

Introduction à Access VBA (article invité)

Suite à des demandes d'abonnés concernant le VBA, j'ai proposé à [Hervé Inisan du blog Le grenier Access](#) d'écrire un article invité à ce sujet, comme je l'ai proposé il y a quelque temps à Thierry, mais, cette fois-ci, sur Access 😊.

Bonjour à tous, je me présente : Hervé Inisan, consultant et formateur en informatique. J'ai été gentiment invité par Lydia à vous proposer un article sur Microsoft Access, qui est l'un de mes domaines d'intervention.

À ce titre, j'ai écrit une dizaine d'ouvrages sur Access, et j'ai depuis 2001 la mention "Most Valuable Professional" (MVP) attribuée par Microsoft.

Enfin, j'anime le blog [Le Grenier Access](#) avec des tutoriels, des forums...

Voici une petite découverte de **Visual Basic pour Applications**, dans sa variante Access (VBA pour Access). Les concepts qui sont expliqués dans la suite sont également applicables à Excel (les exemples, par contre, ne fonctionneront que sous Access).

On démarrera par une introduction (un peu longue, mais nécessaire !) pour les personnes qui ne connaissent pas VBA, avant de terminer par un petit exemple pratique.

Qu'est-ce que VBA ?

Le VBA est un langage de programmation intégré à de nombreux logiciels Microsoft (et non Microsoft, d'ailleurs). Ce langage est fourni pour automatiser des tâches dans votre logiciel ou en étendre les possibilités.

Ai-je vraiment besoin de VBA dans Access ?

À terme, oui, sans doute, si vous construisez une application professionnelle, ergonomique et évolutive.

Les raisons ? Si vous pratiquez Access, vous savez que les formulaires constituent l'interface graphique de votre base de données (l'endroit où les utilisateurs consultent les données, les mettent à jour). Maintenant, dès que votre projet va grandir, vous allez vite vouloir améliorer

l'ergonomie de votre application, ajouter des contrôles de saisie ou automatiser un grand nombre de tâches (par exemple, un publipostage avec Word). C'est là qu'intervient VBA...

Quelle est la différence entre les macros Access et le VBA ?

En fait, il y a trois langages de programmation dans Access, pas moins !

- **Le langage SQL** sert à manipuler les données : c'est grâce à lui que les requêtes fonctionnent ; en d'autres termes, lorsque vous créez une requête graphiquement, Access écrit du code SQL en arrière-plan, pour interroger vos tables.
- **Le langage de macros** permet de programmer assez simplement, en français, et par le biais de listes déroulantes. Malgré son nom, il n'a strictement aucun rapport avec les macros Excel, qui se programment en VBA (ci-dessous).
- **Le langage VBA** (qui nous intéresse ici) : plus complexe que les macros Access, mais également bien plus riche.

Les macros sont tentantes, parce que plus simples. Mais elles ont également des limites, et sont rapidement difficiles à maintenir et à faire évoluer. C'est pour ces raisons que VBA est un choix plus intéressant.

Où taper du code VBA ?

Le code VBA est tapé dans des **modules**. Un module est un simple container de texte, une page blanche façon Word, dans lequel vous tapez du langage VBA.

Un peu comme dans Word, vous devrez respecter une certaine écriture dans votre code informatique : un langage informatique dispose d'une structure, d'une grammaire, d'une orthographe. Un guillemet oublié ou une virgule mal placée, et votre programme ne fonctionnera pas !

La mauvaise nouvelle 😊

Excel dispose d'un enregistreur de macros, qui peut écrire du code VBA automatiquement, pendant que vous faites des actions classiques (déplacement, copier/coller, création d'un

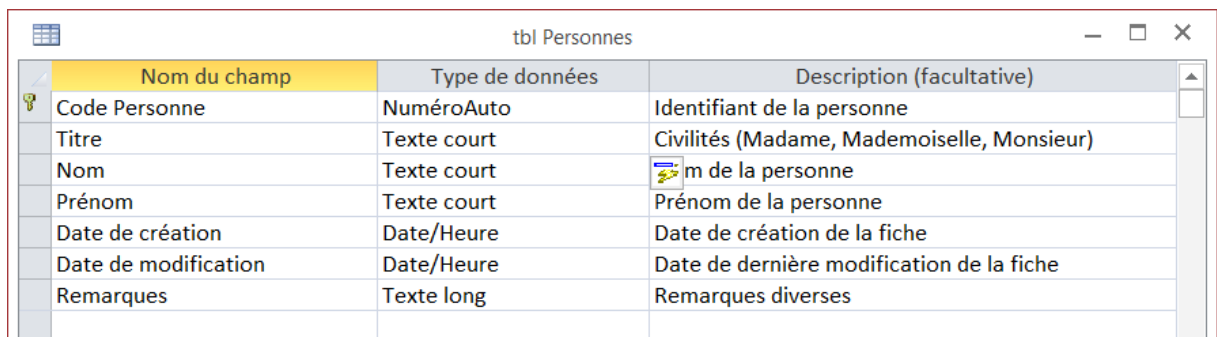
graphique, etc.). Très pratique pour écrire des macros Excel sans connaître le langage VBA sur le bout des doigts !

