

# Utilisation de l'objet OWC PivotTable dans un UserForm Excel

par [SilkyRoad \(silkyroad.developpez.com\)](http://silkyroad.developpez.com)

Date de publication : 20/06/2007

Dernière mise à jour :

Cet article présente l'utilisation du contrôle OWC PivotTable dans un UserForm.

Les exemples ont été testés avec:

\* Excel2007, WinXP et OWC11.

\* Excel2002, WinXP et OWC10.

- I - Introduction
- II - Description générale
  - II-A - La fenêtre 'Liste des champs'
  - II-B - Les zones du PivotTable
  - II-C - La barre d'outils
  - II-D - Les connexions
    - II-D-1 - Les fenêtres de connexion
    - II-D-2 - Les connexions manuelles
    - II-D-3 - Les connexions automatisées
- III - Les événements
  - III-A - BeforeContextMenu
  - III-B - Click
  - III-C - CommandExecute
- IV - Les procédures du classeur démo
- V - Téléchargement

## I - Introduction

Le complément Microsoft Office Web Components (OWC ou Composants Web) est une collection de contrôles permettant de publier sur le Web des feuilles de calcul, des graphiques, des bases de données et des tableaux croisés dynamiques.

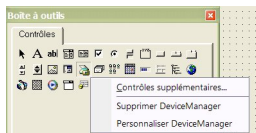
Ces contrôles peuvent aussi être utilisés dans les applications bureautiques Office, et notamment dans **les UserForms**.

Si le complément Microsoft Office Web Components n'est pas disponible sur le poste de travail, téléchargez les composants en fonction de votre version d'Office.

### Complément d'Office 2003 : composants Web Office OWC11

### Office XP Tool: Web Components OWC10

Ensuite, faites un clic droit dans votre UserForm et sélectionnez l'option **Contrôles supplémentaires**.



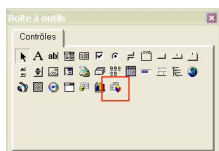
Cochez la ligne **Microsoft Office PivotTable xx.x** et cliquez sur le bouton **OK** pour valider.

xx.x dépend de votre version d'Office:

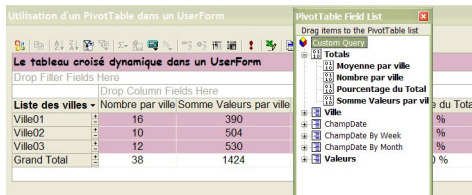
11.0 (OWC11) pour Office 2003 et Office 2007.

10.0 (OWC10) pour Office 2002.

Le contrôle PivotTable est désormais disponible dans la boîte à outils.



La démo présentée dans cet article utilise le contrôle **PivotTable** afin de créer et manipuler un tableau croisé dynamique dans un UserForm.



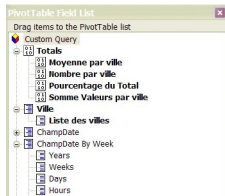
Tout comme un classique TCD de l'application Excel, cet outil permet d'effectuer des calculs (somme, nombre, nb, moyenne, produit, max, min, ecarttype, var) et d'analyser de façon Dynamique vos sources de données.

Le contrôle OWC PivotTable ne remplace pas les tableaux croisés dynamiques de la feuille, mais peut être un support complémentaire si vous souhaitez effectuer des calculs directement depuis un UserForm.

Dans cet exemple, les données sont stockées dans la Feuil2 du même classeur, mais vous pouvez facilement l'adapter et utiliser une source externe (table Access, fichier texte, feuille d'un classeur Excel fermé ...)

## II - Description générale

### II-A - La fenêtre 'Liste des champs'



La fenêtre contient tous les champs qui peuvent être utilisés dans les zones du PivotTable.

Il existe trois types de champs:

- \* Les jeux de champs
- \* Les champs
- \* Les totaux

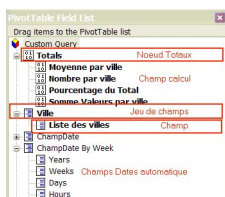
Les **jeux de champs** sont les éléments supérieurs des noeuds extensibles.

Les **Champs** sont les éléments inférieurs des jeux de champs.

Le noeud **Totaux** (Totals) regroupe les champs de calcul automatiques (Somme, Moyenne, Min, Max, Variance...) et les champs de calcul personnalisés.

Les jeux de champs et les champs sont issus de la requête effectuée dans la base de données.

Pour les données de type Date, le jeu de champs contient automatiquement des champs spécifiques (année, trimestre, mois, semaine, jour...) pour faciliter les filtres et les regroupements.



