

Création de rubans personnalisés sous Microsoft Access 2007



Par Christophe WARIN 

Date de publication : 31 août 2006

Dernière mise à jour : 12 décembre 2006

Le passage de la version 2003 à la version 2007 a amené le remplacement des menus par un ruban. Communément appelée **ribbon**, cette barre horizontale propose des groupes de boutons arrangés dans des onglets. A travers ce tutoriel, vous allez remarquer qu'elle représente bien plus qu'un simple menu et qu'elle peut devenir rapidement l'élément principal de vos formulaires. Elle vous permet ainsi de réunir dans un seul et même endroit : les actions, les filtres, les tris, les recherches, etc.

Pas à pas, ce document va vous apprendre à créer votre propre ruban qui viendra s'intégrer avec harmonie dans vos développements.

I - Problématique - énoncé.....	3
II - Les outils.....	4
III - Comment programmer le ruban ?.....	4
IV - Mise en place d'un ruban personnalisé.....	4
IV.A - Chargement du ruban.....	4
IV.B - Association du ruban.....	5
IV.C - Chargement automatique.....	7
V - Structure du fichier XML.....	7
V.A - Structure de base.....	7
V.B - Ruban, onglet, groupe.....	8
V.C - Les contrôles.....	10
V.C.1 - Les boutons.....	10
V.C.1.1 - Les boutons prédéfinis.....	10
V.C.1.2 - Les boutons personnalisés.....	11
V.C.1.3 - Images.....	12
V.C.2 - Les cases à cocher.....	14
V.C.3 - Le contrôle Gallery - tableau de boutons.....	14
V.D - Les contrôles de saisies.....	16
V.D.1 - Les zones de texte.....	16
V.D.2 - Les zones de listes déroulantes.....	17
V.E - Les séparateurs.....	18
VI - Les événements.....	19
VI.A - Les contrôles d'action.....	19
VI.A.1 - Les boutons.....	19
VI.A.2 - Les cases à cocher.....	20
VI.A.3 - Le contrôle Gallery.....	21
VI.B - Les contrôles de saisie.....	22
VII - Agir dynamiquement sur le ruban.....	23
VII.A - Principe de base.....	23
VII.A.1 - Mémorisation du ruban dans un objet.....	23
VII.A.2 - Rafraîchissement du ruban.....	23
VII.B - Activer / Désactiver un élément.....	23
VII.C - Modifier la valeur d'une case à cocher.....	24
VII-D - Autres possibilités.....	25
IX - Conclusion.....	25
X - Annexe A : Liste des idMso.....	25
XI - Annexe B : Liste des fonctions CallBack.....	40

I - Problématique - énoncé

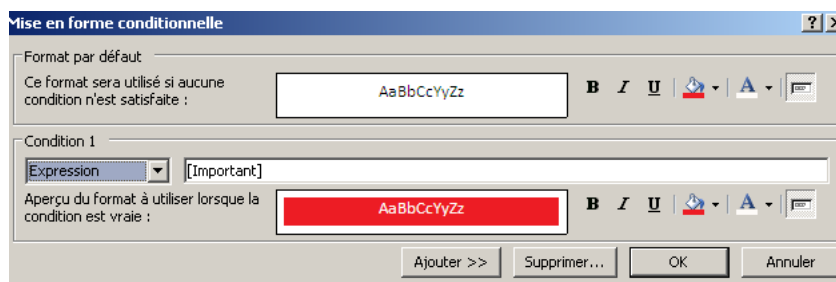
Tout au long de ce tutoriel, les différents rubans créés seront appliqués au même formulaire. Il s'agit d'un formulaire de gestion d'événements.

Il est basé sur une table tblEvenement dont la structure est la suivante :

Nom du champ	Type de données
NumEvenement	NuméroAuto
NomEvenement	Texte
DateEvenement	Date/Heure
Important	Oui/Non

Ci-dessous le formulaire :

Comme le démontre l'image ci-dessus, c'est un formulaire très simple. Il n'y a pas de boutons de commandes : les différentes actions seront exécutées à partir du ruban. La mise en évidence des événements importants est assurée par la **mise en forme conditionnelle**.



Le ruban personnalisé devra gérer :

- L'ajout, la duplication et la suppression des événements. Il doit être impossible de supprimer un événement important;
- Le filtrage des données par mois et par importance;
- La recherche d'événements.


II - Les outils

Comme vous allez le lire un peu plus loin, la création du ruban demande l'écriture de fichiers dans un langage autre que VBA. Bien qu'ils peuvent être rédigés dans n'importe quel éditeur de texte, je vous recommande particulièrement l'éditeur **Notepad++**.



D'autre part, la manipulation du ruban dans cet exemple nécessite l'ajout de deux références à votre projet :

- Microsoft Office 12 Object Library
- Microsoft Scripting Runtime

III - Comment programmer le ruban ?

En lisant le début de cette section, vous allez avoir l'impression que Microsoft a reculé plutôt que d'avancer technologiquement. Je vous rassure, il ne s'agit que d'une impression. Toutefois, je le concède, si le développement de menu via les **commandbars** était à la portée de tous, le développement de rubans se révèle beaucoup plus ardu. Pourquoi ? Tout simplement parce que la création de boutons ne se fait plus en VBA. Le ruban puise son apparence et ses fonctionnalités dans du code  **XML**.

C'est donc un code XML valide (respectant le schéma imposé par Microsoft) qui générera un ruban.

 *La programmation **XML** demande beaucoup de rigueur. Pour être lisible et donc maintenable, le code devra être correctement  **indenté**. D'autre part, contrairement à VBA, la casse des mots-clés doit être respectée.*

IV - Mise en place d'un ruban personnalisé

Avant d'aborder la conception d'un ruban personnalisé, je vous propose de découvrir cette fonctionnalité avec un petit exemple.

La mise en place d'un ruban personnalisé dans vos applications Access nécessite deux étapes :

- Chargement des propriétés du ruban dans l'application;
- Association d'un formulaire (ou d'un état) avec un onglet du ruban.

IV.A - Chargement du ruban

Comme indiqué au chapitre précédent, l'aspect du ruban est codé au sein de données XML. Ce sont ces données XML qui sont à charger dans l'application. Pour cela, l'objet **Application** propose la méthode **LoadCustomUI**. Cette méthode accepte deux arguments :

- Le nom du ruban (il sera utilisé par les formulaires Access)
- Le code XML contenant les informations relatives au ruban

Inutile de vous dire qu'il est peu judicieux d'inscrire le code XML directement dans le code VBA. Si c'est le cas, les opérations de maintenance vont s'avérer très compliquées. Une méthode consiste à écrire les balises XML dans un fichier dédié, de lire ce fichier et de passer son contenu à la méthode **LoadCustomUI**. Dans cet exemple, le fichier en question sera nommé **evenement_ribbon.XML** et sa lecture se fera via l'objet **FileSystemObjet** de la librairie **Microsoft Scripting Runtime**. Vous devez donc l'ajouter à votre projet.



Il est conseillé de donner un nom explicite au fichier. Ceci vous facilitera les corrections futures. Dans le même ordre d'idée, il est préférable de stocker le fichier XML dans le même répertoire que l'application qui l'utilise.

Créez un nouveau module : **mduRibbon**.

VBA

```
Public Function LoadRibbon()
Dim strXML As String
Dim oFso As New FileSystemObject
Dim oFtxt As TextStream
'Charge le fichier XML en mémoire
Set oFtxt = oFso.OpenTextFile(CurrentProject.Path & _
    "\evenement_ribbon.XML", ForReading)
'Récupère le contenu
strXML = oFtxt.ReadAll
'Charge le ruban personnalisé correspondant
Application.LoadCustomUI "rubanperso", strXML
End Function
```

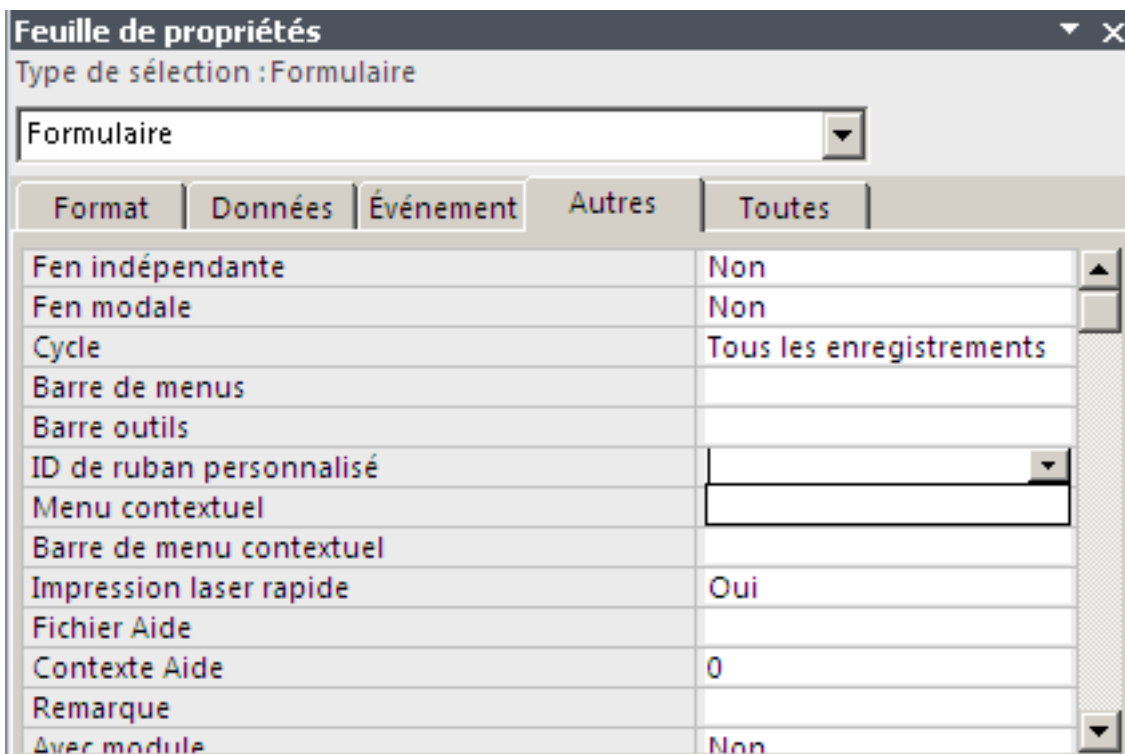
Le fichier XML à utiliser est très simple :

