

Queste de savoir

Faciliter ses développements via le VBE

dimanche 14 juillet 2024

Table des matières

	Introduction	1
1.	L'éditeur de code et ses menus	2
1.1.	Zone d'objet et zone de procédures/événements	2
1.2.	Recherche et remplacement	2
1.3.	Mise en forme	3
1.4.	IntelliSense	4
1.5.	Signets	4
2.	La fenêtre des propriétés	5
2.1.	Écriture et lecture	6
2.2.	Tris	7
2.3.	Accès hors VBE	7
3.	L'explorateur d'objets	8
3.1.	Accès	9
3.2.	Utilisation	9
4.	La fenêtre d'exécution	15
	Conclusion	15

Introduction

Pour programmer en **VBA** sur Microsoft Office, on recourt à l'environnement de développement intégré (**EDI**) par défaut : Visual Basic Editor.

Bien qu'un brin simpliste et vieillot par rapport à d'autres **EDI** modernes, il contient différents menus et fenêtres pour nous faciliter la tâche pendant nos développements. Nous avons notamment eu l'occasion de parler du débogage dans un [précédent billet](#) ↗ .

Au cours de ce billet, nous allons faire un petit tour de cet outil pour gagner en fluidité dans notre programmation.

C'est parti !

i

Pour ce billet, nous nous placerons dans l'environnement Microsoft Office.

i

Pour illustrer ce billet, j'utiliserai du code écrit pour d'autres billets. Ne vous étonnez pas si vous n'avez pas le même affichage que moi !

1. L'éditeur de code et ses menus

L'édition du code est le nerf de la guerre.

La plupart des fonctionnalités étudiées durant cette section sont accessibles depuis le menu "Édition" ou la barre d'outils du même nom ou encore depuis le menu contextuel (clique droit).

?

Comment on rajoute une barre d'outils ? 🍌

Bonne question ! Il est possible d'ajouter et d'enlever des barres d'outils via le sous-menu "Barres d'outils" du menu "Affichage".



FIGURE 1.1. – Barre d'outils "Édition"

1.1. Zone d'objet et zone de procédures/événements

En haut de l'éditeur, nous pouvons trouver deux listes déroulantes.

La zone d'objet, à gauche, liste les différents objets pour le fichier sélectionné. Elle est particulièrement utile pour l'édition de formulaires.

La zone de procédures/événements, à droite, liste les différentes procédures/événements pour l'objet sélectionné.

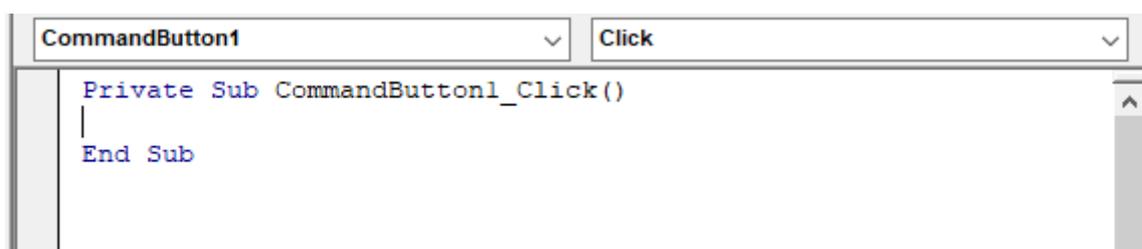


FIGURE 1.2. – Exemple pour un objet bouton dans un UserForm avec son événement de clique

Au travers cette sélection, ces listes nous offrent un moyen d'accéder directement à un élément désiré.

1.2. Recherche et remplacement

La recherche (raccourci **CTRL**+**F**) permet de rechercher pour une procédure, un module ou le projet en intégralité. Ce système est utile pour retrouver des choses, voir où elles sont.

1. L'éditeur de code et ses menus



FIGURE 1.3. – Rechercher (CTRL + F)

Nous pouvons faire du remplacement automatique afin de gagner du temps dans nos modifications de code en cliquant sur "Remplacer..." depuis cette fenêtre ou encore via le raccourci **CTRL**+**H**.

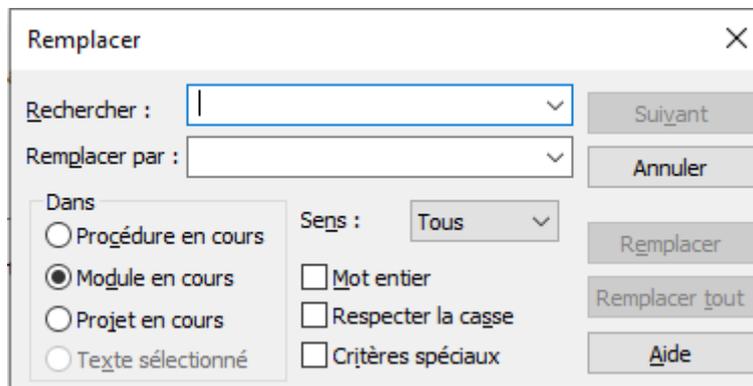


FIGURE 1.4. – Remplacer (CTRL + H)



Concernant les rechercher/remplacer en masse avec "Remplacer tout", attention à être sûr que les options sélectionnées (mot entier, casse, zone, caractères spéciaux) limitent bien à ce que vous voulez effectivement et uniquement remplacer.

1.3. Mise en forme

La mise en forme du code est importante pour faciliter sa lecture.

Pour indenter (raccourci **TAB**) et désindenter (raccourci **MAJ** + **TAB**), nous pouvons nous servir des boutons de la barre d'outils encadrés en orange ci-dessous.

