

CPR Informatique

(poste 3159 ou 3164)



ÉCOLE NATIONALE
DES SCIENCES
GÉOGRAPHIQUES

Septembre 2001

Visual Basic Application pour Excel



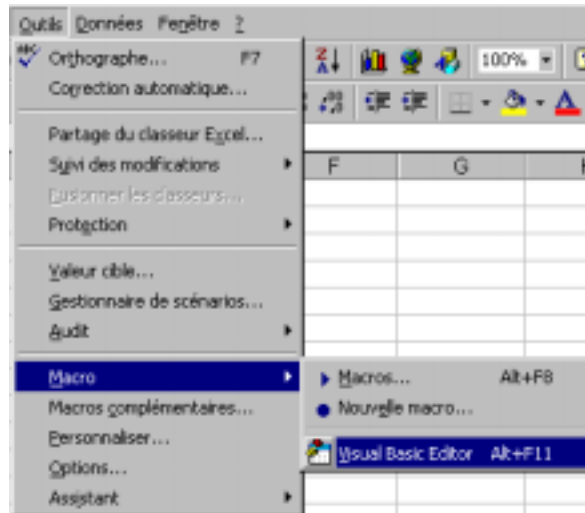
Table des matières

1.- PRISE EN MAIN DE L'ENVIRONNEMENT	3
1.1.- DEMARRAGE DE VISUAL BASIC DANS EXCEL.....	3
1.2.- L'EXPLORATEUR DE PROJET	3
1.3.- L'EXPLORATEUR D'OBJETS	4
1.4.- LA FENETRE CODE.....	6
1.5.- LA FENETRE EXECUTION	7
1.6.- LA FENETRE ESPIONS	8
2.- CREATION DE FORMULAIRES	9
2.1.- CASE D'OPTION.....	9
2.2.- CASE À COCHER	9
2.3.- ZONE DE LISTE	10
2.4.- BOUTON DE COMMANDE	11
2.5.- IDENTIFICATEUR DES CONTROLES	12
3.- LES BOITES DE DIALOGUES.....	13
3.1.- L'OBJET USERFORM.	13
3.1.1.- <i>Création d'une UserForm</i>	13
3.1.2.- <i>Modification du nom d'une UserForm</i>	13
3.1.3.- <i>Modification des propriétés d'une UserForm</i>	13
3.2.- LA BOITE A OUTILS	14
3.2.1.- <i>Affichage ou masquage de la boîte à outils</i>	14
3.2.2.- <i>Les contrôles de la boîte à outils</i>	14
3.3.- LES CONTROLES SUPPLEMENTAIRES	18
3.4.- MODIFICATION D'UN CONTROLE.....	18
3.4.1.- <i>Modification du nom</i>	18
3.4.2.- <i>Modification d'une propriété</i>	19
3.5.- LA SUPERPOSITION DES CONTROLES	19
4.- LES MENUS	20
4.1.- LE « MENU BARRE ».....	20
4.2.- LE « MENU CONTEXTUEL »	20
5.- REPONSES AUX EVENEMENTS.....	22
6.- L'AIDE EN LIGNE SUR LES PROCEDURES EVENEMENTIELLES	23
7.- ACTIONS POSSIBLES SUR UN DOCUMENT EXCEL.....	24
7.1.- OUVERTURE ET FERMETURE D'UN DOCUMENT EXCEL	24
7.2.- ACTIONS SUR L'OBJET « FEUILLE EXCEL »	24
7.3.- ACTIONS SUR L'OBJET « CELLULE EXCEL »	25
7.4.- AUTRES ACTIONS SUR DES OBJETS EXCEL.....	26
7.5.- QUELQUES EXEMPLES D' ACTIONS PLUS COMPLEXES.....	27
ANNEXES	29
A1.- CONSTANTES, BOUTONS ET ICONES DES BOITES DE MESSAGES	29
A2.- CONSTANTES DE CODE DE TOUCHES	31

1.- Prise en main de l'environnement

1.1.- Démarrage de Visual basic dans Excel

Dans le menu « Outils » sélectionnez la commande « Macro – Visual Basic Editor » pour activer l'explorateur de projet.

