

De nouvelles listes pour vos formulaires Access

par Philippe JOCHMANS ([home page de Starec](#))

Date de publication : 17/02/2008

Dernière mise à jour : 17/02/2008

Le but de cet article est de vous montrer comment réaliser de nouvelles listes dans vos formulaires Access. Des zones de listes avec mise en forme conditionnelle et case à cocher, et des listes déroulantes avec mise en forme conditionnelle.

I - INTRODUCTION.....	3
II - PREREQUIS.....	3
III - UNE ZONE DE LISTE AVEC DES CASES A COCHER ET AVEC UNE MISE EN FORME CONDITIONNELLE ILLIMITE.....	3
III-A - Cahier des Charges.....	3
III-B - Préparation du terrain.....	4
III-B-1 - Le contrôle OWC.....	4
III-B-2 - Les boutons de commandes.....	4
III-C - Le code.....	5
III-C-1 - Déclaration des variables.....	5
III-C-2 - Préparation de la liste.....	5
III-C-3 - Ajout des données.....	6
III-C-4 - Finalisation de la liste.....	7
III-C-5 - Les cases à cocher.....	9
III-C-6 - Remise à zéro.....	10
III-C-7 - Récupération de la sélection.....	11
IV - ZONE DE LISTE DEROULANTE AVEC UNE MISE EN FORME CONDITIONNELLE ILLIMITE.....	12
IV-A - Cahier des Charges.....	12
IV-A-1 - Le contrôle OWC.....	12
IV-A-2 - Le bouton de commande.....	12
IV-B - Le code.....	12
IV-B-1 - Déclaration des variables.....	12
IV-B-2 - Préparation de la liste.....	12
IV-B-3 - Ajout des données et finalisation.....	13
IV-B-4 - Dérouler et réduire la liste.....	14
IV-B-5 - Récupération de la sélection.....	15
V - CONCLUSION.....	16
VI - TELECHARGEMENT.....	16
VII - REMERCIEMENTS.....	17

I - INTRODUCTION

Cet article a pour but de vous montrer comment réaliser une zone de liste avec des cases à cocher et un formatage particulier, ainsi qu'une liste déroulante avec formatage. Nous allons donc répondre à deux questions qui reviennent régulièrement dans le forum Access.

- Comment mettre des cases à cocher dans une zone de liste.
- Comment formater les lignes d'une zone de liste ou d'une liste déroulante.

En effet le contrôle zone de liste d'Access ne permet pas de réaliser les deux fonctions demandées, et les zones de listes déroulantes ne permettent pas le formatage conditionnel.

 *La zone de liste Ms Forms 2.0 ListBox permet l'utilisation des cases à cocher.*

Pour cela, nous allons utiliser un complément fourni gratuitement par Microsoft **Microsoft Office Web Components**.

Ce composant permet de faire 3 choses :

- des graphiques.
- des tableaux croisés dynamiques **Utilisation de l'objet OWC PivotTable**
- des feuilles de calculs Excel que nous avons commencé à aborder dans cet article **La mise en forme conditionnelle illimitée dans Access**

En fait, nous allons détourner le composant de sa fonction initiale pour lui faire faire d'autres choses, car ce composant est une feuille de calcul. Nous allons donc profiter d'une partie des propriétés héritées d'Excel.

Tous les essais et codes que vous trouverez ont été réalisés avec Access 2002.

II - PREREQUIS

Une bonne connaissance de VBA est nécessaire pour comprendre l'essentiel de cet article.

Bien que cet article soit pour une utilisation de notre composant sous Access, une bonne partie de code que vous verrez comprend du VBA Excel, une bonne connaissance est donc nécessaire pour la compréhension.

Pour l'implantation du composant et le chargement de celui-ci, je vous invite à aller voir l'article suivant : **La mise en forme conditionnelle illimitée dans Access**

III - UNE ZONE DE LISTE AVEC DES CASES A COCHER ET AVEC UNE MISE EN FORME CONDITIONNELLE ILLIMITE

III-A - Cahier des Charges

Réaliser une zone de liste représentant le personnel d'une société, cette liste devra permettre une sélection multiple (par case à cocher).