Pour commenter et décommenter, nous pouvons nous servir des boutons encadrés en rouge ci-dessous. Cela est très utile pour commenter et décommenter des blocs d'un coup.



FIGURE 1.5. – Mise en forme via l'indentation et les commentaires

1.4. IntelliSense

Le VBE arrive avec des outils d'aide à la saisie de code.

Par défaut, certaines informations s'affichent automatiquement quand nous écrivons le code. Par exemple, les paramètres possibles pour une fonction (**CTRL** + **I**).

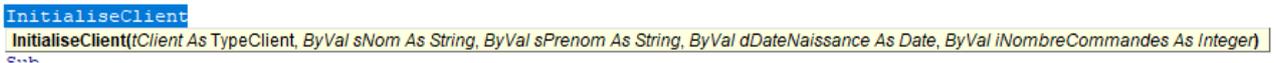


FIGURE 1.6. – Info express (**CTRL** + **I**)

Il y a parfois un petit menu déroulant qui nous liste des méthodes et autres possibilités. Celui-ci peut s'afficher via **CTRL** + **J**.

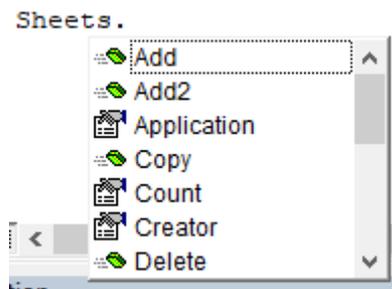


FIGURE 1.7. – Répertoire des propriétés/méthodes (**CTRL** + **J**)

La complétion de code (**CTRL** + **Espace**) liste les possibilités à partir du début du code et complète s'il n'y a qu'une seule possibilité.

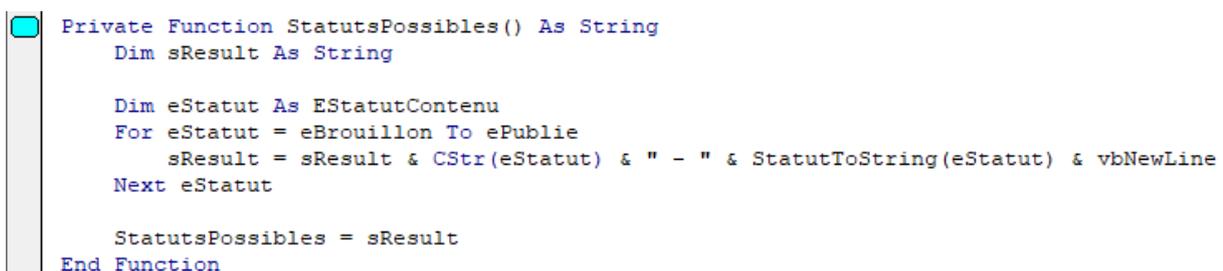
Ces trois fonctionnalités correspondent respectivement aux boutons encadrés en rouge ci-dessous :



FIGURE 1.8. – Aide au développement

1.5. Signets

Les signets servent à créer des marque-pages pour les lignes voulues. Visuellement, ils se traduisent par de petits rectangles colorés dans la marge grise.



2. La fenêtre des propriétés

FIGURE 1.9. – Exemple signet

Cela peut être utile si en écrivant du code, nous savons que nous devons revenir à tel ou tel morceau de code, en naviguant à travers les signets.

Plutôt que de supprimer chaque signet, il est possible de tous les supprimer d'un coup.

Ces options sont listées de cette manière :

