

VBA

POUR

AutoCAD

R14.01, R2000, R2002, R2004

LE LIVRE

Roger ROSEC

VBA

POUR

AutoCAD

Copyright Fasoft 2001-2003

VBA et son environnement

Dans ce chapitre

1.1 - Programmer AutoCAD

1.1.1 – Le VBA

1.1.2 - Comparaison VBA et Visual Basic

1.1.3 - Comparaison VBA et AutoLISP

1.2 – Installation de VBA

1.2.1 – Installer une barre d'icônes VBA

1.3 - Ouvrir l'environnement VBA

1.3.1 - Le gestionnaire VBA

1.3.2 - L'éditeur Visual Basic

1.3.2.1 - La fenêtre de Projet

1.3.2.2 – La fenêtre Propriétés

1.3.2.3 – Les Modules de code

1.3.2.4 – Les Feuilles (*ou UserForms*)

1.3.2.5 – La boîte à outils des Contrôles

1.4 – Obtenir de l'aide

1.4.1 – L'aide VBA générale de Microsoft

1.4.2 – L'aide spécifique VBA pour AutoCAD

1.1 - Programmer AutoCAD

AutoCAD possède deux langages principaux de programmation. Une fois que vous avez décidé de programmer AutoCAD, la première étape consiste à sélectionner le langage de programmation à utiliser.

1.1.1 - Le VBA

Visual Basic pour Applications (VBA) est un langage de programmation et un environnement inclus avec beaucoup d'applications Microsoft — telles que Word, Excel et Access, mais aussi dans d'autres programmes indépendants de Microsoft. C'est ainsi que depuis la version 14, il est également disponible avec AutoCAD. VBA est idéal lorsque vous devez travailler avec des applications différentes ensemble, pour transmettre des données de l'une à l'autre. ActiveX permet d'accéder aux objets qui sont disponibles dans d'autres applications. Cependant, vous pouvez aussi utiliser VBA pour programmer AutoCAD tout seul.

VBA fournit une façon simple de personnaliser AutoCAD, automatiser des tâches et programmer des applications.

1.1.2 - VBA et VB

VBA est comparable, en grande partie au Visual Basic dont il est une émanation. Au premier abord, VBA ressemble beaucoup à Visual Basic, à commencer par l'environnement de développement intégré (EDI). Il y a néanmoins des différences notables.

La principale consiste dans le fait que Visual Basic n'est relié à aucune application spécifique. Il est conçu pour créer des applications autonomes. Le code de Visual Basic est compilé dans un fichier exécutable qui fonctionne seul, sans aucun lien avec une autre application.

VBA, lui, doit être connecté à son application hôte et à un document dans lequel le code a été créé.

Par exemple, un projet VBA créé pour fonctionner avec AutoCAD, son application hôte, ne pourra fonctionner directement dans Word ou Excel mais des connexions entre applications qui le supportent pourront y être facilement établies.

Dans ce livre vous trouverez aussi des exemples d'applications créées avec Visual Basic qui interfèrent sur AutoCAD;

1.1.3 - VBA et AutoLISP

VBA possède les avantages suivants :

- * Il est plus rapide qu'AutoLISP, même quand AutoLISP est compilé.
- * VBA est commun à beaucoup d'autres applications. Si vous avez utilisé VBA auparavant, vous pourrez facilement transférer vos connaissances pour utiliser VBA dans AutoCAD. Il est aussi plus facile de trouver d'autres programmeurs qui connaissent VBA plutôt qu'AutoLISP.
- * VBA est généralement plus facile à apprendre qu'AutoLISP à cause de sa syntaxe nettement plus simple et de son langage objet plus facilement compréhensible.
- * VBA est plus graphique. L'ergonomie des applications est beaucoup plus aisée à programmer.

De l'autre côté, AutoLISP a l'avantage d'une compatibilité descendante avec les versions précédentes d'AutoCAD.

Il n'est pas possible de traduire un programme AutoLISP en VBA et inversement, les concepts étant beaucoup trop différents.

Néanmoins, chaque fois que cela sera possible, nous indiquerons les correspondances entre ces deux modes de programmation.

Note :

Les commandes d'AutoCAD correspondent à la version française. Néanmoins nous essaierons d'indiquer la correspondance anglaise la première fois qu'il en ait fait mention, entre parenthèses en style `courrier`.

1.2 – Installation de VBA

Vous devez posséder au moins la version 14 d'AutoCAD. Mais pour VBA, la mise à jour 14.01 est indispensable. Si vous avez la version 14, votre revendeur vous fournira cette mise à jour gratuitement.

Depuis AutoCAD 2000, l'installation est maintenant automatique, le menu déroulant est en français.

Vérifiez quand même si le VBA a été installé sur votre machine. Pour cela, déroulez le menu *Outils* et voyez si vous avez une ligne d'option indiquant *Macro VBA* (ou *Macro* pour les versions anglaises).

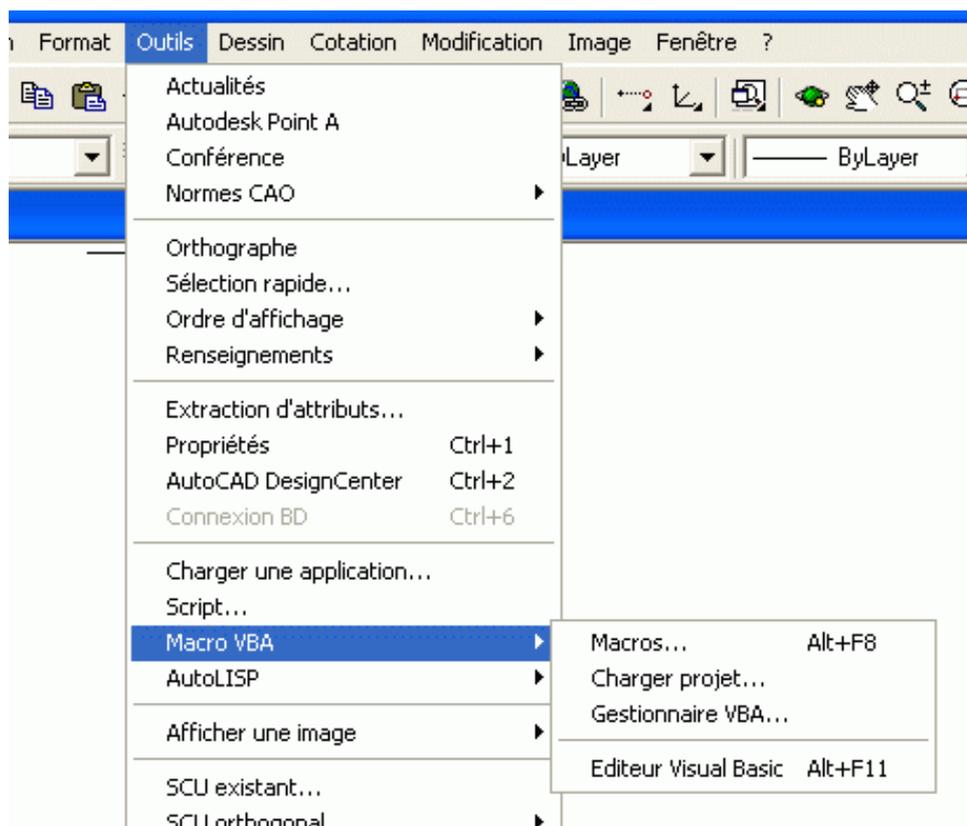


Figure 1-1 Le menu déroulant de VBA qui apparaît quand il a été installé.