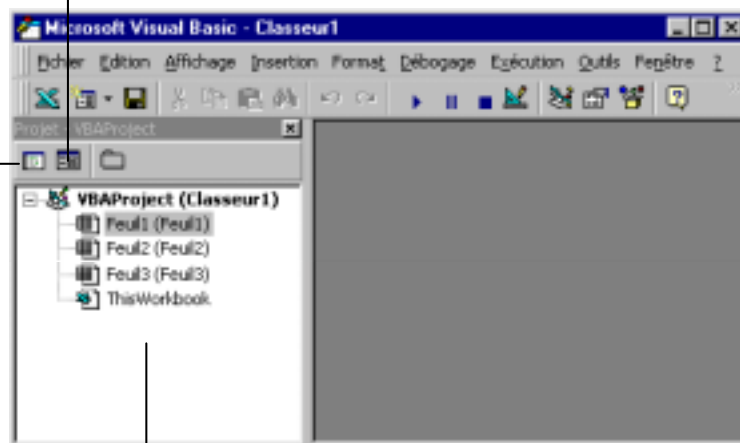


1.2.- L'Explorateur de projet

L'explorateur de projet affiche une liste hiérarchisée des projets et de tous les éléments référencés par chaque projet et contenus dans un projet.

Ouvre une fenêtre destinée à recevoir le code Visual Basic

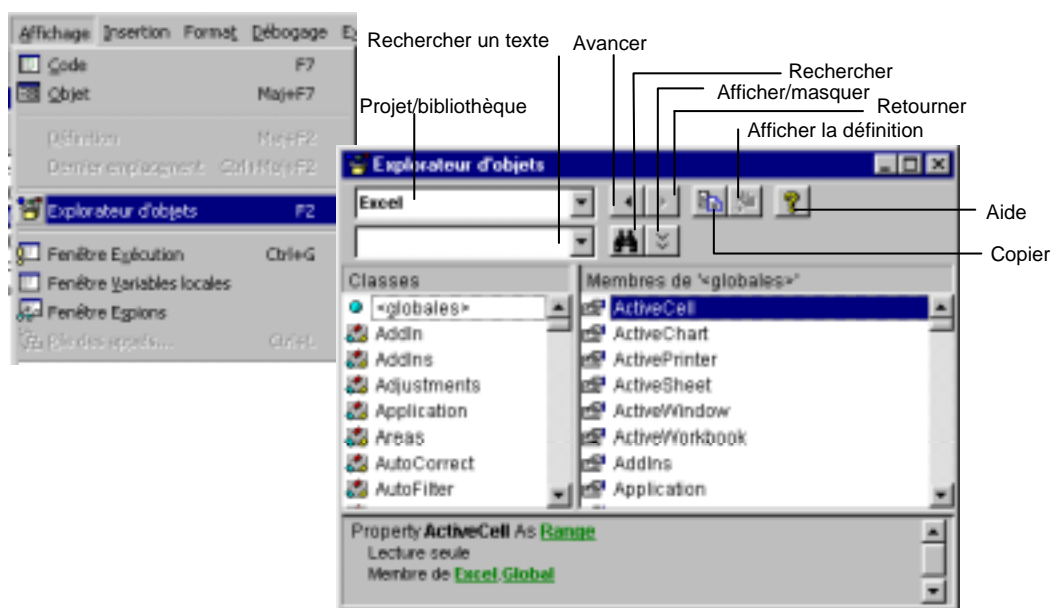
Affiche l'objet sélectionné



Le projet et les éléments qu'il contient

1.3.- L'Explorateur d'objets

Affiche les classes, les propriétés, les méthodes, les événements et les constantes disponibles dans les bibliothèques d'objets et les procédures de votre projet. Il permet de rechercher et d'utiliser des objets que vous créez ainsi que des objets provenant d'autres applications.



Zone Projet/bibliothèque

Affiche les bibliothèques actuellement référencées pour le projet actif. Vous pouvez ajouter des bibliothèques dans la boîte de dialogue Références. « All Libraries » permet d'afficher simultanément toutes les bibliothèques.

Zone Rechercher un texte

Contient la chaîne à utiliser dans votre recherche. Vous pouvez taper ou choisir la chaîne à utiliser. La zone Rechercher un texte conserve jusqu'à la fermeture du projet les 4 dernières chaînes de recherche que vous avez entrées. Vous pouvez utiliser les caractères génériques Visual Basic standard lorsque vous tapez une chaîne.

Pour rechercher un mot entier, utilisez la commande Mot entier du menu contextuel.

Bouton Retourner

Permet de revenir à la sélection précédente dans les listes « Classes et Membres de ». Vous remontez d'une sélection chaque fois que vous cliquez sur ce bouton, jusqu'à épuisement des choix précédents.

Bouton Avancer

Permet de réutiliser vos sélections d'origine dans les listes « Classes et Membres de » chaque fois que vous cliquez dessus, jusqu'à épuisement de la liste de sélections.

Bouton Copier dans le Presse-papiers

Copie la sélection en cours dans la liste « Membres de » ou le texte du volet Détails vers le Presse-papiers. Vous pouvez ensuite coller la sélection dans votre code.

Bouton Afficher la définition

Déplace le curseur à la position à laquelle est définie la sélection dans la liste « Membres de » ou « Classes » de la fenêtre Code.