XML

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
  <ribbon startFromScratch="true">
    <tabs>
      <tab id="tabEvenement" label="Gestion des événements" visible="true">
        <group id="grpEnregistrement" label="Enregsitrements">
          <button id="btnEssai" label="Essai" size="large"/>
        </group>
      </tab>
    </tabs>
  </ribbon>
</customUI>
```

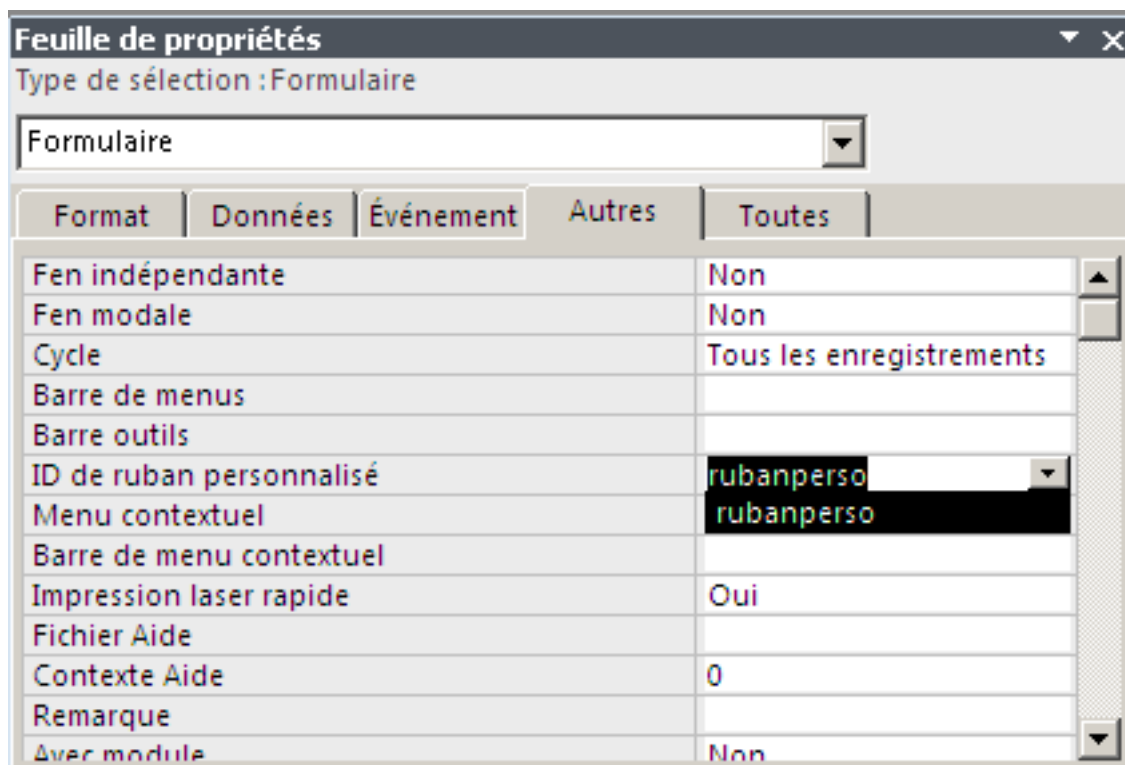
IV.B - Association du ruban

La propriété gérant le ruban se situe dans l'onglet **Autres** des propriétés du formulaire. Il s'agit d'ID de ruban personnalisé. A ce stade, aucune valeur n'est disponible :

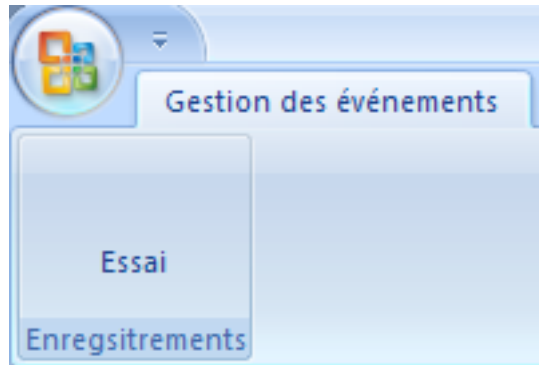


Et pour cause, dans la section précédente, nous avons écrit une fonction VBA permettant de charger le ruban. Mais, cette fonction n'ayant pas été lancée, le ruban n'est pas encore disponible. Il vous faut l'exécuter. Une technique consiste à placer le curseur sur une de ses lignes et d'actionner la touche **F5**.

A présent la zone de liste propose la valeur **rubanperso**.

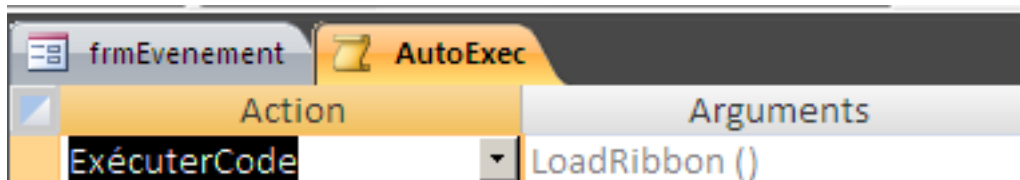


Sélectionnez-la, enregistrez et ouvrez le formulaire en mode visualisation. Le nouveau ruban apparaît.



IV.C - Chargement automatique

Un problème se pose si on ferme la base de données et la ré-ouvre. Le ruban est déchargé. En toute logique, dans ce cas, le ruban personnalisé ne pourra pas être affiché avec le formulaire. La solution : charger le ruban à chaque démarrage de la base de données. Pour cela, il est possible de créer une macro nommée **AutoExec** dont la seule et unique action sera d'exécuter la fonction **LoadRibbon()** vue précédemment.



V - Structure du fichier XML

V.A - Structure de base

Reprenons l'exemple donné plus haut pour examiner en détail la syntaxe d'un ruban:

XML

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
  <ribbon startFromScratch="true">
    <tabs>
      <tab id="tabEvenement" label="Gestion des événements" visible="true">
        <group id="grpEnregistrement" label="Enregistrements">
          <button id="btnEssai" label="Essai" size="large"/>
        </group>
      </tab>
    </tabs>
  </ribbon>
</customUI>
```

Dans ce fichier XML, la balise **ribbon** définit un nouveau ruban personnalisé. La balise **tabs** représente la liste des onglets (par exemple : *Accueil, Créer, Données Externes*). Etant donné qu'il s'agit d'une liste, on retrouve la configuration suivante :

XML

```
<tabs>
  <tab>
</tab>
  <tab>
</tab>
</tabs>
```

Où chaque élément **tab** représente un onglet. Viennent ensuite les groupes (**group**) de contrôles.



Et enfin, chaque groupe est composé de contrôles (**control**). Ici, le groupe **Enregistrements** possède un seul contrôle : un bouton.

Vous remarquerez que toutes les balises sont incluses dans la racine immuable :

XML

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
</customUI>
```



La plupart des balises possèdent un attribut **id**. Il permet d'identifier le composant du ruban de manière unique. Les doublons d'id sont interdits.



Attention : il ne faut pas confondre la balise **group** avec la balise **groups**. **groups** représente la **collection** de groupes, **group** correspond à **un** groupe. Il en va de même pour toutes les collections de noeuds : leur nom est toujours pluriel.

V.B - Ruban, onglet, groupe

Pour répondre à la problématique énoncée au début de ce tutoriel, le ruban va proposer un onglet (Gestion des événements) ainsi qu'un premier groupe de contrôles (Données).

Le code du fichier XML respecte logiquement la hiérarchie des différents éléments.

Etape 1 : balise CustomUI

A ne pas oublier. Pour que le fichier XML soit valide, la balise racine doit être :

XML

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
</customUI>
```

Etape 2 : définition du ruban personnalisé

XML

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
  <ribbon>

  </ribbon>
</customUI>
```

Etape 3 : avant d'ajouter l'onglet, il faut impérativement créer la liste d'onglets.

XML

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
  <ribbon>
    <tabs>

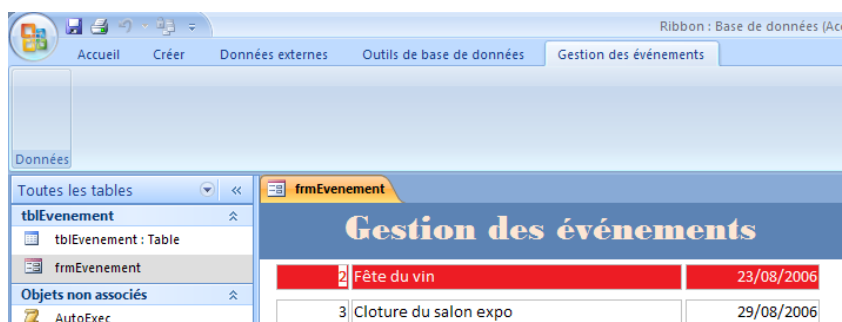
    </tabs>
  </ribbon>
</customUI>
```

Etape 4 : création de l'onglet **Gestion des événements**

Les attributs utilisés sont :

- **id** : nom de l'onglet
- **label** : titre qui sera affiché
- **visible** : l'onglet est visible

Résultat :

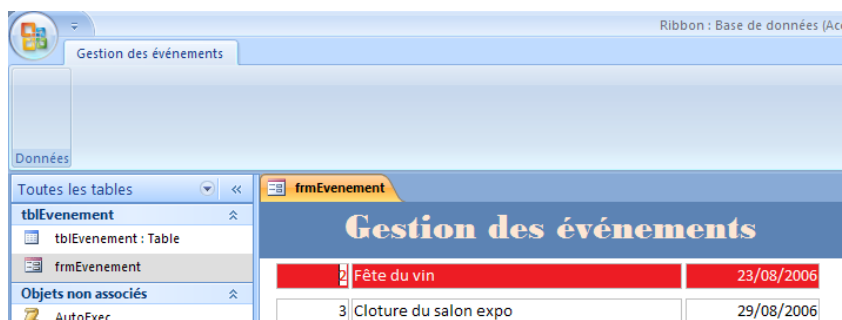


Le ruban personnalisé vient s'ajouter au ruban d'Access. Pour masquer les autres onglets prédéfinis, il faut utiliser l'attribut **startFromScratch** de la balise **ribbon**.

XML

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
  <ribbon startFromScratch="true">
    <tabs>
      <tab id="tabEvenement" label="Gestion des événements" visible="true">
        <group id="grpDonnees" label="Données">

        </group>
      </tab>
    </tabs>
  </ribbon>
</customUI>
```



V.C - Les contrôles

V.C.1 - Les boutons

Éléments de base, les boutons permettent de lancer une action en réponse à leur clic. Dans le XML, ils sont définis par la balise **<button>**. Il existe deux types de boutons :

- Les boutons prédéfinis
- Les boutons personnalisés

Quelque soit le type du bouton, son attribut **size** permet de définir sa taille. Les choix possibles sont **large** et **normal**.

V.C.1.1 - Les boutons prédéfinis

Lors de la création de barres de menu dans les versions antérieures, il était possible de reprendre à son compte des boutons fournis par Access (par exemple les boutons copier, couper, coller).

Avec le ruban, cette fonctionnalité est conservée. Pour insérer un bouton prédéfini, il faut inclure son identifiant dans la balise **<button>** via l'attribut **idMso**.

Par exemple, pour le bouton **Coller** :

XML

```
<button idMso=Paste>
</button>
```

L'identifiant **Mso** correspond au nom anglais du bouton de commande. Vous pouvez consulter la [liste des identifiants Mso](#) en annexe pour obtenir la plupart des commandes.

Dans l'application qui nous intéresse, il est possible de confier des tâches demandées aux boutons prédéfinis :

- Ajouter un nouvel enregistrement (**idMso=GoToNewRecord**)
- Supprimer un enregistrement (**idMso=RecordsDeleteRecord**)

Le code XML du groupe **grpDonnees** sera donc :

XML

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
  <ribbon startFromScratch="true">
    <tabs>
      <tab id="tabEvenement" label="Gestion des événements" visible="true">
        <group id="grpDonnees" label="Données">
          <button idMso="GoToNewRecord" size="normal"/>
          <button idMso="RecordsDeleteRecord" size="normal"/>
        </group>
      </tab>
    </tabs>
  </ribbon>
</customUI>
```

Résultat :



Ce n'est pas parce que l'on utilise un bouton prédéfini que l'on ne peut pas le personnaliser un minimum. Il est par exemple possible de modifier son intitulé afin de le rendre plus explicite.

XML

```
<group id="grpDonnees" label="Données">
  <button idMso="GoToNewRecord" label="Nouvel événement" size="normal"/>
  <button idMso="RecordsDeleteRecord" label="Supprimer l'événement" size="normal"/>
</group>
```



V.C.1.2 - Les boutons personnalisés

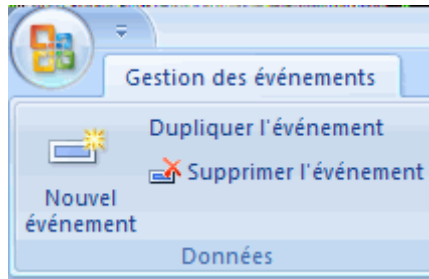
Evidemment, le développeur ne peut pas se limiter aux actions offertes par les boutons inclus dans Access. La possibilité de créer ses propres boutons est donc primordiale. Pour créer un bouton personnalisé, il suffit de ne pas spécifier d'identifiant **idMso** et de le remplacer par un identifiant simple via l'attribut **id**.