Sinon, vous devrez l'installer.

Pour cela, reprenez votre cédérom AutoCAD. Lancez le programme d'installation en double-cliquant sur *Setup.exe*. Vous verrez apparaître l'écran de la figure 1.2 qui vous demandera de choisir une option.

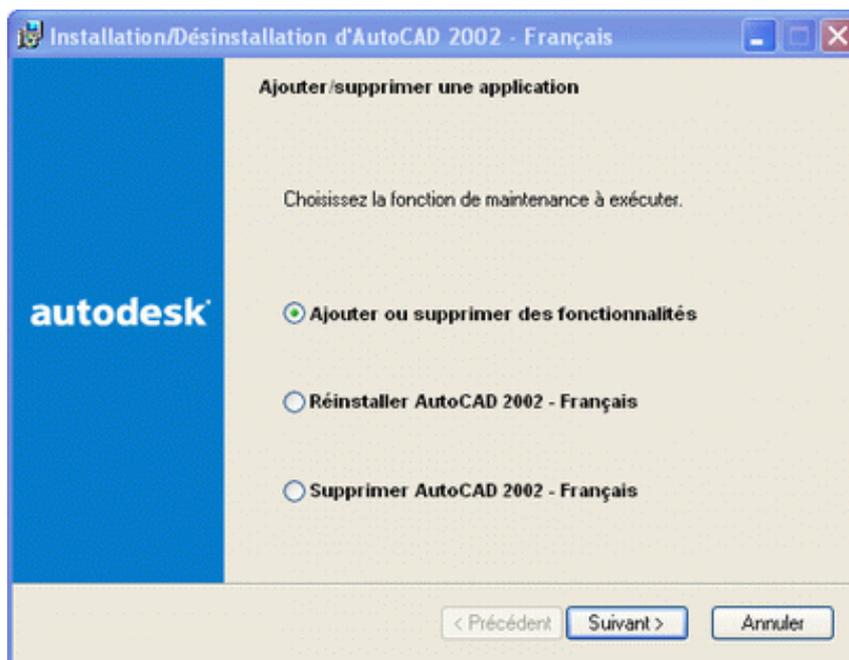


Figure 1-2 L'écran des options d'installation d'AutoCAD.

Choisissez l'option *Ajouter* qui vous permettra d'installer de nouveaux composants à votre configuration actuelle.

Sur l'écran suivant sélectionnez l'option *Support VBA*.

Vous devez disposer de 8 Mo d'espace libre (dans la version R2000). Comme indiqué, si vous travaillez en réseau ou sous Windows XP, vous devez posséder tous les droits pour écrire dans le répertoire d'installation ainsi que dans la base de registres de Windows.

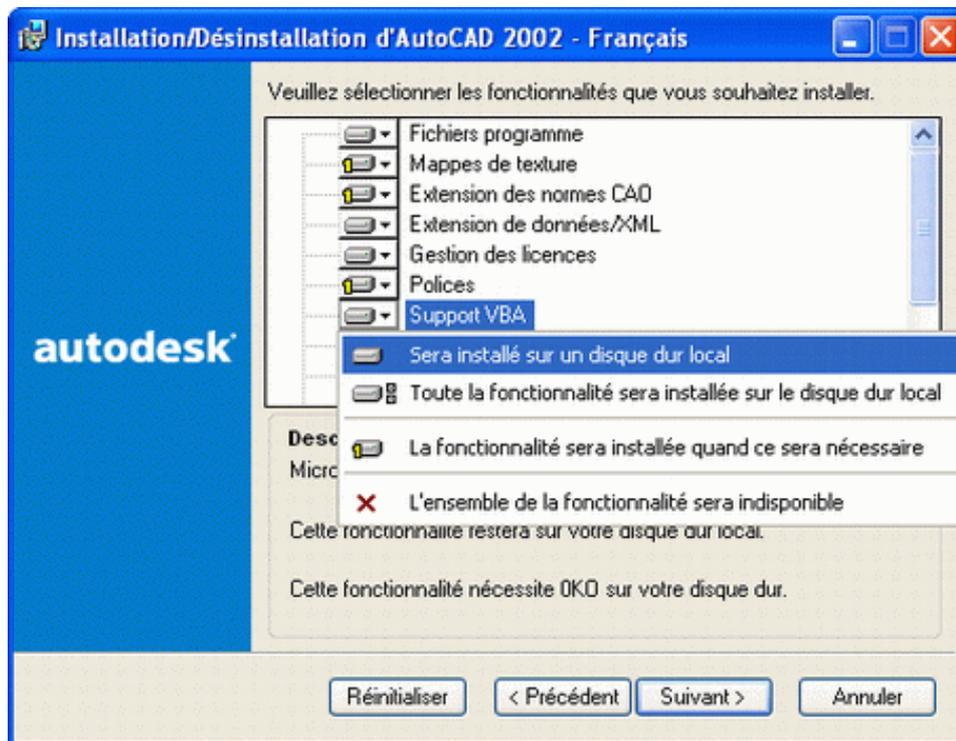


Figure 1-3 Choix des nouveaux composants à installer.

1.2.1 – Une barre d'icônes VBA

Si vous désirez une barre d'icônes VBA toute faite, vous en trouverez une prête à l'emploi dans le répertoire **Fichiers/Chap01/Menu**.

Pour simplifier, placez le fichier **vbaa.mns** et les 5 fichiers d'icônes **.bmp** dans le répertoire Support d'AutoCAD par un *Copier/Coller*. (Plus loin, nous apprendrons à regrouper tous ces fichiers .bmp dans un seul fichier *.dll*.)

La première fois, chargez le menu **VBAA** par le menu *Outils* ⇒ *Personnalisation des menus*, (ou *Personnaliser* puis *Menus* selon la version d'AutoCAD).

Cliquez sur *Parcourir*, choisissez *VBAA.mns* dans le répertoire **Support** puis cliquez sur le bouton *Charger* quand vous êtes de retour dans la fenêtre *Personnalisation des menus*. La barre de menus VBA doit apparaître. Fermez la fenêtre *Personnalisation des menus*.

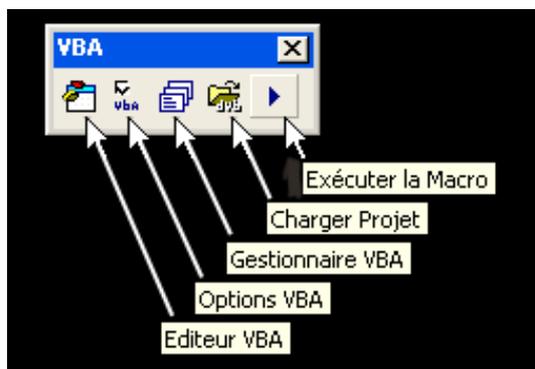


Figure 1-4 La barre d'icônes VBAA.

Les fois suivantes la barre sera chargée automatiquement.

Nota : Pour AutoCAD 2004 et +, il s'agit du répertoire *Support* qui se trouve dans votre espace réservé, c'est-à-dire : `c:\Documents and Settings\Toto\Application Data\Autodesk\AutoCAD 2004\R16.0\Enu\Support\`. Remplacez Toto par votre nom d'utilisateur.

1.3 - Ouvrir l'environnement VBA

AutoCAD offre un environnement facilitant la programmation et l'utilisation du VBA. Pour commencer à travailler avec VBA, vous devez ouvrir l'environnement VBA. VBA possède sa propre interface.