Bouton Aide

Affiche la rubrique d'aide en ligne correspondant à l'élément sélectionné dans la liste « Classes » ou « Membres de ». Vous pouvez également appuyer sur F1.

Bouton Rechercher

Lance une recherche dans les bibliothèques sur la classe ou propriété, méthode, constante ou l'événement correspondant à la chaîne que vous avez tapée dans la zone « Rechercher un texte » et ouvre le volet « Résultats de la recherche », dans lequel figurent les informations demandées.

Bouton Afficher/Masquer les résultats de la recherche

Ouvre ou masque le volet « Résultats de la recherche ». Les informations de ce dernier sont modifiées pour afficher les résultats de la recherche dans la bibliothèque ou dans le projet sélectionné dans la liste « Projet/bibliothèque ». Les résultats de la recherche sont répertoriés par ordre alphabétique de A à Z.

Liste Résultats de la recherche

Affiche la bibliothèque, la classe et le membre correspondant aux éléments indiqués dans votre chaîne de recherche. Les informations du volet « Résultats de la recherche » sont modifiées lorsque vous changez la sélection dans la zone « Projet/bibliothèque ».

Liste « Classes »

Affiche toutes les classes disponibles dans la bibliothèque ou dans le projet sélectionné dans la zone « Projet/Bibliothèque ». S'il existe du code rédigé pour une classe, cette dernière apparaît en gras. La liste commence toujours par <global>, la liste des membres auxquels l'accès est global. Si vous sélectionnez une classe sans préciser de membre, le membre par défaut, s'il existe, est utilisé. Le membre par défaut est indiqué par un astérisque (*) ou par l'icône par défaut spécifique à ce membre.

Liste « Membres de »

Affiche les éléments de la classe sélectionnée dans le volet « Classes » en les triant par groupe, puis dans l'ordre alphabétique au sein de chaque groupe. Les méthodes, propriétés, événements ou constantes pour lesquels du code est rédigé apparaissent en gras. Vous pouvez modifier l'ordre de cette liste avec la commande « Membres du groupe » du menu contextuel de l'Explorateur d'objets.

Volet « Détails »

Affiche la définition du membre. Le volet « Détails » contient un renvoi à la classe ou bibliothèque à laquelle l'élément appartient. Certains membres sont dotés de renvois à leur classe parent. Par exemple, si le texte du volet « Détails » indique que Command1 est déclarée en tant que type de bouton de commande, la classe Bouton de commande apparaît si vous cliquez sur le bouton de commande.

Vous pouvez copier ou faire glisser le texte du volet « Détails » vers la fenêtre Code.

Barre de fractionnement

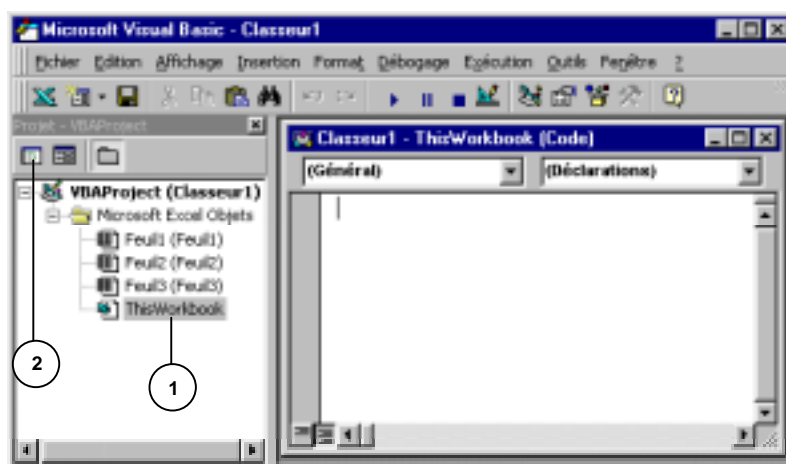
Sépare les volets pour vous permettre de modifier leur taille. Des séparations existent entre :

- la zone « Classes » et la zone « Membres de » ;
- la liste « Résultats de la recherche » et les zones « Classes » et « Membres de » ;
- les zones « Classes » et « Membres de » et le volet « Détails ».

1.4.- La Fenêtre Code

Pour afficher la fenêtre de code :

1. sélectionnez l'objet qui doit être associé au code Visual Basic ;
2. cliquez sur le bouton d'affichage de la fenêtre « code ».



La fenêtre Code permet d'écrire, d'afficher et de modifier le code Visual Basic. Vous pouvez ouvrir autant de fenêtres Code que vous avez de modules, ce qui vous permet de visualiser facilement le code des différentes feuilles ou modules et d'effectuer des opérations de « copier-coller » de l'un à l'autre.