XML

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
  <ribbon startFromScratch="true">
    <tabs>
      <tab id="tabEvenement" label="Gestion des événements" visible="true">
        <group id="grpDonnees" label="Données">

          <button idMso="GoToNewRecord" label="Nouvel événement" size="large"/>
          <button id="btnDupliquer" label="Dupliquer l'événement" size="normal"/>
          <button idMso="RecordsDeleteRecord" label="Supprimer l'événement" size="normal"/>
        </group>
      </tab>
    </tabs>
  </ribbon>
</customUI>
```

Résultat :



A ce stade, le bouton n'est lié à aucune procédure. Un clic dessus n'aura aucun effet. La **gestion des événements** est traitée ultérieurement dans ce tutorial.

V.C.1.3 - Images

Dans un bouton personnalisé, aucune image n'est affectée au bouton. Comme en témoigne l'aperçu d'écran ci-dessus, le rendu graphique est assez disgracieux. Bien entendu, il est possible d'y remédier.

L'insertion d'image dans un bouton peut se faire de deux manières :

- Utiliser l'image d'un bouton prédéfini d'Access
- Utiliser sa propre image

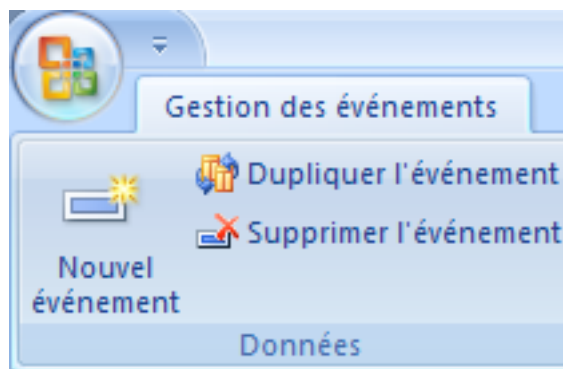
Méthode 1 :

En attribuant l'identifiant idMso d'un bouton prédéfini à l'attribut **imageMso** du bouton personnalisé, Access affiche l'image du bouton prédéfini sur le bouton personnalisé.

Par exemple :

XML

```
<button id="btnDupliquer" label="Dupliquer l'événement" imageMso="SynchronizeData" size="normal"/>
```



Méthode 2 :

Celle-ci est un peu plus complexe. En effet, pour que le bouton puisse afficher l'image, il faut qu'elle soit dans un format acceptable (taille, poids, etc.) par le gestionnaire de ruban d'Office.

Pour convertir l'image dans ce format, il faut utiliser Visual Basic et plus particulièrement les fonctions **CallBack** du ruban. *Qu'est ce que sont ces fameux CallBack ?* Ce sont des fonctions lancées lors du chargement du ruban. Imaginez une fonction nommée **getImageDupliquer** appelée par le ruban à son ouverture et qui retourne une image compatible. Le ruban récupère l'image et l'affecte au bouton **btnDupliquer**. Bien entendu, la fonction en question

doit répondre à un protocole strict, notamment en ce qui concerne ses paramètres pour pouvoir être comprise et exécutée par le ruban.

En VBA, la convention du ruban impose qu'il s'agisse en fait d'une procédure dont l'entête est le suivant :

VBA

```
Public Sub GetImageDupliquer(ByVal control As IRibbonControl, ByRef image)
```

Quand le ruban appelle la procédure, il lui transmet le bouton concerné dans le premier paramètre (ce qui pourra par exemple permettre des vérifications avant l'affectation de l'image) et récupère l'image dans le paramètre **image**. Bien que déclaré en **Variant**, le paramètre **image** est en fait un objet du type **IPictureDisp**. La méthode **LoadPicture** permet de créer un tel objet en fonction du chemin de l'image.

Voici un exemple de fonction **getImageDupliquer** à insérer dans le module **mduRibbon** :

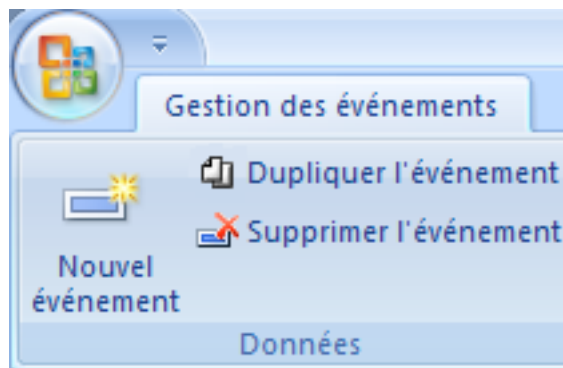
VBA

```
Public Sub GetImageDupliquer(ByVal control As IRibbonControl, ByRef image)
Set image = stdole.LoadPicture(CurrentProject.Path & "\dupliquer.jpg")
End Sub
```

Le lien entre la procédure et le bouton se fait bien entendu dans le XML via l'attribut **getImage** du bouton.

XML

```
<button id="btnDupliquer" label="Dupliquer l'événement" getImage="getImageDupliquer" size="normal"/>
```



Le fait que la fonction **getImageDupliquer** soit à votre charge peut paraître lourd, toutefois, c'est un réel avantage. Il est ainsi possible d'écrire une procédure qui correspond à vos besoins, notamment une procédure générique utilisable par plusieurs boutons :

VBA

```
Public Sub GetImageBouton(ByVal control As IRibbonControl, ByRef image)
Select Case control.Id
Case btnDupliquer:
Set image = stdole.LoadPicture(CurrentProject.Path & "\dupliquer.jpg")
Case btnInsérer:
Set image = stdole.LoadPicture(CurrentProject.Path & "\insérer.bmp")
End Select
End Sub
```

V.C.2 - Les cases à cocher

Une case à cocher est définie dans le code XML avec la balise **checkbox**. Contrairement aux boutons, il n'est pas possible d'utiliser l'attribut **size**. Une telle utilisation provoquerait une invalidation de tout le ruban.

Dans la problématique suivie depuis le début du tutoriel, des cases à cocher peuvent être mises en place pour gérer le filtrage des événements.

C'est ainsi que peut naître le groupe **grpFiltre** dont le XML est :

XML

```
<group id="grpFiltre" label="Filtre">
<checkbox id="chkDate" label="Aujourd'hui"/>
<checkbox id="chkImportant" label="Important"/>
</group>
```



V.C.3 - Le contrôle Gallery - tableau de boutons

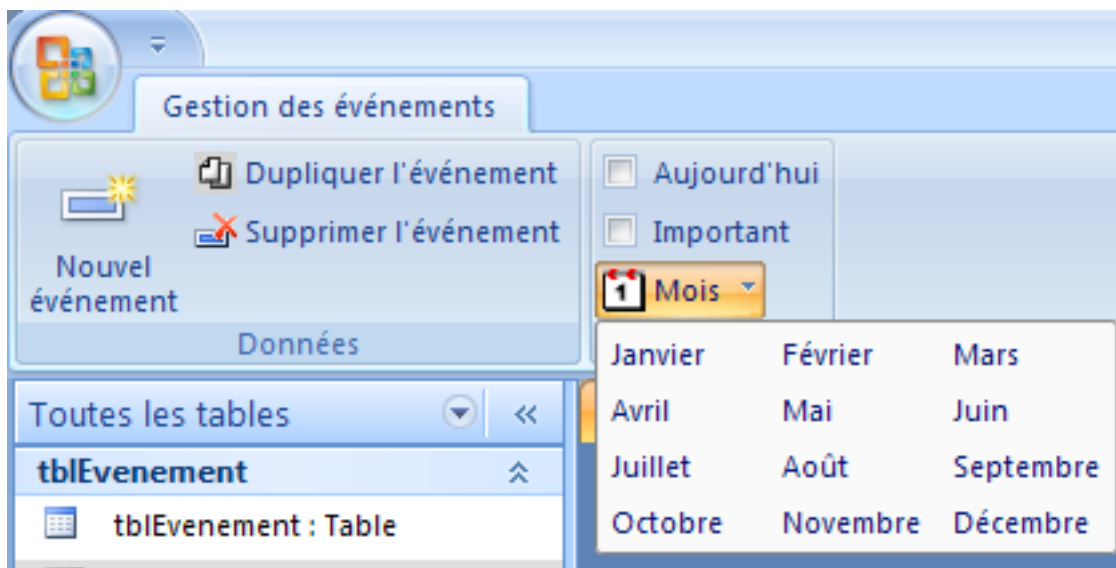
Le contrôle **Gallery** permet d'afficher un composant disposant des boutons en lignes et en colonnes. Il ne s'agit pas d'un sous-menu car il n'est possible d'inclure que des boutons.

Un tel élément est codé dans le XML avec la balise **<gallery>**. Il est alors possible de spécifier le nombre de colonnes (**columns**) et de lignes (**rows**) utilisées pour répartir les différents contrôles. Chaque bouton du contrôle **Gallery** est un **Item** représenté par une balise **<item>**.

L'exemple suivant vient s'ajouter au groupe **grpFiltre** pour permettre à l'utilisateur d'appliquer un filtre sur le mois.

XML

```
<gallery id="galMois" label="Mois ..." columns="3" rows="4" getImage="getImageGalleryMois">
<item id="it01" label="Janvier"/>
<item id="it02" label="Février"/>
<item id="it03" label="Mars"/>
<item id="it04" label="Avril"/>
<item id="it05" label="Mai"/>
<item id="it06" label="Juin"/>
<item id="it07" label="Juillet"/>
<item id="it08" label="Août"/>
<item id="it09" label="Septembre"/>
<item id="it10" label="Octobre"/>
<item id="it11" label="Novembre"/>
<item id="it12" label="Décembre"/>
</gallery>
```



Ici, les éléments (les mois) sont décrits dans le XML mais il est possible de construire le tableau dynamiquement. Tout comme la mise en place d'image sur un bouton, la construction dynamique d'un contrôle **Gallery** se fait à l'aide de fonctions **CallBack**.

Trois fonctions sont nécessaires :

- **getItemCount** : nombre de boutons
- **getItemLabel** : texte du bouton
- **getItemImage** : image du bouton

En VBA, ces **CallBack** doivent correspondre à la norme suivante :

- Sub **getItemCount**(control As IRibbonControl, ByRef count)
- Sub **getItemLabel**(control As IRibbonControl, index As Integer, ByRef label)
- Sub **getItemImage**(control As IRibbonControl, index As Integer, ByRef image)

Pour la liste de mois donnée plus haut, il est possible de créer dynamiquement les différents **items**.

VBA :

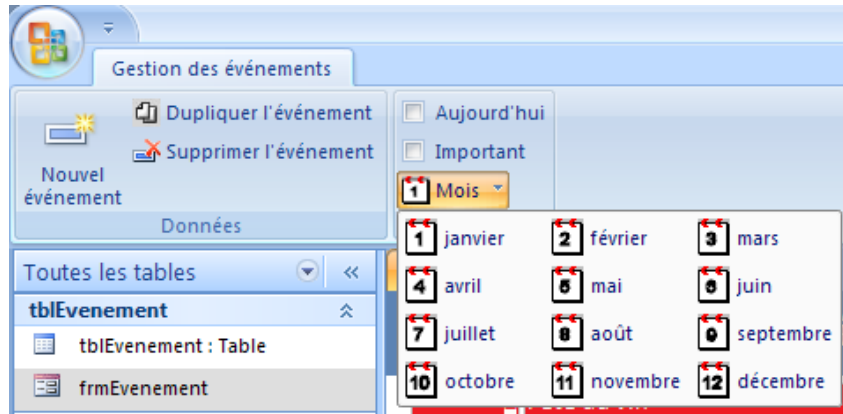
```
VBA
Sub getLabelMois(control As IRibbonControl, index As Integer, ByRef label)
    label = Format(DateSerial(2000, index + 1, 1), "mmmm")
End Sub
Sub getNBMois(control As IRibbonControl, ByRef count)
    count = 12
End Sub
Sub getImageMois(control As IRibbonControl, index As Integer, ByRef image)
    Set image = stdole.LoadPicture("E:\imagesCalendrier\cal" & index + 1 & ".gif")
End Sub
Sub getImageGalleryMois(control As IRibbonControl, ByRef image)
    Set image = stdole.LoadPicture("E:\imagesCalendrier\cal1.gif")
End Sub
```

XML :

```
XML
<gallery id="galMois" label="Mois ..." columns="3" rows="4" getImage="getImageGalleryMois"
```

XML