## II-B - Les zones du PivotTable



Le PivotTable est constitué de 4 zones:

- \* Les champs de ligne
- \* Les champs de colonne
- \* Les champs de filtre
- \* Les totaux et les champs de détails

Pour construire votre TCD, faites glisser/déposer les champs de la fenêtre 'Liste des champs' vers les zones du PivotTable.

## II-C - La barre d'outils



1. Bouton 'A propos de Microsoft Office Web Components'.

Affiche des informations générales sur la version, le Product ID, le copyright du contrôle OWC.

2. Permet de copier la zone sélectionnée dans le PivotTable.

3. Trie la colonne sélectionnée par ordre croissant.

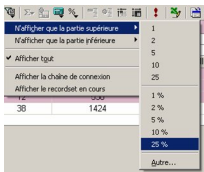
4. Trie la colonne sélectionnée par ordre décroissant.

5. Filtre automatique.

6. Affiche les éléments les plus hauts et les plus bas.

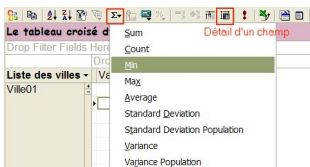
Sélectionnez une cellule du tableau croisé dynamique pour que le bouton soit actif.

Ensuite, cliquez sur le bouton et choisissez le filtre à appliquer.



7. Applique un calcul automatique.

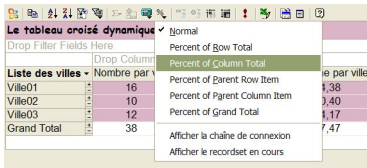
Affichez le détail d'un champ (14) et sélectionnez une cellule pour activer le bouton.



8. Sous-total.

9. Champ calculé.

10. Affiche la sélection sous forme de pourcentage.



Le tableau croisé dynamique	Normal	
Drop Filter Fields Here	Percent of Row Total	
Drop Columns	Percent of Column Total	
Liste des villes • Nombre par ville	Percent of Parent Row Item	le par ville
Ville01 16	Percent of Parent Column Item	1,38
Ville02 10	Percent of Grand Total	0,40
Ville03 12		1,17
Grand Total 38		7,47

**11.** Réduire.

**12.** Développer.

**13.** Masque les détails.

**14.** Affiche les détails.

**15.** Actualisation des données.

**16.** Export vers Excel (pas réussi à faire fonctionner cette transaction).

**17.** Commandes et options pour la mise en forme du PivotTable.

**18.** Affiche la fenêtre 'Liste des champs'.

Faites ensuite glisser les champs dans les différentes zones du PivotTable afin de créer le tableau croisé dynamique.

**19.** Ouvre la fenêtre d'aide des PivotTable. Le fichier d'aide contient de nombreuses informations au sujet de ce contrôle.

**20.** Bouton personnalisé.



Il est possible d'ajouter des boutons personnels dans la barre d'outils, par exemple:

```

Vba

' ----- Ajout et gestion d'un bouton personnalisé dans le pivotTable -----
Private Sub CommandButton5_Click()
    Dim Tb 'As ????
    Dim NouveauBouton 'As ????

    'Définit la barre d'outils du contrôle PivotTable.
    Set Tb = PivotTable1.Toolbar

    On Error Resume Next

    'Définit l'image qui va être associée au bouton.
    'Privilégiez des fichiers .ico pour la visibilité
    'Dans cet exemple, l'image est stockée dans le même répertoire que le classeur.
    Tb.ImageList.ListImages.Add , "NomImage", LoadPicture(ThisWorkbook.Path & "\bouton.ICO")

    If Err.Number > 0 Then
        MsgBox "L'image 'bouton.ICO' n'a pas pu être chargée: Vérifiez le chemin"
        Exit Sub
    End If

    On Error GoTo 0

    'Définition du nouveau bouton
    'Tb.Buttons.Count + 1 permet de placer le bouton en dernier (sur la droite)
    Set NouveauBouton = Tb.Buttons.Add(Tb.Buttons.Count + 1, "Bouton01", , , "NomImage")

    'Applique une info-bulle au nouveau bouton
    NouveauBouton.TooltipText = "Info-bulle du nouveau bouton."

    'Insère le bouton et ajoute une séparation après le dernier bouton existant.
    Tb.Buttons.Add Tb.Buttons.Count, , , 3

    PivotTable1.Refresh
    CommandButton5.Enabled = False
End Sub
    
```

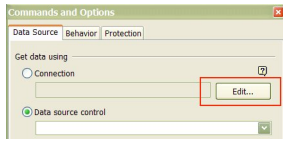
## II-D - Les connexions

### II-D-1 - Les fenêtres de connexion

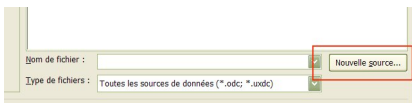
Lorsque le contrôle est inséré dans votre UserForm, cliquez sur le lien pour établir la connexion avec la base de données.