Vous pouvez ouvrir une fenêtre Code à partir de :

- la fenêtre Projet, en sélectionnant une feuille ou un module, et en choisissant le bouton Afficher le code ;
- une fenêtre UserForm, en double-cliquant sur un contrôle ou sur une feuille, en choisissant « Code » dans le menu Affichage ou en appuyant sur F7.

Vous pouvez faire glisser le texte sélectionné vers :

- un autre emplacement de la fenêtre Code en cours ;
- une autre fenêtre Code ;
- les fenêtres Exécution et Espions ;
- la Corbeille.

Éléments de la fenêtre code

Zone Objet

Affiche le nom de l'objet sélectionné. Cliquez sur la flèche située à droite de la zone de liste pour afficher la liste de tous les objets associés à la feuille.

Zone Procédures/Événements

Répertorie tous les événements reconnus par Visual Basic pour la feuille ou le contrôle affiché dans la zone Objet. Lorsque vous sélectionnez un événement, la procédure d'événement qui est associée à son nom apparaît dans la fenêtre Code.

Si « (Général) » est affiché dans la zone Objet, la zone Procédure répertorie l'ensemble des déclarations et des procédures générales qui ont été créées pour la feuille. Si vous modifiez le code du module, la zone Procédure affiche toutes les procédures générales du module. Dans les deux cas, la procédure que vous avez sélectionnée dans la zone Procédure apparaît dans la fenêtre Code.

Toutes les procédures d'un module sont affichées par ordre alphabétique dans une même liste avec défilement. Si vous sélectionnez une procédure à l'aide des zones de liste déroulantes situées en haut de la fenêtre Code, le curseur va immédiatement se placer sur la première ligne de code de la procédure sélectionnée.

Barre de fractionnement

Faire glisser cette barre vers le bas a pour effet de fractionner la fenêtre Code en deux volets horizontaux dont vous pourrez faire défiler le contenu de manière totalement autonome. Vous pouvez ensuite afficher simultanément différentes parties de votre code. Les informations qui apparaissent dans les zones Objet et Procédures/Événements s'appliquent au code du volet qui a le focus. Si vous faites glisser la barre de fractionnement jusqu'en haut de la fenêtre, un volet se ferme.

Barre des indicateurs en marge

Zone grise située à gauche de la fenêtre Code où apparaissent les indicateurs en marge.

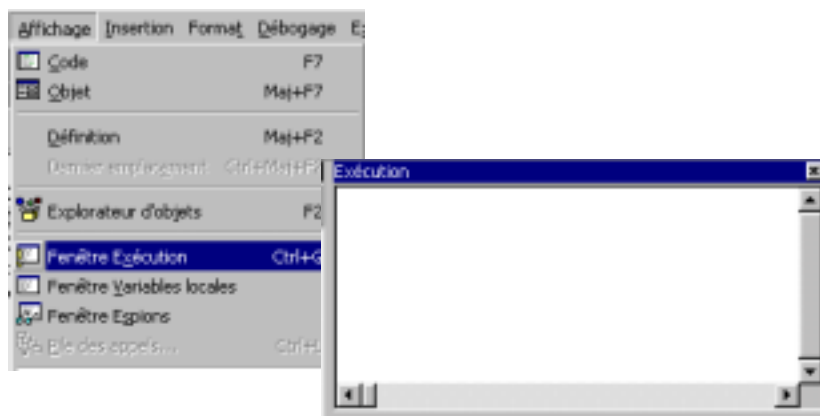
Icône Affichage de la procédure

Affiche la procédure sélectionnée. La fenêtre Code ne peut afficher qu'une procédure à la fois.

Icône Affichage complet du module

Affiche l'ensemble du code du module.

1.5.- La Fenêtre Exécution



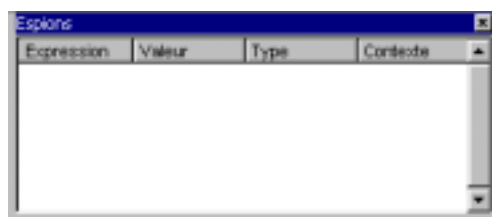
Dans cette fenêtre, vous pouvez :

- taper ou coller une ligne, puis appuyer sur ENTRÉE pour l'exécuter ;
- copier le code de la fenêtre « Exécution » pour le coller dans la fenêtre « Code », mais vous ne pouvez pas enregistrer le code de la fenêtre Exécution ;
- faire glisser et déplacer la fenêtre « Exécution » n'importe où sur votre écran, sauf si vous en faites une fenêtre « ancrable » à l'aide de l'onglet **Ancrage** de la boîte de dialogue Options.

Pour fermer la fenêtre, cliquez sur la case de fermeture. Si elle n'est pas visible, double-cliquez sur la barre de titre pour que la case de fermeture apparaisse, puis cliquez sur cette dernière.