De plus, cette liste devra permettre de visualiser facilement les éléments suivants :

- Les noms devront être classés par ordre alphabétique.

- Chaque service devra avoir sa couleur.

En plus de la case à cocher, chaque sélection devra être mise en surbrillance et de couleur jaune (comme avec un surligneur).

On pourra grâce à un bouton remettre la liste dans sa position initiale.

 Vous trouverez cette zone de liste et tout son code dans le formulaire **frm_ListeCaseCocher** qui se trouve dans la base en téléchargement en bas de cet article.

III-B - Préparation du terrain

Le formulaire sera composé du composant OWC et de deux boutons de commandes.

III-B-1 - Le contrôle OWC

Nous allons paramétrer le contrôle manuellement, cependant, dans le code du formulaire, vous trouverez le code pour le faire par programmation, il est toujours utile de connaître les deux.

Pour accéder aux propriétés du contrôle, il faut faire un clic droit (menu contextuel), et sélectionner **Microsoft Office SpreadSheet xx.0**, et dans le sous-menu sélectionner **Commands et Options**.

Onglet Feuille : Dans cet onglet nous allons décocher :

- En-têtes de lignes (**DisplayRowHeadings** de l'objet Windows).
- En-têtes de colonnes (**DisplayColumnHeadings** de l'objet Windows).
- Quadrillage (**DisplayGridlines** de l'objet Windows).
- Pour figer les volets, nous allons nous positionner sur la cellule A2, et cliquer sur l'icône Figurer les volets.

Onglet Classeur : Dans cet onglet nous allons décocher :

- Barre de défilement horizontale (**DisplayGridlines** de l'objet Windows).
- Sélecteur de feuille (**DisplayWorkbookTabs** de l'objet Windows).
- Barres d'outils (**DisplayToolbar** de l'objet SpreadSheet).
- Nous allons supprimer les feuilles 2 et 3 pour ne garder que la feuille 1.

De plus nous nommerons le contrôle : **spreadListe**

III-B-2 - Les boutons de commandes

Nous allons donc mettre deux boutons de commandes standard sur notre formulaire.

Réinitialisation de la liste :

- Nom : **cmdReset**
- Caption : RAZ

Affichage de la sélection :

- Nom : **cmdAffichageSelection**
- Caption : Affichage de la sélection

III-C - Le code

Nous allons maintenant décortiquer tout le code qui nous permettra de réaliser la zone de liste.

III-C-1 - Déclaration des variables

Nous allons déclarer deux variables en entête du module de formulaire (elles seront donc accessibles partout dans celui-ci).

- **Public wksOwc As OWC10.Spreadsheet** : Cette variable va représenter le composant feuille de calcul du composant OWC.
- **Public wksSheet As OWC10.Worksheet** : Cette variable va représenter la feuille en elle-même.

Les affectations de ces deux variables se feront dans l'évènement **Load** de notre formulaire avant l'appel des routines qui permettront la réalisation de notre zone de liste.

```

VBA

Private Sub Form_Load()
' =====
' Procedure      : Module :Form_frm_ListeCaseCocher // Procédure : Form_Load
' DateTime       : 12/02/2008 19:00
' Auteur         : Starec - Philippe JOCHMANS - http://starec.developpez.com
' Description    : Au chargement on initilase wksOwc, et on lance les routines au fur et
'                  à mesure
' =====

' ===== affectation =====
Set wksOwc = Me.spreadListe.Object
Set wksSheet = wksOwc.Worksheets("Feuille1")

' ===== formatage général du composant =====
Call PrepaOWC

Call AjoutDonnees

Call FinalisationListe

End Sub

```

 *Le fait de déclarer ces variables nous permettra de bénéficier de l'Intellisense.*

