

# Une table des matières pour chacun de vos états c'est maintenant possible. Nouvelle version

par Fabrice CONSTANS ([autres articles](#))


Date de publication : 07/05/2005

Dernière mise à jour : 1/10/2009

Grande absente des fonctionnalités de Microsoft ACCESS, la gestion de tables des matières est pourtant nécessaire. Nous allons, au cours de ce tutoriel, apprendre à en faire une totalement dynamique.

I - Avertissement.....	3
II - Pré requis.....	4
III - Définition du contenu d'une table des matières.....	5
IV - Le générateur.....	6
IV-A - Description du mécanisme.....	6
IV-B - Création du code d'écriture.....	6
IV-C - Création du code de suppression.....	6
V - Intégration.....	8
V-A - Vider la table.....	8
V-B - Le titre.....	9
V-C - Les sous-titres.....	9
V-D - Les items.....	10
VI - L'état Table des matières.....	11
VI-A - Mise en page.....	11
VI-B - Le code.....	12
VIII - Conclusion.....	14
IX - Remerciements.....	15

## I - Avertissement

 *L'utilisation de la touche F1 est vivement conseillée à tous les stades de l'utilisation d'ACCESS. L'amélioration constante de l'aide en fait un partenaire de choix dans l'apprentissage permanent d'ACCESS. Personnellement, je ne peux m'en passer, ne serait-ce que pour mémoire.*

## II - Pré requis

Nous allons utiliser la base de données d'exemple de livraison avec ACCESS : **Les Comptoirs**.  
Ceci, pour nous permettre d'être en condition réelle d'intégration.

### III - Définition du contenu d'une table des matières

Pour commencer, nous devons connaître la physionomie d'une table des matières. Pour rappel, un exemple est présenté ci-dessous.

#### Exemple d'une table des matières

Item	Titre	Page
élément1.....		1
élément2.....		2
élément3.....		2
élément4.....		3
...		

Pour obtenir un système de table des matières véritablement dynamique, son contenu devra être généré à l'impression d'un état.


La table des matières étant un état à part entière, il convient donc de stocker les informations dans une table. Un inventaire rapide permet de définir les informations suivantes :

- Un titre
- Un élément
- Un numéro de page

La table à créer devra donc comporter les champs suivants :

Nom	Type	Longueur/type
tm_Item	texte	100
tm_Page	Numérique	Entier
tm_Type	texte	2

Le champ **tm\_Type** permet de déterminer si l'information doit s'afficher en tant que titre, sous-titre ou item. Ce champ anodin nous permet de définir des niveaux différents dans notre table des matières. Dans notre exemple, **T** représentera un titre, **ST** un sous-titre et **I** un item.

 *Inutile de mettre un type Numérique Long pour le champ **tm\_Page** à moins que l'état d'origine imprime plus de 65000 pages.*

Créez la table avec le nom **tbi\_TableMat** et répondez **non** à la question de la création d'une clef primaire.

## IV - Le générateur

Le générateur fonctionne sur la base de l'impression d'un état. C'est en effet le seul moyen de connaître le numéro de page où se situe l'élément.

### IV-A - Description du mécanisme

Le mécanisme de création doit permettre à chaque impression d'enregistrer en même temps qu'une référence, un numéro de page et un type (Titre, Sous-titre, Item).

L'enregistrement peut se faire à l'aide d'une requête, mais pour plus de confort, nous utiliserons les objets d'accès aux données (ADO).

L'utilisation de l'évènement **Sur impression** pourrait faire l'affaire, mais nous ne pourrions pas faire d'aperçu avant impression de la table des matières avant d'avoir imprimé l'état émetteur.

L'évènement **Au Formatage** nous permettra de créer la table des matières directement à l'aperçu.

### IV-B - Création du code d'écriture

Pour éviter la redondance de code et permettre d'exporter rapidement cette fonctionnalité dans n'importe quelle application, nous allons créer une procédure générique.

Ouvrez un module.

Entrez le code suivant :

#### L'écriture

```
Sub fc_GenereTM(fc_Item As String, fc_Pge As Integer, fc_Typ As String)
Dim rst As Recordset

Set rst = CurrentDb.OpenRecordset("tbl_TableMat", dbOpenDynaset)

rst.FindFirst "tm_Item="" & fc_Item & """"

If rst.NoMatch Then
    rst.AddNew          ' ajout nouvel enregistrement
    rst!tm_item = fc_Item    ' écrit dans le champ
    rst!tm_page = fc_Pge
    rst!tm_type = fc_Typ
    rst.Update          ' sauvegarde
Else
    If rst!tm_page < fc_Pge Then ' problème signalé par
        rst.Edit
        rst!tm_page = fc_Pge    ' on met à jour la page
        rst.Update
    End If
End If

rst.Close          ' ferme

End Sub
```

La procédure **fc\_GenereTM** écrit la référence, la page et le type dans notre table **tbl\_TableMat**.