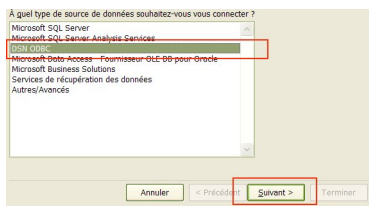
Cliquez sur le bouton de connexion (**Edit**).



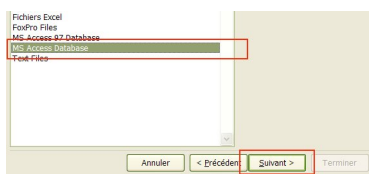
Cliquez sur le bouton **Nouvelle source**.



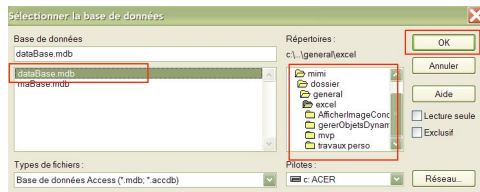
Choisissez **DSN ODBC** dans le type de source et cliquez sur le bouton **Suivant**.



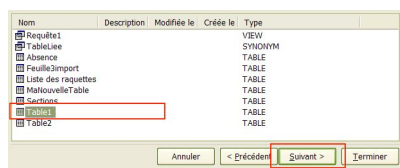
Sélectionnez le format de la base (ici **MS Access Database**) et cliquez sur le bouton **Suivant**.



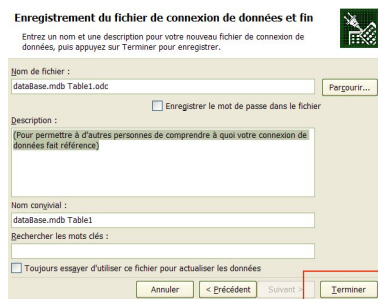
Recherchez la base Access sur votre disque dur et validez en cliquant sur le bouton **OK**.



Sélectionnez la table contenant les données sources et cliquez sur le bouton **Suivant**.



Personnalisez les informations (nom, description...) et cliquez sur le bouton **OK** pour enregistrer la connexion.



## II-D-2 - Les connexions manuelles

Pour lier le PivotTable aux données sources, vous pouvez aussi renseigner directement les propriétés **ConnectionString** et **DataMember**.

BackColor	16777215
CommandText	
ConnectionString	Provider=MSDASQL.1;Persist Security Info=True;DataSource=
ControlText	
DataMember	"C:\DOCUMENTS AND SETTINGS\mimi\dossier\mi
DataSourceName	
DisplayAlerts	True
DisplayDesignTimeUI	False
DisplayExpandIndicator	True
DisplayFieldList	False
DisplayOfficeLogo	True
DisplayPropertyToolbox	False

Exemple de connexion à une table Access:

**ConnectionString:**

```
Provider=MSDASQL.1;Persist Security Info=True;Extended Properties="DSN=MS Access Database;DBQ=C:\DOCUMENTS AND SETTINGS\mimi\dossier\maBase.mdb;DefaultDir=C:\DOCUMENTS AND SETTINGS\mimi\dossier;DriverId=25;FIL=MS Access;MaxBufferSize=2048;PageTimeout=5;";Initial Catalog=C:\DOCUMENTS AND SETTINGS\mimi\dossier\maBase.mdb
```

**DataMember:**

```
`C:\DOCUMENTS AND SETTINGS\mimi\dossier\maBase.mdb`. `Table1`
```

Vous pouvez aussi créer une requête dans la table source. Indiquez ce type de syntaxe dans la propriété **CommandText**:

```
SELECT Champ1,Champ2 FROM Table1
```

Exemple de connexion à un classeur Excel:

Il est possible de se connecter à un classeur Excel à condition que les données soient correctement structurées (la première ligne sert à indiquer le nom des champs, à partir de la première colonne et les informations ne sont pas éparpillées dans la feuille).