Malheureusement, Access ne propose pas vraiment d'outil équivalent : vous devrez donc écrire le VBA par vous-même... 😊

Un petit exemple

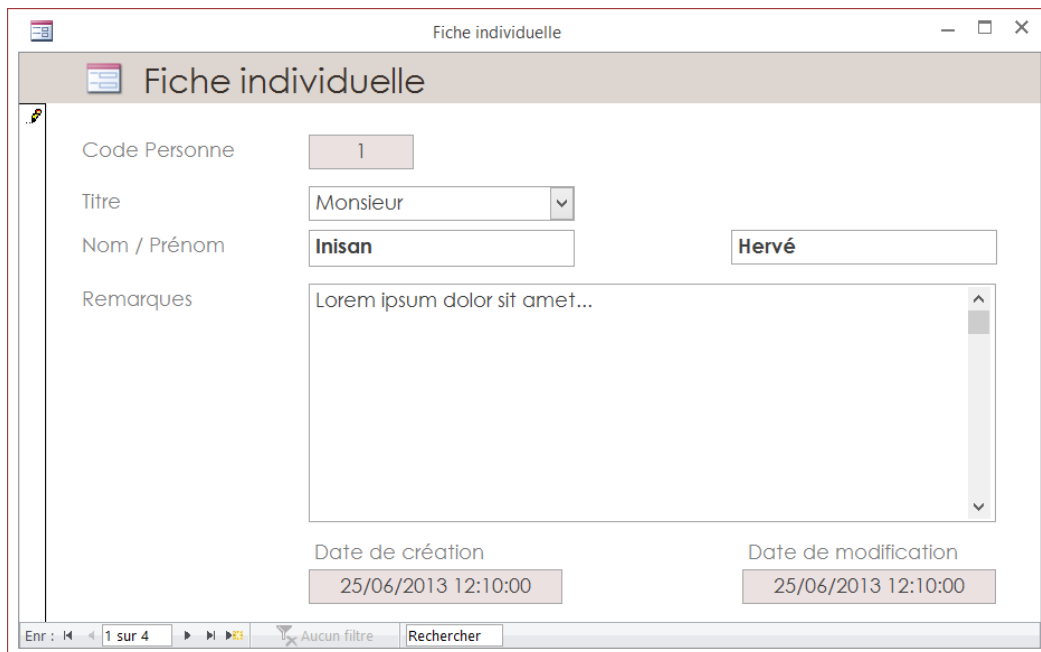
Voici un petit exemple pratique, qui permet d'illustrer certaines techniques VBA. Le scénario est le suivant :

- Ma base de données Access dispose d'une table de personnes (appelée **tbl Personnes**). Cette table contient seulement quelques champs pour la démonstration, vous pouvez bien sûr l'enrichir.



Nom du champ	Type de données	Description (facultative)
Code Personne	NuméroAuto	Identifiant de la personne
Titre	Texte court	Civilités (Madame, Mademoiselle, Monsieur)
Nom	Texte court	m de la personne
Prénom	Texte court	Prénom de la personne
Date de création	Date/Heure	Date de création de la fiche
Date de modification	Date/Heure	Date de dernière modification de la fiche
Remarques	Texte long	Remarques diverses

- J'ai construit un formulaire **frm Personnes** à partir de cette table.



Fiche individuelle

Code Personne: 1

Titre: Monsieur

Nom / Prénom: Inisan Hervé

Remarques: Lorem ipsum dolor sit amet...

Date de création: 25/06/2013 12:10:00

Date de modification: 25/06/2013 12:10:00

Enr : 1 sur 4

- L'idée est de faire en sorte qu'à chaque fois que l'utilisateur se place sur une fiche *Personne*, le *Nom* et *Prénom* de cette personne s'affichent en rappel dans la barre de titre du formulaire. Le résultat attendu est celui-ci :


The screenshot shows a Microsoft Access form window titled "Fiche de : Inisan Hervé". The form contains the following fields:

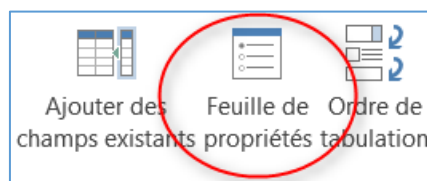
- Code Personne: 1
- Titre: Monsieur
- Nom / Prénom: Inisan Hervé
- Remarques: Lorem ipsum dolor sit amet...
- Date de création: 25/06/2013 12:10:00
- Date de modification: 25/06/2013 12:10:00

Red arrows in the image point to the text "Fiche de : Inisan Hervé" in the title bar of the form window.

