A" & i & ":E" & i).Interior.Color = RGB(160, 210, 160)
            End Select
        Case "Logistique"
            Select Case wks.Range("D" & i).Value
                Case "Responsable"
                    wks.Range("A" & i & ":E" & i).Interior.Color = RGB(200, 200, 250)
                Case "Cariste"
                    wks.Range("A" & i & ":E" & i).Interior.Color = RGB(180, 180, 230)
                Case "Magasinier"
                    wks.Range("A" & i & ":E" & i).Interior.Color = RGB(160, 160, 210)
            End Select
        End Select
    End Select
Next i

' ===== libération =====
Set wks = Nothing

End Sub
    
```

Le résultat :

Nom	Prénom	Service	Fonction	Salaire Mensuel
DURAND	Paul	Administratif	Directeur	5 000.00 €
DUPOND	Chantal	Administratif	Assistante de Direction	2 000.00 €
LEFEVRE	Christine	Administratif	Secrétaire	1 500.00 €
MARTIN	Christophe	Production	Responsable	3 000.00 €
DUMAS	Didier	Production	Chef d'équipe	2 000.00 €
DURAND	Denis	Production	Chef d'équipe	2 000.00 €
ROGER	Alain	Logistique	Responsable	3 000.00 €
PINEAU	Jean-Louis	Logistique	Cariste	1 500.00 €
CARTON	Gérard	Logistique	Magasinier	1 700.00 €
COUVRAT	Christine	Production	Opérateur	1 400.00 €
FERNAND	Didier	Production	Opérateur	1 400.00 €

IV-E - Rajouter un total

Nous allons maintenant créer une ligne qui sera le total des salaires mensuels.

Le code :

```

VBA

Public Sub Total()
    ' =====
    ' Auteur      : Starec - Philippe JOCHMANS - http://starec.developpez.com
    ' Description  : Cette procédure va permettre d'ajouter une ligne comportant la somme
    '               des salaires mensuelles
    ' =====

    ' ===== déclaration =====
    Dim wks As OWC10.Spreadsheet
    Dim i As Integer

    ' ===== affectation =====
    Set wks = Me.SpreadMFC.Object

    ' ===== ajout de la ligne =====
    ' récupération de la dernière ligne +1
    i = (wks.Range("A1").End(xlDown).Row) + 1
    With wks
        .Range("A" & i & ":E" & i).Interior.Color = vbBlack
        .Range("A" & i & ":E" & i).Font.Color = vbWhite
        .Range("A" & i).Value = "Total"
        .Range("E" & i).FormulaLocal = "=Somme(E2:E" & i - 1 & ")"
        .Range("E" & i).NumberFormat = "# ##0.00 $"
    End With

    ' ===== libération =====
    Set wks = Nothing

End Sub
    
```

Le résultat :

Nom	Prénom	Service	Fonction	Salaire Mensuel
DURAND	Paul	Administratif	Directeur	5 000.00 €
DUPOND	Chantal	Administratif	Assistante de Direction	2 000.00 €
LEFEVRE	Christine	Administratif	Secrétaire	1 500.00 €
MARTIN	Christophe	Production	Responsable	3 000.00 €
DUMAS	Didier	Production	Chef d'équipe	2 000.00 €
DURAND	Denis	Production	Chef d'équipe	2 000.00 €
ROGER	Alain	Logistique	Responsable	3 000.00 €
PINEAU	Jean-Louis	Logistique	Cariste	1 500.00 €
CARTON	Gérard	Logistique	Magasinier	1 700.00 €
COUVRAT	Christine	Production	Opérateur	1 400.00 €
FERNAND	Didier	Production	Opérateur	1 400.00 €
Total				24 500.00 €

V - Conclusion

Vous pouvez réaliser des choses remarquables en utilisant les propriétés d'Excel.

- Protection des cellules.
- Formatages différents (Nom de la police, taille, etc...).
- Jouer sur les évènements (Clic sur une cellule pour récupérer la ligne) pour reformater une ligne.

Il n'y a plus qu'à vous mettre au VBA Excel pour réaliser vos superbes listes avec une mise en forme conditionnelle illimitée.

VI - TELECHARGEMENT

Vous trouverez en téléchargement une base Access. Il faut bien sûr avoir installé le composant pour que celle-ci fonctionne.

Base Access

VII - REMERCIEMENTS

Je tiens à remercier **Silkyroad** pour son article qui m'a permis de découvrir ce composant.

Jeannot45 pour sa relecture orthographique.