**ConnectionString:**

```
Provider=MSDASQL.1;Persist Security Info=True;Extended Properties="DSN=Fichiers Excel;DBQ=C:\Documents and Settings\mimi\dossier\PivotTable_UserForm_XL2002.xls;DefaultDir=C:\Documents and Settings\mimi\dossier;DriverId=790;MaxBufferSize=2048;PageTimeout=5;";Initial Catalog=C:\Documents and Settings\mimi\dossier\PivotTable_UserForm_XL2002
```

Nota: L'extension **.xls** n'est pas spécifiée dans le paramètre **Initial Catalog**.

**DataMember:**

```
`C:\Documents and Settings\mimi\dossier\PivotTable_UserForm_XL2002`. `Feuil2$`
```

Ou pour une plage nommée:

```
`C:\Documents and Settings\mimi\dossier\PivotTable_UserForm_XL2002`. `PlageNommee`
```

Nota: L'extension **.xls** n'est pas spécifiée.

#### CommandText:

```
SELECT * FROM [Feuil2$]
```

Ou pour une plage nommée:

```
SELECT * FROM [PlageNommee]
```

## II-D-3 - Les connexions automatisées

La connexion peut aussi être paramétrée par macro. Il s'agit en fait d'automatiser le renseignement des propriétés présentées dans le chapitre précédent. Cela facilitera ensuite le déplacement du classeur sans devoir réécrire manuellement le chemin dans la propriété **ConnectionString**.

Pour l'exemple qui nous intéresse (les données sources sont dans la Feuil2 du classeur), utilisez:

#### Vba

```
'Le catalogue est représenté par le chemin complet du classeur, sans l'extension (.xls).
Cible = Left(ThisWorkbook.FullName, Len(ThisWorkbook.FullName) - 4)
'Empêche l'affichage de la fenêtre gestion des champs
PivotTable1.DisplayFieldList = False

'Connexion
PivotTable1.ConnectionString = _
    "Provider=MSDASQL.1;Persist Security Info=True;Extended Properties=" & Chr(34) & _
    "DSN=Fichiers Excel;DBQ=" & ThisWorkbook.FullName & ";DefaultDir=" & ThisWorkbook.Path & _
    ";DriverId=790;MaxBufferSize=2048;PageTimeout=5;" & Chr(34) & _
    ";Initial Catalog=" & Cible

'Requête dans la feuille (Feuil2) contenant les données sources
PivotTable1.CommandText = "SELECT * FROM [Feuil2$]"
'Exemple pour filtrer les champs
'PivotTable1.CommandText = "SELECT [Feuil2$].[Ville] FROM [Feuil2$]"
```

## III - Les évènements

Le contrôle PivotTable dispose de plusieurs évènements pour interagir avec les actions de l'utilisateur.

Quelques exemples:

### **BeforeContextMenu**

Survient lors de l'utilisation du menu contextuel (clic droit) dans le PivotTable.

### **Click**

Survient lorsque l'utilisateur clique dans le PivotTable.

### **CommandExecute**

Survient après la sélection d'une commande de la barre de menus ou de la barre d'outils.

### **OnConnect**

Survient lorsque le PivotTable se connecte à la source de données.

### **PivotTableChange**

Survient lorsqu'un champ, un jeu de champs ou un total est ajouté ou supprimé du PivotTable.

### **Query**

Survient lorsque le PivotTable fait une requête dans la source de données.

### **SelectionChange**

Survient à chaque sélection dans le PivotTable.

### **ViewChange**

Survient à chaque fois que le PivotTable est redessiné.

Les sous chapitres suivants décrivent les événements utilisés dans le classeur démo.

### III-A - BeforeContextMenu

L'évènement **BeforeContextMenu** gère le menu contextuel (clic droit dans le PivotTable).

Vous pouvez:

- \* Désactiver totalement le clic droit.
- \* Ajouter des transactions personnalisées dans le menu, avec la possibilité de masquer les sous menus prédéfinis.

