



Créer des macros sous Visual Basic pour Excel

Par Henry P. Aubert

Sommaire

1	Qu'est qu'une macro ?	2
2	Enregistrer la séquence des opérations à répéter	2
3	Visualiser le code d'une macro	3
4	Ecrire ou modifier le code d'une macro	4
5	Consulter l'aide de Microsoft VBA	5
6	Les deux types de macro d'Excel	5
7	Importer un module dans un classeur	6
8	Créer le fichier Perso.xls	7
9	Supprimer un module	7
10	Exporter un module	8
11	Renommer un module	9
12	Associer un bouton ou un élément de menu à une procédure	10
13	Créer un nouveau menu personnalisé	13
14	Créer une nouvelle barre d'outils personnalisée	14
15	Associer un raccourci clavier à une procédure	15
16	Supprimer des boutons personnalisés	16
17	Supprimer une barre d'outils	16
18	Editer le résultat de l'exportation d'un module	17

1 Qu'est qu'une macro ?

Les macros⁽¹⁾, dans Excel, sont des opérations sur les classeurs Excel qui ont été enregistrées dans un programme parce qu'elles sont longues et fastidieuses à exécuter à la souris ou au clavier. Les macros permettent aussi de réaliser des opérations sur les classeurs impossibles à la souris ou au clavier. Ces programmes sont eux-mêmes enregistrés comme des sous-programmes d'Excel, au choix :

- Dans le classeur dans lequel on veut pouvoir les utiliser.
- Dans le classeur **Perso.xls** de macros, dites personnelles, d'Excel.
- Dans un classeur de macros complémentaires, dont l'extension est **.xla**.

Il suffit ensuite de "lancer" la macro pour que les opérations qu'elle contient soient exécutées à très grande vitesse, comme indiqué au paragraphe 6, page 5 ci-dessous.

Les *opérations* sont enregistrées dans un langage de programmation appelé **Visual Basic for Applications**, en abréviation **VBA**. On parle alors d'*instructions*.

Pour créer une macro on a trois possibilités :

- **Enregistrer** la séquence des opérations au clavier, comme expliqué au paragraphe 2, page 2, ci-dessous.

Dans ce cas certaines opérations intéressantes sont impossibles à réaliser.

- **Importer**, comme expliqué au paragraphe 7, page 6, ci-dessous, dans le classeur où l'on veut la stocker, une macro proposée sur le site de la page Excel'Ense :

<http://www-rocq.inria.fr/axis/modulad/excel.htm>

ou sur tout autre site Internet *de confiance*, car une macro peut aller modifier le registre, donc le fonctionnement du système, ou de tout autre programme, effacer ou créer des fichiers, etc.

- **Modifier**, comme expliqué au paragraphe 4, page 4, ci-dessous, le code d'une macro déjà existante pour l'améliorer.
- **Ecrire** directement le code d'une macro, comme expliqué au paragraphe 4, page 4, ci-dessous.

2 Enregistrer la séquence des opérations à répéter

- Cliquer sur le **menu Outils**, puis sur l'option **Macro**, puis sur l'option **Nouvelle Macro...** La boîte de dialogue **Enregistrer une macro** s'ouvre :

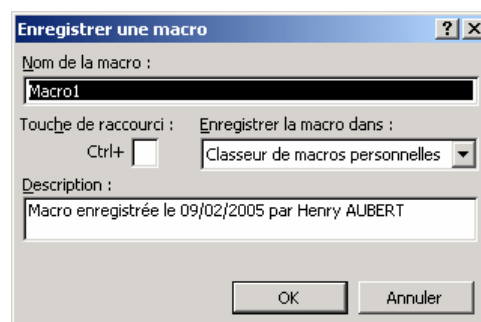
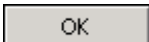


Figure 1 : Boîte de dialogue Enregistrer une macro

- Dans la zone de texte **Nom de la macro** :, taper un nom de macro ou laisser **Macro1**.
- Dans la liste déroulante **Enregistrer la macro dans** :, sélectionner l'option **Classeur de macros personnelles**.
- Cliquer sur le bouton . La barre d'outils **Arrêter l'enregistrement** s'affiche :



- Effectuer sur le classeur Excel la séquence de toutes les opérations que l'on souhaite enregistrer.

(1) D'une façon générale les "macro-commandes" ou **macros**, sont des séquences de commandes informatiques enregistrées dans des fichiers reconnus comme tels par le système ou une application, et qu'il suffit donc de "lancer" pour exécuter ces opérations. Ainsi des anciens fichiers dont l'extension était **.bat** pour les commandes **dos**.

- Cliquer sur le bouton  **Arrêter l'enregistrement** de la barre d'outils **Arrêter l'enregistrement** ci-dessus.

On peut ensuite visualiser le code, c'est-à-dire les instructions du programme qui en résulte, et commencer à comprendre le langage **Microsoft Visual Basic for Applications (VBA)** pour apprendre à modifier, puis écrire directement des macros.

3 Visualiser le code d'une macro

- Appuyer simultanément sur les touches **Alt** et **F11**. L'éditeur de macros pour Excel, **Visual Basic Editor**, s'ouvre.
- Appuyer simultanément sur les touches **Ctrl** et **R**⁽²⁾. On obtient :

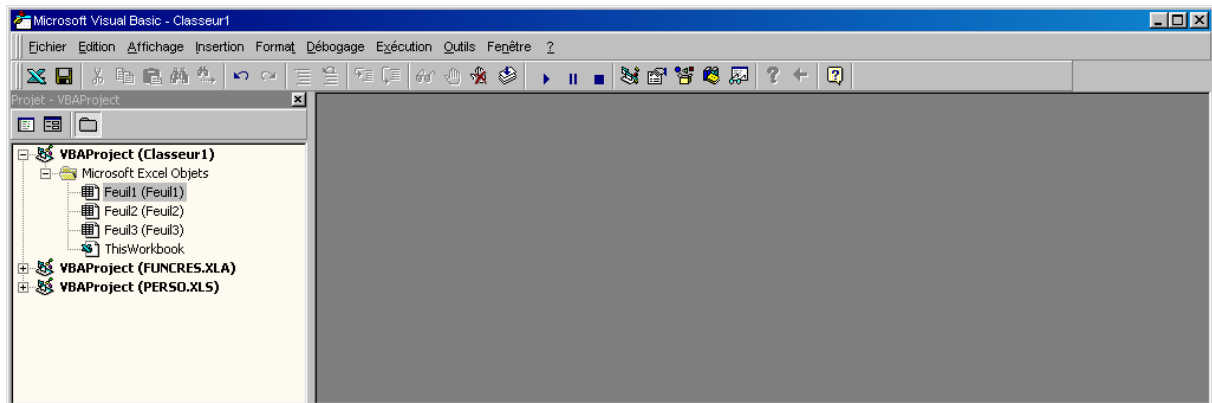


Figure 2 : Fenêtre de l'éditeur de macros Excel divisé en deux panneaux

- Dans l'**Explorateur de projet** (le panneau de droite) développer le dossier **VBA Project (PERSO.XLS)**. On obtient :