J'ai procédé à la correction d'une erreur importante signalée par **Olivera**. En effet il apparait que le dernier item de la page est enregistré avec le numéro de la page précédente. C'est la partie du code figurant après le Else qui gère ce problème. Il semble que l'enregistrement soit préparé avant que la place sur la page courante ne soit vérifiée. Pour preuve le 1er FormatCount (=1) indique la page précédente, le 2ème FormatCount (=2) indique la bonne page.

### IV-C - Création du code de suppression

Lorsque l'on remplit une table temporaire, il faut également songer à la vider. Nous allons créer une procédure à cet effet.

## La suppression

```
Sub fc_SupprTM()  
Dim sql As String  
  
DoCmd.SetWarnings False           ' pas de message de confirmation  
  
sql = "DELETE * FROM tbl_TableMat;" ' requete de suppression  
DoCmd.RunSQL sql  
  
DoCmd.SetWarnings True           ' message confirmation  
  
End Sub
```

Cette procédure exécute une simple requête suppression totale sur la table **tbl\_TableMat**.  
Le module étant terminé, vous pouvez l'enregistrer sous le nom **mdl\_TableMat**.

## V - Intégration

Pour notre démonstration, nous allons utiliser l'état **Liste alphabétique des produits**. Ouvrez-le en mode création.