Vba

```
Private Sub PivotTable1_BeforeContextMenu(ByVal x As Long, ByVal y As Long, _
    ByVal Menu As OWC11.ByRef, ByVal Cancel As OWC11.ByRef)

    Dim MonMenu()
    Dim j As Integer

    '----
    'Permet de désactiver le menu contextuel
    'Cancel = True
    '----

    If CheckBox1.Value = False Then
        'Ajoute des nouveaux éléments à la suite des éléments existants, dans
        'Le menu contextuel.
        MonMenu = Menu.Value
        j = UBound(MonMenu) + 3

        ReDim Preserve MonMenu(j)

        MonMenu(j - 2) = Empty
        'Array(Description, MotClé)
        MonMenu(j - 1) = Array("Afficher la chaîne de connexion", "MenuPerso01")
        MonMenu(j) = Array("Afficher le recordset en cours", "MenuPerso02")

    Else

        'Crée un nouveau menu contextuel personnalisé
        ReDim Preserve MonMenu(1)

        'Array(Description, MotClé)
        MonMenu(0) = Array("Afficher la chaîne de connexion", "MenuPerso01")
        MonMenu(1) = Array("Afficher le recordset en cours", "MenuPerso02")

    End If

    Menu.Value = MonMenu
End Sub
```

### III-B - Click

La procédure ci dessous gère l'évènement **Click** dans le PivotTable.

La macro vérifie le type d'objet cliqué (**TypeName(oPvtSel)**) et affiche dans un Label le contenu de la cellule sélectionnée dans la zone **totaux et champs de détails**.

Vba

```
'Evènement Clic sur le TCD
'Cet exemple extrait le contenu de la cellule sélectionnée.
Private Sub PivotTable1_Click()
    Dim oPvtSel As Object
    Dim PivotAgg As PivotAggregate
    Dim strTotal As String
    Dim NomCol As String, NomLigne As String
    Dim Resultat As String

    'Définit la sélection
    Set oPvtSel = PivotTable1.Selection

    If TypeName(oPvtSel) = "PivotAggregates" Then
        Set PivotAgg = oPvtSel.Item(0)
        Resultat = PivotAgg.Value

        strTotal = PivotAgg.Total.Caption
        'nom colonne
        NomCol = PivotAgg.Cell.ColumnMember.Caption
        'nom élément ligne
        NomLigne = PivotAgg.Cell.RowMember.Caption

        'Remise en forme du résultat en fonction de la colonne cliquée
        If strTotal = "Pourcentage du Total" Then Resultat = Format(Resultat, "0.00 %")
        If strTotal = "Moyenne par ville" Then Resultat = Format(Resultat, "00.00")

        Label1.Caption = Resultat & vbCrLf & strTotal & vbCrLf & NomLigne & vbCrLf & NomCol
    End If
End Sub
```

### III-C - CommandExecute

L'évènement **CommandExecute** se déclenche après la sélection d'une commande de la barre de menus ou de la barre d'outils.

L'argument **Command** identifie la commande exécutée et permet par exemple de lancer une macro en fonction du bouton ou du menu déclenché.

Vba

```
Private Sub PivotTable1_CommandExecute(ByVal Command As Variant, ByVal Succeeded As Boolean)

    'L'argument "Command" permet de contrôler la commande exécutée.
```



Vba

```
If VarType(Command) = vbString Then
    Select Case Command
        'Déclenche la macro du bouton personnalisé.
        Case "Bouton01"
            NomMacro_01

        'Déclenche les macros associées aux nouveaux éléments du menu contextuel
        Case "MenuPerso01"
            NomMacro_02

        Case "MenuPerso02"
            NomMacro_03
    End Select
End If
End Sub
```

## IV - Les procédures du classeur démo

Les codes suivants correspondent à une utilisation dans Excel2003/2007 et OWC11.

Le tableau croisé dynamique est créé automatiquement lors du chargement du UserForm et permet de visualiser:

- \* Le nombre d'enregistrements par Ville.
- \* La somme du champ "Valeur" par ville.
- \* La moyenne du champ "Valeur" par ville.
- \* Un champ calculé "pourcentage du total pour chaque ville".
- \* L'exemple montre aussi comment:
  - Extraire des résultats particuliers en cliquant sur les cellules du PivotTable.
  - Mettre en forme les données du tableau (couleur des cellules, type de caractères, formats des valeurs ...).
  - Une option permet de créer un filtre par semaine sur le champ "ChampDate".
  - Il est ainsi possible de choisir les semaines à afficher dans le PivotTable.