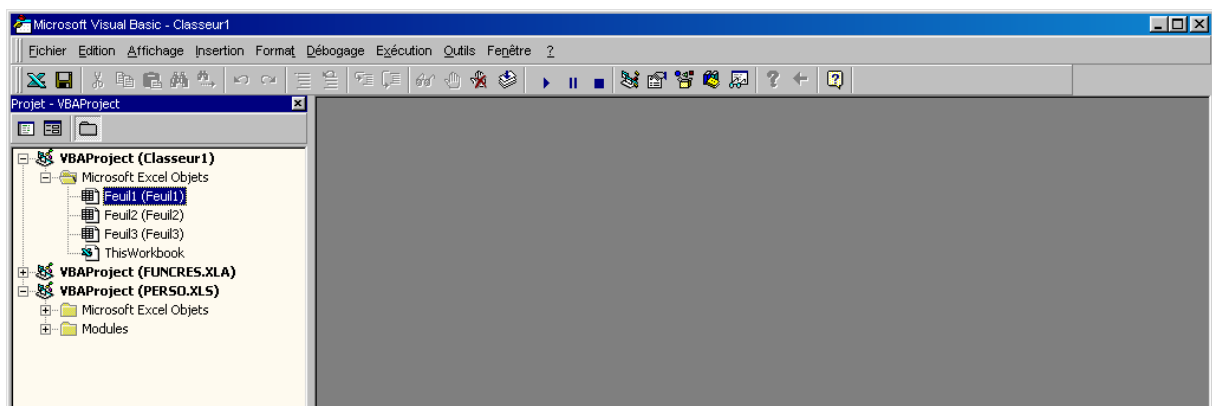


Figure 3 : Développement du dossier **VBA Project (PERSO.XLS)**

⁽²⁾ Ce raccourci clavier ne sert qu'à afficher l'**explorateur de projet** (le panneau de droite de la Figure 5 ci-dessus). Une fois affiché, il le sera à la prochaine ouverture de l'éditeur.

- Encore dans le panneau de droite, développer le dossier **Modules** du projet **VBA Project (PERSO.XLS)** :

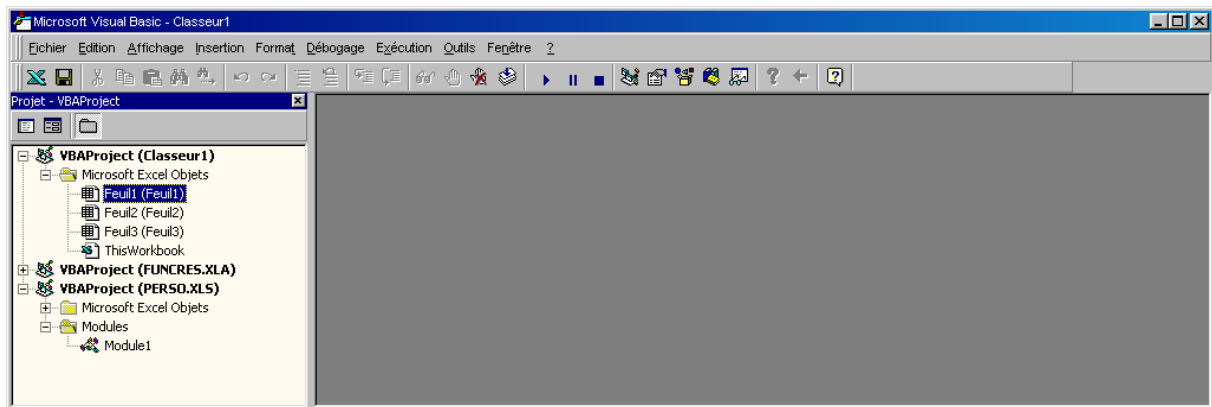


Figure 4 : Développement du dossier **Modules** du dossier **VBA Project (PERSO.XLS)**

Figure 5 : Fenêtre de l'éditeur de macros Excel

- Double cliquer sur le module **Module1** qui contient la macro **Macro1** enregistrée :

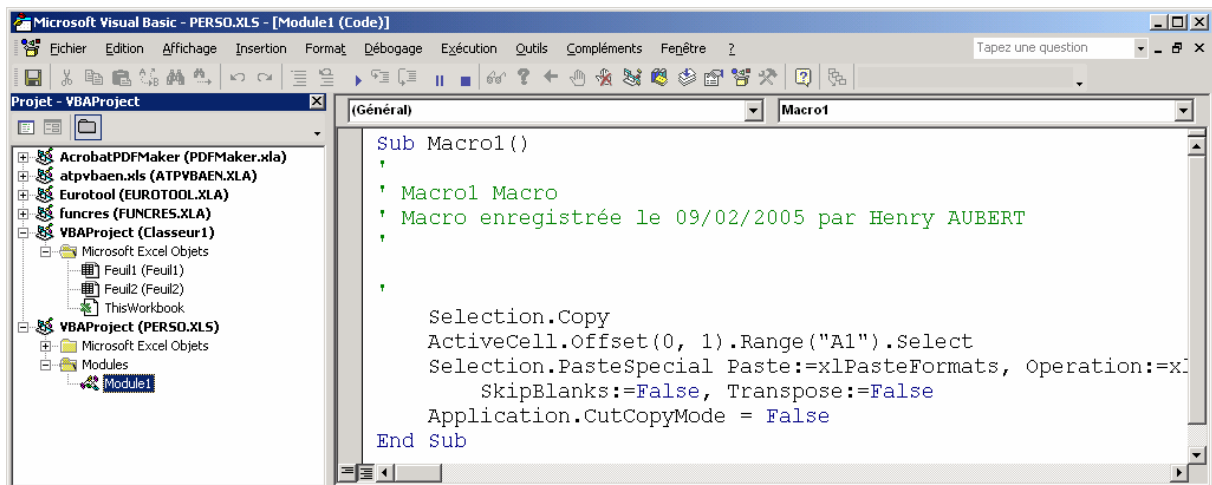


Figure 6 : Visualisation du code d'une macro enregistrée

4 Ecrire ou modifier le code d'une macro

- Ouvrir l'éditeur de macros pour Excel, **Visual Basic Editor** comme au paragraphe 3 ci-dessus.
- Développer le dossier **VBA Project (PERSO.XLS)** comme au paragraphe 3, page 3.
- Développer le dossier **Modules** du projet **VBA Project (PERSO.XLS)** comme au paragraphe 3, page 3.
- Cliquer du bouton droit sur le module contenant la macro à modifier ou devant contenir la macro à écrire, par exemple **Module1**.
- Ecrire ou modifier le code en s'inspirant d'autres macros de l'aide en ligne, et du manuel VBA pour Excel et Word bientôt disponible sur <http://hpa.free.fr>.

5 Consulter l'aide de Microsoft VBA

- Sélectionner, dans une macro, un mot d'une instruction⁽³⁾, par exemple **ActiveCell**.
- Appuyer sur la touche **F1**. La fenêtre de Microsoft **Visual Basic : Aide** s'ouvre :

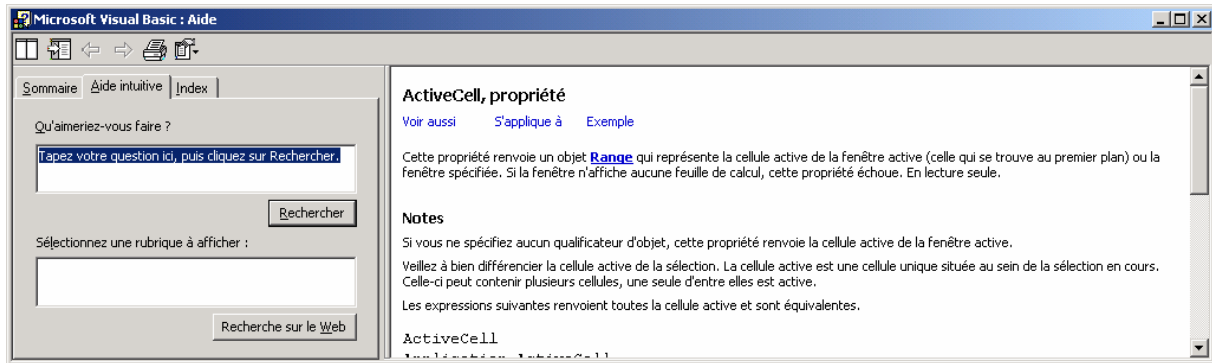


Figure 7 : Fenêtre de Microsoft Visual Basic : Aide

6 Les deux types de macro d'Excel

Il y a, pour Excel, deux types de macros :