III-C-2 - Préparation de la liste

Nous allons ici voir le code qui permettra de préformater la liste.

```

VBA

Public Sub PrepaOWC()
' =====
' Procedure      : Module :Form_frm_ListeCaseCocher // Procédure : PrepaOWC
' DateTime       : 12/02/2008 18:59
' Auteur         : Starec - Philippe JOCHMANS - http://starec.developpez.com
' Description    : Cette routine va permettre de préparer le composant en enlevant ce
'                  qui ne nous convient pas, et on prépare l'entête
' =====

```

VBA

```

' ===== préparation de l'aspect =====
With wksOwc
    .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
        .DisplayGridlines = False         ' désactivation du quadrillage
    End With
    .Worksheets("Feuille2").Delete        ' on supprime la feuille 2
    .Worksheets("Feuille3").Delete        ' on supprime la feuille 3
    .Windows(1).ViewableRange = "A1:G200000"
End With

' ===== on affecte les noms aux entêtes de colonnes =====
With wksSheet
    .Range("A1").Value = "Choix"
    .Range("B1").Value = "ID"
    .Range("C1").Value = "Civilité"
    .Range("D1").Value = "Nom"
    .Range("E1").Value = "Prénom"
    .Range("F1").Value = "Service"
    .Range("G1").Value = "Fonction"
End With

' ===== formatage de l'entête =====
With wksOwc.Range("A1:G1")
    .Interior.Color = RGB(220, 200, 250) ' on colorie l'entête en violet
    .Borders.LineStyle = xlContinuous    ' on encadre les cellules
    .Font.Bold = True                    ' on met le texte en gras
End With

End Sub

```

Ce code nous permet donc de :

- Préparer l'aspect de notre liste (barres, etc.) par le code.
- Mettre les entêtes de colonnes
- Formater cette entête (couleur, police, cadrage).

 *Nous avons figé les volets manuellement, ce qui permettra de laisser toujours notre entête visible.*

III-C-3 - Ajout des données

Nous allons maintenant ajouter les données à notre liste, pour cela nous allons utiliser la technologie **DAO**, et réaliser le remplissage de notre feuille.

 *Vous trouverez les données dans la base en téléchargement.*

VBA

```

Public Sub AjoutDonnees()
' =====
' Procedure      : Module :Form_frm_ListeCaseCocher // Procédure : AjoutDonnees
' DateTime       : 12/02/2008 20:01
' Auteur        : Starec - Philippe JOCHMANS - http://starec.developpez.com
' Description    : Cette procédure va permettre d'ajouter les données
' =====

```

VBA

```

' ===== déclaration =====
Dim rst As DAO.Recordset
Dim strSql As String
Dim i As Long

' ===== affectation =====

strSql = "SELECT tbl_Personnel.IdPersonnel, tbl_Civilite.strCivAbrege, tbl_Personnel.strNom, tbl_Personnel.strPr
-
      " tbl_Service.strService, tbl_Fonction.strFonction, tbl_Service.lngCouleur " & _

"FROM tbl_Service INNER JOIN (tbl_Fonction INNER JOIN (tbl_Civilite INNER JOIN tbl_Personnel " & _
      "ON tbl_Civilite.IdCivilite = tbl_Personnel.IdCivilite) " & _
      "ON tbl_Fonction.IdFonction = tbl_Personnel.IdFonction) " & _
      "ON tbl_Service.IdService = tbl_Personnel.IdService " & _
      "ORDER BY tbl_Personnel.strNom;"

Set rst = CurrentDb.OpenRecordset(strSql)

' ===== on met les données dans la feuille =====
i = 2
While Not rst.EOF
    With wksSheet
        ' affectation des données et formatage
        With .Range("A" & i)
            .HorizontalAlignment = xlHAlignCenter
            .Font.Bold = True
        End With
        .Range("B" & i).Value = rst("IdPersonnel")
        .Range("C" & i).Value = rst("strCivAbrege")
        .Range("D" & i).Value = rst("strNom")
        .Range("E" & i).Value = rst("strPrenom")
        .Range("F" & i).Value = rst("strService")
        .Range("G" & i).Value = rst("strFonction")
        ' on affecte à la ligne la couleur du service
        .Range("A" & i & ":G" & i).Interior.Color = rst("lngCouleur")
        ' on entoure les cellules
        .Range("A" & i & ":G" & i).Borders.LineStyle = xlContinuous
    End With
    i = i + 1
    rst.MoveNext
Wend

' ===== libération =====
rst.Close
Set rst = Nothing

End Sub