Vous retrouverez cette base de données au bas de l'[article](#).

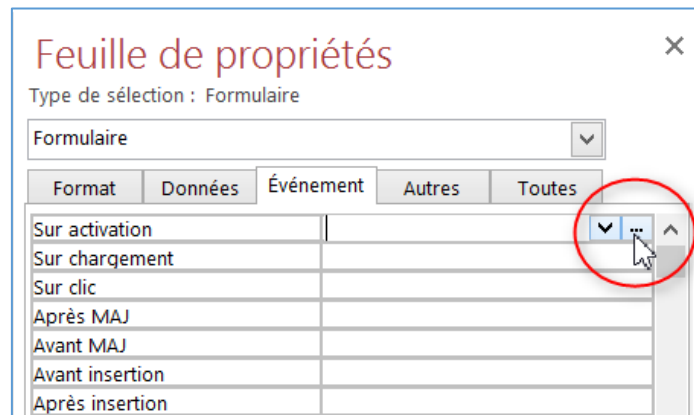
Pour obtenir ce résultat, sur Access 2007 à 2013 :

- Ouvrez votre formulaire en **Mode Création** .
- Faites apparaître les propriétés du formulaire, en cliquant sur l'icône *Feuille de propriétés*, sous l'onglet **Création** du ruban.

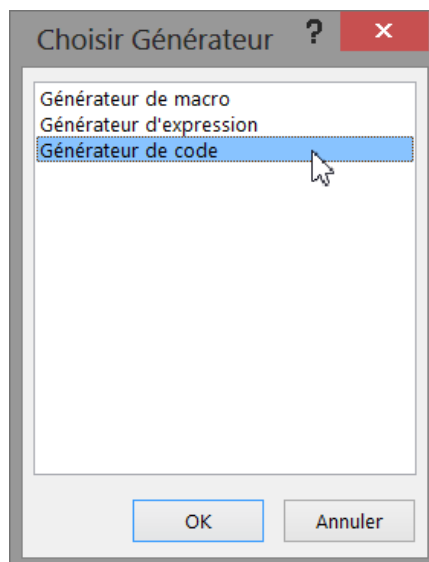


- Activez l'onglet **Événement**, dans cette fenêtre **Propriétés**.
- Cliquez dans l'événement **Sur activation** qui se produit à chaque fois que l'utilisateur change de fiche dans un formulaire (que ce soit par le clavier ou par les boutons de déplacement en bas du formulaire). Voir plus loin le paragraphe "À propos des événements".

- Cliquez sur les points de suspension à droite de l'événement.

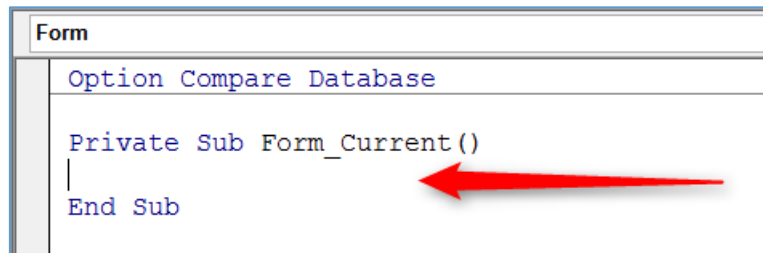


- Dans la boîte de dialogue qui se présente, choisissez l'option **Générateur de code** (sous-entendu : **Générateur de code VBA**) et cliquez sur **Ok**.



- Vous basculez dans un nouveau logiciel appelé VBE (**Visual Basic Editor**, ou l'Éditeur Visual Basic). Un module est affiché (il s'agit du module associé à notre formulaire), un bloc de code VBA est créé, et votre curseur positionné à l'intérieur.

Un bloc de code délimité ainsi par **Sub** et **End Sub** s'appelle une procédure en VBA (vous l'appelleriez une macro dans Excel ; on emploie moins ce terme "macro" dans Access VBA du fait de la confusion avec l'autre langage évoqué plus haut).



```
Form
Option Compare Database

Private Sub Form_Current()
|
End Sub
```

- À la position du curseur, ajoutez du code VBA de façon à obtenir ce qui suit. Les lignes démarrant par une apostrophe sont des **commentaires** (vous pouvez taper ce que vous voulez après l'apostrophe, comme notes personnelles ; Access ne prendra pas ces lignes en compte).

```
1 Private Sub Form_Current()
2 'Modifier la barre de titre (légende) du formulaire
3 Me.Caption = "Fiche de : " & Me.Nom & " " & Me.Prénom
4
5 'Modifier le titre placé en en-tête
6 'en y recopiant la légende du formulaire
7 Me.Auto_EnTete0.Caption = Me.Caption
8 End Sub
```