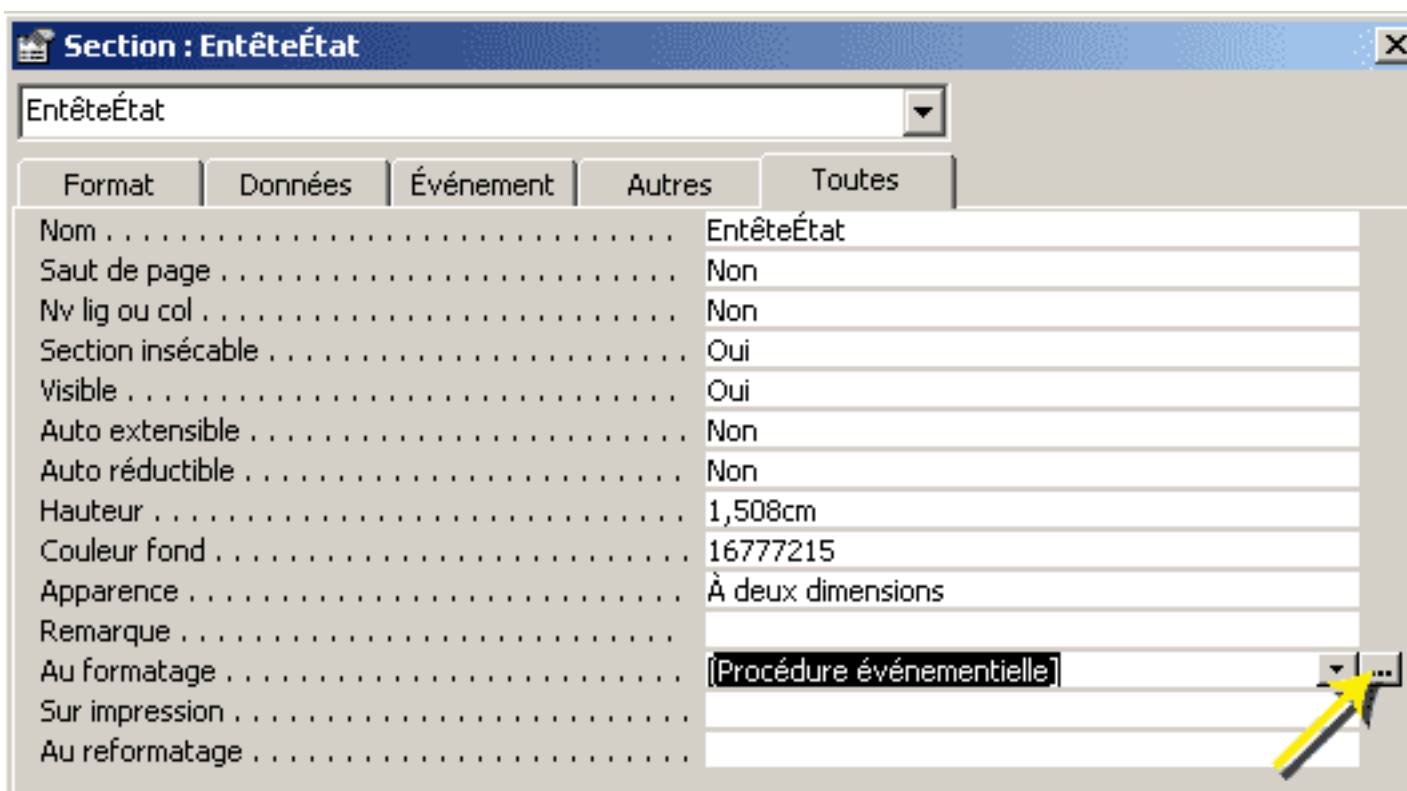
### V-A - Vider la table

Avant tout, il convient de vider la table **tbl\_TableMat**. Nous allons utiliser l'événement **Avant Ouverture** pour cette opération. Cliquez dans le carré en haut à gauche de l'état.



*Carré incontournable*

Ouvrez la fenêtre de propriétés et cliquez sur l'onglet **Évènement**. Dans la propriété **Avant ouverture**, sélectionnez **[Procédure événementielle]** et ouvrez le module à l'aide du bouton suivant.



*Un bouton vraiment utile !*

Insérez ce code :

Petit code, gros effet

```
fc_SupprTM
```

Le code est maintenant en place. A chaque ouverture de l'état, la table **tbl\_TableMat** sera vidée.



## V-B - Le titre

La première information concerne le titre de la table des matières.

Cliquez sur la zone **En-tête d'état** et dans la fenêtre propriété, cliquez sur l'onglet **Evènement**. Dans la propriété **Au formatage**, sélectionnez [**Procédure événementielle**] et ouvrez le module correspondant.

Insérez ce code :

Le code qui enregistre le titre.

```
fc_GenereTM Me.Titre.Caption, Me.Page, "T" ' écriture dans la table "table des matières"
```

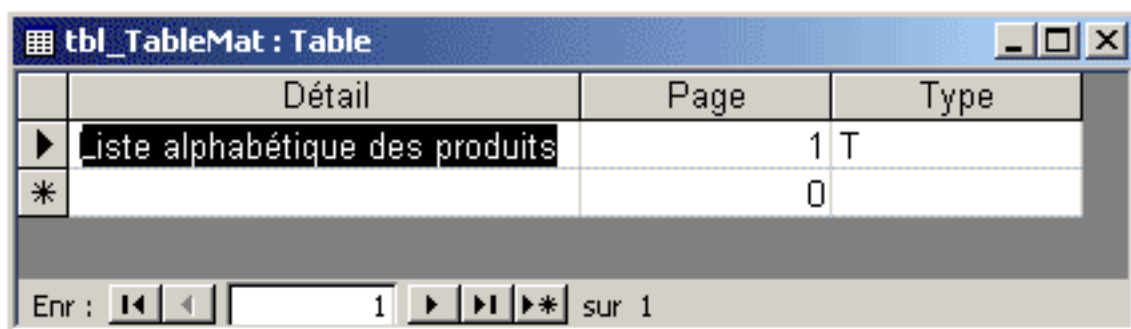
Le détail de chaque commande est commenté ci-dessous.

Commande	Description
fc_GenereTM	Appel de la procédure d'écriture de l'enregistrement.
Me.Titre.Caption	Valeur de l'étiquette nommée <b>Titre</b>
Me.Page	Le numéro de la page courante
"T"	Identification du titre de la table des matières

Faites un essai pour vérifier que la table **tbl\_tableMat** est correctement renseignée.

Pour cela, cliquez sur l'icône d'**aperçu avant impression**, puis, avec la touche **F11**, faites apparaître la fenêtre **Base de données**. Cliquez sur l'onglet **Table**, puis sur l'objet **tbl\_TableMat**.

Vous devriez voir ceci.



*La table tbl\_TableMat*

La génération du titre est terminée.

## V-C - Les sous-titres

Cliquez sur l'état et passez en mode **Création**. Sélectionnez la zone d'**En-tête de groupe Nom du produit**. Dans la propriété **Au formatage**, sélectionnez [**Procédure événementielle**] et ouvrez le module.

Insérez ce code :

Récupération des sous-titres.

```
fc_GenereTM Me.PremièreLettreDuNom, Me.Page, "ST"
```

Comme précédemment, chaque commande est commentée ci-dessous.