```
getItemCount="getNBMois" getItemLabel="getLabelMois" getItemImage="getImageMois"/>
```



V.D - Les contrôles de saisies

Le ruban n'accepte pas que des contrôles de commande mais aussi des zones de saisies. Il est ainsi possible d'utiliser des zones de texte et des zones de listes déroulantes.

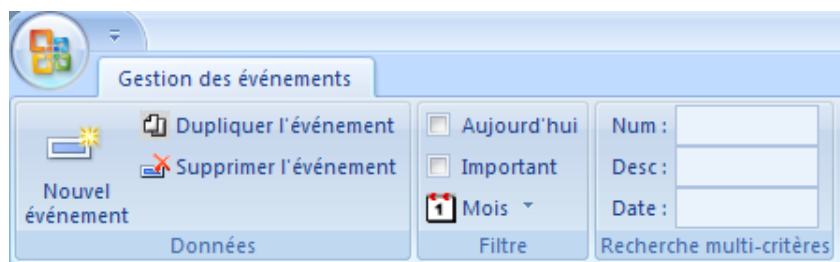
V.D.1 - Les zones de texte

Il aurait été légitime de penser qu'il s'agissait de **TextBox**. Malheureusement, il y a un piège, ce sont en fait des **editBox**. Le risque de confusion avec leurs homologues VBA est donc important, soyez attentif. Comme pour les autres contrôles, il est possible de définir une légende (**label**), à la différence près qu'elle ne serait pas placée dans la zone de texte mais sur son côté gauche.

Grâce à ces éléments, nous pouvons rajouter un groupe de recherche à notre ruban personnalisé.

XML

```
<group id="grpRecherche" label="Recherche multi-critères">
  <editBox id="txtNum" label="Num :"/>
  <editBox id="txtDesc" label="Desc :"/>
  <editBox id="txtDate" label="Date :"/>
</group>
```



Bien entendu, il est possible d'affecter dynamiquement du texte à une zone grâce à sa fonction **CallBack getText**.

Exemple :

Dans le module :

VBA

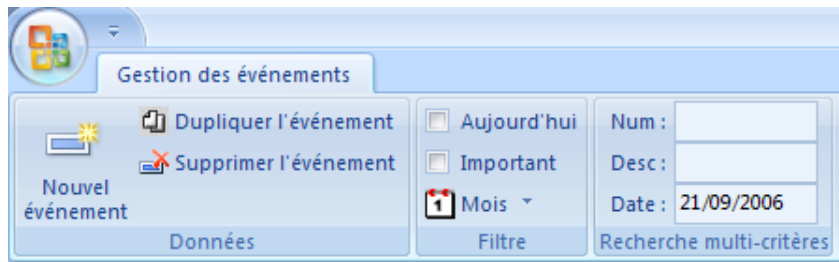
VBA

```
Sub getTextDate(control As IRibbonControl, ByRef text)
text = Format(Date, "dd/mm/yyyy")
End Sub
```

Dans le XML :

XML

```
<editBox id="txtDate" label="Date :" getText="getTextDate"/>
```



L'attribut **sizeString** de la zone de texte définit sa largeur. Elle est exprimée en nombre de caractères. L'attribut **maxLength** correspond au nombre de caractères maximum autorisés à la saisie.

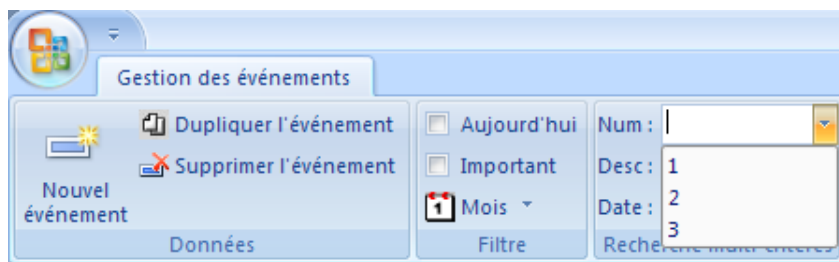
V.D.2 - Les zones de listes déroulantes

Là, le nom coïncide bien avec celui des zones de listes déroulantes VBA, à savoir **comboBox**. Comme pour le contrôle **Gallery**, les éléments sous-jacents sont définis dans des balises **item**.

Exemple :

XML

```
<comboBox id="cmbNum" label="Num :">
<item id="itNum1" label="1"/>
<item id="itNum2" label="2"/>
<item id="itNum3" label="3"/>
</comboBox>
```



Les éléments peuvent aussi être créés dynamiquement. Prenons l'exemple donné plus haut qui consiste à proposer la liste des numéros d'événements.

VBA

```
Private oRst As DAO.Recordset

Sub getNBNum(control As IRibbonControl, ByRef count)
Set oRst =
CurrentDb.OpenRecordset("SELECT NumEvenement FROM tblevenement ORDER BY NumEvenement")
'Récupère le nombre d'enregistrements
```

VBA

```
With oRst
    .MoveLast
    count = .RecordCount
    .MoveFirst
End With
End Sub

Sub getNumLabel(control As IRibbonControl, index As Integer, ByRef label)
On Error GoTo err
    With oRst
        label = .Fields("NumEvenement")
        .MoveNext
    End With
Exit Sub
err:
MsgBox err.Description
End Sub
```

Ci-dessous, le code XML permettant de mettre en place ce comportement.

XML

```
<comboBox id="cmbNum" label="Num :" getItemCount="getNBNum" getItemLabel="getNumLabel"/>
```

Analysons en détail le code VBA. La variable **oRst** est déclarée au niveau du module. La méthode **getNBNum** (destinée à compter le nombre d'items à créer) est lancée en premier par le ruban. Elle instancie le recordset stocké dans la variable privée du module. La méthode **getNumLabel** n'a plus qu'à parcourir le recordset (disponible, puisque déclaré dans le module) pour retourner le numéro de l'événement (**champ NumEvenement**).

V.E - Les séparateurs

Autre type de contrôle disponibles : les **séparateurs**. Ce sont des barres verticales divisant deux groupes de boutons. La balise XML permettant de les identifier est **<separator>**. Ces composants n'affichant rien, le seul attribut à valoriser est l'identifiant (**id**).

Exemple :

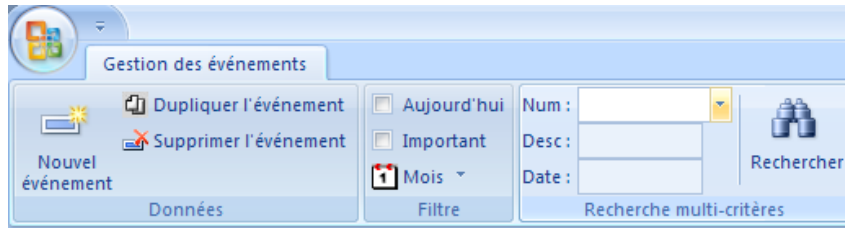
XML

```
<separator id="sepRecherche"/>
```

Le groupe de recherche devient alors :

XML

```
<group id="grpRecherche" label="Recherche multi-critères">
<comboBox id="cmbNum" label="Num :" getItemCount="getNBNum" getItemLabel="getNumLabel"/>
<editBox id="txtDesc" label="Desc :"/>
<editBox id="txtDate" label="Date :"/>
<separator id="sepRecherche"/>
<button id="btnRechercher" label="Rechercher" imageMso="FindDialog" size="large"/>
</group>
```



VI - Les événements

Jusqu'à présent, seul l'interface a été dessinée. Aucun code n'est affecté au ruban. Seuls les boutons prédéfinis sont opérationnels. Dans cette section, nous allons voir comment Access permet de lier du code VBA aux autres éléments d'un ruban personnalisé.

VI.A - Les contrôles d'action

Il s'agit des éléments qui nécessitent un clic pour déclencher leur action. On y retrouve les boutons, les cases à cocher ainsi que le contrôle Gallery. Le nom de la procédure qu'ils doivent lancer est inscrit dans leur attribut **onAction**. Pour que l'événement soit déclenché correctement, la procédure (**CallBack**) doit répondre à une nomenclature précise dépendante du type de contrôle.

VI.A.1 - Les boutons

La procédure événementielle **onAction** d'un bouton doit correspondre à la syntaxe suivante :

```
VBA
Sub onAction(ByVal control As IRibbonControl)
```

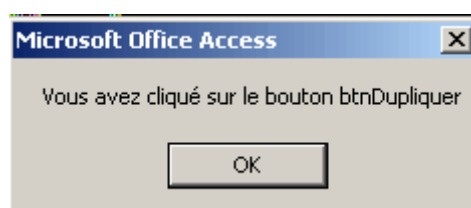
L'argument **control** de la procédure représente le contrôle à l'origine du traitement.

Prenons le couple VBA/XML qui suit.

```
VBA
Public Sub btnDupliquer_action(ByVal control As IRibbonControl)
    MsgBox "Vous avez cliqué sur le bouton " & control.Id
End Sub
```

```
XML
<button id="btnDupliquer" label="Dupliquer l'événement"
imageMso="SynchronizeData" size="normal" onAction="btnDupliquer_action"/>
```

Un clic sur le bouton **Dupliquer l'enregistrement** provoque l'affichage de la boîte de dialogue ci-dessous.



Pour revenir au projet qui nous intéresse depuis le début de ce tutoriel, et plus particulièrement la duplication de l'enregistrement, le code de la méthode **btnDupliquer_action** pourrait être :

VBA

```

Public Sub btnDupliquer_action(ByVal control As IRibbonControl)
    Dim strNom As String, bolImp As Boolean
    Dim oFrm As Form
    'Interroge l'utilisateur
    If MsgBox("Voulez-
vous dupliquer cette événement aujourd'hui ?", vbYesNo + vbQuestion, "Dupliquer ?") = vbYes Then
        'Accède au formulaire
        Set oFrm = Forms("frmEvenement")
        With oFrm.Recordset
            'Récupère le nom de l'événement sélectionné
            strNom = .Fields("NomEvenement")
            bolImp = .Fields("Important")
            'Crée le nouvel enregistrement
            .AddNew
            .Fields("NomEvenement") = strNom
            .Fields("DateEvenement") = Date
            .Fields("Important") = bolImp
            .Update
        End With
    End If
End Sub
    
```

VI.A.2 - Les cases à cocher

La procédure événementielle des cases à cocher est analogue à celle des boutons à la différence près qu'un paramètre supplémentaire est nécessaire. Il permet de savoir si la case est cochée ou pas.

La norme définit l'entête suivant à appliquer :

VBA

```

Sub onAction(ByVal control As IRibbonControl, pressed As Boolean)
    
```

Le filtrage des événements de notre projet sur la date et leur importance peut être obtenu par :

XML

```

<checkBox id="chkDate" label="Aujourd'hui" onAction="chkDate_action"/>
<checkBox id="chkImportant" label="Important" onAction="chkImp_action"/>
    
```

VBA

```

Public Sub chkDate_action(ByVal control As IRibbonControl, pressed As Boolean)

    'mémorise la valeur de la case à cocher
    bolFiltreDate = pressed
    'Applique le filtre
    AppliquerFiltre
End Sub
Public Sub chkImp_action(ByVal control As IRibbonControl, pressed As Boolean)

    'mémorise la valeur de la case à cocher
    bolFiltreImp = pressed
    'Applique le filtre
    AppliquerFiltre
End Sub

Private Sub AppliquerFiltre()
    Dim oFrm As Form
    Dim strFiltre As String
    'Accède au formulaire
    Set oFrm = Forms("frmEvenement")
    'Construit le filtre
    