Vba

```
Option Explicit
```

```
Private Sub UserForm_Initialize()
```

```
    Dim Cible As String  
    Dim oPvtView As OWC11.PivotView  
    Dim oFieldSets As OWC11.PivotFieldSets  
    Dim oPvtTotal As PivotTotal  
    Dim GrdTotal As Integer
```

```
    'Le catalogue est représenté par le chemin complet du classeur, sans l'extension (.xls).  
    Cible = Left(ThisWorkbook.FullName, Len(ThisWorkbook.FullName) - 4)  
    'Empêche l'affichage de la fenêtre gestion des champs  
    PivotTable1.DisplayFieldList = False
```

```
    'Connexion
```

```
    PivotTable1.ConnectionString = _  
        "Provider=MSDASQL.1;Persist Security Info=True;Extended Properties=" & Chr(34) & _  
        "DSN=Fichiers Excel;DBQ=" & ThisWorkbook.FullName & ";DefaultDir=" & ThisWorkbook.Path & _  
        ";DriverId=790;MaxBufferSize=2048;PageTimeout=5;" & Chr(34) & _  
        ";Initial Catalog=" & Cible
```

Vba

```
'Requête dans la feuille (Feuil2) contenant les données sources
PivotTable1.CommandText = "SELECT * FROM [Feuil2$]"
'Exemple pour filtrer les champs
'PivotTable1.CommandText = "SELECT [Feuil2$].[Ville] FROM [Feuil2$]"

Set oPvtView = PivotTable1.ActiveView
Set oFieldSets = oPvtView.FieldSets

With oPvtView
    'Ajoute le champ "Ville" dans la zone Row (ligne)
    .RowAxis.InsertFieldSet oFieldSets("Ville")
    .FieldSets("Ville").Fields(0).Caption = "Liste des villes"

    'Ajoute le champ "Valeurs" dans la zone données (Data)
    .DataAxis.InsertFieldSet oFieldSets("Valeurs")

    'Ajoute une fonction Nombre
    Set oPvtTotal = oPvtView.AddTotal("Nombre par ville", _
        oFieldSets("Valeurs").Fields(0), plFunctionCount)
    .DataAxis.InsertTotal oPvtTotal
    'Alignement centré
    oPvtTotal.HAlignment = plHAlignCenter
    'Couleur de fond
    oPvtTotal.View.TotalBackColor = RGB(220, 180, 200)

    'Ajoute une fonction Somme
    Set oPvtTotal = oPvtView.AddTotal("Somme Valeurs par ville", _
        oFieldSets("Valeurs").Fields(0), plFunctionSum)
    .DataAxis.InsertTotal oPvtTotal
    oPvtTotal.HAlignment = plHAlignCenter

    ' ---- Ajoute un champ calculé Moyenne
    Set oPvtTotal = oPvtView.AddCalculatedTotal("Moyenne", "Moyenne par ville", _
        "[Somme Valeurs par ville] / [Nombre par ville]")
    .DataAxis.InsertTotal oPvtTotal
    oPvtTotal.HAlignment = plHAlignCenter
    'application d'un format
    oPvtTotal.NumberFormat = "00.00"
    '-----

    ' ---- Ajoute un champ calculé 'Pourcentage'

    With PivotTable1.ActiveData
        'récupère le total du champ "Somme Valeurs par ville"
        GrdTotal = .Cells(.RowAxis.RowMember.TotalMember, .ColumnAxis.ColumnMember). _
            Aggregates("Somme Valeurs par ville").Value
    End With

    Set oPvtTotal = oPvtView.AddCalculatedTotal("Pourcentage", "Pourcentage du Total", _
        "([Somme Valeurs par ville] / " & GrdTotal & ")")
    .DataAxis.InsertTotal oPvtTotal
    oPvtTotal.HAlignment = plHAlignCenter
    oPvtTotal.NumberFormat = "0.00 %"

    '-----

    'Mise en forme de la barre de titre
```

Vba

```
With .TitleBar
    'La police de caractères
    .Font.Name = "comic sans ms"
    'Le titre
    .Caption = "Le tableau croisé dynamique dans un UserForm"
    'Couleur de fond
    .BackColor = RGB(220, 180, 200)
    'Couleur du texte
    .ForeColor = RGB(0, 0, 0)
End With

'Masque les détails
PivotTable1.ActiveData.HideDetails

End Sub

'Evènement Clic sur le TCD
'Cet exemple extrait le contenu de la cellule sélectionnée.
Private Sub PivotTable1_Click()
    Dim oPvtSel As Object
    Dim PivotAgg As PivotAggregate
    Dim strTotal As String
    Dim NomCol As String, NomLigne As String
    Dim Resultat As String

    'Définit la sélection
    Set oPvtSel = PivotTable1.Selection

    If TypeName(oPvtSel) = "PivotAggregates" Then
        Set PivotAgg = oPvtSel.Item(0)
        Resultat = PivotAgg.Value

        strTotal = PivotAgg.Total.Caption
        'nom colonne
        NomCol = PivotAgg.Cell.ColumnMember.Caption
        'nom élément ligne
        NomLigne = PivotAgg.Cell.RowMember.Caption

        'Remise en forme du résultat en fonction de la colonne cliquée
        If strTotal = "Pourcentage du Total" Then Resultat = Format(Resultat, "0.00 %")
        If strTotal = "Moyenne par ville" Then Resultat = Format(Resultat, "00.00")

        Label1.Caption = Resultat & vbCrLf & strTotal & vbCrLf & NomLigne & vbCrLf & NomCol
    End If
End Sub

Private Sub CommandButton1_Click()
    If CommandButton1.Caption = "Bloquer la barre d'outils" Then
        PivotTable1.Toolbar.Enabled = False
        CommandButton1.Caption = "Débloquer la barre"
    Else
        PivotTable1.Toolbar.Enabled = True
        CommandButton1.Caption = "Bloquer la barre d'outils"
    End If
End Sub
```