1.3.1 - Le gestionnaire VBA

Les programmes VBA sont sauvegardés dans les *projets*. Un projet contient tous les éléments dont on a besoin pour exécuter le programme. Vous pouvez utiliser le *Gestionnaire VBA* pour visualiser vos projets VBA. Le Gestionnaire VBA vous permet aussi de charger, décharger, enregistrer, créer, incorporer et extraire des projets VBA. Pour ouvrir le Gestionnaire VBA, utilisez une des manières suivantes :

- Choisissez *Outils Macro VBA Gestionnaire VBA*
- Cliquez sur l'icône  de la barre d'outils VBA.
- Tapez **gestvba** (ou **vbaman** en version US) sur la ligne de commande.

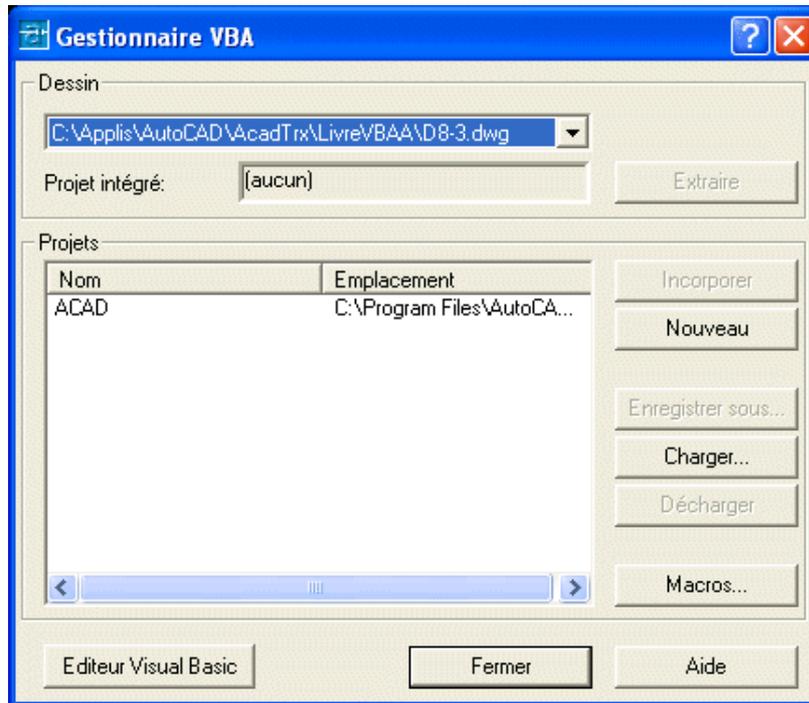


Figure 1-5 Le gestionnaire VBA.

Le Gestionnaire VBA nous sera très utile pour gérer les programmes VBA voire même indispensable pour les renommer ou les décharger de la mémoire.

1.3.2 - L'éditeur Visual Basic

Pour ouvrir VBA dans AutoCAD, vous avez le choix entre :

- Taper **vbaide** _↑ sur la ligne de commande d'AutoCAD (toutes versions).
- Choisir *Outils* ⇒ *Macro VBA* ⇒ *Editeur Visual Basic*.
- Cliquer sur l'icône  de la barre d'outils VBA
- Cliquer sur le bouton *Editeur Visual Basic* dans la fenêtre du *Gestionnaire VBA*

AutoCAD affiche alors la fenêtre d'environnement VBA appelé IDE (ou EDI), pour Environnement de Développement Intégré.

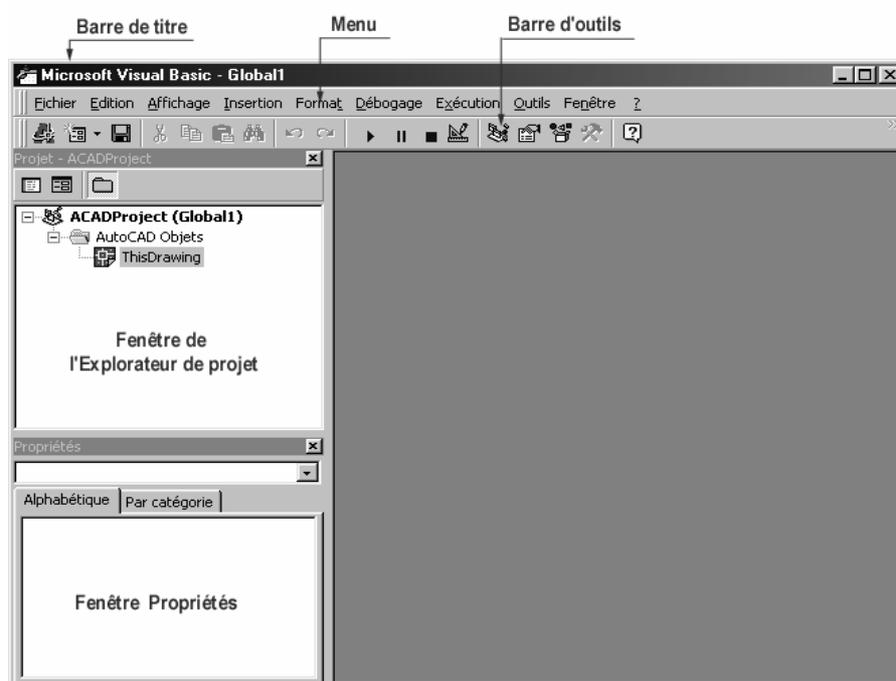


Figure 1-6 L'Environnement de Développement Intégré de VBA, version 2000 fr

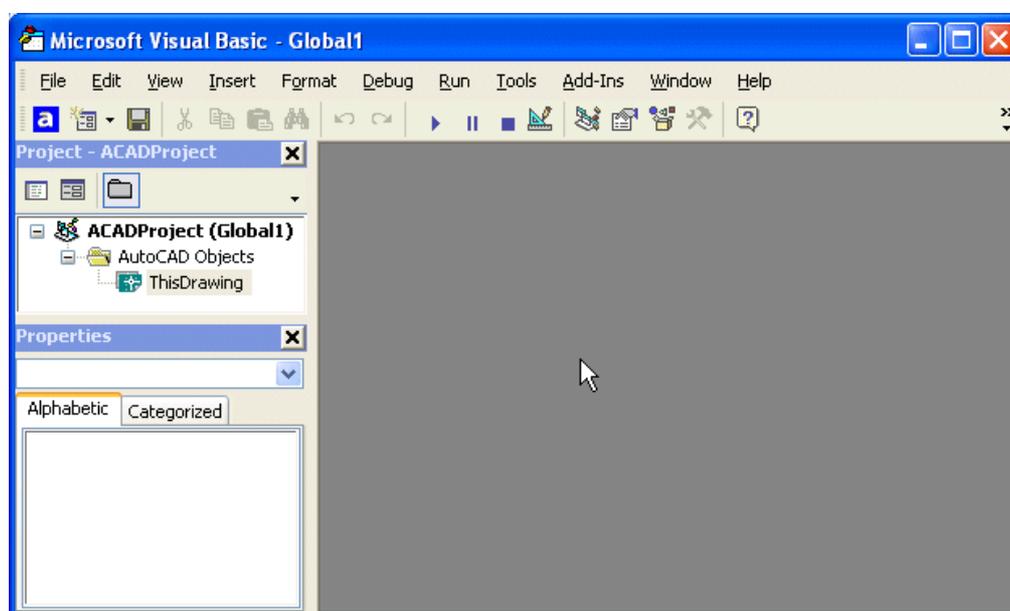


Figure 1-6b L'Environnement de VBA, version 2004 US sous XP

Une des caractéristiques de Visual Basic est de posséder de multiples fenêtres. VBA n'y déroge pas. Pour le moment, deux fenêtres apparaissent sur la gauche de l'écran. Il se peut que les fenêtres apparaissent sur la droite de l'écran. VBA permet à l'utilisateur de choisir l'endroit approprié pour les fenêtres qui peuvent se déplacer à loisir.

