

Utiliser un lecteur de code-barres (douchette) sur Excel et Access

Si vous gérez des stocks, vous vous servez probablement des codes-barres. Pour gagner du temps, vous pouvez utiliser un **lecteur de code-barres (ou douchette)** afin de ne pas saisir les numéros un par un et risquer de commettre des erreurs de frappe. Cet outil peut être également utilisé avec **Excel** ou **Access** et c'est ce que nous allons découvrir dans ce tutoriel (si vous souhaitez utiliser le même lecteur de code-barres que moi, [rendez-vous à la fin de l'article](#)).

Vous retrouverez, dans l'[Espace Membres](#), dans un fichier txt, les codes que je vais utiliser dans la suite de ce tutoriel.

Une **douchette** est un périphérique d'entrée et s'utilise de la même manière qu'un clavier, c'est-à-dire que vous scannez les données pour que votre ordinateur les traite.

Que ce soit sur Excel ou Access, le scan unique d'un code-barre permet la recherche du code-barres dans votre base de données. Pour cet usage, vous pouvez utiliser comme base (il faudra bien sûr personnaliser les champs) l'un des tutoriels suivants :

- [Créer un formulaire personnalisé pour saisir des données sur Excel](#) ainsi que [son explication](#) si vous gérez vos stocks sur Excel.

Saisie des coordonnées clients

Code client

Civilité

Prénom

Nom

Adresse

Code Postal

Ville

Téléphone

E-mail

Liste déroulante de recherche

Nouveau contact Modifier Quitter

Si le paramètre *MatchEntry* de votre liste de recherche est réglé sur *1 - fmMatchEntryComplete*, lors du scan du code-barre, les données correspondantes seront automatiquement reportées dans les champs grâce au code que nous avons programmé pour la liste des numéros client. ;

- [Rechercher un enregistrement avec une liste déroulante dans un formulaire Access](#) si c'est sur Access.



Clients	
	<input type="text" value=""/>
ID Client	<input type="text" value=""/>
Raison sociale	ENTREPRISE A
Civilité	Mme
Prénom	Emma
Nom du contact	MARTIN
Adresse	2, rue de l'Église
Code postal	95690
Ville	NESLES-LA-VALLÉE
Téléphone	01 25 68 97 45
Email	emma.martin@yahoo.fr
Date de démarrage	12/01/2012

En scannant votre code-barre tout en étant placé sur la liste déroulante de recherche, cela vous affiche les données correspondantes de l'enregistrement si votre liste est basée sur votre champ de code-barres.

Dans ce tutoriel-ci, c'est plutôt l'utilisation en mode inventaire que nous allons étudier afin de scanner "en rafale" tout votre stock (ou en tout cas, plusieurs code-barres à la suite). De cette manière, vous pouvez faire l'inventaire de votre stock et ne pas avoir forcément votre fichier de stock à portée de main. De plus, c'est beaucoup plus rapide de tout scanner en une fois avant

d'envoyer tous les codes-barres sur votre application, sauf si vous souhaitez saisir la quantité stockée à chaque entrée.

Utiliser un lecteur de code-barres pour faire un inventaire sur Excel :

Pour la démonstration avec Excel, je vais utiliser le même fichier que lors du [tutoriel sur la création de formulaire Excel VBA](#) auquel j'ai simplement modifié le nom des champs (dans la feuille Excel et dans le formulaire) et remplacé les données présentes pour que cela ressemble plus à une gestion de stocks. J'ai également modifié la source de la ComboBox2 pour mettre A, B et C et réglé le *MatchEntry* de la liste déroulante des codes-barres sur 2 – *fmMatchEntryNone*. Enfin, dans la feuille Excel, j'ai créé un [format personnalisé](#) pour les données de la colonne A (pour les codes-barres) : # ##### ######.

Vous pouvez donc, vous aussi, partir du même fichier qu'utilisé dans le tutoriel de base et modifier la boucle *For I = 1 To 7* en *For I = 1 To 3* aux 3 endroits où elle se trouve.

Pour ajouter un code-barre, je vais choisir le champ *ComboBox1* et je vais programmer son retour à la ligne dans le code VBA, car les codes-barres, une fois le transfert lancé, seront insérés, par défaut, les uns à la suite des autres (ce qui n'est pas le but).