Vba

```
Private Sub CommandButton2_Click()  
    'Affichage fenêtre gestion des champs  
    PivotTable1.DisplayFieldList = True  
End Sub  
  
'Création d'un TCD et d'un filtre pour gérer les dates (par numéro de semaine)  
Private Sub CommandButton3_Click()  
    Dim oPvtView As OWC11.PivotView  
    Dim oFieldSets As OWC11.PivotFieldSets  
    Dim oPvtTotal As PivotTotal  
  
    'affichage de la fenêtre gestion des champs  
    PivotTable1.DisplayFieldList = True  
  
    'Requête dans la feuille (Feuil2) contenant les données sources  
    'Exemple pour filtrer les champs  
    PivotTable1.CommandText = "SELECT [Feuil2$].[Ville],[Feuil2$].[ChampDate] FROM [Feuil2$]"  
  
    Set oPvtView = PivotTable1.ActiveView  
    Set oFieldSets = oPvtView.FieldSets  
  
    With oPvtView  
        'Ajoute le champ "Ville" dans la zone Row (ligne)  
        .RowAxis.InsertFieldSet oFieldSets("Ville")  
        .FieldSets("Ville").Fields(0).Caption = "Liste des villes"  
  
        'Ajoute le champ "ChampDate by Week" dans la zone de filtre  
        .FilterAxis.InsertFieldSet oFieldSets("ChampDate by Week")  
        '.FieldSets("ChampDate by Week").Fields(0).ExcludedMembers = Array("")  
  
        'Ajoute le champ "ChampDate" dans la zone données (Data)  
        .DataAxis.InsertFieldSet oFieldSets("ChampDate")  
  
        'Ajoute une fonction Nombre  
        Set oPvtTotal = oPvtView.AddTotal("Nombre de dates par ville", _  
            oFieldSets("ChampDate").Fields(0), plFunctionCount)  
        .DataAxis.InsertTotal oPvtTotal  
        'Alignement centré  
        oPvtTotal.HAlignment = plHAlignCenter  
        'Couleur de fond  
        oPvtTotal.View.TotalBackColor = RGB(220, 180, 200)  
    End With  
  
    'Emplêche l'affichage de la fenêtre gestion des champs  
    PivotTable1.DisplayFieldList = False  
    'Masque les détails  
    PivotTable1.ActiveData.HideDetails  
End Sub  
  
'Affiche un TCD vierge en ne prenant en compte que la ville "Ville01" dans la requête.  
Private Sub CommandButton4_Click()  
    PivotTable1.CommandText = "SELECT * FROM [Feuil2$] WHERE Ville='Ville01'"  
  
    'Affiche la fenêtre de gestion des champs  
    PivotTable1.DisplayFieldList = True
```