```

VBA

```

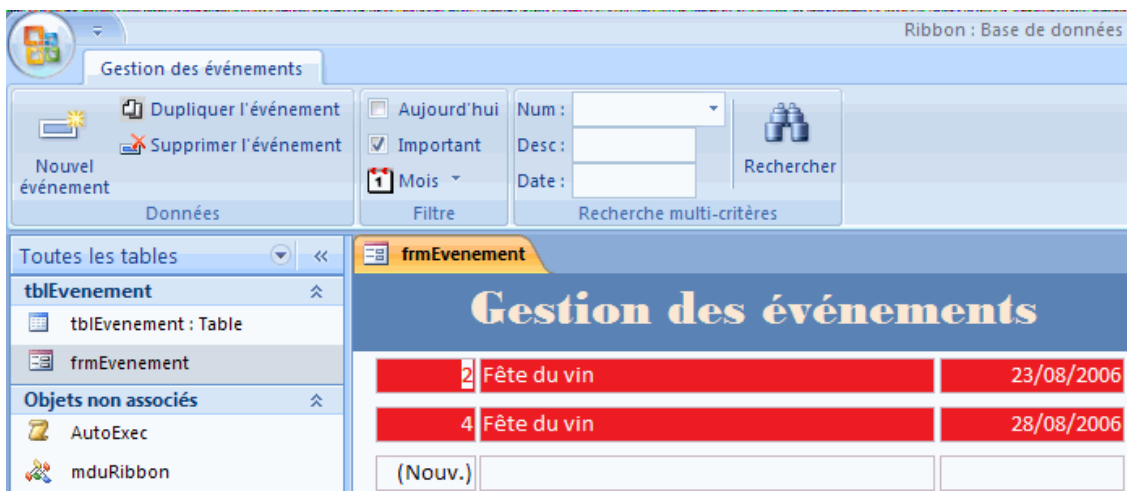
If bolFiltreDate Then
    strFiltre = "DateEvenement=Date()"
End If
'Tiens compte du filtre sur Important
If bolFiltreImp Then
    strFiltre = strFiltre & IIf(strFiltre <> "", " AND ", "") & "Important"
End If
'Affecte le filtre au formulaire
oFrm.Filter = strFiltre
oFrm.FilterOn = True
End Sub
    
```

Où **bolFiltreDate** et **bolFiltreImp** sont déclarées en entête de module par :

VBA

```

Private bolFiltreDate As Boolean
Private bolFiltreImp As Boolean
    
```



Comme vous avez pu le remarquer, les procédures **chkDate_action** et **chkImp_action** sont identiques. Si vous le souhaitez, vous pouvez les réunir :

VBA

```

Public Sub chkDateImp_action(ByVal control As IRibbonControl, pressed As Boolean)
    'Quel bouton ?
    Select Case control.Id
        Case "chkDate": bolFiltreDate = pressed
        Case "chkImp": bolFiltreImp = pressed
    End Select
    'Applique le filtre
    AppliquerFiltre
End Sub
    
```

Ce qui donne dans le XML :

XML

```

<checkBox id="chkDate" label="Aujourd'hui" onAction="chkDateImp_action"/>
<checkBox id="chkImp" label="Important" onAction="chkDateImp_action"/>
    
```

VI.A.3 - Le contrôle Gallery

La méthode **onAction** d'un contrôle **Gallery** est encore plus complète que les précédentes. Elle propose :

- Le contrôle **Gallery** à l'origine de l'événement
- L'**identifiant** de l'élément sélectionné
- L'**index** de l'élément sélectionné

Le tout est encapsulé dans la nomenclature suivante :

VBA

```
Sub onAction(ByVal control As IRibbonControl, selectedId As String, selectedIndex As Integer)
```

Ce qui peut donner pour le contrôle **galMois** :

VBA

```
Public Sub galMois_action(ByVal control As IRibbonControl, _
    selectedId As String, selectedIndex As Integer)
    Dim oFrm As Form
    Dim strFiltre As String
    'Accède au formulaire
    Set oFrm = Forms("frmEvenement")
    strFiltre = "Month(DateEvenement)=" & selectedIndex + 1
    'Affecte le filtre au formulaire
    oFrm.Filter = strFiltre
    oFrm.FilterOn = True
End Sub
```

XML

```
<gallery id="galMois" label="Mois ..." columns="3" rows="4"
    getImage="getImageGalleryMois" getItemCount="getNBMois" getItemLabel="getLabelMois" getItemImage="getImageMois" />
</group>
```

VI.B - Les contrôles de saisie

Qu'ils s'agissent de zones de texte ou bien de zones de listes déroulantes, la procédure événementielle est la même. Nommée **onChange**, elle se déclenche lorsque l'utilisateur valide sa saisie (changement de contrôle, appui sur la touche *Entrée*). Son entête est défini comme suit :

VBA

```
Sub onChange(ByVal control As IRibbonControl, text As String)
```

Exemple :

VBA

```
Public Sub cmbNum_change(ByVal control As IRibbonControl, text As String)
    MsgBox "Valeur de " & control.Id & ":" & text
End Sub
```

XML

```
<comboBox id="cmbNum" label="Num :" getItemCount="getNBNum" getItemLabel="getNumLabel" onChange="cmbNum_change" />
```

Le code à appliquer aux contrôles **cmbNum**, **txtDesc** et **txtDate** est similaire à celui des cases à cocher du groupe **Filtre**. Je ne vais donc pas rentrer plus en détails. Il s'agit d'une simple recherche multi-critères dont le **tutoriel de Cafeine** constitue un bon exemple.

VII - Agir dynamiquement sur le ruban

A ce stade le ruban est opérationnel, toutefois il est encore possible de l'améliorer. Il permet à l'utilisateur de déclencher des actions, mais ne réagit pas aux actions de l'application. Par exemple, le bouton de suppression ne se désactive pas lorsque l'événement sélectionné est important.

VII.A - Principe de base

L'objet **Access.Application** ne fournit pas de méthode permettant d'accéder à un élément du ruban. En fait, rien ne permet d'agir directement sur un attribut de celui-ci. La seule solution disponible est de lui demander d'aller lui-même chercher les informations dont il a besoin. Grâce à des fonctions nommées **getXXX**, le ruban est capable de rafraîchir sa propriété **XXX**. Vous vous en doutez, cela passe une nouvelle fois par des fonctions **CallBack**. La mise au point d'un tel système passe par deux étapes :

- Au chargement, il faut instancier un objet correspondant au ruban. Cet objet déclaré dans un module sera accessible partout dans l'application.
- Appeler la méthode de rafraîchissement du ruban (ou d'un de ses contrôles) au moment voulu.

VII.A.1 - Mémorisation du ruban dans un objet

La balise racine **CustomUI** possède un attribut **onLoad** définissant une fonction de rappel exécutée après le chargement du ruban. Cette fonction retourne un objet encapsulant le ruban.

En VBA, il s'agit d'une procédure :

VBA

```
Sub getMonRuban(ribbon As IRibbonUI)
    Set oMonruban = ribbon
End Sub
```

Où **oMonruban** est une variable de type **IRibbonUi** déclarée dans l'entête du module **mduRibbon**.

Le XML devient :

XML

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" onLoad="getMonRuban">
...
...
</customUI>
```

VII.A.2 - Rafraîchissement du ruban

Pour rafraîchir le ruban, l'objet **IRibbonUi** offre deux méthodes :

- **Invalidate** : Rafraîchit tout le ruban
- **InvalidateControl**(ControlID as String) : Rafraîchit le contrôle dont l'identifiant est passé en paramètre.

VII.B - Activer / Désactiver un élément

Comme indiqué dans l'énoncé, il ne doit pas être possible de supprimer un événement important. Le bouton doit donc être inactif dans ce cas. La propriété gérant cet état est nommée **enabled**. L'attribut définissant la fonction **CallBack** modifiant cette propriété est donc **getEnabled**.

Ce qui donne :

XML

```
<button idMso="RecordsDeleteRecord" label="Supprimer l'événement"
size="normal" getEnabled="getBtnSupprimer_enabled"/>
```

VBA

```
Sub getBtnSupprimer_enabled(control As IRibbonControl, ByRef enabled)
On Error GoTo err:
Dim oFrm As Form
'Accède au formulaire
Set oFrm = Forms("frmEvenement")
enabled = Not oFrm.Recordset.Fields("Important").Value
Exit Sub
err:
'en cas d'erreur, rendre la suppression impossible
enabled = False
End Sub
```

Cependant, il n'est pas possible de modifier le comportement d'un bouton prédéfini. Il faut donc remplacer le bouton **DeleteRecord2** par un bouton personnalisé :

XML

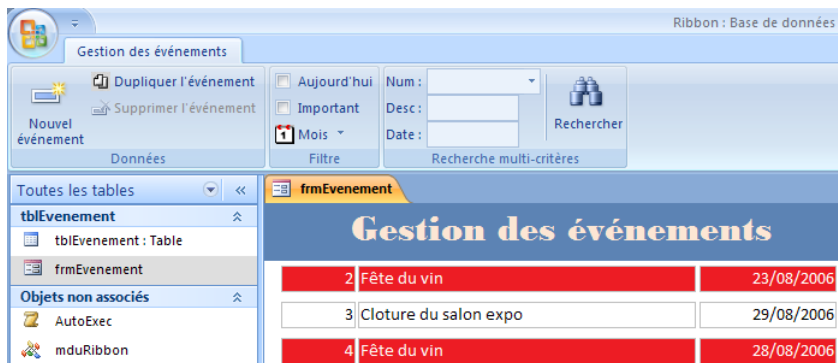
```
<button id="btnSupprimer" label="Supprimer l'événement"
size="normal" getEnabled="getBtnSupprimer_enabled" onAction="btnSupprimer_action" imageMso="DeleteRecord2"/>
```

Il ne reste plus qu'à demander le rafraîchissement du bouton lorsque l'utilisateur change d'enregistrement. Cela passe par une procédure événementielle sur l'événement **Current (Sur Activation)** du formulaire.

VBA

```
Private Sub Form_Current()
If Not (oMonruban Is Nothing) Then
oMonruban.InvalidateControl "btnSupprimer"
End If
End Sub
```

Le test **Not (oMonruban Is Nothing)** permet d'éviter d'éventuelles erreurs au cas où le ruban ne serait pas chargé.



VII.C - Modifier la valeur d'une case à cocher

Dans notre exemple, le fait de filtrer par mois doit réinitialiser les deux cases à cocher. En effet, le filtrage avec les cases à cocher et celui avec le contrôle **Gallery** sont indépendants. L'attribut définissant la valeur d'une case à cocher est **pressed**, il est donc nécessaire d'implémenter la fonction **CallBack getPressed**.

VBA

```
Sub getchkDateImp_pressed(control As IRibbonControl, ByRef pressed)
    'Quel bouton ?
    Select Case control.Id
        Case "chkDate": pressed = bolFiltreDate
        Case "chkImp":  pressed = bolFiltreImp
    End Select
End Sub
```

XML

```
<checkBox id="chkDate" label="Aujourd'hui" onAction="chkDateImp_action" getPressed="getchkDateImp_pressed"/>
<checkBox id="chkImp" label="Important" onAction="chkDateImp_action" getPressed="getchkDateImp_pressed"/>
```

Reste à définir le moment où les cases à cocher doivent être rafraîchies. Réponse : lorsque l'utilisateur utilise le contrôle **Gallery**.

Il faut donc rajouter ces quelques lignes à la procédure **galMois_action** :

VBA

```
'Vide les cases à cocher
bolFiltreDate = False
bolFiltreImp = False
oMonruban.InvalidateControl "chkDate"
oMonruban.InvalidateControl "chkImp"
```

VII-D - Autres possibilités

Il est possible de modifier n'importe quelle propriété d'un contrôle via une procédure de rappel **getXXX** où **XXX** est le nom de la propriété en question. **La liste des fonctions Callback** est disponible dans l'annexe B.

IX - Conclusion

Comme vous avez pu vous en rendre compte, la création de rubans personnalisés est plus complexe que celle des menus sous les versions précédentes. Cela reste toutefois normal étant donné le nombre de fonctionnalités disponibles.

Dorénavant, le ruban ne sera plus une barre disgracieuse dans vos développements mais un véritable composant intégré à vos applications offrant à portée de la main la globalité des actions envisageables. Telle en a été la volonté de Microsoft en l'incluant dans Office, telle en sera la votre en l'incluant dans vos développements. Pour ma part, les possibilités offertes me séduisent de jour en jour.

Je tiens à remercier l'équipe Access de www.developpez.com pour leur relecture approfondie.