Vous avez également des menus déroulants et la barre d'outils *Standard* qui vous seront utiles pour écrire, déboguer et lancer vos programmes. D'autres barres d'outils sont disponibles. Nous les étudierons au fur et à mesure de nos besoins.

Selon votre version d'AutoCAD et votre système d'exploitation, votre éditeur s'affichera soit en français, soit en anglais.

1.3.2.1 - La fenêtre Explorateur de Projet (project explorer)

 La fenêtre *Explorateur de projet* montre tous les projets qui sont ouverts. Chaque projet peut contenir plusieurs classeurs et cette fenêtre y affiche tous les composants. Au lancement de l'éditeur, vous n'avez qu'un seul projet que VBA nomme d'office *Global1* dans lequel vous trouvez déjà un classeur nommé *AutoCAD Objets*.

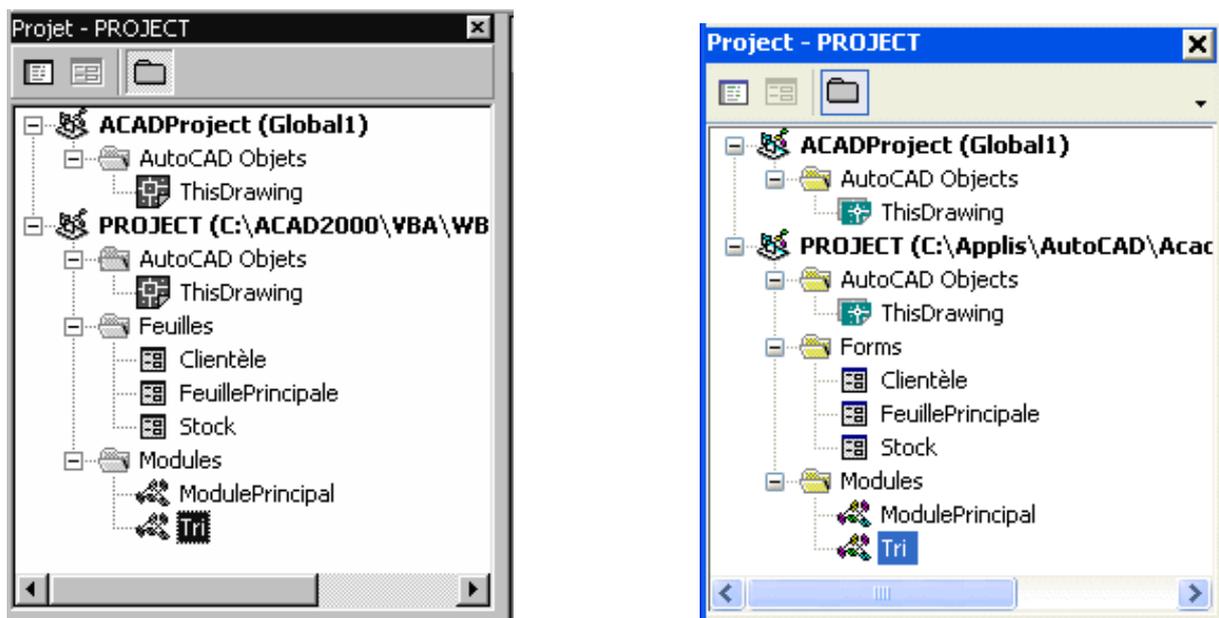


Figure 1-7 Exemples (versions 2000 et 2004) de fenêtre de l'Explorateur de projet qui comporte plusieurs classeurs, ici tous ouverts.

Vous utiliserez cette fenêtre pour vous déplacer facilement d'un élément à l'autre.

1.3.2.2 – La fenêtre Propriétés (properties)

 La fenêtre *Propriétés* permet de définir et d'éditer les propriétés des modules, feuilles et contrôles insérés dans votre projet. Cette fenêtre va lister toutes les propriétés de l'objet ou de la feuille sélectionnée ainsi que leurs valeurs correspondantes.

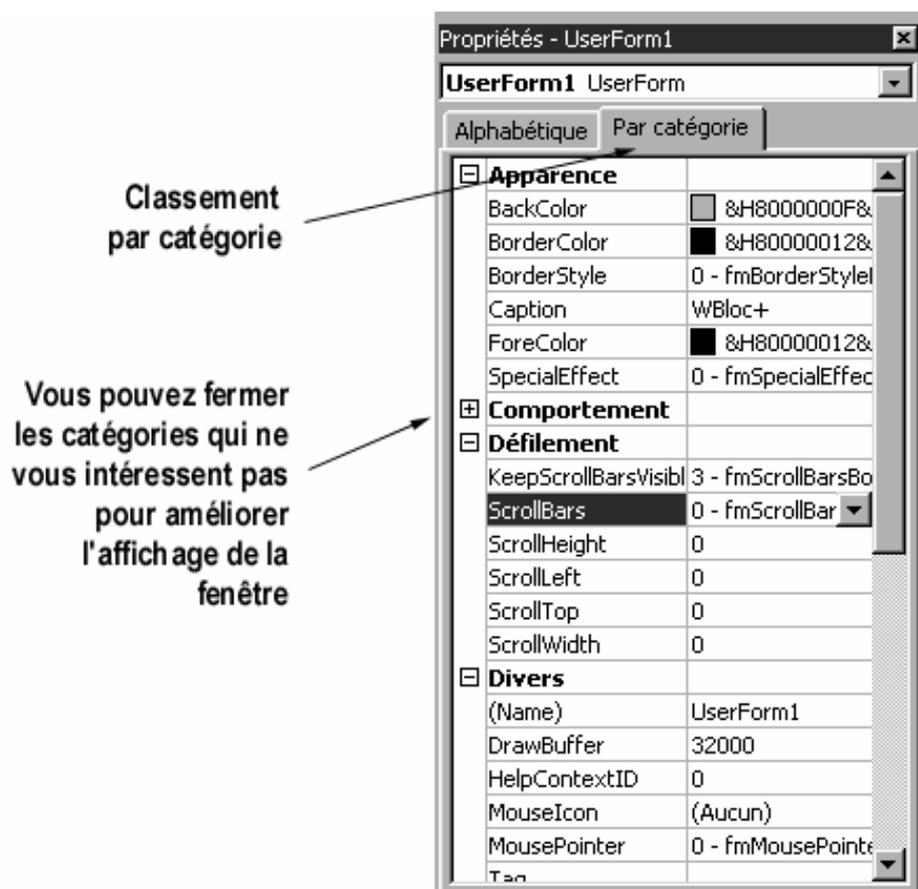


Figure 1-8 Exemple de fenêtre Propriétés, avec ici les propriétés triées par catégorie.

Grâce aux deux formes d'affichage des propriétés (*Alphabétique* et *Par catégorie*) il est facile de définir les caractéristiques des objets insérés dans la feuille. Vous pouvez sélectionner directement les objets dans la liste déroulante affichée dans la partie haute de la boîte.