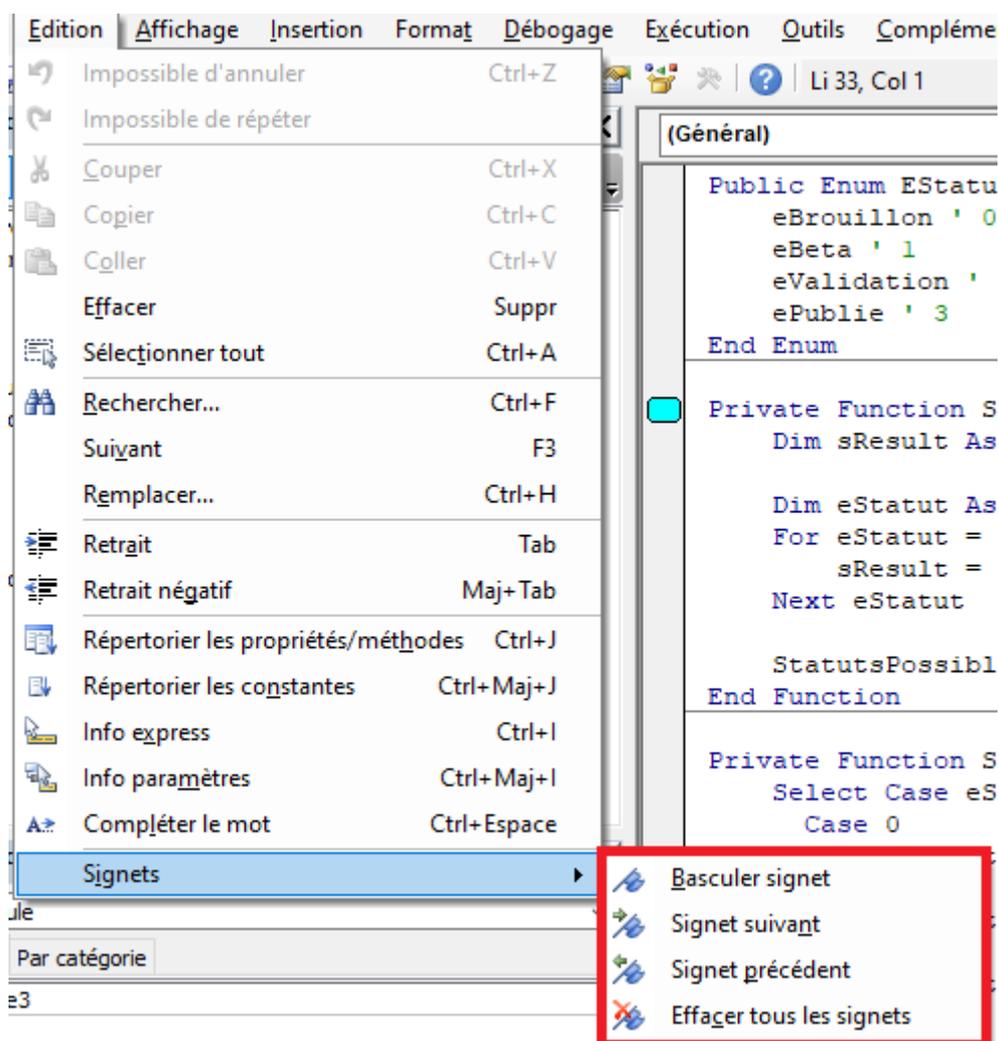


FIGURE 1.10. – Les actions possibles pour les signets

Pendant cette première section, nous avons vu comment utiliser les outils liés à l'édition de code.

2. La fenêtre des propriétés

Les éléments constituant le projet (feuilles, classeur, formulaires et leurs contrôles, ainsi que modules) ont des propriétés.

2. La fenêtre des propriétés

Ces propriétés peuvent être consultées et modifiées via la fenêtre du même nom (sous-menu "Fenêtre Propriétés" du menu "Affichage" ou depuis la barre d'outils "Standard" ou raccourci **F4**).

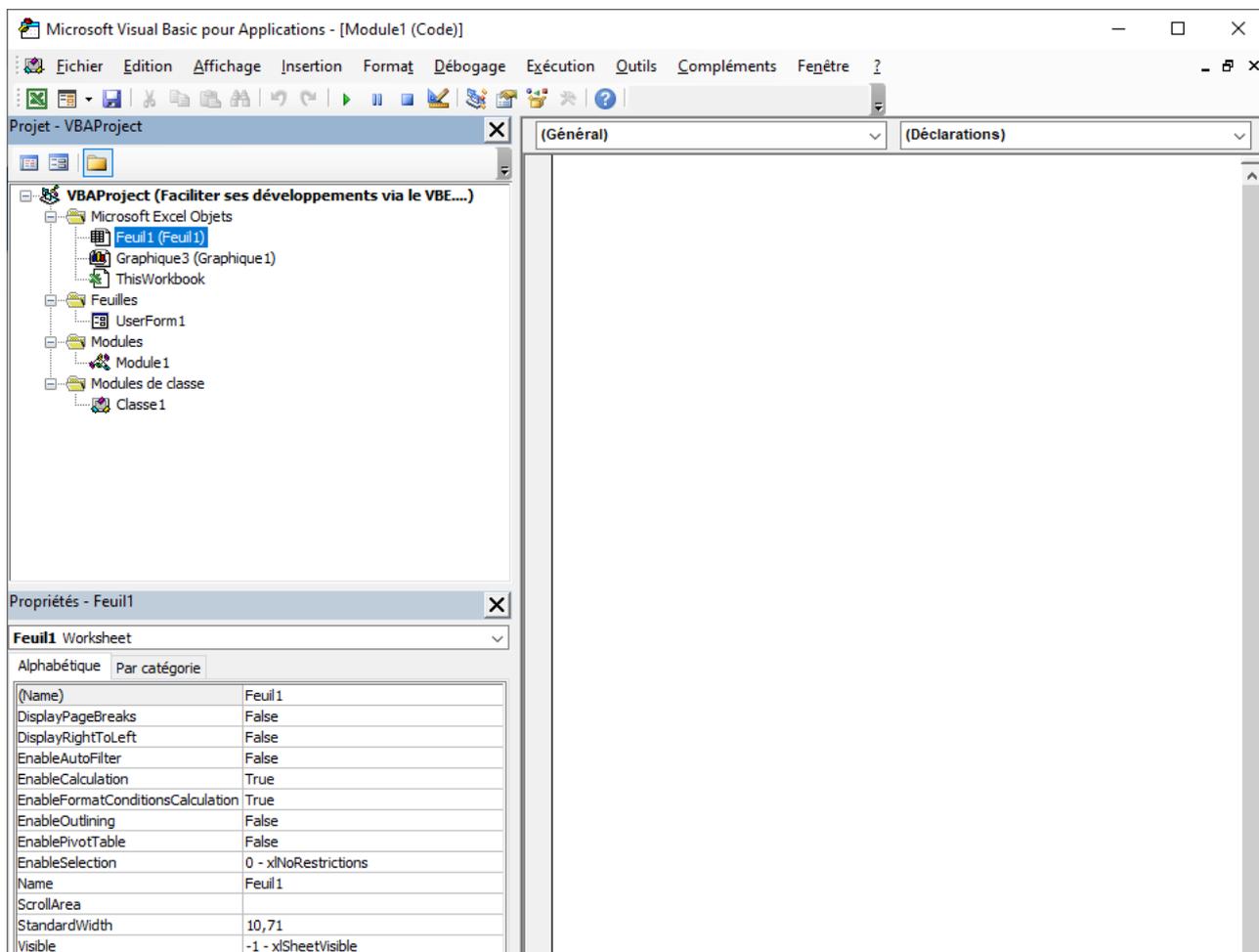


FIGURE 2.11. – La fenêtre propriétés pour un objet feuille de calcul

Comme nous pouvons le voir dans l'illustration ci-dessus, ces propriétés sont listées dans un tableau avec leurs valeurs.