```

Des petits commentaires s'imposent :

- La requête SQL nous permet de récupérer toutes nos données dans les tables composants la base, vous noterez que nous récupérons le champ **lngCouleur** de la table **tbl_Service**. En effet, chaque service ayant sa propre couleur.
- Ensuite grâce à une boucle **While .. Wend**, nous remplissons notre grille, et vous noterez donc que nos dernières lignes intègrent la couleur, et réalisent le quadrillage.

III-C-4 - Finalisation de la liste

Nous allons maintenant finaliser notre liste.

VBA

```

Public Sub FinalisationListe()
' =====
' Procedure      : Module :Form_frm_ListeCaseCocher // Procédure : FinalisationListe

```

```

VBA
' DateTime      : 13/02/2008 05:21
' Auteur        : Starec - Philippe JOCHMANS - http://starec.developpez.com
' Description    : Le but de cette routine est de finaliser la liste
' =====

Dim lngDerRow As Long
lngDerRow = wksSheet.Range("B1").End(xlDown).Row

' ===== enlever les lignes et les colonnes non remplies =====
wksOwc.Windows(1).ViewableRange = "A1:G" & lngDerRow

' ===== réduire la colonne Id pour que celle-ci ne soit pas visible =====
wksSheet.Range("B:B").ColumnWidth = 0

' ===== empêcher l'utilisateur de saisir dans la grille =====
wksSheet.Protectio.Enabled = True

End Sub

```

Ce code permet donc de (dans l'ordre) :

- Récupérer la dernière ligne.
- Enlever les lignes et les colonnes non remplies pour ne permettre que l'affichage des lignes nécessaires.
- Mettre la largeur de la colonne B à 0 (pour cacher le champ ID qui est la clé).
- Protéger la feuille pour éviter toute saisie.

Voici maintenant l'aspect final de notre liste.

avec des case à cocher.

Statut	Nom	Prénom	Service	Fonction
<input type="checkbox"/>	CAMBOUS	Denis	Logistique	Responsable
<input type="checkbox"/>	COURI	Alain	Production	Magasinier
<input type="checkbox"/>	DUM	Didier	Ressources Humaines	Responsable
<input type="checkbox"/>	DUPOND	Brigitte	Administratif	Secrétaire
<input type="checkbox"/>	DURAND	Marcel	Administratif	Directeur
<input type="checkbox"/>	ENGRAND	Christine	Logistique	Secrétaire
<input type="checkbox"/>	GARN	Charlotte	Production	Chef d'équipe
<input type="checkbox"/>	GENDRE	René	Production	Conducteur de Ma
<input type="checkbox"/>	LEFEVRE	Denise	Production	Secrétaire
<input type="checkbox"/>	MOARI	Geneviève	Production	Conducteur de Ma

RAZ

Affichage de la sélection

 La largeur finale de la liste a été faite manuellement.

III-C-5 - Les cases à cocher

Le but de notre liste (en plus de la mise en forme conditionnelle) est de nous permettre de faire des sélections qui doivent être représentées par des cases à cocher et les lignes doivent être mises en surbrillance (comme avec un surligneur).

Pour cela nous allons agir sur l'évènement Click du composant.