Vba

```
End Sub

' ----- Ajout et gestion d'un bouton personnalisé dans le pivotTable -----
Private Sub CommandButton5_Click()
    Dim Tb 'As OWC11.????
    Dim NouveauBouton 'As OWC11.????

    'Définit la barre d'outils du contrôle PivotTable.
    Set Tb = PivotTable1.Toolbar

    On Error Resume Next

    'Définit l'image qui va être associée au bouton.
    'Privilégiez des fichiers .ico pour la visibilité
    'Dans cet exemple, l'image est stockée dans le même répertoire que le classeur.
    Tb.ImageList.ListImages.Add , "NomImage", LoadPicture(ThisWorkbook.Path & "\bouton.ICO")

    If Err.Number > 0 Then
        MsgBox "L'image 'bouton.ICO' n'a pas pu être chargée: Vérifiez le chemin"
        Exit Sub
    End If

    On Error GoTo 0

    'Définition du nouveau bouton
    'Tb.Buttons.Count + 1 permet de placer le bouton en dernier (sur la droite)
    Set NouveauBouton = Tb.Buttons.Add(Tb.Buttons.Count + 1, "Bouton01", , , "NomImage")

    'Applique une info-bulle au nouveau bouton
    NouveauBouton.TooltipText = "Info-bulle du nouveau bouton."

    'Insère le bouton et ajoute une séparation après le dernier bouton existant.
    Tb.Buttons.Add Tb.Buttons.Count, , , 3

    PivotTable1.Refresh
    CommandButton5.Enabled = False
End Sub

'L'événement CommandExecute se déclenche après la sélection
'd'une commande de la barre de menus ou de la barre d'outils.
Private Sub PivotTable1_CommandExecute(ByVal Command As Variant, ByVal Succeeded As Boolean)

    'L'argument "Command" permet de contrôler la commande exécutée.
    If VarType(Command) = vbString Then
        Select Case Command
            'Déclenche la macro du bouton personnalisé.
            Case "Bouton01"
                NomMacro_01

            'Déclenche les macros associées aux nouveaux éléments du menu contextuel
            Case "MenuPerso01"
                NomMacro_02

            Case "MenuPerso02"
                NomMacro_03
        End Select
    End If
End Sub

'La macro associée au nouveau bouton
```

## Vba

```
Sub NomMacro_01()  
    Dim pTable As OWC11.PivotTable  
    Dim pTableView As OWC11.PivotView  
    Dim pFieldset As OWC11.PivotFieldSet  
    Dim pField As OWC11.PivotField  
    Dim Resultat As String  
  
    'MsgBox "Essai utilisation bouton personnalisé."  
  
    Set pTable = PivotTable1  
    Set pTableView = pTable.ActiveView  
  
    'source pour boucler sur les champs:  
    'http://msdn2.microsoft.com/fr-fr/library/aa139960(office.10).aspx  
  
    'Boucle sur chaque Fieldset  
    For Each pFieldset In pTableView.FieldSets  
        Resultat = Resultat & "The '" & pFieldset.Name & _  
            "' FieldSet contains " & _  
            pFieldset.Fields.Count & " field(s):" & vbCrLf  
  
        'Boucle sur chaque Champ  
        For Each pField In pFieldset.Fields  
            Resultat = Resultat & vbTab & pField.Name & vbCrLf  
        Next  
  
        Resultat = Resultat & "----" & vbCrLf  
    Next  
  
    MsgBox Resultat, , "Liste les champs contenus dans le PivotTable."  
End Sub  
'-----  
  
'Gestion du menu contextuel (clic droit dans le PivotTable)  
Private Sub PivotTable1_BeforeContextMenu(ByVal x As Long, ByVal y As Long, _  
    ByVal Menu As OWC11.ByRef, ByVal Cancel As OWC11.ByRef)  
  
    Dim MonMenu()  
    Dim j As Integer  
  
    '-----  
    'Permet de désactiver le menu contextuel  
    'Cancel = True  
    '-----  
  
    If CheckBox1.Value = False Then  
        'Ajoute des nouveaux éléments à la suite des éléments existants, dans  
        'Le menu contextuel.  
        MonMenu = Menu.Value  
        j = UBound(MonMenu) + 3  
  
        ReDim Preserve MonMenu(j)  
  
        MonMenu(j - 2) = Empty  
        'Array(Description, MotClé)  
        MonMenu(j - 1) = Array("Afficher la chaîne de connexion", "MenuPerso01")  
        MonMenu(j) = Array("Afficher le recordset en cours", "MenuPerso02")  
  
    Else  
  
        'Crée un nouveau menu contextuel personnalisé  
        ReDim Preserve MonMenu(1)  
  
        'Array(Description, MotClé)
```

Vba

```
    MonMenu(0) = Array("Afficher la chaîne de connexion", "MenuPerso01")
    MonMenu(1) = Array("Afficher le recordset en cours", "MenuPerso02")

    End If

    Menu.Value = MonMenu
End Sub

'La macro associée au menu contextuel
Sub NomMacro_02()
    'Affiche la chaîne de connexion
    Label1.Caption = PivotTable1.ConnectionString
End Sub

Sub NomMacro_03()
    'Affiche le paramétrage du Recordset pour l'affichage en cours.
    Label1.Caption = PivotTable1.CommandText
End Sub
```



## V - Téléchargement

### Téléchargez le classeur démo

Le fichier zip contient:

Une version Excel2007/Excel2003 et OWC11

Une version Excel2002 et OWC10