2.1. Écriture et lecture

Nous pouvons donc nous en servir tant en écriture qu'en lecture.

Pour l'écriture, nous modifierons bien souvent la propriété **Name** par exemple afin d'avoir un nom explicite et savoir à quoi correspond l'élément.

Pour la lecture, c'est une source de renseignement supplémentaire pour nous aider dans nos programmes.

2. La fenêtre des propriétés

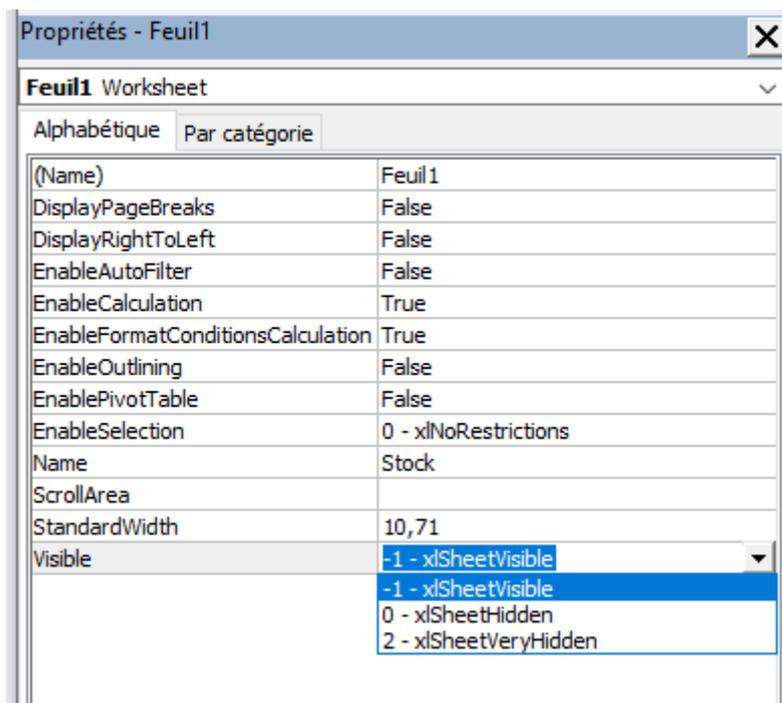


FIGURE 2.12. – Valeurs possibles pour la propriété "Visible" de la feuille sélectionnée

2.2. Tris

Nous pouvons trier ces propriétés par ordre alphabétique ou par catégorie. Cette dernière option est utile lors de la conception de formulaires.

2.3. Accès hors VBE

Enfin, cette fenêtre de propriétés peut aussi être utilisée hors VBE, via le bouton "Propriétés" dans le ruban "Développeur".

Pour des contrôles *ActiveX*, il faut également penser à se mettre en "Mode Création" :