Voici donc le code qui va nous permettre de réaliser cette opération.

```
VBA
Private Sub spreadListe_Click()
' =====
' Procedure      : Module :Form_frm_ListeCaseCocher // Procédure : spreadListe_Click
' DateTime      : 13/02/2008 05:31
' Auteur        : Starec - Philippe JOCHMANS - http://starec.developpez.com
' Description    : Nous allons utiliser l'évènement click pour mettre la ligne en évidence
' =====

' ===== déclaration =====
Dim lngRow As Long ' pour stocker la ligne sélectionnée

' ===== affectation =====
lngRow = wksOwc.ActiveCell.Row

' ===== réalisation de la ligne sélectionnée sur le click =====
If wksSheet.Range("A" & lngRow).Value = "x" Then
' on a déjà coché la case, donc on reprend les formatages d'origine
wksSheet.Protection.Enabled = False
With wksSheet
.Range("A" & lngRow).Value = ""
With .Range("A" & lngRow & ":G" & lngRow)
.Interior.Color = CLng(DLookup("lngCouleur", "tbl_Service", "strService='" &
wksSheet.Range("F" & lngRow).Value & "'"))
End With
End With
wksSheet.Protection.Enabled = True
Else
' on met en évidence la ligne et on coche la case
wksSheet.Protection.Enabled = False
With wksSheet
.Range("A" & lngRow).Value = "x"
With .Range("A" & lngRow & ":G" & lngRow)
.Interior.Color = vbYellow
End With
End With
wksSheet.Protection.Enabled = True
End If
End Sub
```

Quelques commentaires sur le principe :

Nous allons donc dans la colonne A mettre la lettre x qui représentera une croix, donc de visualiser que la case est cochée.

Nous détectons si la cellule contient cette lettre x.

Si elle contient cette lettre :

- On enlève le x et on met une chaîne vide.
- On remet la couleur du service en cherchant dans la table **tbl_Service** la couleur correspondante à l'aide de la fonction de domaine **DLookup**

Si la cellule ne contient pas de x

- Nous mettons un x dans la cellule A de la ligne
- Nous mettons la couleur de fond de toute la ligne en jaune.

 Avant de réaliser les opérations de modification, nous déprotégeons la feuille, et ensuite nous la reprotégeons.

Voici donc le résultat visuel avec des lignes cochées.

avec des case à cocher.

Statut	Nom	Prénom	Service	Fonction
	CAMBOUS	Denis	Logistique	Responsable
	COURI	Alain	Production	Magasinier
X	DUM	Didier	Ressources Humaines	Responsable
X	DUPOND	Brigitte	Administratif	Secrétaire
X	DURAND	Marcel	Administratif	Directeur
X	ENGRAND	Christine	Logistique	Secrétaire
	GARN	Charlotte	Production	Chef d'équipe
	GENDRE	René	Production	Conducteur de Ma
X	LEFEVRE	Denise	Production	Secrétaire
	MOARI	Geneviève	Production	Conducteur de Ma

RAZ

Affichage de la sélection

III-C-6 - Remise à zéro

Nous allons maintenant créer un code qui nous permettra de ré initialiser notre liste.

Le code est le suivant :

VBA

```
Private Sub cmdReset_Click()
' =====
' Procédure      : Module :Form_frm_ListeCaseCocher // Procédure : cmdReset_Click
' DateTime      : 13/02/2008 19:33
' Auteur        : Starec - Philippe JOCHMANS - http://starec.developpez.com
' Description    : Le click sur ce bouton va réinitialiser la liste
' =====

' ===== déclaration =====
Dim lngDerRow As Long ' pour stocker la dernière ligne
Dim i As Long ' pour boucler sur les lignes

' ===== affectation =====
lngDerRow = wksSheet.Range("B1").End(xlDown).Row
```

VBA

```

' ===== réinitialisation des lignes =====
' on enlève la protection

wksSheet.Protection.Enabled = False
For i = 2 To lngDerRow
    With wksSheet
        .Range("A" & i).Value = ""
        With .Range("A" & i & ":G" & i)
            .Font.Color = vbBlack
            .Interior.Color = CLng(DLookup("lngCouleur", "tbl_Service", "strService='" &
wksSheet.Range("F" & i).Value & "'"))
        End With
        DoEvents
    End With
Next i
' on remet la protection
wksSheet.Protection.Enabled = True

End Sub
    
```

Nous allons tout simplement parcourir la liste et remettre la couleur en fonction du service grâce à la fonction de domaine **DLookup**, et remplacer la croix par une chaîne vide.