X - Annexe A : Liste des idMso

Cette liste correspond à la version Access 2007 Béta 2 :

Commande	idMso
10 %	PrintPreviewZoom10
1000%	PrintPreviewZoom1000
150 %	PrintPreviewZoom150
200 %	PrintPreviewZoom200
25 %	PrintPreviewZoom25
3D enfoncé	ControlSpecialEffectSunken
50 %	PrintPreviewZoom50
500%	PrintPreviewZoom500
75 %	PrintPreviewZoom75
Access	ImportAccess
Accueil	TabHomeAccess
Actualiser	RecordsRefreshMenu
Actualiser	RecordsRefreshRecords
Actualiser la liste	TableSharePointListsRefreshList
Actualiser le tableau croisé dynamique	PivotRefresh
Actualiser l'état	SourceControlRefreshStatus
Actualiser tout	DataRefreshAll
Administrer	GroupAdminister
Affichage	ViewsSwitchToDefaultView
Affichage par défaut...	DefaultView
Affichage personnalisé	AdpDiagramCustomView
Affichages	GroupViews
Afficher en tant que	PivotShowAsMenu
Afficher haut/bas	PivotShowTopAndBottomItemsMenu
Afficher la table...	QueryShowTable
Afficher les colonnes...	RecordsUnhideColumns
Afficher les détails	PivotShowDetails
Afficher les étiquettes de relation	AdpDiagramShowRelationshipLabels
Afficher les marges	ShowMargins
Afficher les relations directes	RelationshipsDirectRelationships
Afficher les sauts de page	AdpDiagramViewPageBreaks
Afficher tout	PivotShowAll
Afficher toutes les actions	MacroShowAllActions
Afficher toutes les relations	RelationshipDesignAllRelationships
Afficher...	WindowUnhide
Afficher/Masquer	GroupAdpDiagramShowHide
Afficher/Masquer	GroupLayoutShowHide
Afficher/Masquer	GroupMacroShowHide
Afficher/Masquer	GroupPivotChartShowHide
Afficher/Masquer	GroupPivotTableShowHideAccess
Afficher/Masquer	GroupQueryShowHide
Afficher/Masquer	GroupTableDesignShowHide
Afficher/Masquer	GroupViewsShowHide
Ajout	QueryAppend
Ajouter à partir d'Outlook	RecordsAddFromOutlook
Ajouter aux données en sortie	AdpOutputOperationsAddToOutput
Ajouter des champs existants	FieldList
Ajouter des objets à SourceSafe	SourceControlAddObjects
Ajouter des tables connexes	AdpDiagramAddRelatedTables
Ajouter une base de données dans SourceSafe	SourceControlAddDatabase
Ajouter une table...	AdpDiagramAddTable

Ajuster à la fenêtre	ZoomFitToWindow
Ajuster à la taille du formulaire	PositionFitToWindow
Ajuster au contenu	SizeToFit
Alignement du contrôle	GroupControlAlignment
Aligner à droite	AlignRight
Aligner à droite	ObjectsAlignRight
Aligner à gauche	AlignLeft
Aligner à gauche	ObjectsAlignLeft
Aligner en bas	ObjectsAlignBottom
Aligner en haut	ObjectsAlignTop
Aligner sur la grille	ControlSnapToGrid
Analyse croisée	QueryCrosstab
Analyser	GroupAnalyze
Analyser la table	DatabaseAnalyzeTable
Analyser les performances	DatabaseAnalyzePerformance
Ancrage	PositionAnchoringGallery
Annuler	Undo
Annuler Extraire	SourceControlUndoCheckOut
Annuler l'ordre personnalisé	PivotClearCustomOrdering
Aperçu avant impression	FilePrintPreview
Aperçu avant impression	TabPrintPreviewAccess
Aperçu avant impression	ViewsAdpDiagramPrintPreview
Aperçu des 10 premiers enregistrements	First10RecordsPreview
Aperçu et impression	FilePrintMenu
Appliquer le filtre	FilterToggleFilter
Appliquer le filtre/tri	ApplyFilter
Appliquer le format des nombres avec virgule	ApplyCommaFormat
Appliquer le format monétaire	ApplyCurrencyFormat
Appliquer le format pourcentage	ApplyPercentageFormat
Appliquer un filtre serveur	ServerFilterApply
Après la sélection	FilterAfterSelection
Archiver	SourceControlCheckIn
Arguments	MacroArguments
Assistant Carte postale	PostcardWizard
Assistant État	CreateReportFromWizard
Assistant Format automatique...	AutoFormatWizard
Assistant Formulaire commercial	BusinessFormWizard
Assistant Requête	CreateQueryFromWizard
Assistant Sécurité au niveau utilisateur...	DatabaseUserLevelSecurityWizard
Atteindre	GoToMenuAccess
Au contenu	SizeToFitAccess
Au plus étroit	SizeToNarrowest
Au plus grand	SizeToTallest
Au plus large	SizeToWidest
Au plus petit	SizeToShortest
Augmentation horizontale	HorizontalSpacingIncrease
Augmentation verticale	VerticalSpacingIncrease
Augmenter le nombre de valeurs décimales	FormattingIncreaseDecimals
Augmenter le retrait	IndentIncrease
Autorisations	TableListPermissions
Autorisations d'accès...	DatabasePermissions

Autre	GroupCreateOther
Autre couleur d'arrière-plan/remplissage	FontAlternateFillBackColorPicker
Avant la sélection	FilterBeforeSelection
Barre des messages	ViewMessageBar
Barres d'outils personnalisées	GroupAddInsCustomToolbars
Basculer tout hors ligne	SharePointListsWorkOffline
Base de données Access	DatabaseAccessBackEnd
Base de données Access	ExportAccess
Base de données Access 2000	FileSaveAsAccess2000
Base de données Access 2002 - 2003	FileSaveAsAccess2002_2003
Base de données Access 2007	FileSaveAsAccess2007
Base de données ODBC	ExportOdbcDatabase
Base de données ODBC	ImportOdbcDatabase
Bouton	FormControlButton
Bouton bascule	ControlToggleButton
Boutons d'extraction	PivotExpandIndicators
Cadre d'objet dépendant	ControlBoundObjectFrame
Cadre d'objet indépendant	ControlUnboundObjectFrame
Calcul automatique	PivotAutoCalcMenu
Cascade	WindowsCascade
Case à cocher	FormControlCheckBox
Case d'option	FormControlRadioButton
Centrer	AlignCenter
Champ actif	GroupPivotChartActiveFieldAccess
Champ actif	GroupPivotTableActiveFieldAccess
Champs et colonnes	GroupFieldsAndColumns
Changement de fenêtre	WindowsSwitch
Charger à partir d'une requête...	LoadFromQuery
Ciselé	ControlSpecialEffectChiseled
Clé primaire	AdpPrimaryKey
Clés	AdpDiagramKeys
Coder/décoder une base de données...	DatabaseEncodeDecode
Collage spécial...	PasteSpecialDialog
Collecter les données	GroupCollectData
Coller	Paste
Coller	PasteSpecial
Coller par ajout	PasteAppend
Colonne de recherche...	DatasheetColumnLookup
Colonnes	PrintColumns
Combinaisons multiples	PivotChartMultiplePlots
Commandes de la barre d'outils	GroupAddInsToolbarCommands
Commandes de menu	GroupAddInsMenuCommands
Commence par la sélection	FilterBeginsWithSelection
Compacter une base de données	FileCompactAndRepairDatabase
Compléments	AddInsMenu
Compléments	TabAddIns
Compléments COM...	ComAddInsDialog
Compter les enregistrements	TotalsCountRecords
Conditions	MacroConditions
Connexion	ServerConnection
Contient la sélection	FilterContainsSelection
Contraintes	AdpConstraints
Contrôle de la source	TabSourceControl

Contrôle de la source des bases de données	GroupDatabaseSourceControl
Contrôle de la source des objets	GroupSourceControlShow
Contrôle des tabulations	ControlTabControl
Contrôler l'alignement	GroupControlAlignmentLayout
Contrôles	GroupControlsAccess
Contrôles	GroupFormattingControls
Contrôles ActiveX	ControlActiveX
Convertir	ConvertDatabaseFormat
Convertir les macros en Visual Basic	MacroConvertMacrosToVisualBasic
Copie du fichier de base de données	DatabaseCopyDatabaseFile
Copier	Copy
Correspond à la sélection	FilterEqualsSelection
Couleur	GridlinesColorPicker
Couleur d'arrière-plan/remplissage	FontFillBackColorPicker
Couleur de police	FontColorPicker
Couleur de surbrillance du texte	TextHighlightColorPicker
Couleur du trait	ControlLineColorPicker
Couper	Cut
Courrier électronique	FileSendAsAttachment
Création	TabAdpDiagramDesign
Création	TabAdpFunctionAndViewToolsDesign
Création	TabAdpSqlStatementDesign
Création	TabAdpStoredProcedureToolsDesign
Création	TabFormToolsDesign
Création	TabTableToolsDesignAccess
Création de formulaire	CreateFormInDesignView
Création de requête	CreateQueryInDesignView
Création de table	CreateTableInDesignView
Création de table	QueryMakeTable
Création de table	TableDesign
Création d'état	CreateReportInDesignView
Création du total calculé...	PivotCreateCalculatedTotal
Création d'un champ de détail calculé...	PivotCreateCalculatedField
Créer	TabCreate
Créer	TabMacroToolsDesign
Créer	TabPivotChartDesign
Créer	TabPivotTableDesign
Créer	TabQueryToolsDesign
Créer	TabRelationshipToolsDesign
Créer	TabReportToolsDesign
Créer à partir de SourceSafe	SourceControlCreateDatabaseFromProject
Créer un courrier électronique	CreateEmail
Créer un fichier MDE...	DatabaseMakeMdeFile
Créer un menu contextuel à partir de la macro	CreateShortcutMenuFromMacro
Créer un réplica...	ReplicationCreateReplica
Croissant	SortUp
Date et heure...	DateAndTimeInsert
De droite à gauche	TextDirectionRightToLeft
De gauche à droite	AlignLeftToRightMenu
De gauche à droite	TextDirectionLeftToRight
Décroissant	SortDown
Définir la police des étiquettes...	LabelFontDialog

Définir le mot de passe de la base de données...	SetDatabasePassword
Définir les paramètres par défaut du contrôle	ControlSetControlDefaults
Définition des données	QueryDataDefinition
Définition du mot de passe de connexion	DatabaseSetLogonSecurity
Démarrer SourceSafe	SourceControlRun
Dépendances d'objet	DatabaseObjectDependencies
Déplacer le champ	PivotMoveField
Déplacer les données	GroupMoveData
Déplacer vers la zone de champ	PivotMoveToFieldArea
Déplacer vers la zone de colonnes	PivotMoveToColumnArea
Déplacer vers la zone de détails	PivotMoveToDetailArea
Déplacer vers la zone de filtres	PivotMoveToFilterArea
Déplacer vers SharePoint	DatabaseMoveToSharePoint
Dernier	MailMergeGotToLastRecord
Dernière version	SourceControlGetLatestVersion
Deux pages	PrintPreviewZoomTwoPages
Développer le champ	PivotExpandField
Développer tout	RecordsExpandAllSubdatasheets
Différences	SourceControlShowDifferences
Dimensionner automatiquement les tables sélectionnées	AdpDiagramAutosizeSelectedTables
Diminuer le nombre de valeurs décimales	FormattingDecreaseDecimals
Diminuer le retrait	IndentDecrease
Diminution horizontale	HorizontalSpacingDecrease
Diminution verticale	VerticalSpacingDecrease
Disposition	GroupAdpDiagramLayout
Dissocier	ObjectsUngroup
Dissocier	PivotUngroupItems
Document HTML	ExportHtmlDocument
Document HTML	ImportHtmlDocument
Documentation de base de données	DatabaseDocumenter
Données	GroupPivotChartDataAccess
Données	GroupPivotTableDataAccess
Données	GroupPrintPreviewData
Données de la liste en cache	CacheListData
Données externes	TabExternalData
Données seulement	PrintDataOnly
Dossier Outlook	ImportOutlook
Douze pages	PrintPreviewTwelvePages
Dupliquer	PasteDuplicate
Écart-type	PivotAutoCalcStandardDeviation
Écart-type	TotalsStandardDeviation
Écart-type (population)	PivotAutoCalcStandardDeviationPopulation
Échelle unifiée	PivotChartMultipleUnified
Effacer	ClearMenuAccess
Effacer la grille	ClearGrid
Effacer la mise en page	RelationshipsClearLayout
Effacer tous les filtres	FilterClearAllFilters
Effet spécial	ControlSpecialEffectMenu
Empaqueter signer	FilePackageAndSign