En mode Arrêt, une instruction figurant dans la fenêtre Exécution est exécutée dans le contexte ou la portée indiqué dans la zone Procédure. Par exemple, si vous tapez « Print variablename », la sortie est la valeur d'une variable locale, comme si la méthode Print s'était trouvée dans la procédure que vous exécutiez quand le programme s'est arrêté.

1.6.- La Fenêtre Espions



Cette boîte apparaît automatiquement lorsque des expressions espionnes sont définies dans le projet. Vous pouvez :

- modifier la taille des en-têtes de colonnes en faisant glisser la bordure vers la droite pour les agrandir ou vers la gauche pour les réduire ;
- faire glisser une variable sélectionnée vers la fenêtre Exécution ou la fenêtre Espion ;
- fermer la fenêtre en cliquant sur la case de fermeture. Si elle n'est pas visible, double-cliquez sur la barre de titre pour que la case de fermeture apparaisse, puis cliquez sur cette dernière.

Expression : affiche l'expression espionne avec l'icône Espion, à sa gauche.

Valeur : indique la valeur de l'expression au moment du passage au mode Arrêt. Vous pouvez modifier la valeur puis appuyer sur ENTRÉE, sur les touches fléchées HAUT et BAS, TAB, MAJ+TAB, ou cliquer sur l'écran pour valider la modification. Si la valeur est incorrecte, le champ Modifier reste actif et la valeur est mise en évidence. Un message décrivant l'erreur apparaît également. Pour annuler une modification, appuyez sur ÉCHAP.

Type : indique le type de l'expression.

Contexte : affiche le contexte de l'expression espionne. Si le contexte de l'expression n'est pas dans la portée lors du passage au mode Arrêt, la valeur en cours n'est pas indiquée. Pour fermer la fenêtre, cliquez sur la case de fermeture. Si elle n'est pas visible, double-cliquez sur la barre de titre pour que la case de fermeture apparaisse, puis cliquez sur cette dernière.

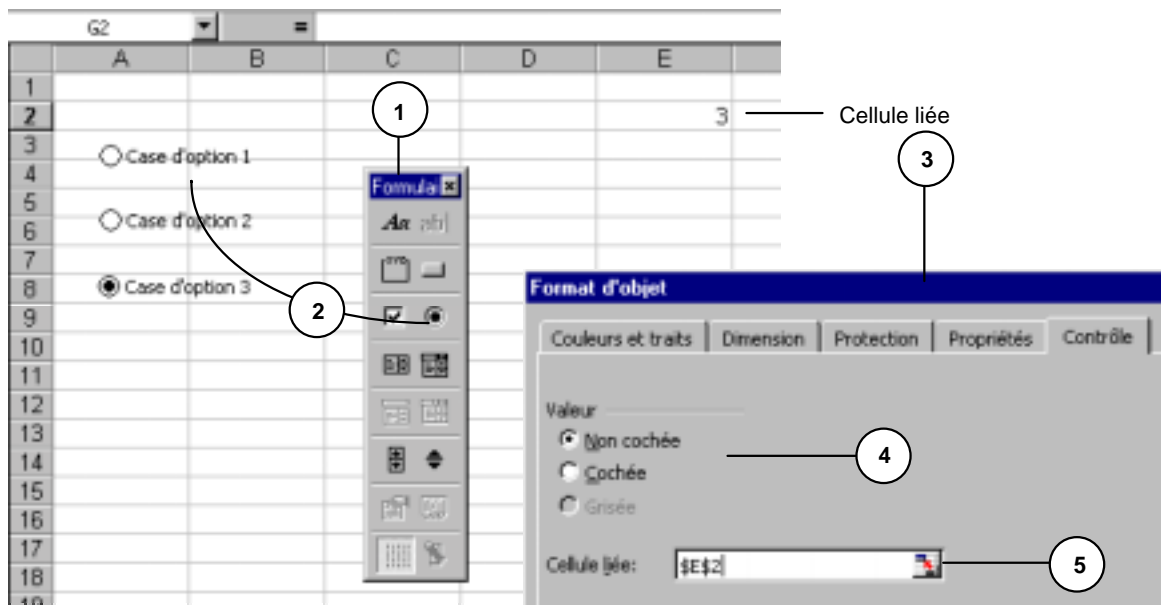
2.- Création de formulaires

Sur une feuille de calcul vous pouvez placer des case à cocher, des boutons d'options, des listes déroulantes ou des boutons de commande (qui serviront à déclencher des macros).

Ces contrôles placés sur une feuille de calcul peuvent ne pas être visible au moment de l'impression.

