

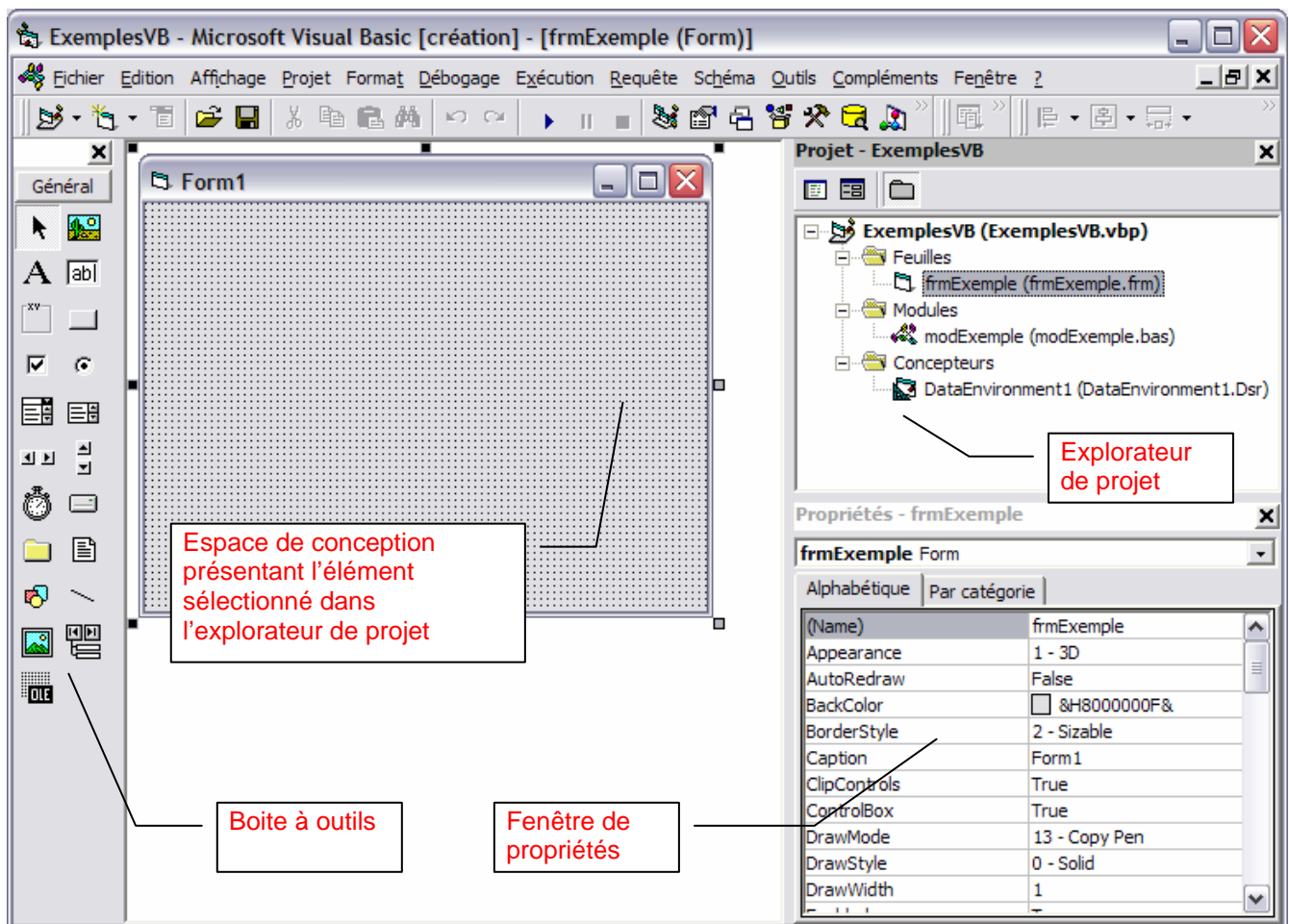
2 L'INTERFACE DE VISUAL BASIC 6

2.1 Environnement de programmation

L'éditeur VB se compose d'une fenêtre principale qui peut-être **SDI** ou **MDI**. Dans le premier cas, la fenêtre n'est pas conteneur des autres, contrairement au second cas (Outil/Options/Etendues/Environnement de développement).