3. L'explorateur d'objets

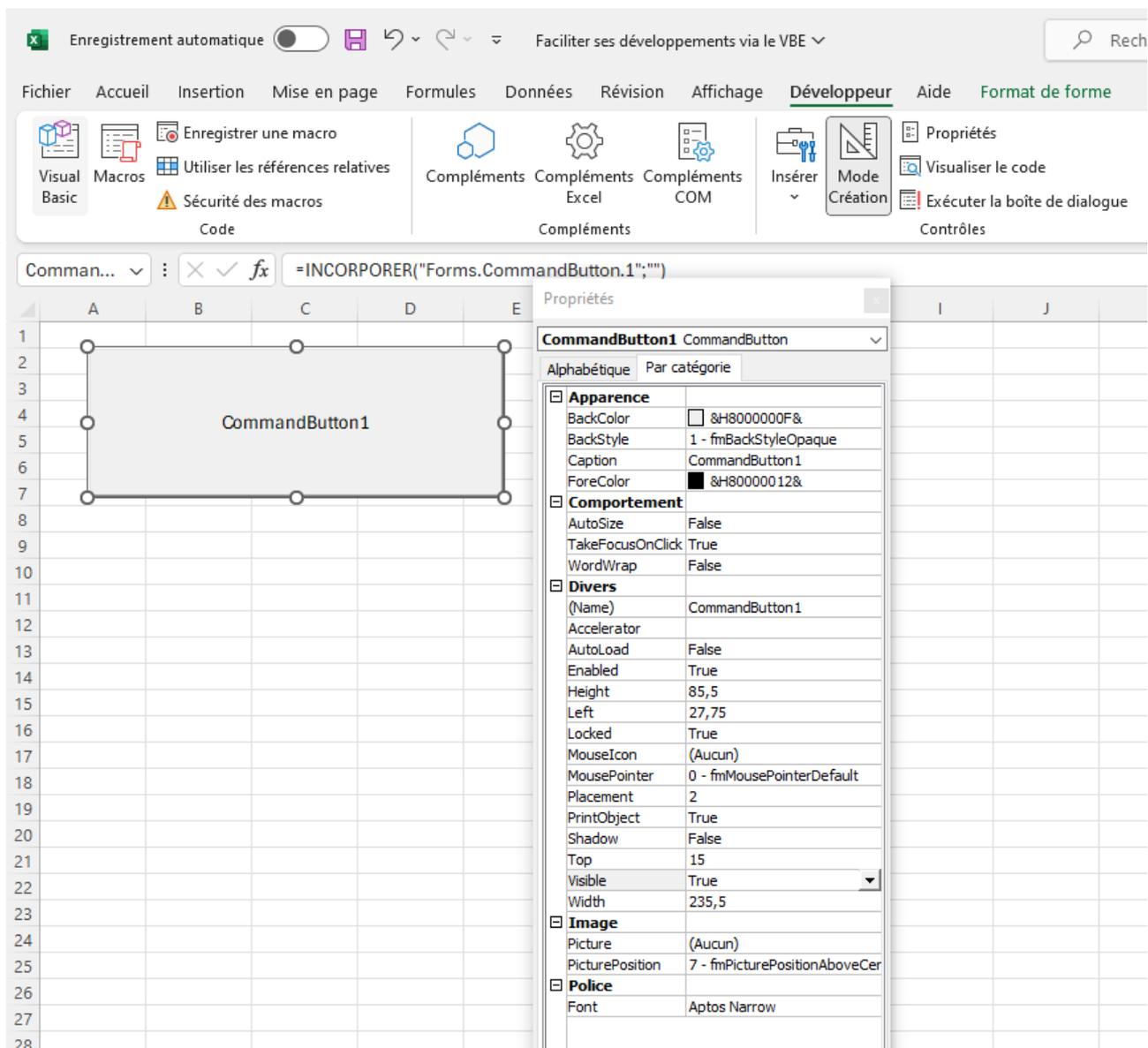


FIGURE 2.13. – Fenêtre de propriétés pour un élément de contrôle ActiveX

Remarquez que nous avons trié les propriétés par catégorie dans ce dernier exemple ce qui est assez pratique.

Durant cette seconde section, nous avons présenté la fenêtre des propriétés.

3. L'explorateur d'objets

L'explorateur d'objets sert, comme son nom l'indique, à explorer les objets du projet, qu'ils fassent partie de nos bibliothèques (VBAProject, ...) ou de bibliothèques tierces (Excel, ...).

3. L'explorateur d'objets

3.1. Accès

Pour ouvrir cette fenêtre, il faut soit passer par le sous-menu "Explorateur d'objets" du menu "Affichage" ou depuis la barre d'outils Standard ou par le menu contextuel ou encore par le raccourci **F2**.

3.2. Utilisation

La fenêtre se présente ainsi :

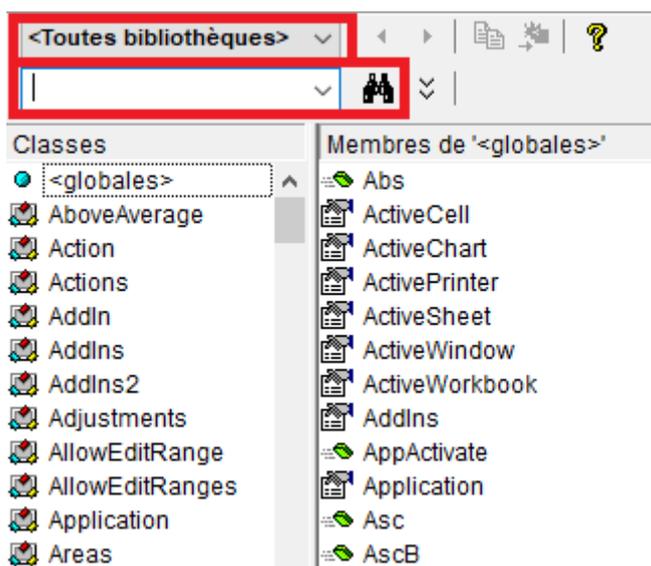


FIGURE 3.14. – Explorateur d'objets

3.2.1. Sélection de bibliothèque et recherche

La liste en haut à gauche sert à choisir la bibliothèque à explorer tandis que la zone de texte dessous permet d'effectuer une recherche pour la ou les bibliothèques sélectionnées.