III-C-7 - Récupération de la sélection

Maintenant que nous avons coché nos lignes, il nous faut les récupérer, cela sera réalisé avec le code suivant.

VBA

```

Private Sub cmdAffichageSelection_Click()
' =====
' Procedure      : Module :Form_frm_ListeCaseCocher // Procédure : cmdAffichageSelection_Click
' DateTime       : 13/02/2008 19:49
' Auteur         : Starec - Philippe JOCHMANS - http://starec.developpez.com
' Description    : Va permettre d'afficher les lignes sélectionnées
' =====

' ===== déclaration =====
Dim lngDerRow As Long      ' pour récupérer la dernière ligne
Dim i As Long             ' pour réaliser la boucle sur l'ensemble des lignes
Dim strMessage As String  ' préparation du message qui donnera les lignes sélectionnés

' ===== affectation =====
lngDerRow = wksSheet.Range("B1").End(xlDown).Row

' ===== boucle sur les lignes =====
For i = 2 To lngDerRow
    If wksSheet.Range("A" & i).Value = "x" Then
        strMessage = "Identifiant : " & wksSheet.Range("B" & i).Value & _
            " Nom : " & wksSheet.Range("D" & i).Value & vbCrLf & strMessage
    End If
Next i

' ===== affichage des lignes sélectionnées =====
MsgBox "Les lignes sélectionnées" & vbCrLf & vbCrLf & strMessage, vbOKOnly

End Sub
    
```

Nous allons juste parcourir les lignes, détecter la présence du x, stocker les données dans une chaîne et les afficher dans une msgbox (bien sûr en fonctionnement normal, on affiche pas de boîtes de dialogue).

IV - ZONE DE LISTE DEROULANTE AVEC UNE MISE EN FORME CONDITIONNELLE ILLIMITE

Nous allons voir comment, sur le même principe, réaliser une liste déroulante (Combobox) qui possède bien sûr une mise en forme conditionnelle.

IV-A - Cahier des Charges

Réaliser une liste déroulante représentant le personnel d'une société.

Cette liste devra permettre de visualiser facilement les éléments suivants (Nom, Prénom).

- Les noms devront être classés par ordre alphabétique.
- Chaque service devra avoir sa couleur.

 Vous trouverez cette liste déroulante et tout son code dans le formulaire **frm_LiseDeroulante** qui se trouve dans la base en téléchargement en bas de cet article.

IV-A-1 - Le contrôle OWC

Nous allons préparer ce contrôle de la même manière que la zone de liste, ce contrôle sera nommé **spreadCombo**.

IV-A-2 - Le bouton de commande

Nous allons donc mettre un bouton de commande standard sur notre formulaire, celui-ci simulera la flèche d'une liste déroulante.

Nous nommerons notre bouton **cmdDropDownListe**.

Pour avoir une flèche vers le bas, nous allons tout simplement utiliser la police **Wingdings 3** et la lettre **q** en minuscule.

 Vous avez d'autres possibilités avec des images, etc.

IV-B - Le code

IV-B-1 - Déclaration des variables

La déclaration est identique à la zone de liste.

IV-B-2 - Préparation de la liste

Voici le code qui permet de préparer la liste déroulante, celui-ci est en tout point similaire à celui de la zone de liste.

VBA

```
Public Sub PrepaCombo ( )
' =====
' Procedure      : Module :Form_frm_LiseDeroulante // Procédure : PrepaCombo
' DateTime      : 16/02/2008 07:10
' Auteur        : Starec - Philippe JOCHMANS - http://starec.developpez.com
```

VBA

```
' Description : Ce code va permettre de préparer les entêtes de la liste
' =====

' ===== on met les entêtes =====
With wksSheet
    .Range("A1") = "Id"
    .Range("B1") = "Nom"
    .Range("C1") = "Prénom"
    With .Range("A1:C1")
        .Interior.Color = RGB(180, 180, 240)
        .Borders.LineStyle = xlContinuous
    End With
End With

End Sub
```

IV-B-3 - Ajout des données et finalisation

Ce code va ajouter et finaliser la liste déroulante. Un petit ajustement en création est nécessaire pour finaliser la largeur des colonnes sans passer par le code.