- Les *procédures*, dont le code débute par `Sub ()` et se termine par `End Sub`. Elles peuvent être enregistrées, modifiées, ou directement écrites. Elles ne sont exécutables qu'au travers de la boîte de dialogue **Macro**. Pour cela :
 - Appuyer simultanément sur les touches **Alt** et **F8** (ou cliquer sur le menu **Outils** puis sur le sous-menu **Macros** puis sur l'option **Macros...**). La boîte de dialogue **Macro** s'ouvre :

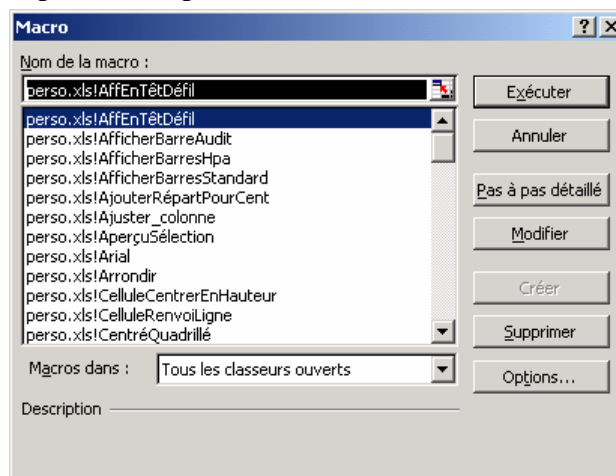



Figure 8 : Sélection d'une macro pour l'Exécuter, la Modifier ou la Supprimer

- Sélectionner la macro à exécuter.
- Cliquer sur le bouton **Exécuter**.
- Les *fonctions* dont le code débute par `Function ()` et se termine par `End Function`. Elles ne peuvent pas être enregistrées : elles doivent être écrites directement. Elles ne peuvent qu'être insérées dans une cellule. Pour cela :
 - Cliquer sur le bouton  **Insérer une fonction** ou appuyer simultanément sur **Shift** (ou **Maj**) et **F3**. La boîte de dialogue **Insérer une fonction** s'ouvre.

⁽³⁾ Il suffit de placer le pointeur de la souris au milieu du mot sur lequel on veut obtenir de l'aide.

- Sélectionner la catégorie **Personnalisées** de la liste des **catégories** :. On obtient :

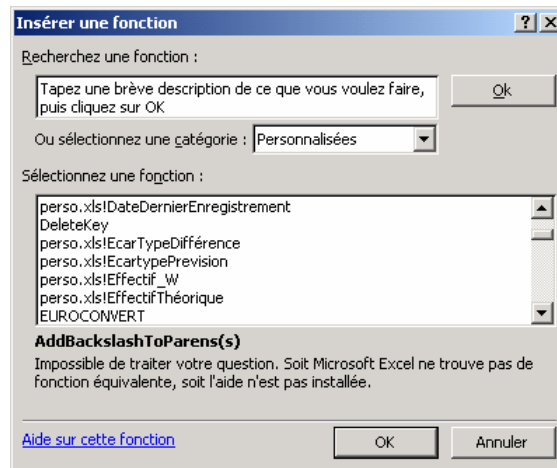


Figure 9 : Sélection d'une fonction personnalisée à insérer dans une cellule

- Sélectionner la fonction à insérer.

Les procédures et les fonctions peuvent être regroupées et déplacées de façon tout à fait quelconque dans un ou plusieurs **modules**.

7 Importer un module dans un classeur

Une procédure ou une fonction ne peut être lancée que dans le classeur dans lequel elle est enregistrée.

Une procédure ou une fonction enregistrée dans le classeur **PERSO.XLS** peut être lancée dans tout classeur ouvert.

On suppose qu'on veut importer un module dans le classeur **Perso.xls**. Dans ce cas, il faut que celui-ci existe. Pour importer le module dans un classeur autre que **Perso.xls**, il faut qu'il soit ouvert.

- Dans tous les cas, ouvrir un classeur Excel.
- Ouvrir l'éditeur de macros pour Excel, **Visual Basic Editor** comme au paragraphe 3, page 3.
 - ◆ Si on veut importer le module dans le classeur **Perso.xls** et que le projet **VBA Project (PERSO.XLS)** n'apparaît pas dans le panneau de droite comme dans la Figure 5, page 4, fermer l'éditeur de macros, puis créer **PERSO.XLS** comme au paragraphe 8, page 7 puis, après, rouvrir l'éditeur de macros.
 - ◆ Pour importer le module dans un des classeurs *ouverts*, la manœuvre suivante est identique, au nom du classeur près.
- Cliquer du bouton droit sur un module *quelconque*. Un menu contextuel s'ouvre :

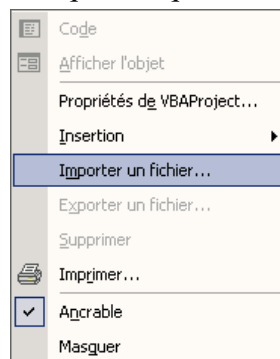


Figure 10 : Le menu contextuel de gestion des modules Visual Basic

- Cliquer sur **Importer un fichier...** La boîte de dialogue **Importer un fichier** s'ouvre :

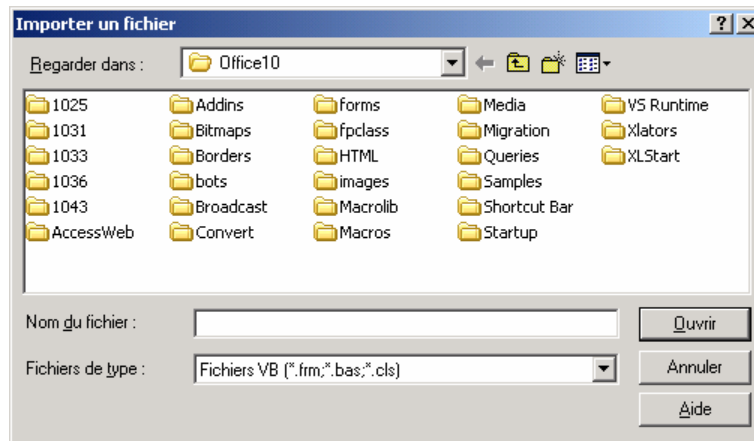


Figure 11 : Boîte de dialogue Importer un fichier

- Rechercher le dossier, puis le puis le fichier **.bas** contenant la ou les macros à importer.
- Cliquer sur le bouton **Ouvrir**. Le module est placé dans le dossier **Modules**.
Le nom du module est automatiquement modifié si un module de même nom existe déjà.

8 Créer le fichier Perso.xls

- Enregistrer une macro **Macro1** quelconque, comme au paragraphe 2, page 2, mais sans aucune opération.
- Supprimer le module **Module1** dans lequel la macro **Macro1** a été enregistrée, comme indiqué au paragraphe 9, page 7 ci-dessous.

9 Supprimer un module

On suppose qu'il s'agit d'un module du classeur **Perso.xls**. Pour les autres modules, la manœuvre est identique, au nom du classeur près.