Commande	Description
fc_GenereTM	Appel de la procédure d'écriture de l'enregistrement.
Me.PremièreLettreDuNom	Valeur du contrôle <b>PremièreLettreDuNom</b>
Me.Page	Le numéro de la page courante
"ST"	Identification d'un sous-titre de la table des matières

Vous noterez que le travail est répétitif, cela est essentiellement dû au fait que nous avons créé un code générique.

## V-D - Les items

Pour les items, procédez de la même manière que pour les cas précédents, mais avec la zone **Détails**. Dans la procédure, insérez le code suivant :

### Récupération des items.

```
fc_GenereTM Me.Nom_du_produit, Me.Page, "I"
```

Commande	Description
fc_GenereTM	Appel de la procédure d'écriture de l'enregistrement.
Me.Nom_du_produit	Valeur du contrôle <b>Nom du produit</b>
Me.Page	Le numéro de la page courante
"I"	Identification d'un item de la table des matières

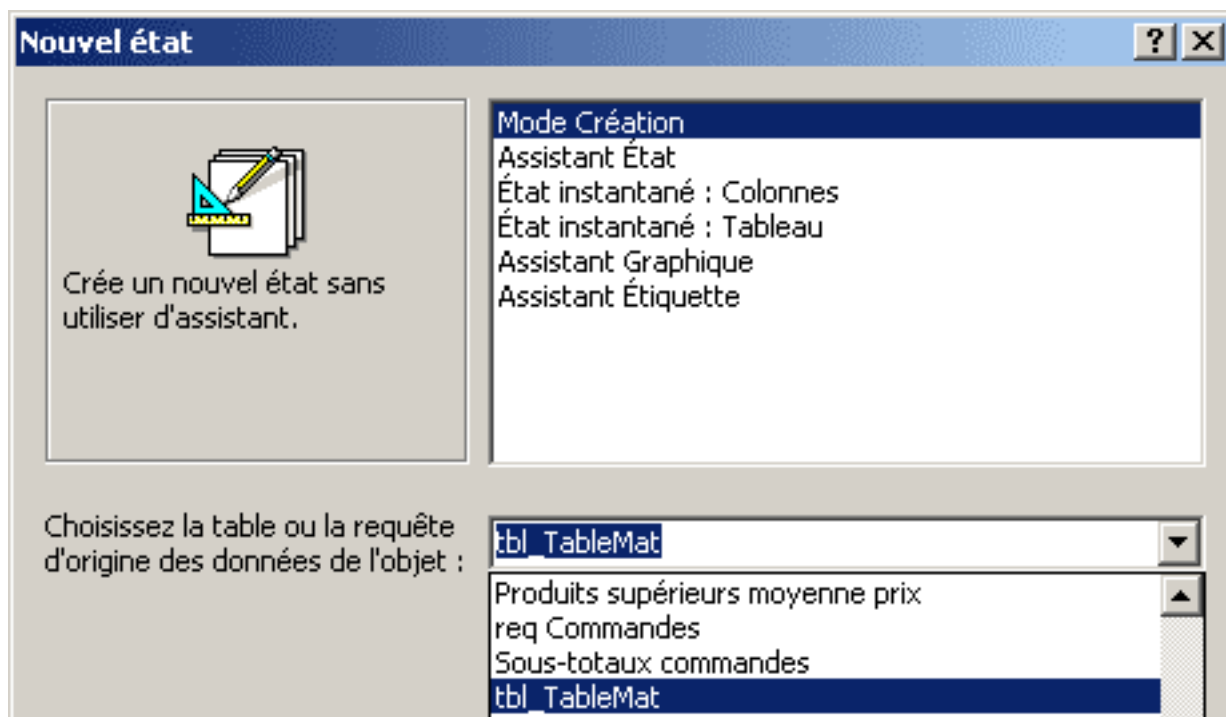
La partie qui génère la table des matières est terminée. La quantité de code nécessaire n'est pas très importante, comme vous pouvez le constater.

## VI - L'état Table des matières

Ce chapitre est l'aboutissement de ce tutoriel. En effet, nous allons maintenant créer l'état **Table des matières**.

### VI-A - Mise en page

Créez un nouvel état.



*Comment créer un état.*

Ajoutez une zone d'**En-tête d'état**. *Menu Affichage/En-tête et pied de rapport.*  
 A l'aide de la liste des champs, insérez dans l'**En-tête d'état** le champ **tm\_Item**.  
 Changez les propriétés suivantes de **tm\_Item** :

Propriété	Valeur
Nom	tm_titre
Taille de police	18
Epaisseur police	Gras

Réduisez la hauteur de l'**En-tête de page** et du **Pied d'état** le plus possible et le **Pied de page** de moitié.  
 Dans le **Détail** insérez les champs suivants :

- tm\_Type
- tm\_Item
- tm\_Page

Insérez une étiquette avec une série de points à intituler **tm\_Points**.  
 Ajustez le tout pour avoir la présentation suivante.