3. L'explorateur d'objets

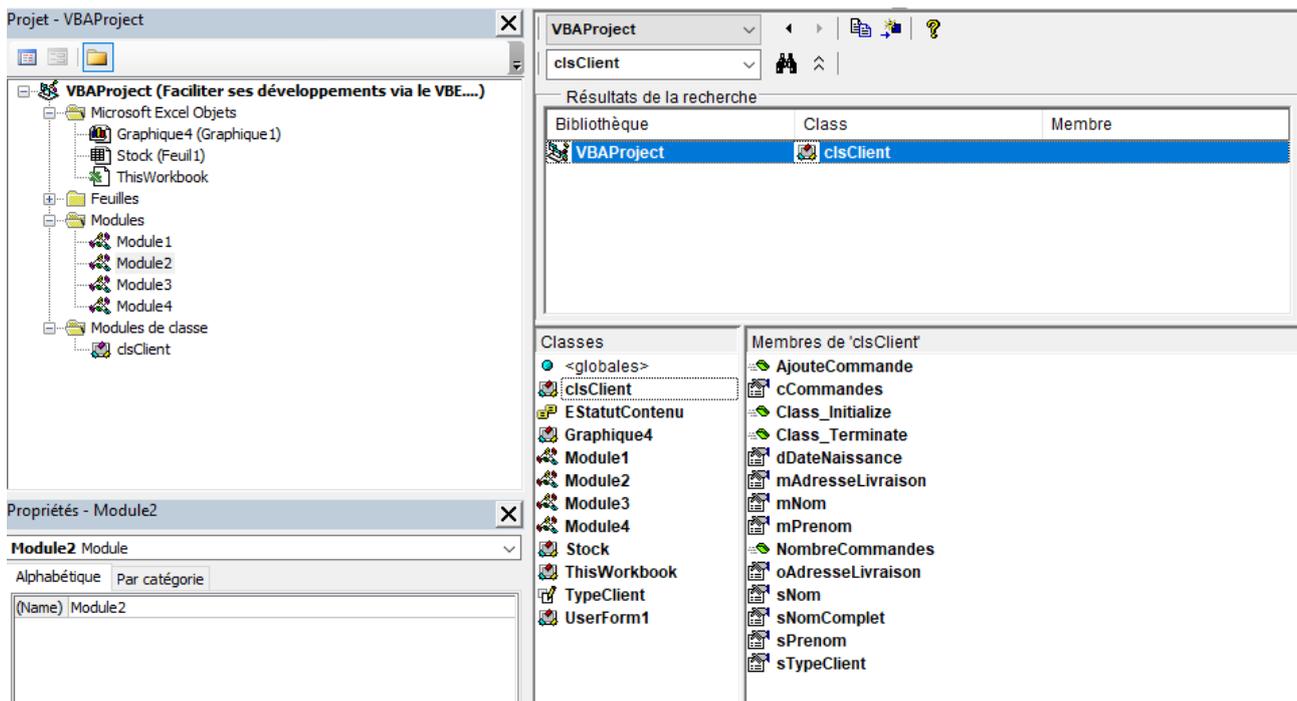


FIGURE 3.15. – Résultat d'une recherche dans l'explorateur d'objets



Encore une fois, la bibliothèque VBAProject est le nom par défaut pour notre projet.

3.2.2. Classes vs membres

Les différents éléments sont représentés par des icônes spécifiques en fonction de ce qu'ils sont (type personnalisé, énumération, classe, etc.)

La partie gauche contient les classes tandis que la partie droite liste les membres pour la classe sélectionnée.

3. L'explorateur d'objets

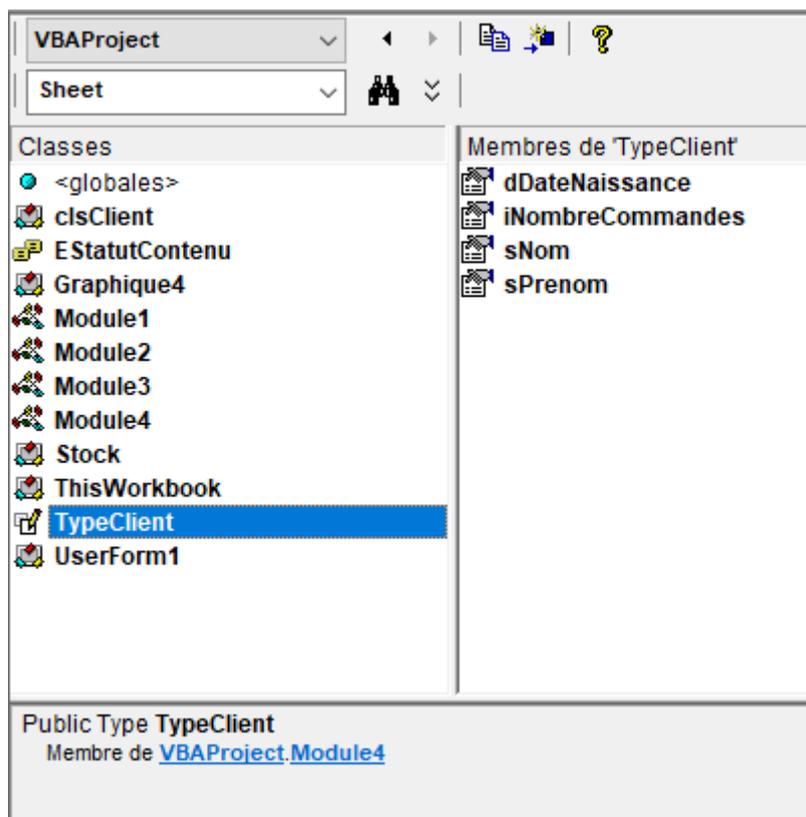


FIGURE 3.16. – Les classes à gauche et les membres à droite



Certains types d'éléments sont listés dans la partie gauche (type personnalisé par exemple) même s'ils sont définis dans ce qui pourrait être une classe (module, fichier classe, etc.).

3.2.3. Copie, accès définition et petite information

En haut de la fenêtre dans la partie droite, nous trouvons trois boutons, dont deux nous intéressent dans l'immédiat.

Tout à gauche se trouve un bouton pour copier le nom dans le presse papier. À droite de celui-ci, il y a un bouton pour accéder aux lignes de code définissant l'élément (possible pour notre code uniquement).

En bas de la fenêtre se situe une petite information selon l'élément sélectionné (le prototype pour une fonction par exemple) ainsi que l'appartenance de l'élément.