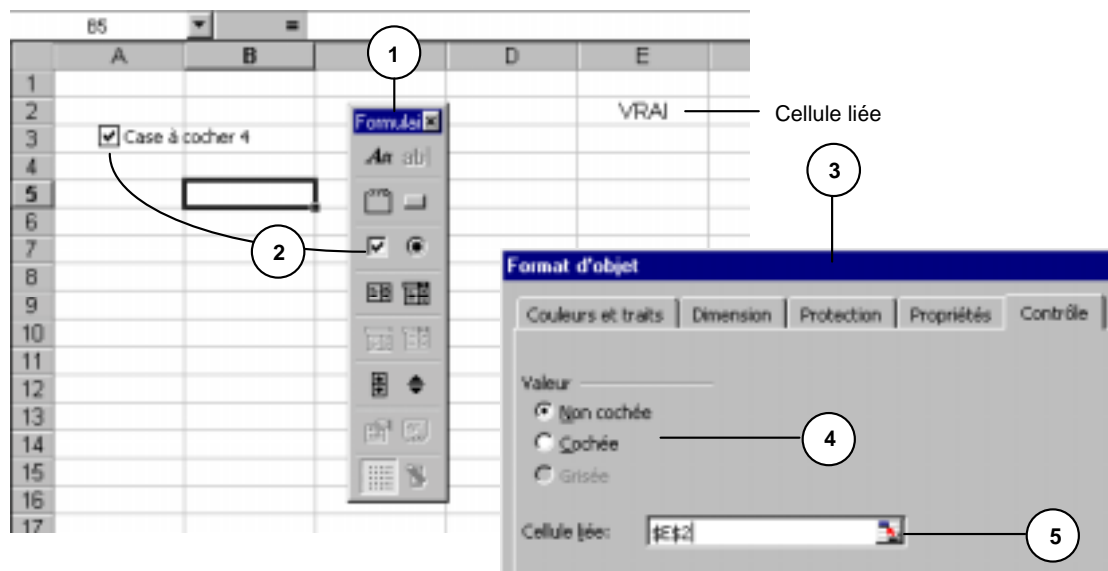
2.1.- Case d'option

1. dans le menu « *Affichage* », activez la commande « *Barre d'outils* », puis affichez la barre « *Formulaire* » ;
2. cliquez sur le bouton « *Case d'option* », puis placez l'objet sur la feuille de calcul ;
3. lorsque l'objet est sélectionné, dans le menu « *Format* », activez la commande « *Contrôle..* » ;
4. cliquez sur l'onglet « *Contrôle* », puis indiquez la valeur souhaitée (Non coché, Coché ou Grisé) ;
5. indiquez la cellule liée. La cellule liée étant la cellule qui reçoit le n° de la case d'option activée.