- Ouvrir l'éditeur de macros pour Excel, **Visual Basic Editor** comme au paragraphe 3, page 3.
- Développer le dossier **VBA Project (PERSO.XLS)** comme au paragraphe 3, page 3.
- Développer le dossier **Modules** du **projet VBA Project (PERSO.XLS)** comme au paragraphe 3, page 3.
- Cliquer du bouton droit sur le module à supprimer, par exemple **Module1**. Un menu contextuel s'ouvre :

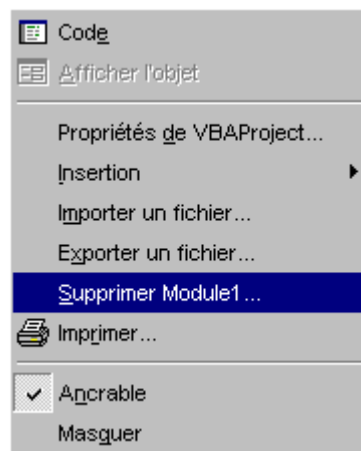


Figure 12 : Le menu contextuel d'un module Visual Basic

- Sélectionner puis cliquer sur l'option **Supprimer Module1**. Une boîte de dialogue s'ouvre :

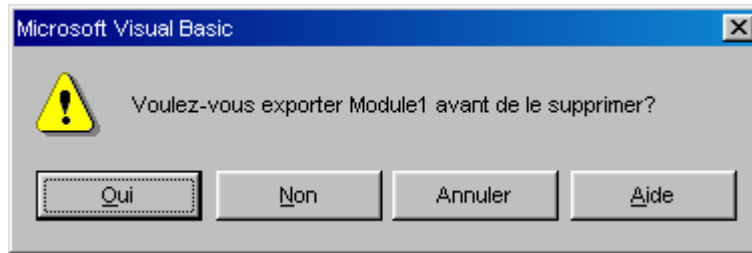
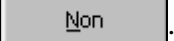


Figure 13 : Boîte de dialogue pour exporter un module avant de le supprimer

- Cliquer sur le bouton .

10 Exporter un module

Il est prudent de sauvegarder les modules créés en les exportant.

L'exportation se fait dans un fichier texte dont l'extension est **.bas**.

On suppose qu'il s'agit d'un module du classeur **Perso.xls**. Pour les autres modules, la manœuvre est identique, au nom du classeur près.

- Ouvrir l'éditeur de macros pour Excel, **Visual Basic Editor** comme au paragraphe 3, page 3.
- Développer le dossier **VBA Project (PERSO.XLS)** comme au paragraphe 3, page 3.
- Développer le dossier **Modules** du **projet VBA Project (PERSO.XLS)** comme au paragraphe 3, page 3.
- Cliquer du bouton droit sur le module à supprimer, par exemple **Module1**. Un menu contextuel s'ouvre :

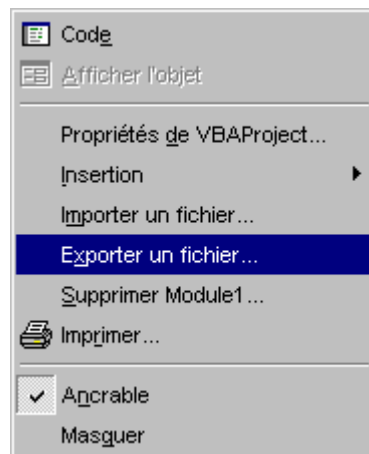


Figure 14 : Le menu contextuel d'un module Visual Basic

- Sélectionner puis cliquer sur l'option **Exporter un fichier**. La boîte de dialogue **Exporter un fichier** s'ouvre :

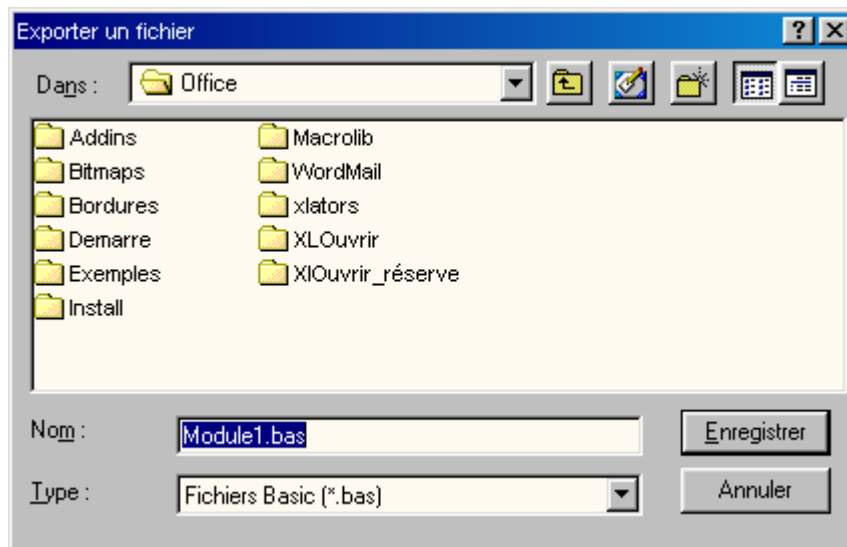


Figure 15 : Boîte de dialogue **Exporter un fichier**

- Sélectionner le dossier dans lequel on souhaite enregistrer le fichier **.bas** contenant la ou les macros.
- Cliquer sur le bouton **Enregistrer**.
Remarque : Changer le nom du fichier **.bas**, si l'on ne veut pas remplacer une précédente sauvegarde. Le nom du fichier **.bas** est indépendant du nom du module.

11 Renommer un module

- Ouvrir l'éditeur de macros pour Excel, **Visual Basic Editor** comme au paragraphe 3, page 3.
- Développer le dossier **VBA Project (PERSO.XLS)** comme au paragraphe 3, page 3.
- Développer le dossier **Modules** du **projet VBA Project (PERSO.XLS)** comme au paragraphe 3, page 3.
- Cliquer du bouton droit sur le module à renommer, par exemple **Module1**.
- Appuyer sur la touche F4. La fenêtre **Propriétés** de l'objet s'ouvre, et affiche les propriétés de l'objet sélectionné, ici, le module **Module1** :