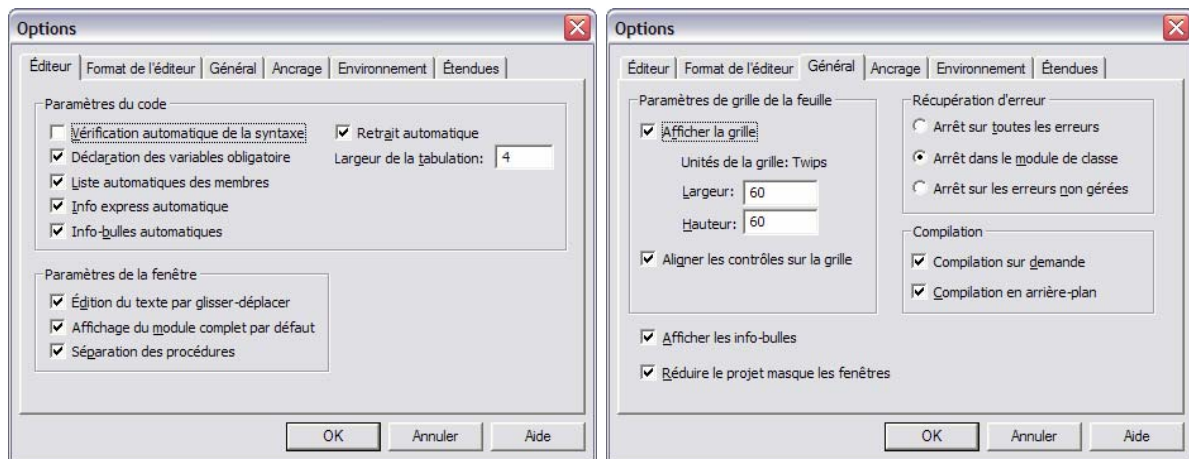
Nous trouvons ensuite de nombreuses fenêtres nous informant sur le projet en cours, les objets qu'il contient, l'exécution...

Il est important d'avoir VB durant ce chapitre afin de visualiser les écrans décrits. Voici les fenêtres principales de l'éditeur.



2.1.1 Positionner les options par défaut

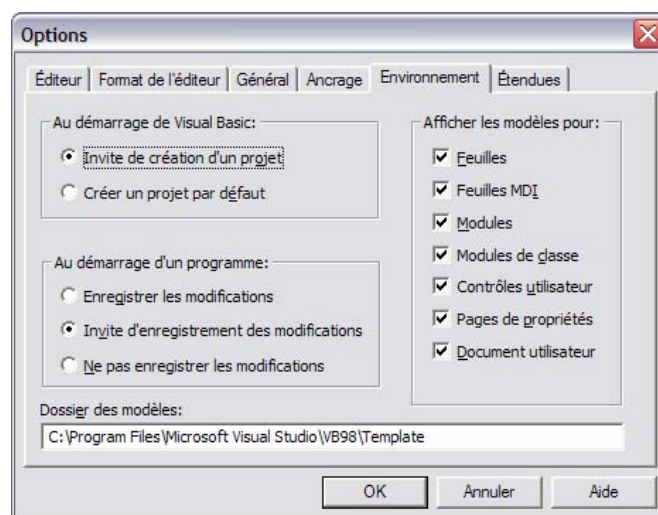
Avant de commencer à utiliser VB, il convient de positionner quelques paramètres importants. Le menu Outils / Options affiche la fenêtre suivante :



Il faut d'abord décocher la « Vérification automatique de la syntaxe. » Cette option affiche une boîte de message à chaque erreur de syntaxe lors de la frappe du code. Ces messages sont vite gênants (surtout au début, lorsqu'on a beaucoup d'erreurs.) Cette case décochée, les erreurs apparaissent en rouge sans les messages.

Il faut cocher la « Déclaration des variables obligatoire. » VB n'impose pas la déclaration des variables. Ainsi, dès que le compilateur trouve un mot inconnu, il considère que c'est une variable. Le type par défaut est Variant. Nous verrons plus loin ses caractéristiques. Il est impératif d'imposer la déclaration des variables, le cas contraire est source d'erreurs difficiles à trouver car ce sont des erreurs logiques et donc non détectées par le compilateur. Cette option est active au démarrage suivant.