Empilé	ControlLayoutStacked
En relief	ControlSpecialEffectRaised
Enregistrements	GroupRecords
Enregistrer	FileSave
Enregistrer	RecordsSaveRecord
Enregistrer comme contact Outlook	RecordsSaveAsOutlookContact
Enregistrer en tant que requête	SaveAsQuery
Enregistrer l'objet sous	SaveObjectAs
Enregistrer sous	FileSaveAs
Enregistrer sous un autre format	FileSaveAsMenuAccess
En-tête et pied de page	PageHeaderOrFooterShowHide
En-tête/pied de formulaire	FormHeaderOrFooterShowHide
Entre...	FilterBetween
Épaisseur	GridlinesWidthGallery
Épaisseur de ligne	ControlLineThicknessGallery
État	CreateReport
État vide	CreateReportBlankReport
États	GroupCreateReports
Étiquette	FormControlLabel
Étiquettes	CreateLabels
Excel	ExportExcel
Excel	ImportExcel
Exécuter	QueryRunQuery
Exécuter une macro	MacroRun
Exportations enregistrées	ExportSavedExports
Exporter	GroupExport
Exporter vers Excel	PivotExportToExcel
Extraire	DrillOut
Extraire	SourceControlCheckOut
Extraire dans	DrillInto
Fenêtre	GroupWindowAccess
Fermer	CloseDocument
Fermer	GroupMacroClose
Fermer	GroupQueryClose
Fermer	MasterViewClose
Fermer la base de données	FileCloseDatabase
Fermer l'aperçu	GroupPrintPreviewClosePreview
Fermer l'aperçu avant impression	PrintPreviewClose
Feuille de données	TabTableToolsDatasheet
Feuille des propriétés	PropertySheet
Fichier dBASE	ExportDBase
Fichier dBASE	ImportDBase
Fichier Lotus 1-2-3	ExportLotus
Fichier Lotus 1-2-3	ImportLotus
Fichier Paradox	ExportParadox
Fichier Paradox	ImportParadox
Fichier texte	ExportTextFile
Fichier texte	ImportTextFile
Fichier XML	ExportXmlFile
Fichier XML	ImportXmlFile
Figier	RecordsFreezeColumns
Filtre automatique	PivotAutoFilter
Filtre serveur par formulaire	ServerFilterByForm

Filtre/tri avancé...	SortAndFilterAdvanced
Filtrer	FiltersMenu
Filtrer	PivotFilterBySelection
Filtrer et trier	GroupPivotChartFilterAndSort
Filtrer et trier	GroupPivotTableFilterAndSort
Filtrer hors sélection	FilterExcludingSelection
Filtrer par formulaire	FilterAdvancedByForm
Filtrer par sélection	FilterBySelection
Format	TabFormToolsFormatting
Format	TabReportToolsFormatting
Format de cellule...	FormatCellsDialog
Format :	FormattingFormat
Formulaire	CreateForm
Formulaire double affichage	CreateFormSplitForm
Formulaire vierge	CreateFormBlankForm
Formulaires	GroupCreateForms
Formules	PivotFormulasMenu
Fractionner	WindowSplit
Fusion avec Microsoft Office Word	MergeToWord
Générateur	QueryBuilder
Gérer	FileManageMenu
Gérer	GroupSourceControlManage
Gérer les réponses	ManageReplies
Gestion des index...	AdpManageIndexes
Gestion des utilisateurs et des groupes...	DatabaseUserAndGroupAccounts
Gestionnaire de Menu Général	DatabaseSwitchboardManager
Gestionnaire de tables liées	DatabaseLinedTableManager
Graphique croisé dynamique	CreateFormPivotChart
Graphique...	ControlChart
Gras	Bold
Gravé	ControlSpecialEffectEtched
Grille	AdpViewGridPane
Grille	GridShowHide
Grouper	ObjectsGroup
Grouper	PivotGroupItems
Grouper par	AdpOutputOperationsGroupBy
Hauteur de ligne...	RowHeight
Historique	SourceControlShowHistory
Horizontal équilibré	HorizontalSpacingMakeEqual
Huit pages	PrintPreviewEightPages
Ignorer les modifications	SharePointListsDiscardChangesMenu
Ignorer toutes les modifications	SharePointListsDiscardAllChanges
Ignorer toutes les modifications et actualiser	SharePointListsDiscardAllChangesAndRefresh
Image	ControllImage
Importations enregistrées	ImportSavedImports
Importer	GroupImport
Impression rapide	FilePrintQuick
Imprimer	GroupPrintPreviewPrintAccess
Imprimer	PrintDialogAccess
Index	TableIndexes
Index / Clés	AdpDiagramIndexesKeys
Insérer	QueryInsertColumn

Insérer des colonnes	QueryInsertColumns
Insérer des lignes	TableRowsInsertWord
Insérer un saut de page	PageBreakInsertOrRemove
Insérer une page	ControlPage
Italique	Italic
Largeur de colonne...	ColumnWidth
Légende	PivotChartLegendShowHide
Liaisons OLE/DDE	OleDdeLinks
Libérer	RecordsUnfreeze
Lien hypertexte...	HyperlinkInsert
Lier les tables	FileServerLinkTables
Ligne/colonne	PivotSwitchRowColumn
Lignes	GroupMacroRows
Liste des champs	PivotFieldList
Liste SharePoint	ExportSharePointList
Liste SharePoint	GroupSharePointList
Liste SharePoint	ImportSharePointList
Listes SharePoint	CreateTableUsingSharePointListsGallery
Listes SharePoint	GroupSharepointLists
Logo	ControlLogo
Macro	CreateMacro
Macro	GroupMacro
Marges	PageMarginsGallery
Marges du contrôle	ControlMarginsGallery
Masquer	CustomizeHide
Masquer la table	AdpDiagramHideTable
Masquer la table	RelationshipsHideTable
Masquer les colonnes	RecordsHideColumns
Masquer les détails	HideDetails
Masquer les détails	PivotHideDetails
Max.	AutoSumMax
Max.	PivotAutoCalcMax
Me prévenir	TableListAlertMe
Mettre à l'arrière-plan	ObjectSendToBack
Mettre au premier plan	ObjectBringToFront
Min.	AutoSumMin
Min.	PivotAutoCalcMin
Mise à jour	QueryUpdate
Mise en forme	GroupFormatting
Mise en forme auto.	AutoFormatGallery
Mise en forme auto.	GroupAutoFormatAccess
Mise en forme conditionnelle	FontConditionalFormatting
Mise en forme de contrôle	GroupMarginsAndPadding
Mise en forme de contrôle	GroupMarginsAndPaddingControlLayout
Mise en page	GroupPageLayoutAccess
Mise en page	TabReportToolsPageSetupDesign
Mise en page	TabReportToolsPageSetupLayout
Mise en page...	PageSetupDialog
Mode	ViewsModeMenu
Mode Création	ViewsDesignView
Mode État	ViewsReportView
Mode Feuille de données	ViewsDatasheetView
Mode Formulaire	ViewsFormView

Mode Graphique croisé dynamique	ViewsPivotChartView
Mode Page	ViewsLayoutView
Mode SQL	ViewsAdpDiagramSqlView
Mode Tableau croisé dynamique	ViewsPivotTableView
Modèles de tables	CreateTableTemplatesGallery
Modes Tables	AdpDiagramTableModesMenu
Modifier des relations	RelationshipsEditRelationships
Modifier l'affichage personnalisé	AdpDiagramModifyCustomView
Modifier le flux de travail	TableSharePointListsModifyWorkflow
Modifier le langage SQL	AdpStoredProcedureEditSql
Modifier le type de graphique	PivotChartType
Modifier les colonnes et les paramètres	TableSharePointListsModifyColumnsAndSettings
Module	CreateModule
Module de classe	CreateClassModule
Mosaïque horizontale	WindowsTileHorizontally
Mosaïque verticale	WindowsTileVertically
Moyenne	AutoSumAverage
Moyenne	PivotAutoCalcAverage
N'afficher que la partie inférieure	PivotShowOnlyTheBottomMenu
N'afficher que la partie supérieure	PivotShowOnlyTheTopMenu
NB	AutoSumCount
NB	PivotAutoCalcCount
Ne commence pas par la sélection	FilterDoesNotBeginsWithSelection
Ne contient pas la sélection	FilterDoesNotContainSelection
Ne correspond pas à la sélection	FilterNotEqualsSelection
Ne se termine pas par la sélection	FilterDoesNotEndWithSelection
Nom de fenêtre	WindowNameGoesHere
Nom uniquement	AdpDiagramNameOnly
Noms de colonne	AdpDiagramColumnNameNames
Noms de macro	MacroNames
Noms des tables	QueryTableNamesShowHide
Non sélectionné	FilterIsNotSelected
Normal	PivotShowAsNormal
Nouveau	FileNewDatabase
Nouveau	GoToNewRecord
Nouveau champ	DatasheetNewField
Nouvelle étiquette	AdpDiagramNewLabel
Nouvelle table	AdpDiagramNewTable
Numéro de page	HeaderFooterPageNumberInsert
Numérotation	Numbering
Objet :	ObjectGallery
Ombre	ControlSpecialEffectShadowed
Opérations de sortie	GroupAdpOutputOperations
Options	SourceControlOptions
Options avancées	CreateOtherObjectsMenu
Options avancées	FilterAdvancedMenu
Options de réplication	ReplicationOptionsMenu
Ordre de tabulation...	TabOrder
Organisation de la sélection	AdpDiagramArrangeSelection
Organisation des tables	AdpDiagramArrangeTables
Orthographe	SpellingAccess
Outils	GroupAdpQueryTools
Outils	GroupAdpSqlStatementDesignTools

Outils	GroupFieldsTools
Outils	GroupMacroTools
Outils	GroupPivotChartTools
Outils	GroupPivotTableToolsAccess
Outils	GroupRelationshipsTools
Outils	GroupTableDesignTools
Outils	GroupToolsAccess
Outils de base de données	GroupDatabase Tools
Outils de base de données	TabDatabaseTools
Outils de création de formulaire	TabSetFormTools
Outils de création d'état	TabSetReportTools
Outils de diagramme	TabSetAdpDiagram
Outils de fonction et affichage	TabSetAdpFunctionAndViewTools
Outils de graphique croisé dynamique	TabSetPivotChartAccess
Outils de macro	TabSetMacroTools
Outils de présentation de formulaire	TabSetFormToolsLayout
Outils de présentation d'état	TabSetReportToolsLayout
Outils de procédure stockée	TabSetAdpStoredProcedure
Outils de relation	TabSetRelationshipTools
Outils de requête	TabSetQueryTools
Outils de schéma	GroupSchemaTools
Outils de table	TabSetTableToolsDatasheet
Outils de table	TabSetTableToolsDesign
Outils de tableau croisé dynamique	TabSetPivotTableAccess
Outils d'instruction SQL	TabSetAdpSqlStatement
Ouvrir	FileOpenDatabase
Par ordre croissant	AdpOutputOperationsSortAscending
Par ordre décroissant	AdpOutputOperationsSortDescending
Par total	PivotChartSortByTotalMenu
Paramétrage de requête	GroupQuerySetup
Paramètres	QueryParameters
Partager des objets	SourceControlShareObjects
Pas à pas	MacroSingleStep
Paysage	PageOrientationLandscape
PDF ou XPS	PublishToPdfOrEdoc
Personnalisé	TabSetFormReportExtensibility
Personnaliser la barre d'outils Accès rapide...	QuickAccessToolbarCustomization
Pièce jointe	ControlAttachment
Plat	ControlSpecialEffectFlat
Plus	ExportMoreMenu
Plus	ImportMoreMenu
Plus	RecordsMoreRecordsMenu
Plus de formulaires	CreateFormMoreFormsGallery
Plus de pages	PrintPreviewMultiplePagesMenu
Plus grand que la sélection	FilterLargerThanSelection
Plus petit que la sélection	FilterSmallerThanSelection
Plusieurs éléments	CreateFormWithMultipleItems
Police	GroupFontAccess
Police	GroupTextFormatting
Police :	Font
Portrait	PageOrientationPortrait
Position	GroupControlPositionLayout