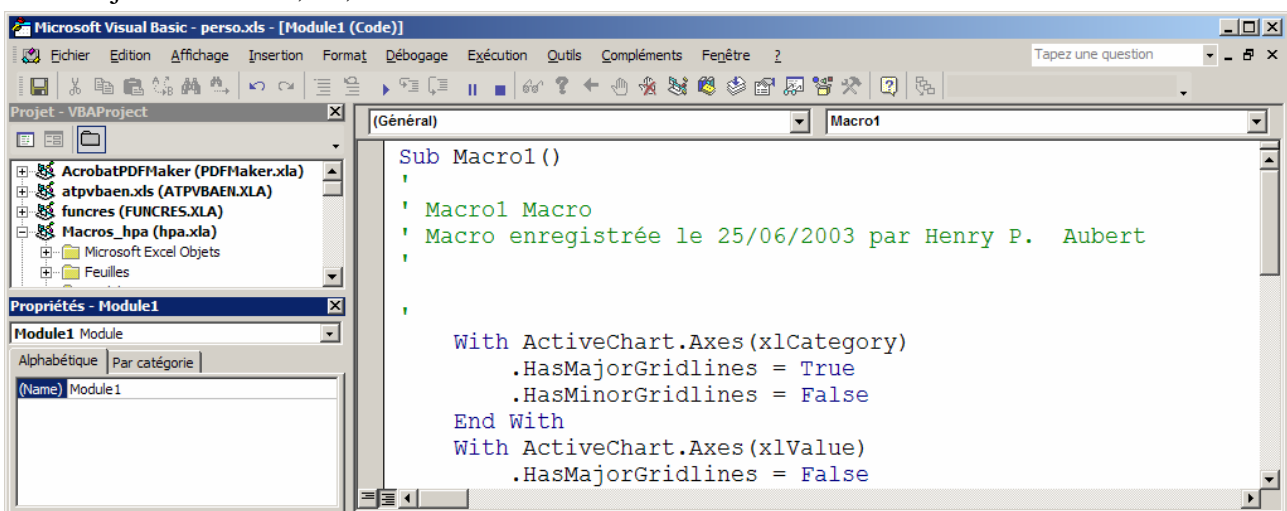


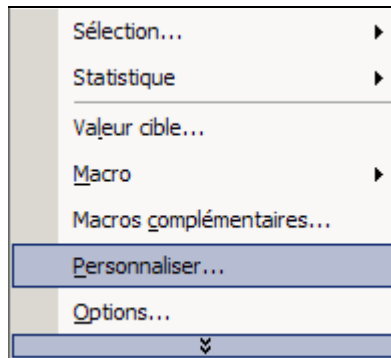
Figure 16 : L'éditeur de macros pour Excel avec la fenêtre **Propriétés**

- Double cliquer sur la valeur **Module1** de **Name**, la seule propriété du module, son nom.
- Taper le nouveau nom. La modification est immédiate.
- Pour fermer la fenêtre **Propriété**, cliquer sur son bouton

12 Associer un bouton ou un élément de menu à une procédure

Il est impossible d'associer une fonction à un bouton.

- Cliquer sur le menu **Outils**. Sélectionner l'option **Personnaliser...** On obtient :



- Cliquer sur cette option. La boîte de dialogue **Personnaliser** s'ouvre :

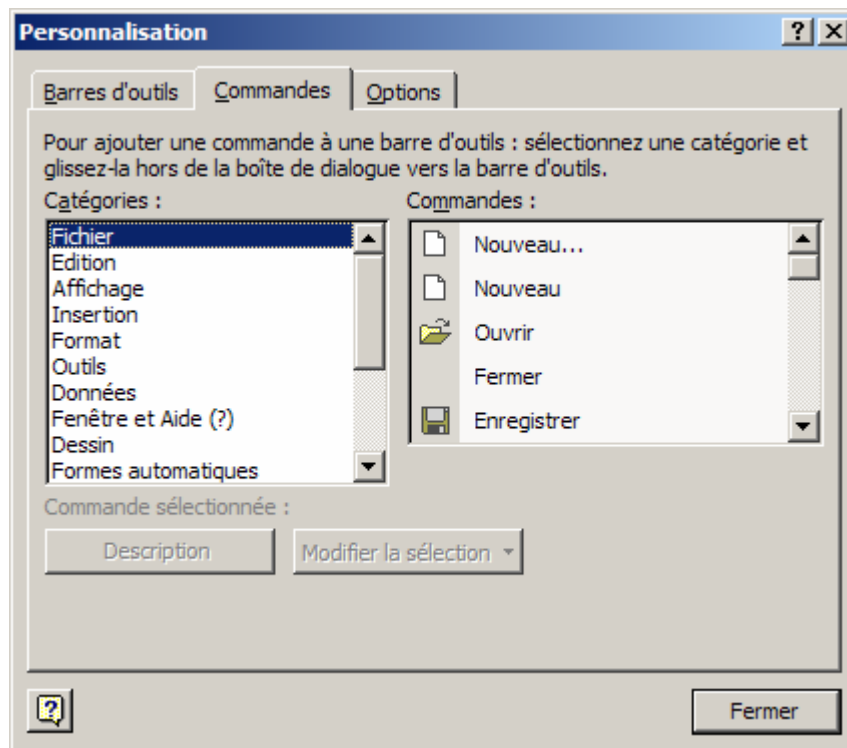


Figure 17 : Boîte de dialogue **Personnaliser**

On peut aussi cliquer du bouton droit n'importe où dans la zone des menus et des barre d'outils, pour trouver l'option **Personnaliser**, et ouvrir la boîte de dialogue ci-dessus.

- Dans la liste **Catégories** :, sélectionner **Macros** :

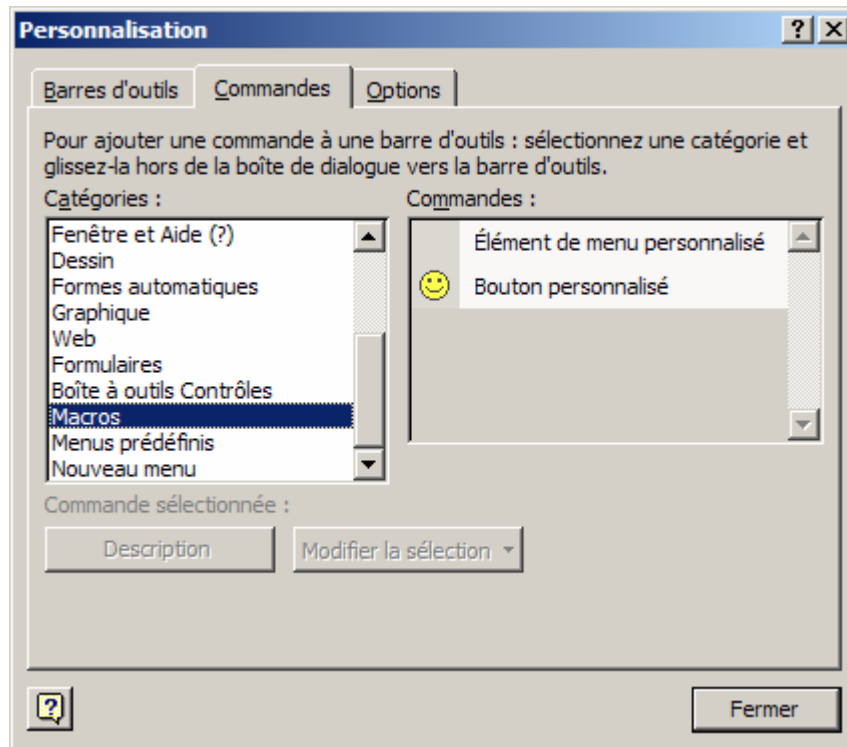


Figure 18 : Les commandes disponibles pour la personnalisation avec des macros

- Dans la liste **Commandes**, sélectionner indifféremment **Bouton Personnalisé** ou **Element de menu personnalisé** pour ajouter une procédure à une barre d'outils existante ou à un menu existant. Le résultat est, à la présentation de départ près, le même.
- Traîner la commande sélectionnée à l'endroit de la barre d'outils existante ou du menu existant où on veut la placer. Exemple :

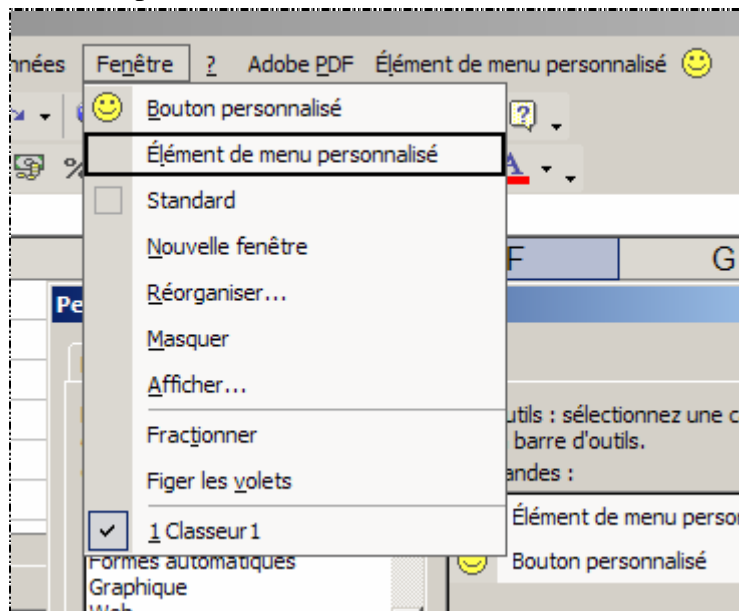


Figure 19 :

Dans cet exemple, un Bouton personnalisé et un Élément de menu personnalisé **ont été** ajoutés à la **Barre des menus**, et dans le menu **Fenêtre**.