On peut sélectionner une unité de grille plus petite pour pouvoir positionner les contrôles plus finement sur les formulaires.



Il faut enfin sélectionner « Invite d'enregistrement des modifications. » C'est essentiel pour ne pas oublier d'enregistrer les modifications des éléments modifiés. Nous avons qu'un projet était constitué de plusieurs fichiers. Cette option affiche une liste des éléments modifiés lors du lancement de l'application. Il est possible de sélectionner les éléments à enregistrer ou non.

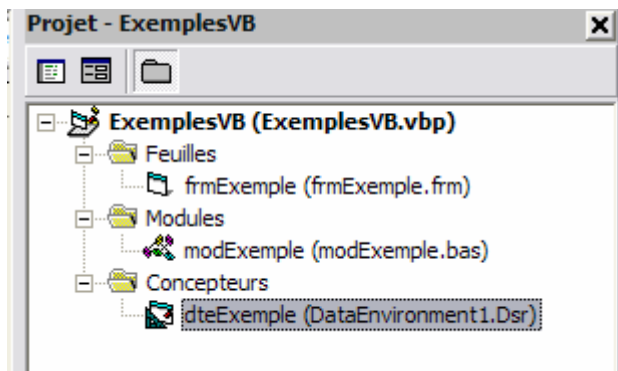
L'onglet format de l'éditeur permet de personnaliser l'éditeur. N'abusez pas de ces fonctionnalités au risque de rendre l'IHM pénible à utiliser.

2.1.2 Espace de conception

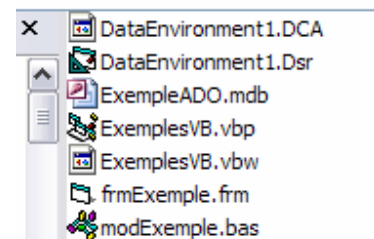
Il affiche l'élément sélectionné dans l'explorateur de projet. On peut y manipuler notamment les formulaires, le code des formulaires et des modules, les classes personnalisées, les concepteurs...

2.1.3 L'explorateur de projet

Une fenêtre importante est l'**explorateur de projet** (ctrl + R). Cette fenêtre contient la liste de toutes les feuilles, les modules, les modules de classe, les références etc. inclus dans le projet.



Explorateur de projet



Explorateur Windows

Un clic droit sur cette fenêtre permet d'ajouter des objets (formulaires, modules etc.) nouveaux ou existants.

Le fichier « .vbp » est le point d'entrée d'un projet. Il contient les informations de contexte. Un clic droit sur le projet permet de positionner les propriétés du projet.

Le libellé à gauche correspond à la propriété *Name* de chaque élément alors que le libellé entre parenthèses correspond au nom du fichier sur le disque. A chaque élément correspond un ou deux fichiers dans l'explorateur Windows. Il est important de vérifier l'enregistrement de chaque élément avant de quitter.

2.1.4 La fenêtre de propriétés

La **fenêtre de propriétés** (F4) renseigne sur l'état de l'objet sélectionné (on parle aussi de focus.) Elle permet de définir les valeurs des propriétés durant la conception de l'application. Ces propriétés sont aussi accessibles par programme durant l'exécution si la classe le prévoit.

Cette fenêtre est affichée lorsque le formulaire *frmExemple* est sélectionné dans l'explorateur de projet. On trouve toutes les propriétés disponibles à la conception triées par ordre alphabétique ou par catégorie.

2.1.5 La boîte à outils

La **boîte à outils** contient les **contrôles** disponibles. On peut ajouter autant de contrôles que l'on souhaite dans cette boîte (à condition qu'ils soient installés sur le poste.)

Les contrôles sont des objets compilés autonomes qui possèdent généralement une interface utilisateur (IHM) à l'exécution et qui peuvent être utilisés dans un projet. Un bouton est un contrôle, une zone de texte... Certains d'entre eux peuvent être très sophistiqués comme le contrôle Internet qui est un navigateur tout fait. Certains contrôles n'ont pas d'interface à l'exécution, mais ils mettent à disposition un ensemble de méthodes dédiées à un domaine. Le contrôle Winsock permet de gérer une connexion réseau. Il se pose sur une feuille en création, mais n'a pas d'interface à l'exécution.