3. L'explorateur d'objets

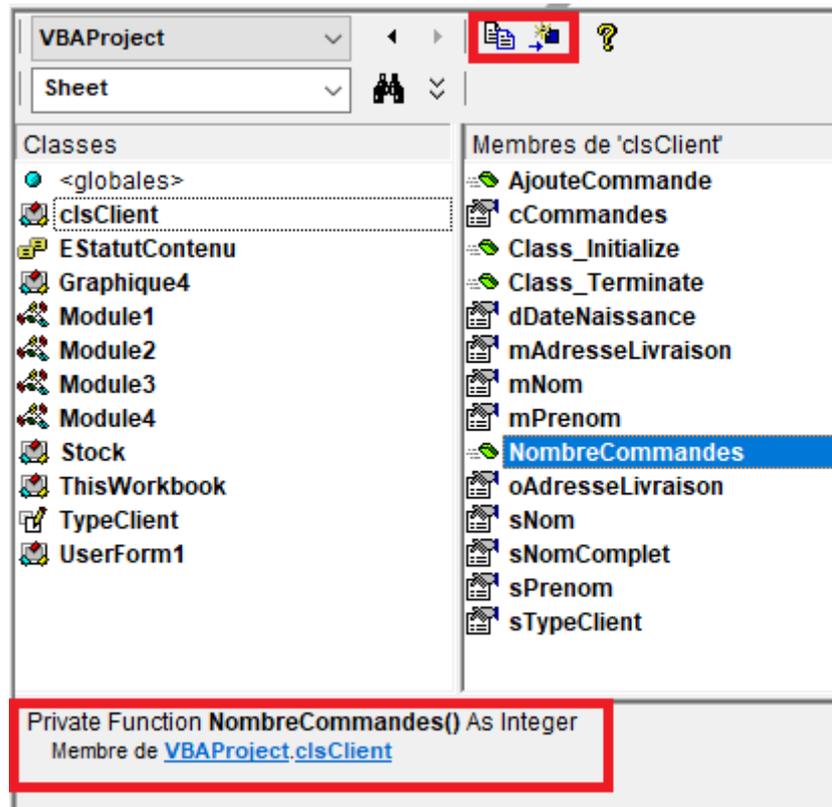


FIGURE 3.17. – Copie, accès à la définition et prototype

3.2.4. Éléments masqués

Parfois, des éléments sont volontairement masqués. Pour afficher ou masquer ceux-ci de l'explorateur d'objets, il faut respectivement cocher ou décocher l'option "Afficher des membres masqués" du menu contextuel.

3. L'explorateur d'objets

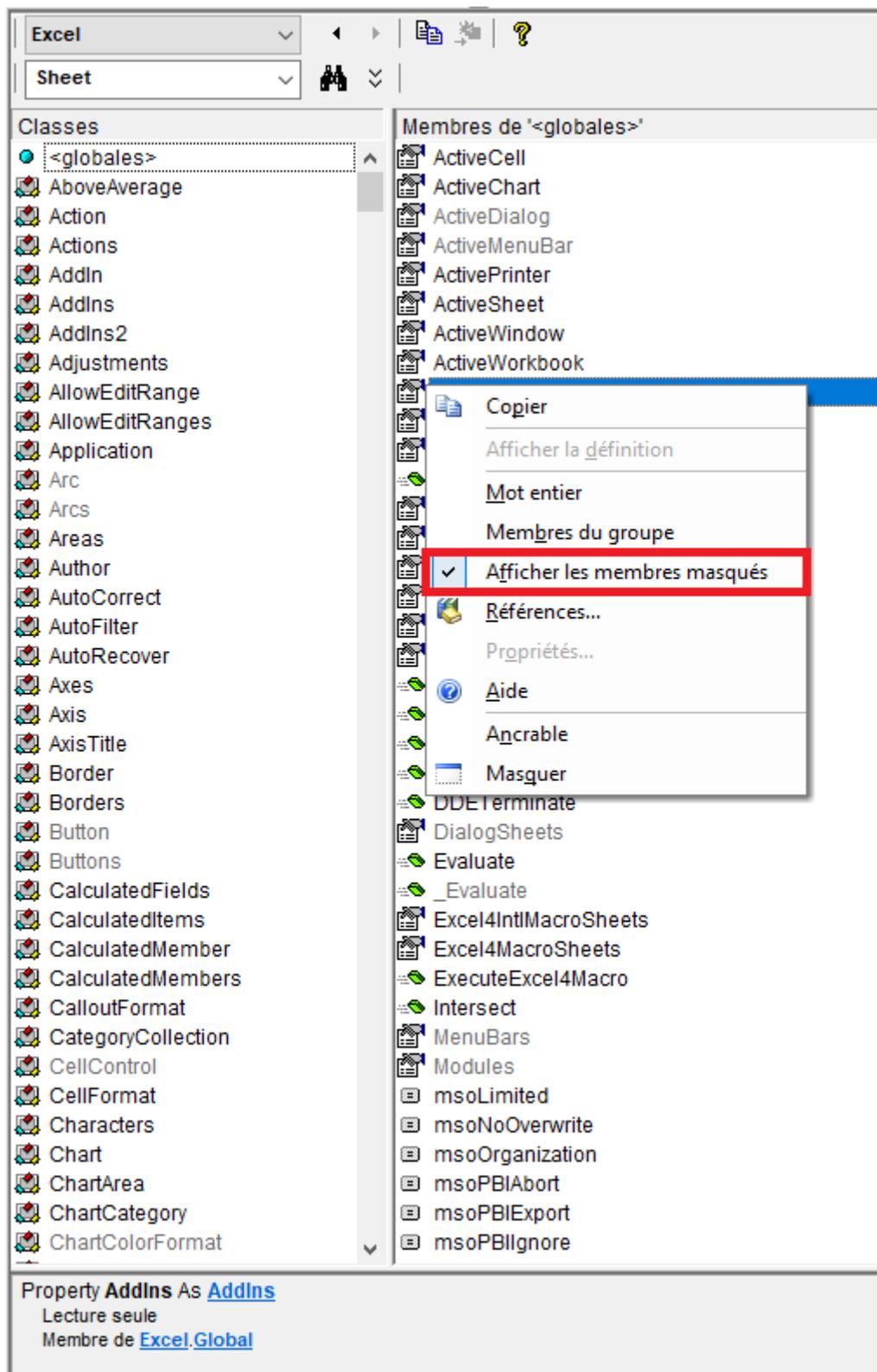


FIGURE 3.18. – Option "Afficher les membres masqués"

3.2.5. Aide en ligne

Nous pouvons nous dire qu'il manque tout de même une description courte ou même une plus grande explication avec des exemples. C'est là que l'aide en ligne intervient !