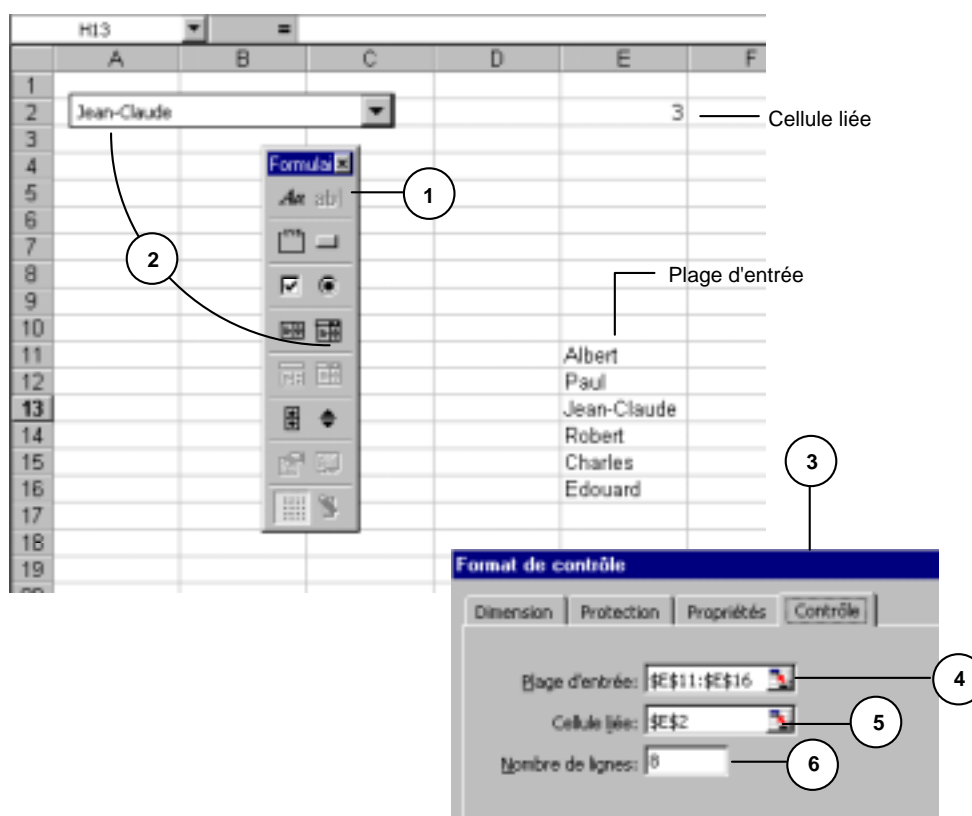
2.2- Case à cocher

1. dans le menu « *Affichage* », activez la commande « *Barre d'outils* », puis affichez la barre « *Formulaire* » ;
2. cliquez sur le bouton « *Case à cocher* », puis placez l'objet sur la feuille de calcul ;
3. lorsque l'objet est sélectionné, dans le menu « *Format* », activez la commande « *Contrôle* » ;
4. cliquez sur l'onglet « *Contrôle* », puis indiquez la valeur souhaitée (Non coché, Coché ou Grisé) ;
5. indiquez la cellule liée. La cellule liée étant la cellule qui reçoit « *VRAI* » lorsque la case est cochée, « *FAUX* » dans l'autre cas.



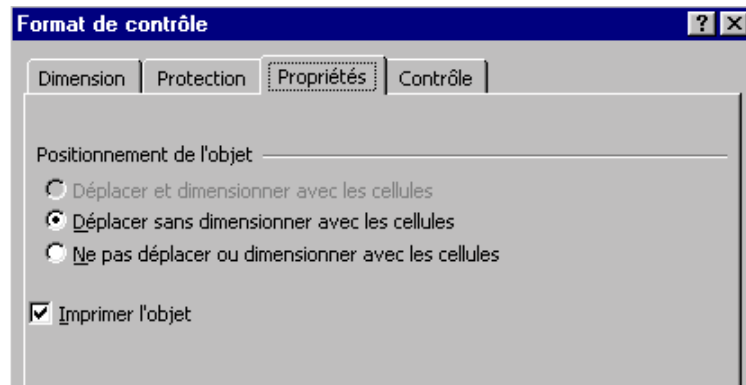
2.3.- Zone de liste

1. dans le menu « Affichage », activez la commande « Barre d'outils », puis affichez la barre « Formulaire » ;
2. cliquez sur un des boutons « Zone de liste», puis placez l'objet sur la feuille de calcul ;
3. lorsque l'objet est sélectionné, dans le menu « Format », activez la commande « Contrôle » ;
4. cliquez sur l'onglet « Contrôle », puis indiquez la « Plage d'entrée ». La plage d'entrée étant la plage des cellules contenant les informations qui doivent figurer dans la liste ;
5. indiquez la cellule liée. La cellule liée étant la cellule qui reçoit le n° du rang de l'information sélectionnée dans la liste ;
6. indiquez le « Nombre de lignes ». Le nombre de lignes correspond aux nombre de lignes affichées lorsque l'on clique sur la zone de liste déroulante.



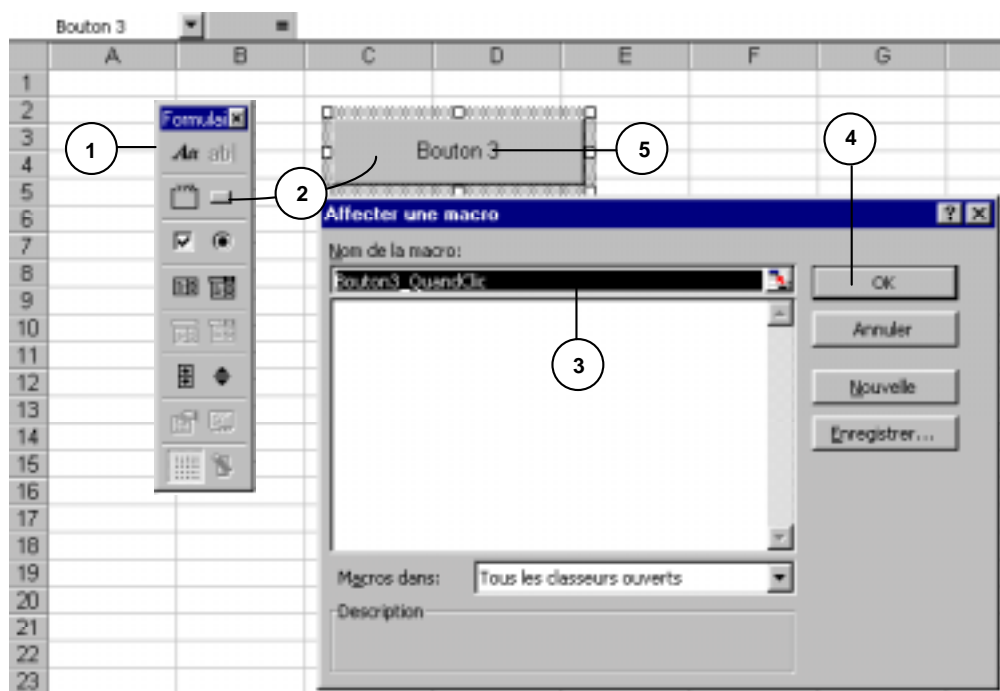
Remarque :