- Cliquer du bouton droit de la souris sur l'élément ou le bouton personnalisé ajouté. Un menu contextuel s'affiche :

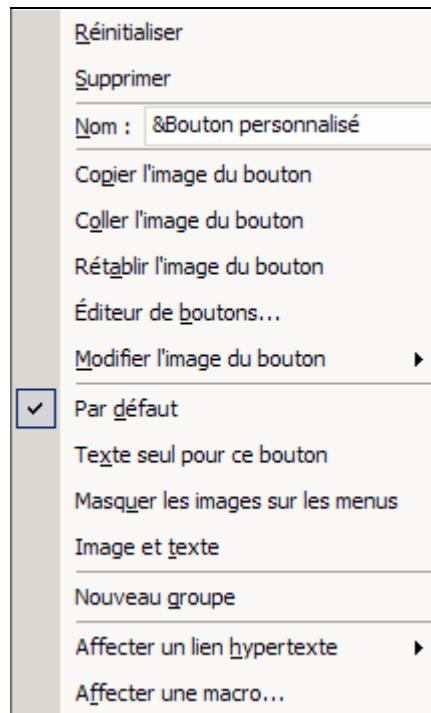


Figure 20 : Menu contextuel de personnalisation d'un bouton ou un élément de menu personnalisé

La seule différence est que, pour un **Bouton personnalisé**, l'option **Par défaut** est sélectionnée, comme ci-dessus, alors que pour un **Élément de menu personnalisé** c'est l'option **Texte seul pour ce bouton** qui l'est. On passe de l'un à l'autre en cliquant sur une de ces deux options.

- Cliquer sur l'option **Affecter une macro...** La boîte de dialogue **Affecter une macro...** s'ouvre :

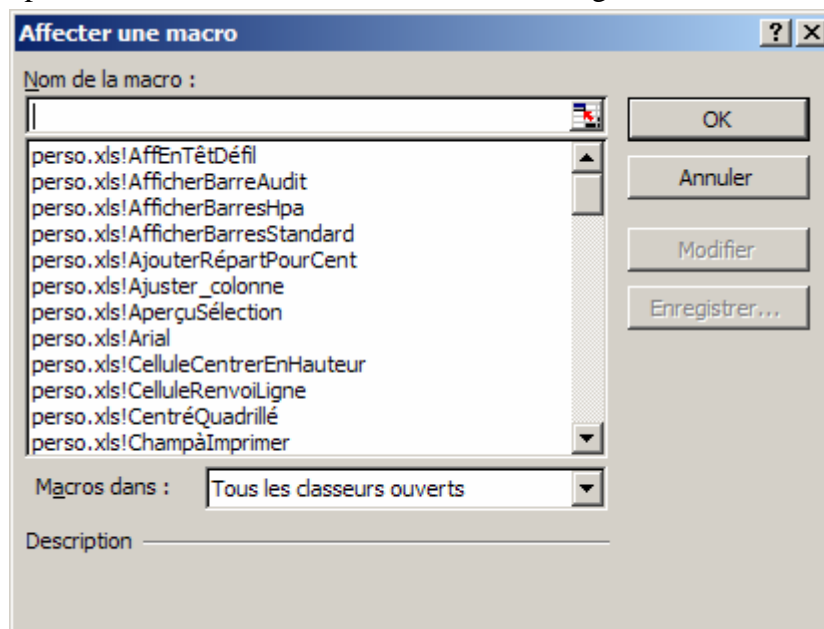

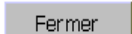


Figure 21 : Boîte de dialogue **Affecter une macro...**

- Sélectionner la "macro" c'est-à-dire la procédure à affecter au **Bouton** ou **Élément de menu personnalisé**.
- cliquer sur le bouton .
- Personnaliser (Image du bouton, Nom, etc.) le au moyen du menu contextuel précédent.
- Dans la boîte de dialogue **Personnaliser**, cliquer sur .

13 Créer un nouveau menu personnalisé

Les procédures personnalisées peuvent aussi être associées à des éléments d'un menu personnalisé. Pour cela :

- Ouvrir la boîte de dialogue **Personnaliser**, comme au paragraphe 12, page 10, ci-dessus.
- Dans la liste **Catégories** :, sélectionner **Nouveau Menu** :

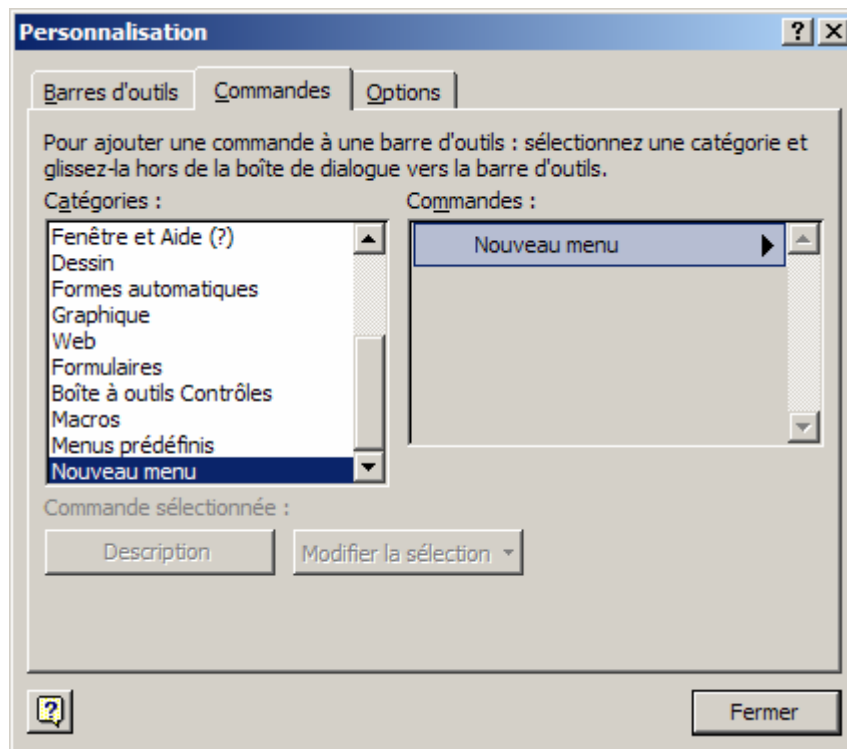


Figure 22 : La boîte de dialogue **Personnalisation**

- Dans la liste **Commandes** : sélectionner puis traîner l'option **Nouveau menu** à l'emplacement souhaité :