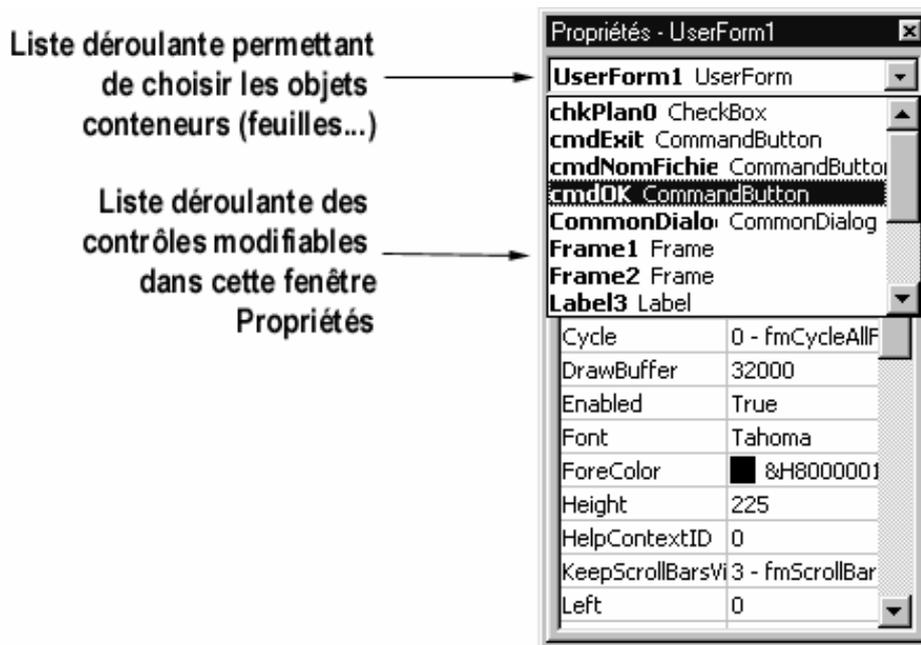


Figure 1-9 Les différents éléments de la fenêtre Propriétés ; liste triée par ordre alphabétique.

1.3.2.3 – Les Modules de code

 Certains éléments du projet VBA sont contenus dans des *modules*. Un module est un morceau indépendant de code de programmation. Le projet VBA peut contenir un ou plusieurs modules. Il peut également n'en contenir aucun.

Pour ouvrir un module, choisissez *Insertion* ⇒ *Module*



Figure 1-10 Une première manière pour ouvrir un nouveau module.

ou cliquez sur la flèche déroulante à droite du second bouton sur la barre d'outils Standard de VBA, puis choisissez *Module*.

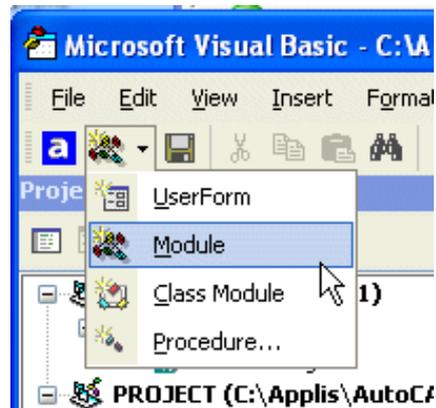


Figure 1-11 Une autre manière pour ouvrir un nouveau module.

VBA ouvre alors une fenêtre de l'éditeur de texte de code ; vous pouvez ainsi commencer à entrer le code. Vous pouvez constater que dans la fenêtre du Projet, VBA ajoute un nouveau module à la liste des modules. Par défaut, le premier module est appelé Module1. La figure 1-12 affiche l'interface VBA qui apparaît alors, y compris l'éditeur de texte servant à écrire ou modifier le code.

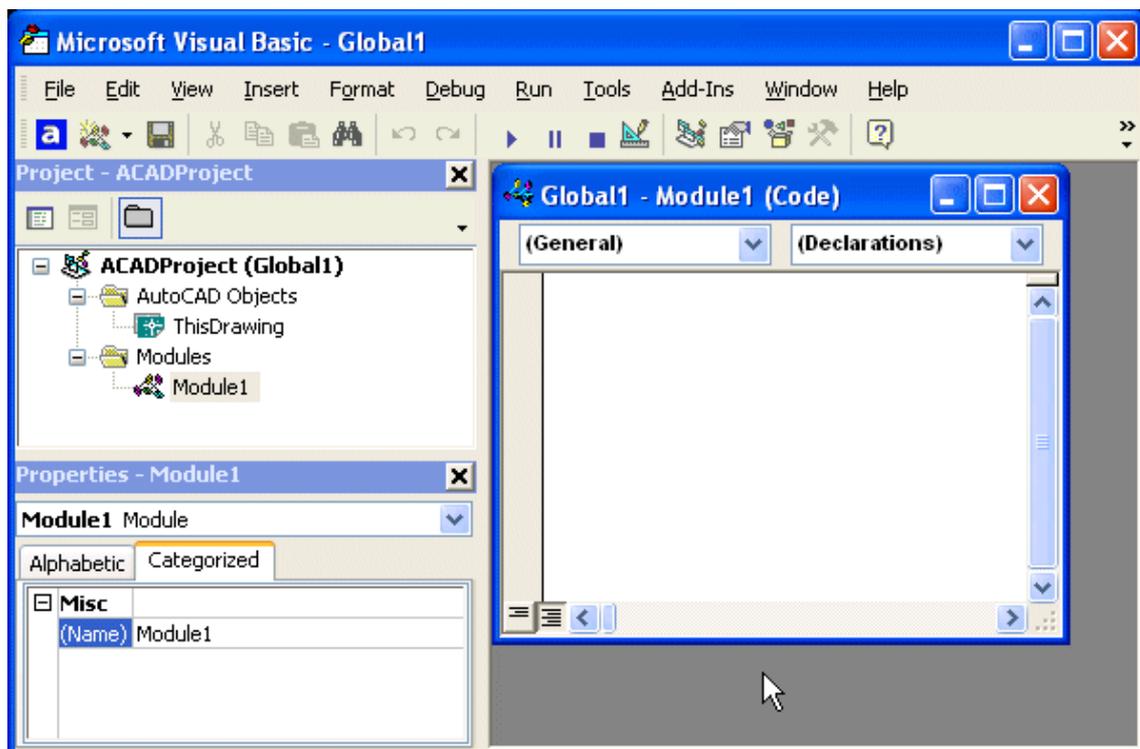


Figure 1-12 Un premier module est maintenant inséré dans le projet.

Astuce

Vous pouvez redimensionner l'éditeur de texte du module comme toute autre fenêtre. Il est plus facile de travailler avec une grande fenêtre lorsque vous écrivez du code dans l'éditeur de texte. Cliquez sur le bouton de maximalisation de la fenêtre (en haut, à droite, bouton central) pour agrandir l'éditeur de texte à sa dimension maximale.

1.3.2.4 – Les Feuilles (ou *UserForms*)

 Sous VBA on appelle *Feuille* les fenêtres ou boîtes de dialogue dans le projet. Pour VBA, le mot clé qui désigne une feuille est *UserForm*. Les feuilles servent à recevoir les *contrôles* qui y sont dessinés et affichés.

Pour ouvrir une nouvelle feuille, choisissez *Insertion* ⇒ *UserForm*



Figure 1-13 Insertion d'une feuille en passant par le menu déroulant *Insertion*.

ou cliquez sur la flèche déroulante à droite du second bouton sur la barre d'outils Standard de VBA, puis choisissez *UserForm*. Deux nouvelles fenêtres surgissent alors et la fenêtre *Propriétés* se remplit.