Certains objets sont accessibles par le biais d'un contrôle ou par programme (dans le code.) Le fait de dessiner le contrôle sur un formulaire à un effet déclaratif. Une instance de la classe à laquelle il fait référence est automatiquement créée lors de l'exécution. Si on fait appel à la classe sans utiliser le contrôle (en général une DLL), il faut explicitement créer une instance de classe dans un module à l'aide du code approprié.

Grâce à ce système (possible par le concept objet), VB peut faire énormément de choses et l'on trouve sur Internet ou sur le marché une multitude de contrôles dédiés à toutes sortes de fonctions. Les contrôles sont souvent développés en C++ pour être plus performants. Cependant, ils peuvent être réalisés dans n'importe quel langage dès lors qu'il respecte les préconisations ActiveX une fois compilés.

2.1.6 Fenêtres de mise au point

La fenêtre **espion express** (maj+F9) permet de connaître l'état des propriétés d'un ou plusieurs objets, de variables, au cours d'une d'exécution pas à pas.

La fenêtre **Exécution** (ctrl+G) permet d'interroger les propriétés ou les variables, ou encore de modifier ces mêmes propriétés ou variables durant l'exécution pas-à-pas.

Ces deux fenêtres sont particulièrement utiles pour le débogage. L'exécution pas-à-pas facilite énormément la mise au point. On doit cette fonctionnalité aux versions précédentes de VB qui était un langage interprété et non compilé.

2.1.7 Fenêtre de présentation des feuilles

La fenêtre de **présentation des feuilles** permet de définir graphiquement la position par défaut des fenêtres lors de leur affichage à l'exécution. Il est aussi possible de définir cette position dans la fenêtre de propriétés ou par programme, lors du chargement de la feuille.

Cette fenêtre affiche des repères de définition (clic droit.) Cela est très pratique pour vérifier que les écrans conçus respectent les préconisations du cahier des charges. En effet, si on développe des écrans en 1280 x 1024, il y a peu de chance qu'ils soient correctement affichés sur des écrans en 800 x 600.

2.1.8 L'explorateur d'objet

Enfin, **l'explorateur d'objets** (F2) permet d'afficher l'ensemble des propriétés et méthodes des objets, des procédures et des fonctions internes ou personnalisées. On y trouve aussi les constantes intégrées et personnalisées.

Cet écran est très utile pour explorer les bibliothèques référencées dans le projet actif.

2.1.9 L'aide en ligne

Un élément essentiel de VB est l'aide en ligne ; elle est indispensable pour travailler. VB a l'avantage d'offrir de nombreuses procédures et fonctions toutes faites. La contrepartie de ce confort est la multiplicité des mots réservés et des syntaxes. L'aide en ligne contient de nombreux exemples que l'on peut copier et coller dans son propre code pour les modifier.

L'utilisation d'une nouvelle bibliothèque, d'un nouveau contrôle nécessite un apprentissage pour le manipuler et l'exploiter au mieux.

2.2 Les éléments d'un programme

Une application VB se constitue d'un **fichier de projet** (.vbp). Il s'agit d'un fichier texte contenant les propriétés du projet et la liste des modules, formulaires, contrôles etc. de l'application au stade de la conception.

Pour fonctionner, un projet nécessite au moins un formulaire ou un module.

2.2.1 Les formulaires

Les **formulaires** sont des éléments d'interface qui contiennent des contrôles. C'est ce qu'on appelle une fenêtre au stade de l'exécution. Ils génèrent un fichier **.FRM** au format texte qui contient le code et d'autres informations. Peut y être associé un fichier **.FRX** qui contient des informations binaires telles que des images. Le fichier *.frx* n'apparaît pas dans l'explorateur de projet.

Dans les propriétés du projet, on doit définir la feuille de démarrage, notamment dans le cas où plusieurs feuilles existent dans un même projet.

Il existe **deux types de formulaires** : les SDI et les MDI. Une fenêtre MDI contient des fenêtres SDI. On ne peut mettre sur une feuille MDI que des contrôles qui possèdent une propriété *Alignment*. Il ne peut y avoir qu'une seule fenêtre MDI par projet.