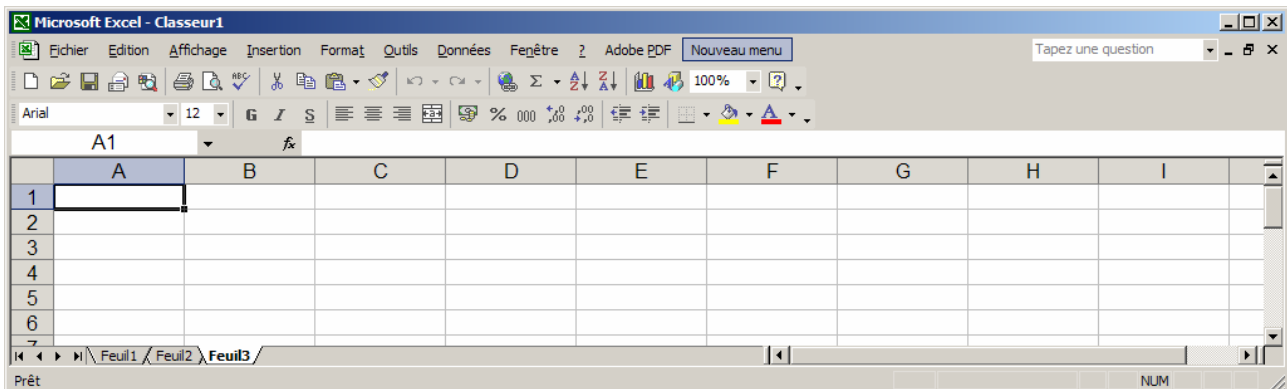


Figure 23 : Ajout d'un nouveau menu dans la barre des menus

- Personnaliser ce nouveau menu en cliquant dessus du bouton droit de la souris.
- Ajouter des boutons personnalisés associés à des procédures ou des commandes Excel à ce menu, comme au paragraphe 12, page 10, ci-dessus.

14 Créer une nouvelle barre d'outils personnalisée

Les procédures personnalisées peuvent aussi être associées à des boutons d'une barre d'outils personnalisée. Pour cela :

- Ouvrir la boîte de dialogue **Personnaliser**, comme au § 13, ci-dessus.
- Sélectionner l'onglet **Barres d'outils**. On obtient :

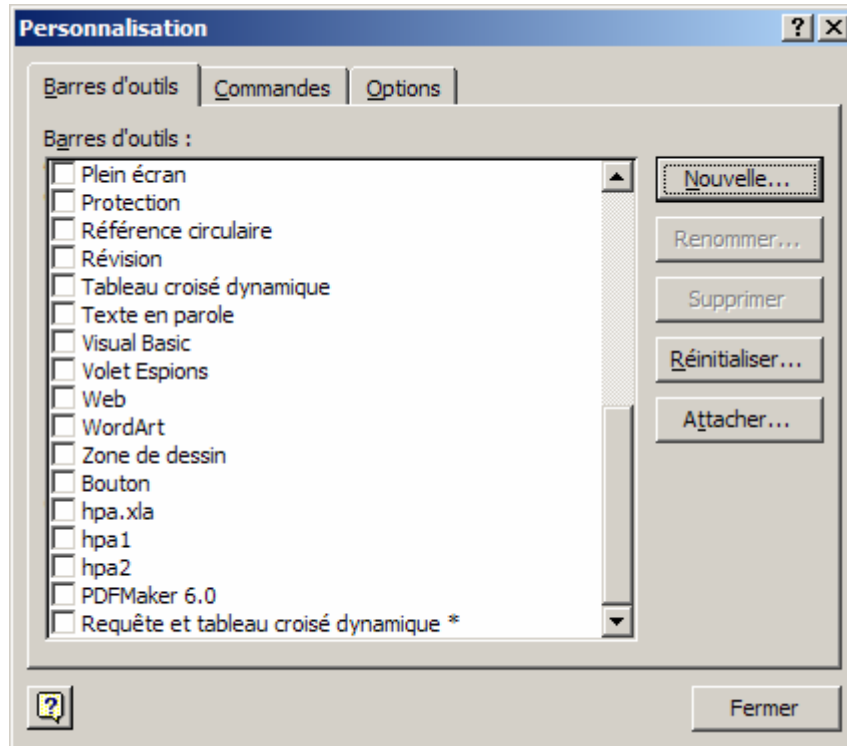


Figure 24 : Feuille **Barre d'outils** de la boîte de dialogue **Personnalisation**

- Cliquer sur le bouton **Nouvelle...**. On obtient :

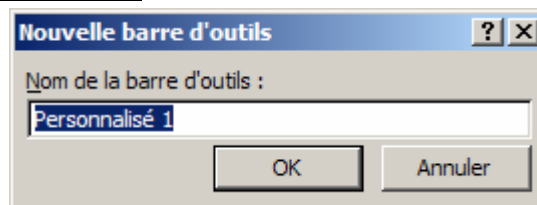
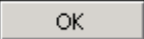


Figure 25 : Boîte de dialogue **Nouvelle barre d'outils**

- Donner un nom à cette barre.

- Cliquer sur le bouton . Une nouvelle barre d'outils sans boutons s'affiche :

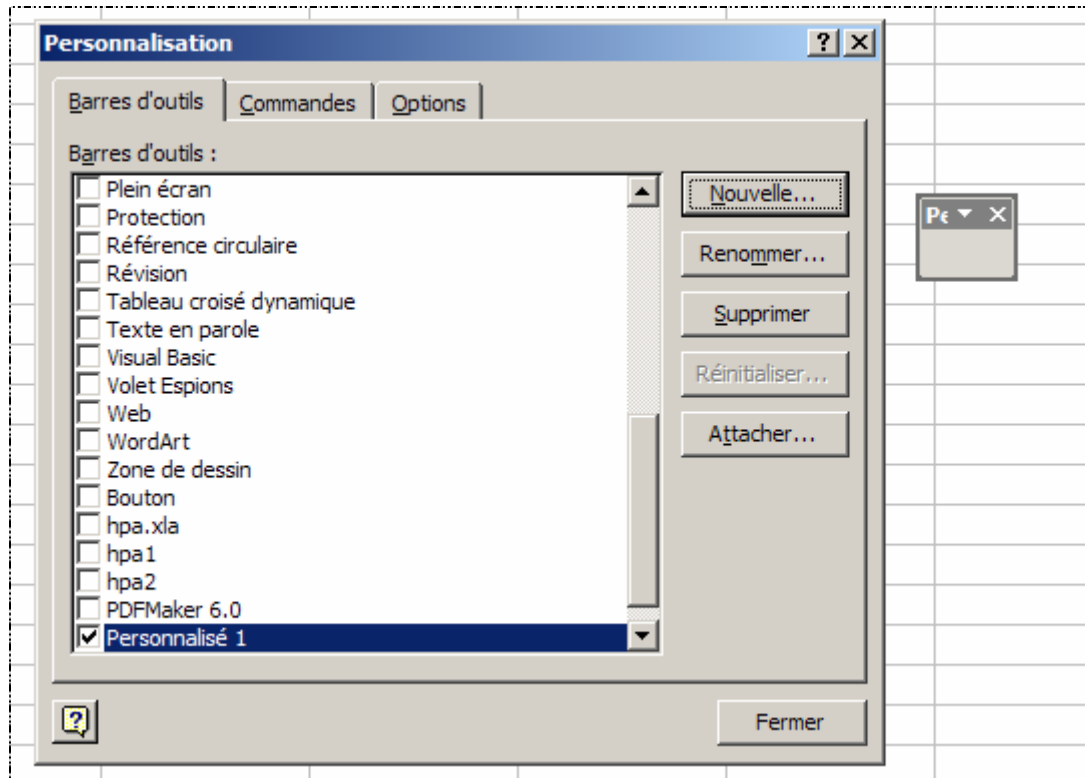
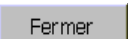


Figure 26 : Une nouvelle barre d'outils sans boutons s'affiche

Et cette nouvelle barre est listée et cochée dans la liste **Barres d'outils** :

- Cliquer sur le bouton  de la boîte de dialogue **Personnalisation**.
- Ajouter des boutons personnalisés ou des commandes Excel à cette barre, comme au paragraphe 12, page 10, ci-dessus.