Quand vous insérez une nouvelle feuille :

- Chaque fenêtre de feuille comporte des boutons *Agrandissement*, *Réduction* et *Fermeture*.
- La feuille affiche une grille destinée à faciliter l'insertion des contrôles. Vous pouvez désactiver la grille de feuille ou définir l'espacement de ses lignes dans l'onglet *Général* de la boîte de dialogue *Options* du menu *Outils*, (*Show Grid* dans l'onglet *General* de la boîte *Options* du menu *Tools*).

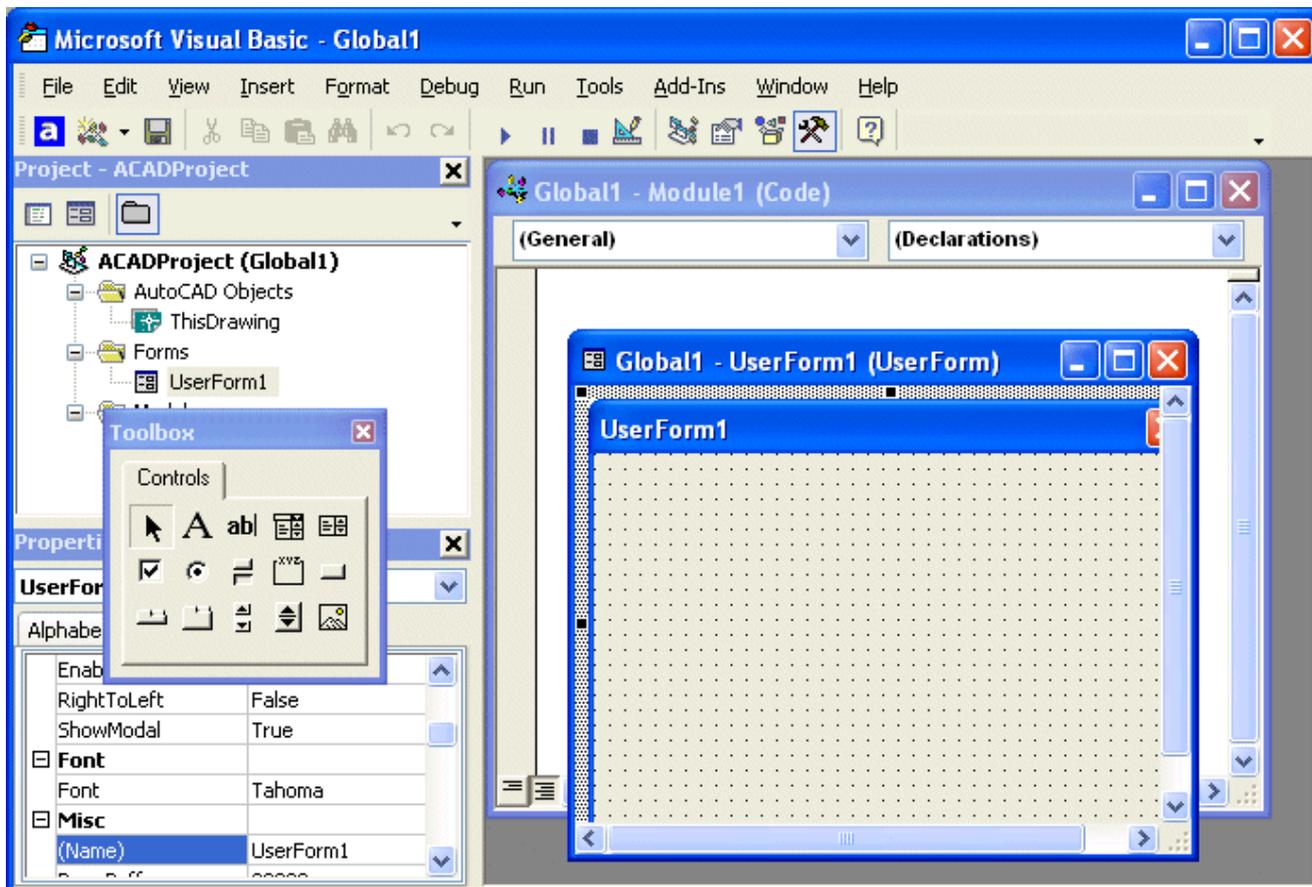


Figure 1-14 A l'insertion d'une feuille, une nouvelle feuille apparaît ainsi que la boîte à outils Contrôles.

1.3.2.5 – La boîte à outils des Contrôles

 On appelle *contrôles* (*controls*) les éléments que l'on peut sélectionner dans la **boîte à outils** et disposer sur une feuille. Les contrôles peuvent être dimensionnés et rangés comme on veut. Vous pouvez aligner automatiquement les contrôles sur la grille en sélectionnant l'option correspondante dans l'onglet *Général* de la boîte de dialogue *Options*.

Il y a 15 contrôles standard qui sont fournis par Autodesk. On peut en ajouter autant que vous en voulez. Nous verrons plus loin comment en créer.

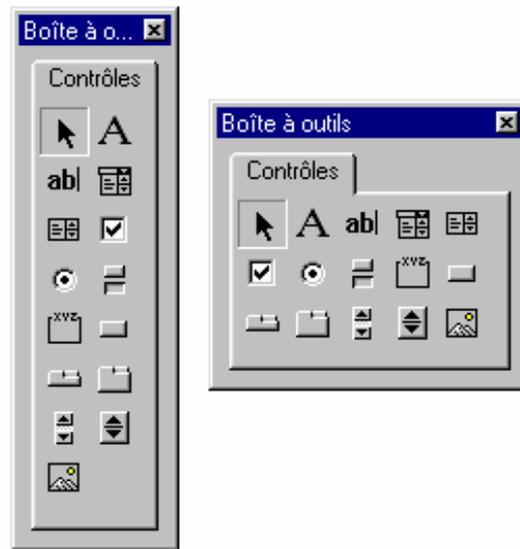


Figure 1-15 La boîte à outils *Contrôles* peut prendre différentes formes. Il suffit de tirer sur le cadre extérieur.

A ce stade, vous pouvez constater que la fenêtre de l'Explorateur de projet s'est enrichie :

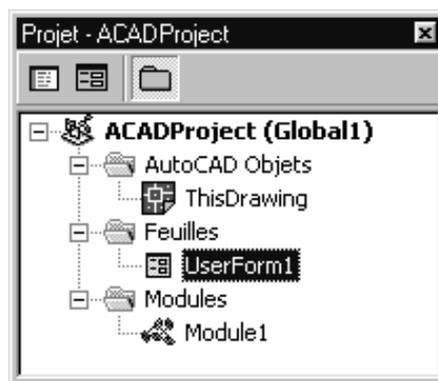


Figure 1-16 La fenêtre *Projet* a évolué.

Deux nouveaux classeurs ont été ajoutés à votre projet :

- Un classeur **Feuilles** qui contient la feuille *UserForm1*.
- Un classeur **Modules** qui contient le module de code *Module1*

Si vous avez ajouté d'autres feuilles ou modules, vous devez les y retrouver.

 AutoCAD tourne toujours en arrière-plan. Vous pouvez y revenir en cliquant à n'importe quel moment sur son bouton sur la barre des Tâches de Windows ou en cliquant sur *Affichage AutoCAD* sur la barre d'outils Standard de VBA.