VBA

```
Public Sub RemplirCombo()
' =====
' Procedure : Module :Form_frm_LiseDeroulante // Procédure : RemplirCombo
' DateTime : 16/02/2008 07:19
' Auteur : Starec - Philippe JOCHMANS - http://starec.developpez.com
' Description : Nous allons maintenant remplir notre liste
' =====

' ===== déclaration =====
Dim rst As DAO.Recordset
Dim strSql As String
Dim i As Integer
Dim lngDerRow As Long

' ===== affectation =====
strSql = "SELECT tbl_Personnel.IdPersonnel, tbl_Personnel.strNom, tbl_Personnel.strPrenom, " & _
"tbl_Service.lngCouleur " & _

"FROM tbl_Service INNER JOIN tbl_Personnel ON tbl_Service.IdService = tbl_Personnel.IdService " & _
"ORDER BY tbl_Personnel.strNom;"
Set rst = CurrentDb.OpenRecordset(strSql)

' ===== on remplit la liste =====
i = 2
While Not rst.EOF
    With wksSheet
        .Range("A" & i).Value = rst("IdPersonnel")
        .Range("B" & i).Value = rst("strNom")
        .Range("C" & i).Value = rst("strPrenom")
        With .Range("A" & i & ":C" & i)
            .Interior.Color = rst("lngCouleur")
            .Borders.LineStyle = xlContinuous
        End With
    End With
    i = i + 1
    rst.MoveNext
Wend

' ===== on finalise l'aspect de la liste =====
lngDerRow = wksSheet.Range("B1").End(xlDown).Row

' ===== enlever les lignes et les colonnes non remplies =====
wksOwc.Windows(1).ViewableRange = "A1:C" & lngDerRow
```

VBA

```

' ===== empêcher l'utilisateur de saisir dans la liste =====
wksSheet.Protection.Enabled = True

' ===== réduire la liste =====
Me.spreadCombo.Height = 500
Me.wksOwc.Windows(1).DisplayVerticalScrollBar = False

' ===== libération =====
rst.Close
Set rst = Nothing

End Sub

```

Complément.

Deux nouvelles lignes viennent d'apparaître :

VBA

```

Me.spreadCombo.Height = 500
Me.wksOwc.Windows(1).DisplayVerticalScrollBar = False

```

Ce qui nous permet de réduire la liste à deux lignes (l'entête et la première ligne) et de supprimer la barre de défilement verticale.

Ce qui nous donne :



 Si vous mettez la propriété **Height** à 0, vous verrez toujours apparaître l'entête, car la ligne est figée.

IV-B-4 - Dérouler et réduire la liste

Pour dérouler la liste, nous allons simplement mettre les lignes suivantes sur l'évènement **Click** du bouton de commande.

VBA

```

Private Sub cmdDropDownListe_Click()
' ===== agrandissement de la liste =====
Me.spreadCombo.Height = 2000
Me.wksOwc.Windows(1).DisplayVerticalScrollBar = True
End Sub

```

Pour réduire la liste, nous allons mettre les lignes suivantes sur l'évènement **MouseMove** de la zone où est située le contrôle OWC (ici, la zone détail de notre formulaire).

VBA

```
Private Sub Détail_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
' ===== réduire la liste =====
Me.spreadCombo.Height = 500
Me.wksOwc.Windows(1).DisplayVerticalScrollBar = False
End Sub
```

Nous avons maintenant une liste déroulante parfaitement opérationnelle.

IV-B-5 - Récupération de la sélection

Maintenant que notre liste déroulante est opérationnelle, nous allons voir comment lui donner les mêmes fonctions qu'une liste déroulante classique en récupérant les données, et en les affichant en entête de la liste sous notre ligne figée.

Le code suivant va nous permettre cela, il s'effectue sur l'évènement **Click** du composant.