15 Associer un raccourci clavier à une procédure

- Appuyer simultanément sur les deux touches Alt et F8. La boîte de dialogue **Macro** s'ouvre :

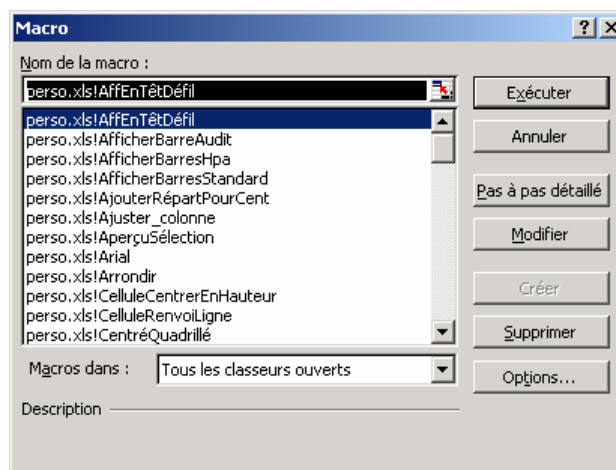
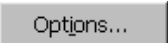


Figure 27 : Boîte de dialogue **Macro**

- Sélectionner la procédure à laquelle associer un raccourci clavier.

- Cliquer sur le bouton . La boîte de dialogue **Options de macro** s'ouvre :

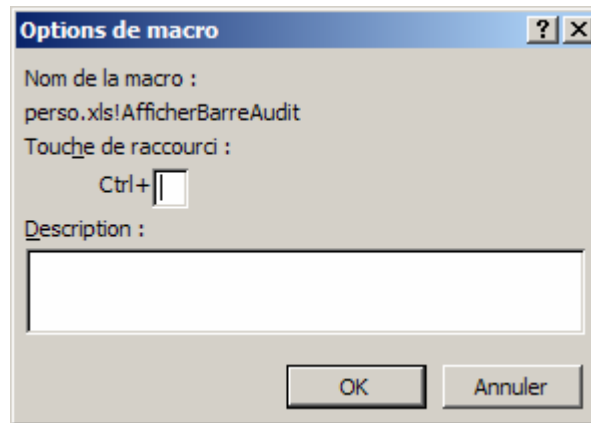
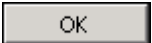
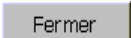


Figure 28 : Boîte de dialogue **Options de macro**

- Taper une lettre ou un chiffre et, éventuellement une description.
- cliquer sur le bouton .

16 Supprimer des boutons personnalisés

- Ouvrir la boîte de dialogue **Personnaliser**, comme au paragraphe 12, page 10, ci-dessus.
- Traîner l'élément à supprimer en dehors de la zone des menus et des barres d'outils.
- Cliquer sur le bouton  de la boîte de dialogue **Personnalisation**.

17 Supprimer une barre d'outils

- Ouvrir la boîte de dialogue **Personnaliser**, comme au paragraphe 12, page 10, ci-dessus.
- Sélectionner l'onglet **Barres d'outils**.
- Dans la liste des barres d'outils, sélectionner la barre d'outils personnalisée à supprimer.
- Cliquer sur le bouton. La boîte de dialogue ci-dessous s'affiche.

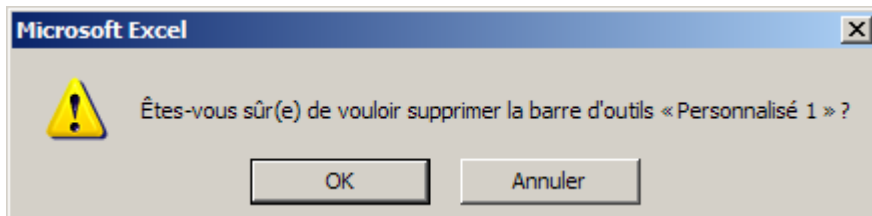
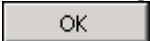
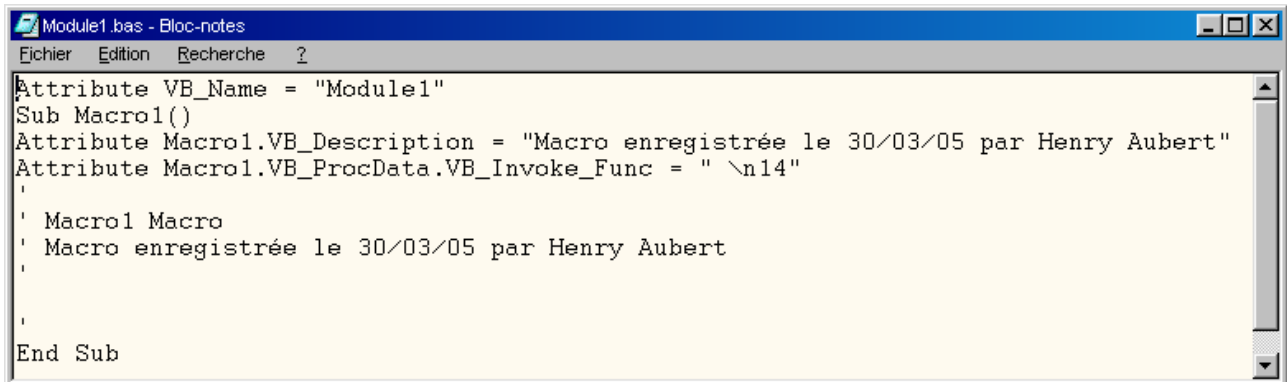


Figure 29 : Boîte de dialogue de suppression d'une barre d'outils personnalisée

- cliquer sur le bouton .

18 Editer le résultat de l'exportation d'un module

- Ouvrir le fichier **.bas** qui résulte de l'exportation du module du même nom effectuée au paragraphe 10, page 8 ci-dessus, avec l'éditeur de texte **NotePad.exe**. On obtient, pour **Module1** par exemple :



```

Module1.bas - Bloc-notes
Fichier  Edition  Recherche  ?
Attribute VB_Name = "Module1"
Sub Macro1()
Attribute Macro1.VB_Description = "Macro enregistrée le 30/03/05 par Henry Aubert"
Attribute Macro1.VB_ProcData.VB_Invoke_Func = " \n14"
'
' Macro1 Macro
' Macro enregistrée le 30/03/05 par Henry Aubert
'
End Sub

```

*Figure 30 : Edition du fichier texte résultant de l'exportation du module **Module1***

- Modifier le texte en argument dans l'instruction :
`Attribute VB_Name = "Module1"`
pour modifier le nom du module en cas de réimportation.
- Modifier le texte en argument dans l'instruction :
`Attribute Macro1.VB_Description =`
`"Macro enregistrée le 31/03/2005 par Henry AUBERT"`
pour modifier la description de la procédure en cas de réimportation. Ce texte correspond à celui qu'on saisit dans la zone **Description** de la boîte de dialogue de création ou de modification des options de la procédure, comme à la Figure 28, page 16, ci-dessus.
- Modifier l'espace du texte en argument dans l'instruction :
`Attribute Macro1.VB_ProcData.VB_Invoke_Func = " \n14"`
pour attribuer un raccourci clavier à la procédure. Le caractère saisi correspond à celui qu'on saisit dans la zone **Touche de raccourci** de la boîte de dialogue de création ou de modification des options de la procédure, comme à la Figure 28, page 16, ci-dessus.