1.4 – Obtenir de l'aide

VBA offre plusieurs formes d'aide dont certaines en anglais même dans une version française d'AutoCAD. On peut distinguer deux catégories :

- L'aide VBA générale de Microsoft. Selon la version, cette partie est complètement en français (à l'exception des exemples qui n'ont pas été entièrement francisés).
- L'aide VBA et ActiveX spécifique pour AutoCAD. La plus grande partie est malheureusement en anglais.

1.4.1 – L'aide VBA générale de Microsoft

L'aide de Microsoft Visual Basic pour Applications fournit des informations sur l'environnement VBA en général, applicable à tous les programmes sachant utiliser VBA : Word, Excel ...

Cliquez sur le bouton  *Aide de Microsoft Visual Basic* sur la barre d'outils VBA, ou sur la barre de menu choisissez *Aide* ⇒ *Microsoft Visual Basic*.

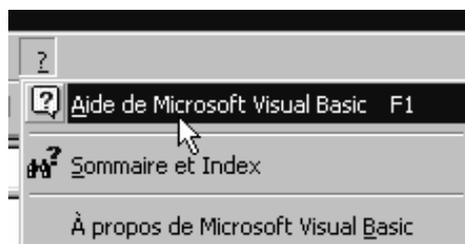


Figure 1-17 Pour obtenir l'aide VBA générale de Microsoft

Vous obtiendrez une documentation concernant les bases, les fonctions et instructions du Visual Basic qui sont appliquées dans le VBA pour AutoCAD. Cette partie vous sera utile au début, si vous ne connaissez pas Visual Basic.

Vous trouverez également dans cette partie l'aide pour les autres applications VBA que vous pouvez avoir. Vous en aurez besoin quand vous devrez écrire du code VBA qui intègre plus d'une application.

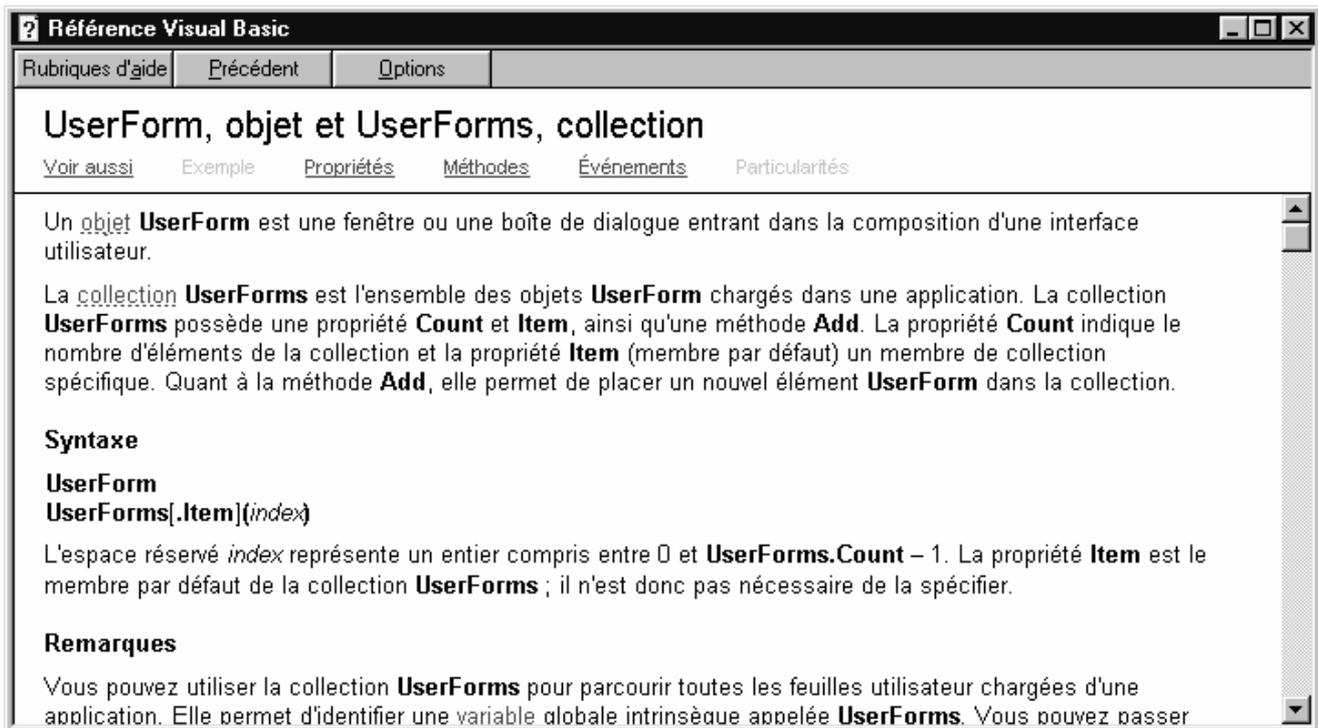


Figure 1-18 L'aide Visual Basic fournie ici est adaptée à VBA.

1.4.2 – L'aide spécifique VBA pour AutoCAD

L'*Explorateur d'objets* vous fournit une liste d'objets ainsi que leurs propriétés et méthodes. Ouvrez cet explorateur en cliquant *Explorateur d'objets* sur la barre d'outils ou en choisissant *Affichage* ⇒ *Explorateur d'objets*, (*View* ⇒ *Project Explorer*). Les objets sont organisés en bibliothèques. Tous les objets avec lesquels vous travaillez dans AutoCAD sont listés dans la bibliothèque *AutoCAD*.

Pour voir la liste d'objets AutoCAD, cliquez le liste déroulante <*Toutes bibliothèques*> et choisissez *AutoCAD*. Vous pouvez voir le résultat dans la figure 1-19.

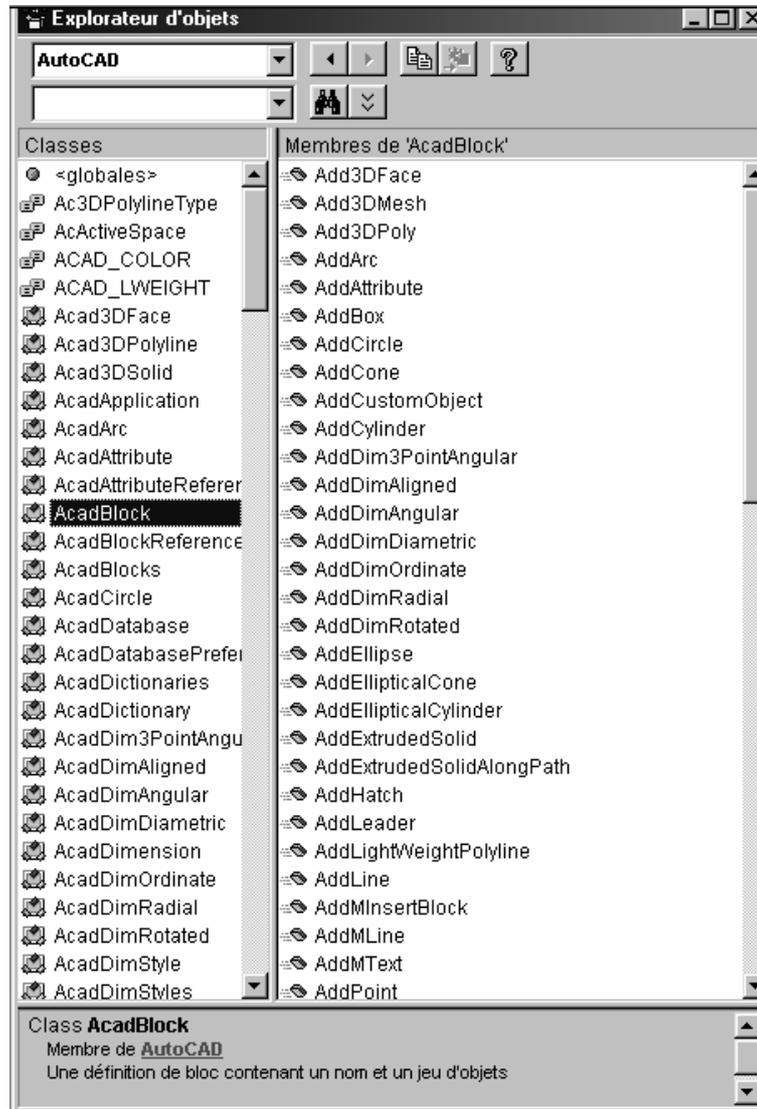


Figure 1-19 L'Explorateur d'objets fournit la liste des objets ainsi que leurs propriétés et méthodes.