VBA

```
Private Sub spreadCombo_Click()
' =====
' Procédure      : Module :Form_frm_LiseDeroulante // Procédure : spreadCombo_Click
' DateTime      : 16/02/2008 08:02
' Auteur        : Starec - Philippe JOCHMANS - http://starec.developpez.com
' Description    : Sur la sélection dans la liste, va remonter celui les données
' =====

' ===== déclaration =====
Dim lngRowSelect As Long           ' récupération de la ligne sélectionnée
Dim idSelect As Long              ' on récupère l'Id de la sélection
Dim strNomSelect As String        ' on récupère le nom de la sélection
Dim strPrenomSelect As String     ' on récupère le prénom de la sélection
Dim lngColSelect As Long         ' on récupère la couleur de la sélection

' ===== récupération de la ligne et des données de celle-ci =====
lngRowSelect = wksOwc.ActiveCell.Row
idSelect = wksSheet.Range("A" & lngRowSelect).Value
strNomSelect = wksSheet.Range("B" & lngRowSelect).Value
strPrenomSelect = wksSheet.Range("C" & lngRowSelect).Value
lngColSelect = wksSheet.Range("A" & lngRowSelect).Interior.Color

' ===== on déprotège la feuille =====
wksSheet.Protection.Enabled = False

' ===== on supprime la ligne sélectionnée =====
wksSheet.Range("A" & lngRowSelect).EntireRow.Delete

' ===== on insère une ligne =====
wksSheet.Range("A2").Select
wksSheet.Range("A2").EntireRow.Insert (xlShiftDown)

' ===== on remet les données =====
With wksSheet
    .Range("A2").Value = idSelect
    .Range("B2").Value = strNomSelect
    .Range("C2").Value = strPrenomSelect
    With .Range("A2:C2")
        .Interior.Color = lngColSelect
        .Borders.LineStyle = xlContinuous
    End With
End With
```

VBA

```

' ===== on remet la protection =====
wksSheet.Protection.Enabled = True

' ===== on réduit la liste =====
Me.spreadCombo.Height = 500
Me.wksOwc.Windows(1).DisplayVerticalScrollBar = False

End Sub

```

Le code est entièrement commenté, et ne comporte pas de difficultés particulières.

Nous avons maintenant une liste déroulante qui se comporte correctement, il suffit de récupérer la valeur des variables pour lancer les procédures.

Il manque cependant une chose, c'est faire revenir la liste dans son état initial lorsqu'on la déroule à nouveau, pour cela il suffit de rajouter deux lignes à notre bouton de commande qui déroule la liste.

Le code de ce bouton devient donc :

VBA

```

Private Sub cmdDropDownListe_Click()

' ===== mise à jour de la liste =====
wksSheet.Protection.Enabled = False
Call RemplirCombo

' ===== agrandissement de la liste =====
Me.spreadCombo.Height = 2000
Me.wksOwc.Windows(1).DisplayVerticalScrollBar = True

End Sub

```

Je vous invite à tester la base en téléchargement pour voir l'effet visuel.

 *Lorsque vous avez cliqué sur le bouton pour déployer la liste, il faut vous positionner tout de suite sur la liste, car si vous allez sur le formulaire, la liste se rétracte.*

V - CONCLUSION

Nous venons de voir comment réaliser de nouvelles zones de liste et listes déroulantes qui offrent plus de possibilités (graphiquement) que les contrôles fournis avec Access.

Le seul regret que j'ai, c'est que le composant ne permet pas l'ajout d'images comme sur une feuille Excel, car à la place des x on aurait pu afficher de belles coches, et que le composant ne possède pas de méthodes pour ajouter des commentaires, comme avec la méthode **Add.Comment** d'Excel.

N'oublions pas que c'est un composant Web que nous détournons de son usage initial.

Vous pouvez bien sûr décliner cet exemple à l'infini en jouant sur la couleur de la police, la fonte, l'italique, le gras et d'autres paramètres.

Tout ce que vous avez vu ici est entièrement applicable sur les UserForm Excel et Word.

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 remercie **Jeannot45** pour sa relecture.

Et enfin toute l'équipe de **developpez.com** pour ce remarquable site.