En-tête d'état

# tm\_Item

En-tête de page

Détail

tm_Item	.....	tm_Pag
---------	-------	--------

Pied de page

Pied d'état

### L'état en mode création

Sélectionnez la totalité des champs du **Détail** un par un (**Shift+clac gauche**) et changer les propriétés suivantes :

Propriété	Valeur
Style fond	Transparent
Style bordure	Transparent

Une fois cette mise en forme effectuée, passez à l'écriture du code.

## VI-B - Le code

Sélectionnez le **Détail** et créez une procédure événementielle sur la propriété **Au formatage**. Dans la section des déclarations générales, créez les constantes suivantes :

### La déclaration des constantes.

```
Option Compare Database
Const Blanc = 16777215
Const Gris = 12632256
```

Ces constantes contiennent les couleurs gris et blanc, qui permettent d'alterner sous-titres et items dans le détail de l'état.

Dans la procédure **Détail\_Format** entrez le code suivant :

### Le code principal

```
' rend visible les contrôles et étiquettes
Me.tm_Item.Visible = True
Me.tm_Page.Visible = True
Me.tm_Points.Visible = True
' cache systématiquement tm_Type
Me.tm_Type.Visible = False
' La hauteur d'origine
Me.Détail.Height = 0.423

' suivant le type T,ST,I
Select Case Me.tm_Type
Case "T" ' traitement du titre (ne doit pas apparaître dans détail)
Me.tm_Item.Visible = False
Me.tm_Type.Visible = False
Me.tm_Page.Visible = False
Me.tm_Points.Visible = False
```

### Le code principal

```

Me.Détail.Height = 0
Case "ST" ' traitement des sous-titre (apparaît en grisé)
Me.tm_Item.FontSize = 10
Me.tm_Item.FontBold = True
Me.Détail.BackColor = Gris
Case "I" ' traitement des items (apparaît sur fond blanc)
Me.tm_Item.FontSize = 8
Me.tm_Item.FontBold = False
Me.Détail.BackColor = Blanc
End Select
    
```

Lorsque vous définissez la valeur d'une propriété, celle-ci reste en l'état jusqu'à son changement. C'est pour cette raison que le code suivant, exécuté pour chaque enregistrement, définit certaines propriétés systématiquement.

- 1 Le titre ne doit s'afficher que dans l'**En-tête**. Il sera afficher également dans la première ligne du **Détail**. C'est pourquoi nous devons obligatoirement le cacher et réduire le **Détail** pour ne pas laisser d'espace disgracieux.
- 2 Les sous-titres sont grisés avec la propriété **BackColor**
- 3 Les items sont sur fond blanc toujours avec la propriété **BackColor**.

Notez que les sous-titres et les items s'intercalent et qu'il convient de définir les propriétés à chaque passage.

## Liste alphabétique des produits

<b>A</b>	.....	<b>1</b>
Aniseed Syrup	.....	1
<b>B</b>	.....	<b>1</b>
Boston Crab Meat	.....	1
<b>C</b>	.....	<b>1</b>
Camembert Pierrot	.....	1
Camarvon Tigers	.....	1
...	.....	.

*Un résultat du plus bel effet...*

Faites un essai en cliquant sur le bouton **aperçu avant impression**. Nous pouvons sauvegarder l'état.

## VIII - Conclusion

Nous avons une nouvelle fois démontré la souplesse d'ACCESS au travers de ce tutoriel. Cette fonctionnalité se veut la plus générique possible pour permettre une intégration facile et rapide dans d'autres états ou applications. Nous avons volontairement utilisé un type alphabétique explicite (T,ST,I) pour une meilleure compréhension du mécanisme de fonctionnement. Cependant, sachez que vous pouvez créer plus de niveaux en utilisant un type numérique.

## IX - Remerciements

Je tiens à remercier : **Cerbères** pour la qualité de ses corrections.

A l'équipe de **Developpez.com** pour ce site génial.

A **Nono40** pour son super éditeur XML qui se bonifie avec le temps, comme un vieux Pommard.

A **Maxence Hubiche** pour le temps qu'il passe à publier mes articles. Je présente mes plus plates excuses à ceux que j'aurais omis de remercier. A **Olivera** pour m'avoir indiqué cette erreur de numérotation de page.