Les formulaires possèdent 3 types de code :

- Déclarations (variables, constantes etc.) ;
- Procédures événementielles ;
- Procédures générales.

Le code des formulaires est stocké dans un module de formulaire accessible par le bouton « code » dans l'explorateur de projet. En fait, pour un objet formulaire listé dans l'explorateur de projet, on peut accéder à deux composantes du formulaire : l'interface et le code.

Les modules de formulaire sont différents des modules que nous aborderons plus tard.

2.2.2 Les contrôles ActiveX complémentaires

On appelle contrôles les objets disponibles dans la boîte à outil que l'on peut poser sur un formulaire. Il s'agit de classes que l'on trouve essentiellement dans des fichiers **.ocx**, parfois dans des **.dll**. Pour être accessibles de n'importe où, les contrôles doivent être inscrits dans la base de registre de Windows. En général, les procédures d'installation s'en chargent.

Le fait de « dessiner » un contrôle sur un formulaire a un effet déclaratif. Cela signifie qu'une instance du contrôle est créée et qu'elle est accessible par la propriété *Name* du contrôle. Dans l'ensemble, les contrôles offrent une interface visible à l'exécution, mais ce n'est pas toujours le cas (CommonDialog, Winsock...) Dans ce cas, ils sont destinés à être piloté par programme exclusivement et n'interagissent pas directement avec l'utilisateur final.

On trouve d'abord les contrôles par défaut de VB dans la boîte à outils. Ces contrôles peuvent être ajoutés au projet puis utilisés dans des formulaires. On en trouve de nombreux, gratuits ou payant, dédiés à toutes sortes de tâches. Il est également possible de créer ses propres contrôles avec VB.

2.2.3 Les modules

Il s'agit de fichiers texte ayant l'extension **.BAS**. Ils contiennent des déclarations (variables, constantes, structures, API...) et des procédures générales.

Du fait qu'il n'y a pas d'IHM (interface homme / machine), ils ne peuvent pas contenir de procédures événementielles.

Ils sont listés dans la rubrique « modules » de l'explorateur de projet. **Ils sont différents des modules de formulaires.**

2.2.4 Les modules de classe

Ce sont également des fichiers texte ayant l'extension **.CLS**. Ils sont listés dans l'explorateur de projet dans la rubrique « module de classe. » Ils permettent de créer des objets personnalisés dans les projets *EXE standard* ou dans les projets *DLL*, des contrôles *ActiveX* dans les projets *OCX* (cf. IV création d'un projet).

On y trouve 3 types de code :

- Déclarations ;
- Procédures générales ;
- Procédures de l'objet (Property, Initialize, Terminate...).

2.2.5 Les références

Il est possible de mettre en référence d'autres composants (souvent des dll) afin de pouvoir exploiter les procédures qu'ils contiennent. Toutefois, ces fichiers ne sont pas inclus dans l'exécutable lors de la compilation. Il faut donc veiller à les distribuer avec l'exécutable ou s'assurer qu'ils sont installés sur le poste qui recevra l'application.

Par exemple, on peut mettre en référence Excel, Outlook, MAPI (classes liées à la messagerie) etc. On peut ensuite utiliser tous les objets disponibles dans les bibliothèques référencées. Dans le cas d'Excel, on peut désormais entièrement le manipuler à partir de VB. Bien entendu, il faut qu'Excel soit installé sur le poste.

2.2.6 Les modèles

Ils se trouvent dans les dossiers \Templates\Forms ou \Templates\Projects. Ils permettent de réaliser des écrans types de base. Cependant, ils sont très loin de permettre la réalisation d'applications professionnelles. On y trouve quelques exemples lorsque l'on débute en VB.

2.2.7 Les assistants

Il s'agit de modèles interactifs qui permettent de créer rapidement des bases de projets ou d'objets. Comme les modèles, il s'agit d'éléments basiques.

L'essentiel

Prise en main de l'éditeur VB. Identifier les principales fenêtres : explorateur de projet, fenêtre de propriétés, boîte à outils...

Fonctionnement et organisation de l'aide en ligne.

Les principaux éléments d'un projet :

- Les formulaires ;
- Les modules ;
- Les contrôles et références.