Position	GroupPosition
Position	GroupPositionLayout
Pourcentage d'éléments de colonnes parentes	PivotShowAsPercentOfParentColumnItem
Pourcentage d'éléments de lignes parentes	PivotShowAsPercentOfParentRowItem
Pourcentage du total de la colonne	PivotShowAsPercentOfColumnTotal
Pourcentage du total de la ligne	PivotShowAsPercentOfRowTotal
Pourcentage du total général	PivotShowAsPercentOfGrandTotal
Précédent	MailMergeGoToPreviousRecord
Premier	MailMergeGoToFirstRecord
Presse-papiers	GroupClipboard
Presse-papiers Office...	ShowClipboard
Procédure stockée	CreateStoredProcedure
Propriétés	SourceControlProperties
Propriétés de la base de données	FileDatabaseProperties
Propriétés des colonnes	AdpDiagramColumnProperties
Propriétés du serveur	ServerProperties
Publier	MenuPublish
Publier au format PDF ou XPS	FileSaveAsPdfOrXps
Puces	Bullets
Quadrillage	GridlinesGallery
Quadrillage	GroupDesignGridlines
Quadrillage	GroupFormattingGridlines
Quatre pages	PrintPreviewFourPages
Rapport de relations	RelationshipsReport
Recalculer les sauts de page	AdpDiagramRecalculatePageBreaks
Rechercher	GroupFindAccess
Rechercher des compléments pour d'autres formats de fichier	AdvertisePublishAs
Rechercher...	FindDialog
Rectangle	ControlRectangle
Récupérer un document de création maître...	ReplicationRecoverDesignMaster
Réduire le champ	PivotCollapseFieldAccess
Réduire tout	RecordsCollapseAllSubdatasheets
Règle	RulerShowHide
Regroupement et totaux	GroupGroupingAndTotals
Regrouper et trier	Grouping
Relations	AdpDiagramRelationships
Relations	DatabaseRelationships
Relations	GroupDatasheetRelationships
Relations	GroupRelationships
Remplacer...	ReplaceDialog
Remplissage du contrôle	ControlPaddingGallery
Renommer	DatasheetColumnRename
Renvoyer :	QueryReturnGallery
Réorganiser	TabControlLayout
Réorganiser	TabFormToolsLayout
Réorganiser	TabReportToolsAlignment
Réorganiser	TabReportToolsLayout
Réorganiser les icônes	WindowsArrangeIcons
Réplica partiel	DatabasePartialReplica
Reproduire la mise en forme	FormatPainter

Requête Ajout de valeurs	AdpStoredProcedureQueryAppendValues
Requête Ajout...	AdpStoredProcedureQueryAppend
Requête Création de table...	AdpStoredProcedureQueryMakeTable
Requête Mise à jour	AdpStoredProcedureQueryUpdate
Requête Sélection	AdpStoredProcedureQuerySelect
Requête Suppression	AdpStoredProcedureQueryDelete
Requis	FormattingRequiredField
Résoudre les conflits...	ReplicationResolveConflicts
Restaurer la base de données SQL	ServerRestoreSqlDatabase
Résultats	GroupQueryResults
Rétablir	Redo
Rétablir la liaison des listes	AccessRelinkLists
Revenir	Revert
Saisie de données	WindowsDataEntry
Sauvegarde de la base de données SQL	FileBackUpSqlDatabase
Sauvegarder la base de données	FileBackupDatabase
Schéma	AdpViewDiagramPane
Schéma	CreateDiagram
Se termine par la sélection	FilterEndsWithSelection
Sélection	QuerySelectQueryType
Sélection	SortSelectionMenu
Sélectionné	FilterIsSelected
Sélectionner	SelectMenuAccess
Sélectionner	SelectRecord
Sélectionner les objets	ObjectsSelect
Sélectionner tout	SelectAllAccess
Sélectionner tout	SelectAllRecords
Sélections	GroupPivotTableSelections
Serveur	FileServerMenu
Serveur de gestion des documents	FilePublishToSharePoint
Snapshot Viewer	ExportSnapshot
Somme	PivotAutoCalcSum
Somme	TotalsSum
Souligner	Underline
Sous-feuille de données	RecordsSubdatasheetMenu
Sous-feuille de données...	RecordsInsertSubdatasheet
Sous-formulaire	SubformMenu
Sous-formulaire dans une nouvelle fenêtre	SubformInNewWindow
Sous-formulaire/état	ControlSubFormReport
Sous-total	PivotSubtotal
SQL	AdpViewSqlPane
SQL direct	QuerySqlPassThroughQuery
SQL Server	DatabaseSqlServer
Style	GridlinesStyleGallery
Suite des fenêtres...	WindowMoreWindowsDialog
Suivant	FindNext
Suivant	MailMergeGoToNextRecord
Suppression	QueryDelete
Supprimer	AdpOutputOperationsTableRemove
Supprimer	ControlLayoutRemove
Supprimer	DatasheetColumnDelete
Supprimer	Delete

Supprimer	RecordsRemoveSubdatasheet
Supprimer la colonne	RecordsDeleteColumn
Supprimer la table de la base de données	AdpDiagramDeleteTable
Supprimer la tabulation	DeleteTab
Supprimer le champ	PivotRemoveField
Supprimer l'enregistrement	RecordsDeleteRecord
Supprimer les colonnes	TableColumnsDelete
Supprimer les lignes	TableRowsDelete
Supprimer tous les tris	SortRemoveAllSorts
Supprimer une base de données SQL	FileDropSqlDatabase
Sur la grille	ControlAlignToGrid
Sur la grille	SizeToGridAccess
Synchroniser	SynchronizeData
Synchroniser maintenant...	ReplicationSynchronizeNow
Table	CreateTable
Tables	GroupCreateTables
Tabulaire	ControlLayoutTabular
Taille	GroupControlSize
Taille	GroupSizeAndPosition
Taille	PageSizeGallery
Taille de police :	FontSize
Tester les règles de validation	TableTestValidationRules
Texte enrichi	GroupRichText
Titre	ControlTitle
Totaux	QueryTotalsShowHide
Totaux	RecordsTotals
Totaux	TotalsMenu
Totaux croissants	PivotChartSortByTotalAscending
Totaux décroissants	PivotChartSortByTotalDescending
Trait	ControlLine
Transfert de la base de données	FileServerTransferDatabase
Trier et filtrer	GroupSortAndFilter
Type de données et mise en forme	GroupDataTypeAndFormatting
Type de données :	FormattingDataType
Type de ligne	ControlLineTypeGallery
Type de requête	GroupAdpQueryType
Type de requête	GroupQueryType
Type	GroupPivotChartType
Une page	ZoomOnePage
Union	QueryUnionQuery
Unique	FormattingUnique
Utilisateurs et autorisations	DatabasePermissionsMenu
Utiliser les Assistants contrôle	ControlWizards
Variance	PivotAutoCalcVariance
Variance	TotalsVariance
Variance (population)	PivotAutoCalcVariancePopulation
Vérifier la syntaxe SQL	AdpVerifySqlSyntax
Vertical équilibré	VerticalSpacingMakeEqual
Visual Basic	VisualBasic
Visualiser le code	ViewVisualBasicCodeAccess
Word	ExportWord
Zone de groupe	FormControlGroupBox

Zone de liste	FormControlListBox
Zone de liste déroulante	FormControlComboBox
Zone d'édition	FormControlEditBox
Zones de dépôt	PivotDropAreas
Zoom	GroupZoom
Zoom	PrintPreviewZoomMenu
Zoom	ZoomPrintPreviewExcel
Zoom 100 %	Zoom100

XI - Annexe B : Liste des fonctions CallBack

Contrôle	Attribut	Procédure
all controls	getEnabled	sub GetEnabled(control as IRibbonControl, ByRef enabled)
all controls	getImage	sub GetImage(control as IRibbonControl, ByRef image)
all controls	getLabel	sub GetLabel(control as IRibbonControl, ByRef label)
all controls	getSize	sub GetSize(control as IRibbonControl, ByRef size)
all controls	getSupertip	sub GetSupertip(control as IRibbonControl, ByRef screentip)
all controls	getScreentip	sub GetScreentip(control as IRibbonControl, ByRef screentip)
all controls	getVisible	sub GetVisible(control as IRibbonControl, ByRef visible)
all controls	getDescription	sub GetDescription(control as IRibbonControl, ByRef description)
Button	onAction	sub OnAction(control as IRibbonControl)
Button	onAction - repurposed	sub OnAction(control as IRibbonControl, byRef fCancelDefault)
checkBox	getPressed	sub GetPressed(control as IRibbonControl, ByRef return)
checkBox	onAction	sub OnAction(control as IRibbonControl, pressed as Boolean)
comboBox	getItemCount	sub GetItemCount(control as IRibbonControl, ByRef count)
comboBox	getItemImage	sub GetItemImage(control as IRibbonControl, index as Integer, ByRef image)
comboBox	getItemLabel	sub GetItemLabel(control as IRibbonControl, index as Integer, ByRef label)
comboBox	getItemScreen	sub GetItemScreen(control as

		IRibbonControl, index as Integer, ByRef screen)
comboBox	getText	sub GetText(control as IRibbonControl, ByRef text)
comboBox	onChange	sub OnChange(control as IRibbonControl, text as String)
customUI	onLoad	sub OnLoad(ribbon as IRibbonUI)
dropDown	getItemCount	sub getItemCount(control as IRibbonControl, ByRef count)
dropDown	getItemLabel	sub GetItemLabel(control as IRibbonControl, index as Integer, ByRef label)
dropDown	getItemScreen	sub GetItemScreen(control as IRibbonControl, index as Integer, ByRef screen)
dropDown	getSelectedItemIndex	sub GetSelectedItemIndex(control as IRibbonControl, ByRef index)
dropDown	getSelectedItemID	sub GetSelectedItemID(control as IRibbonControl, ByRef index)
dropDown	onAction	sub OnAction(control as IRibbonControl, selectedId as String, selectedIndex as Integer)
dropDown	getItemImage	sub GetItemImage(control as IRibbonControl, index as Integer, ByRef image)
dropDown	getItemID	sub GetItemID(control as IRibbonControl, index as Integer, ByRef id)
editBox	getText	sub GetText(control as IRibbonControl, ByRef text)
editBox	onChange	sub OnChange(control as IRibbonControl, text as String)
gallery	getItemLabel	sub GetItemLabel(control as IRibbonControl, index as Integer, ByRef label)
gallery	onAction	sub OnAction(control as IRibbonControl, selectedId as String, selectedIndex as Integer)
gallery	getSelectedItemIndex	sub GetSelectedItemIndex(control

		as IRibbonControl, ByRef index)
gallery	getSelectedItemID	sub GetSelectedItemID(control as IRibbonControl, ByRef index)
gallery	getItemWidth	sub getItemWidth(control as IRibbonControl, ByRef width)
gallery	getItemHeight	sub getItemHeight(control as IRibbonControl, ByRef height)
gallery	getItemScreen	sub GetItemScreen(control as IRibbonControl, index as Integer, ByRef screen)
gallery	getItemImage	sub GetItemImage(control as IRibbonControl, index as Integer, ByRef image)
gallery	getItemCount	sub GetItemCount(control as IRibbonControl, ByRef count)
gallery	getItemID	sub GetItemID(control as IRibbonControl, index as Integer, ByRef id)
toggleButton	getPressed	sub GetPressed(control as IRibbonControl, ByRef return)
toggleButton	onAction—repurposed	sub OnAction(control as IRibbonControl, pressed as Boolean, byRef fCancelDefault)
toggleButton	onAction	sub OnAction(control as IRibbonControl, pressed as Boolean)