3. L'explorateur d'objets

En cliquant sur le bouton "Aide" pour l'élément sélectionné, la page correspondante de l'aide en ligne s'ouvre. Nous pouvons aussi y accéder via le menu contextuel.

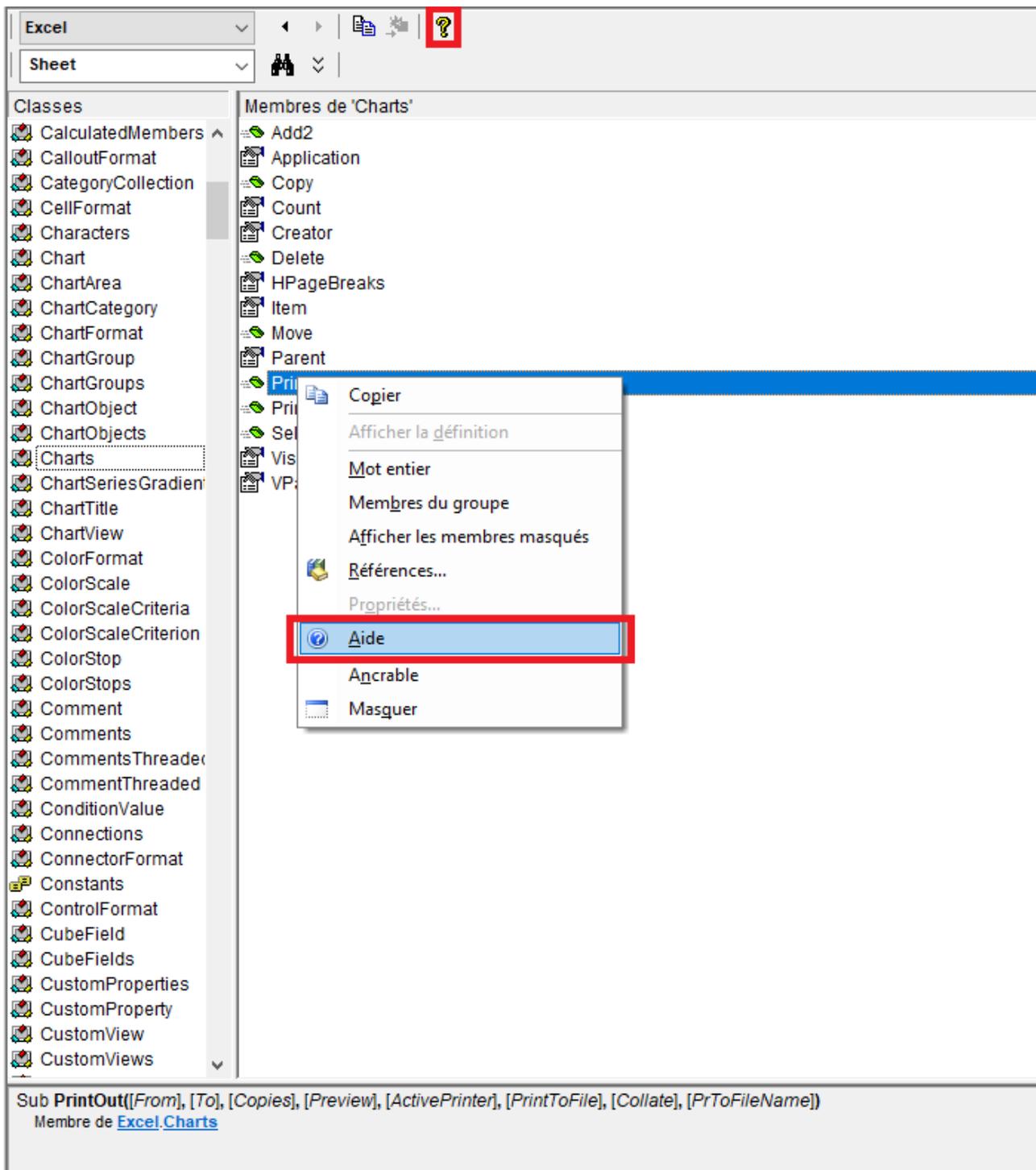


FIGURE 3.19. – Aide en ligne pour l'élément sélectionné

Au fil de cette section, vous avez pu vous familiariser avec l'explorateur d'objets.

4. La fenêtre d'exécution

4. La fenêtre d'exécution

Nous avons déjà eu l'occasion de parler de la fenêtre d'exécution dans un précédent billet sur le débogage.

En VBA, nous pouvons écrire dedans via `Debug.Print`.

Il est aussi possible d'exécuter du code directement dedans pour tester ou vérifier des choses :



```
Exécution
? Sheets(1).Name
Graphique1
? Sheets(2).Name
Stock
? 5 + 5
10
|
```

FIGURE 4.20. – Fenêtre d'exécution (CTRL + G)

i

Au sein de cette fenêtre, le `?` a la même effet que `Debug.Print`.

Voilà pour cette courte section sur la fenêtre d'exécution.

Conclusion

C'est déjà la fin de ce billet.

Au cours de celui-ci, nous avons vu comment faciliter notre travail sur VBE via l'éditeur de code et ses menus, la fenêtre des propriétés, l'explorateur d'objets ou encore la fenêtre d'exécution.

Pour être davantage complet, il faudrait également aborder l'aide à la conception de formulaires (`UserForm`), mais ce sujet dépasse l'étendue de ce billet.

À bientôt !

Quelques ressources :

- [Cette page](#) 
- [Cette page](#) 

Liste des abréviations

EDI Environnement de Développement Intégré. 1

VBA Visual Basic for Applications. 1