- Par précaution, il est toujours utile de cliquer sur le menu **Débogage > Compiler**. Cette option permet de préparer votre code VBA pour l'exécution, et elle en profite pour vérifier si la syntaxe est correcte. En cas d'erreur, pointez les lignes de code surlignées, rectifiez-les, puis recompilez !
- C'est terminé ! Vous pouvez ouvrir votre formulaire, vous déplacer de fiche en fiche : la barre de titre se modifie automatiquement. Certes, ce n'est pas encore spectaculaire, mais

l'ergonomie du formulaire est meilleure. Et vous avez pu tester la notion d'événement, ainsi que l'impact de VBA sur l'interface graphique.

The screenshot shows a Microsoft Access form window titled 'Fiche de : Inisan Hervé'. The form has a light gray header bar with the title. Below the header, there are several fields: 'Code Personne' with the value '1', 'Titre' with a dropdown menu showing 'Monsieur', 'Nom / Prénom' with two text boxes containing 'Inisan' and 'Hervé', and 'Remarques' with a text area containing 'Lorem ipsum dolor sit amet...'. At the bottom, there are two date fields: 'Date de création' and 'Date de modification', both showing '25/06/2013 12:10:00'. The status bar at the bottom indicates 'Enr : 1 sur 4', 'Aucun filtre', and a search box labeled 'Rechercher'. Three red arrows point to the title bar and the title text.

À propos des événements

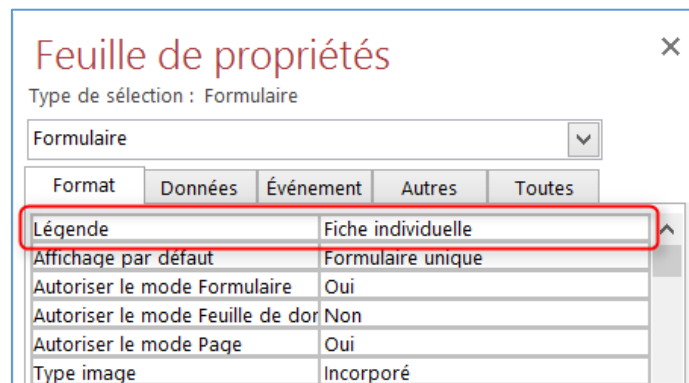
Comme dans la vraie vie, un **événement VBA** est "quelque chose qui se produit à un instant précis". Dans Access, l'événement se déclenche généralement suite à une action de l'utilisateur : ouvrir ou fermer un formulaire, se déplacer dans un formulaire, cliquer sur un bouton, etc.

Associer du VBA à un événement permet d'intervenir à un moment clef du fonctionnement d'Access et de lui ajouter un comportement sur mesure. La difficulté étant, lorsqu'on est débutant, de trouver l'événement adapté au besoin !

Quelques explications complémentaires

Pour aller plus loin dans la technique, voici quelques compléments :

- **Me** ("Moi", en anglais) désigne le formulaire en cours (celui dans lequel vous avez tapé votre code VBA). D'autres syntaxes sont possibles, l'idée est d'aller à l'essentiel dans cet article.
- **Caption** est la légende d'un formulaire (ce même réglage que vous trouvez dans les propriétés du formulaire). L'intérêt ici est que nous modifions cette légende dynamiquement, en VBA.



- **Me.Nom** et **Me.Prénom** sont les champs du formulaire (et, par extension, de la table associée).
- Enfin, l'opérateur "&" (qui existe également dans Excel) permet de "concaténer" (coller) plusieurs éléments de texte pour en faire une phrase complète. Si vous écrivez **x = "lundi" & "mardi"**, **x** est alors égal à **"lundimardi"**. Pour ajouter une espace, on écrirait : **x = "lundi" & " " & "mardi"**. Comme dans Excel, les chaînes de texte sont délimitées par des guillemets.
- J'ai supposé que le titre d'en-tête (sur le fond gris en haut du formulaire) avait été créé automatiquement par Access, et qu'il s'appelait par conséquent **Auto_Entete0**. Si tel n'est pas le cas, notre petit programme VBA échouera à l'exécution. Si votre titre s'appelle autrement, corrigez-le dans le volet **Propriétés** (onglet **Autres**, propriété **Nom**).

J'espère que ce tutoriel vous a donné le goût d'approfondir le langage VBA. Ne vous formalisez pas si vous débutez et que certains ordres vous échappent : une langue ne s'apprend pas en un jour...mais vous avez franchi la première étape !

Tutoriel réalisé avec Access 2013

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

Article écrit par Hervé Inisan du site

Le Grenier Access : <https://grenier.self-access.com>