Dans la fenêtre VBA, je saisis ce code :

```
Private Sub ComboBox1_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)
Dim L As Integer
If KeyCode = 13 Then
L = Sheets("Clients").Range("A65536").End(xlUp).Row + 1
Range("A" & L).Value = Format(ComboBox1, "# ##### ######")
Unload Me
UserForm1.Show
End If
End Sub
```

Explication de la procédure VBA :

L'évènement programmé est l'appui sur une touche (*KeyDown*).

Comme il ne faut pas que n'importe quelle touche déclenche l'action, on ajoute la condition *KeyCode = 13* qui correspond au retour à la ligne.

La déclaration et la valeur de *L* sont les mêmes que dans le tutoriel de base pour pouvoir ajouter notre saisie à la suite des autres déjà présentes dans la feuille Excel.

Pour insérer le code-barre avec les mêmes espaces que dans le code-barre original, je transforme le format de la saisie dans la ComboBox1 par # ##### ######.

Enfin, on vide l'userform en le fermant et on l'ouvre de nouveau avec *Unload Me* et *UserForm1.Show*.

Toutes ces actions vont donc se produire à chaque fois que la douchette passera au code suivant en faisant un retour à la ligne lors du transfert des données du mode inventaire.


Pour terminer, si votre combobox pour la saisie et recherche des codes-barres n'est pas le champ qui récupère le focus lors de l'affichage du formulaire, il vous faudra ajouter le code suivant à l'activation du formulaire pour être positionné sur le bon champ :


```
Private Sub UserForm_Activate()  
Me.ComboBox1.SetFocus  
End Sub
```

N'oubliez pas de désactiver la procédure *ComboBox1_KeyDown* en mettant une apostrophe (') devant chaque ligne de cette dernière (la ligne deviendra verte) lorsque vous aurez fini d'inventorier tous vos produits. En laissant ce code sans le désactiver, quand vous scannerez un code-barre dans la liste pour le rechercher, il sera ajouté à la suite de votre liste comme lors de votre inventaire. Quand vous souhaiterez réutiliser ce code, il vous suffira de retirer les apostrophes.

Utiliser un lecteur de code-barres pour faire un inventaire sur Access :

Pour effectuer une saisie de codes-barres en rafale sur Access, je pars du fichier du [tutoriel sur la recherche d'enregistrement via une liste déroulante](#) où je vais importer le fichier d'Excel en utilisant l'onglet **Données externes** d'Access et laisser ce dernier créer la clé primaire (il ne faut pas que la clé primaire soit le code-barres). Bien sûr, pour faire les choses bien, il faudrait renommer les champs pour ne pas avoir d'accents ou d'espaces, mais ce n'est pas le plus important dans ce tutoriel. Créez rapidement le formulaire de saisie basé sur cette nouvelle table et [créez la liste déroulante de recherche](#) si vous le souhaitez.

En **Mode création** , sélectionnez le champ Code-barre et, dans la **Feuille**

de propriétés , à l'onglet *Évènement*, cliquez sur les 3 points au bout de la ligne *Sur touche activée*, choisissez le **Générateur de code** et saisissez le code suivant :

```
If KeyAscii = vbKeyReturn Then  
DoCmd.GoToRecord , , acNext  
End If
```

Explication de la procédure VBA :

Tout comme sur Excel, l'action est déclenchée sur l'appui d'une touche (*KeyPress*). On filtre pour que l'évènement ne se déclenche que s'il s'agit de la touche de retour à la ligne (*vbKeyReturn*).

Puis, on se positionne sur l'enregistrement suivant, mais dans le même champ, contrairement à son comportement de base qui aurait placé les saisies les unes à côté des autres dans des champs différents.

Si vous souhaitez utiliser le même lecteur de code-barres (douchette) que celui que j'ai utilisé dans ce tutoriel, vous pouvez le commander en cliquant sur l'image :



Article sponsorisé : la [marque Aukey](#) m'a offert la douchette utilisée pour réaliser ce tutoriel.

Tutoriel réalisé avec Excel et Access 2016

[Voir la version vidéo de cet article](#)

*Article écrit par Lydia Provin du site
Votre Assistante : <https://www.votreassistante.net>*