Pour obtenir de l'aide, en anglais, sur un objet, sélectionnez-le dans l'Explorateur d'objets et cliquez sur le bouton  Aide de l'explorateur. Procédez de la même manière pour une méthode ou une propriété, comme montré dans la figure 1-20.

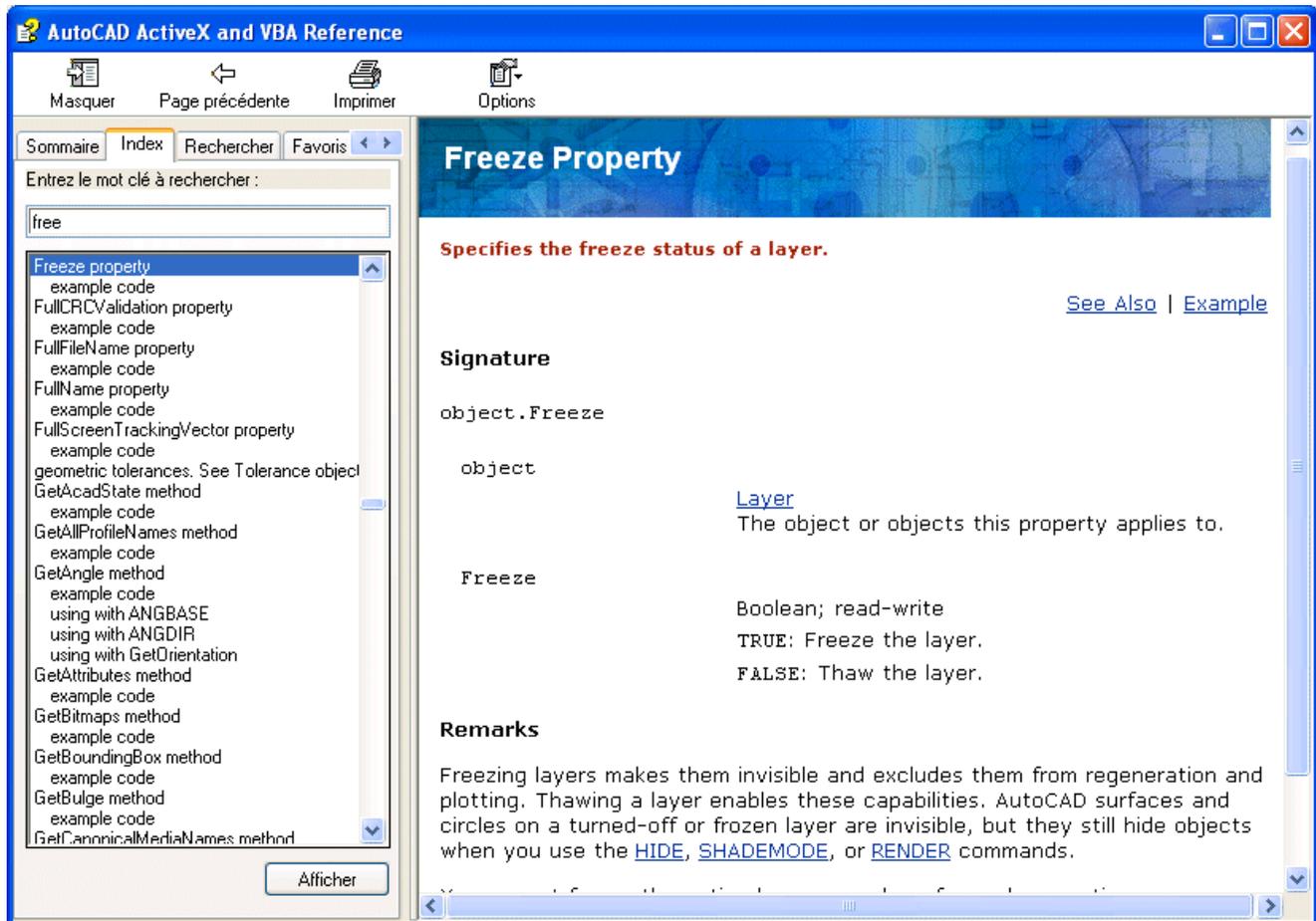


Figure 1-20 La page d'aide pour la propriété Freeze.

Vous pouvez aussi ouvrir cette fenêtre *AutoCAD ActiveX and VBA Reference* en cliquant avec le bouton de droite dans les volets *Classes ou Membres* et en choisissant *Aide*. VBA ouvre alors la page d'aide pour l'objet, la méthode ou la propriété sélectionnée. De là, un index vous permettra d'obtenir un complément d'informations.

Quand vous ouvrez une page d'aide, la plupart du temps vous pouvez cliquer sur *Example* pour afficher un exemple. Ces exemples sont une bonne manière d'apprendre le VBA. Vous pouvez copier un petit morceau de code VBA, le coller dans votre propre routine et ensuite le modifier à souhaits.

Pour une aide plus générale, AutoCAD offre deux méthodes :

- * *ActiveX and VBA Reference*, une liste alphabétique, en anglais, d'objets, méthodes, propriétés et événements.

- * Le *Guide de programmation ActiveX et VBA* qui explique, en français selon les versions, les concepts et les techniques d'ActiveX Automation.

Pour accéder à ces guides de référence, cliquez avec le bouton de droite dans l'un ou l'autre volet et choisissez *Aide*. VBA ouvre l'aide sur l'objet, la propriété ou la méthode sélectionnée. Ensuite choisissez *Sommaire de l'aide* pour accéder aux nombreux sujets d'aide.

Une fois que vous avez commencé à programmer, vous pouvez obtenir de l'aide sur toute expression, que ce soit en Visual Basic ou en VBA spécifique à AutoCAD, en y plaçant le curseur et en appuyant sur F1. Par exemple, si vous tapez **AddLine** et appuyez sur F1, vous obtiendrez l'aide AutoCAD sur la manière de créer une ligne en VBA et si vous tapez **Dim** puis F1, c'est l'aide Microsoft concernant cette instruction qui sera affichée.

1.5 – Résumé

Dans ce chapitre, nous avons comparé VBA à Visual Basic et à AutoLISP. Nous avons vu comment installer VBA, si ce n'était déjà fait.

Plus loin, nous avons eu un premier aperçu de l'Environnement de Développement Intégré de VBA et nous avons inséré un premier module et une nouvelle feuille. Nous avons vu comment obtenir de l'aide générale concernant le Visual Basic ou de l'aide plus spécifique à VBA pour AutoCAD.

Dans le chapitre suivant, nous passerons à l'action en écrivant notre premier programme.