- les cellules liées, les plages d'entrées, peuvent être situées sur une autre feuille ;
- pour ne pas imprimer ces contrôles, décochez la case « Imprimer l'objet ».



2.4.- Bouton de commande

1. dans le menu « Affichage », activez la commande « Barre d'outils », puis affichez la barre « Formulaire » ;
2. cliquez sur le bouton « Bouton », puis placez l'objet sur la feuille de calcul ;
3. la fenêtre « Affecter une macro » s'affiche, sélectionnez la macro ;
4. cliquez sur « Ok » ;
5. sélectionner le texte du bouton, puis taper le texte qui vous convient.



2.5.- Identificateur des contrôles

Contrairement aux boîtes de dialogue que nous verrons dans le chapitre suivant, les contrôles insérés dans une feuille Excel (Case option, Case à cocher, Boutons ...) n'ont pas d'identificateur. La seule manière de les « identifier » est de connaître l'ordre de création du contrôle dans la feuille Excel. Visual Basic range les contrôles dans un tableau de même type. Par exemple, si vous créez trois boutons sur votre feuille Excel, VBA crée un tableau « Buttons » de trois éléments de type « Buttons ».

De même, si vous créez deux cases d'options, vous aurez un tableau « OptionButtons » de deux éléments. Si vous désirez connaître l'état des cases d'options (par exemple laquelle des deux est sélectionnée), vous pouvez écrire le code suivant :

```
For I = 1 To 2
    Valeur = Sheets ("nom de la feuille").OptionButtons(I).Value
    Debug.Print Valeur
Next
```

Dans le cas précédent, « Valeur » vaudra « 1 » pour la case cochée et une valeur négative quelconque pour la case non cochée.

Autre exemple : rendre un bouton visible.

Pour connaître l'ordre de création du bouton (donc son rang dans le tableau « Buttons »), on peut par exemple parcourir l'ensemble des boutons de la feuille Excel (faire une boucle sur le nombre de boutons de la feuille) et comparer l'attribut « Text » de tous les boutons avec celui recherché.

Une fois le rang du bouton trouvé, il suffit d'affecter la valeur « False » à la propriété « Visible » du bouton.

```
Sheets ("nom de la feuille").Buttons(rang).Visible = False
```

Note :

Le tableau des « boutons de commande » a pour nom : « Buttons ».

Le tableau des « Cases à cocher » a pour nom : « CheckBoxes ».

Le tableau des « Cases d'options » a pour nom : « OptionButtons ».

Le tableau des « Zones de liste » a pour nom : « DropDowns ».

Le tableau des « Zones de texte » a pour nom : « Labels ».

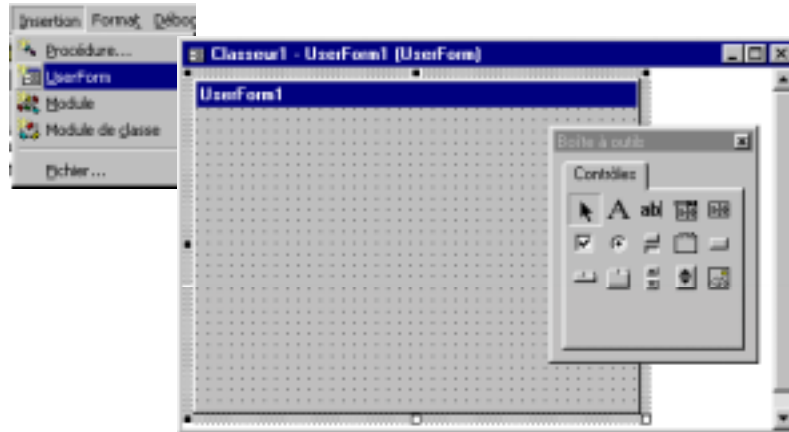
3.- Les boîtes de dialogues

3.1.- L'objet UserForm.

3.1.1.- Création d'une UserForm

La commande « *Userform* » du menu « *Insertion* » vous permet de créer des fenêtres ou des boîtes de dialogue pour votre projet.

Pour créer une fenêtre, activez la commande « *UserForm* » du menu « *Insertion* ».

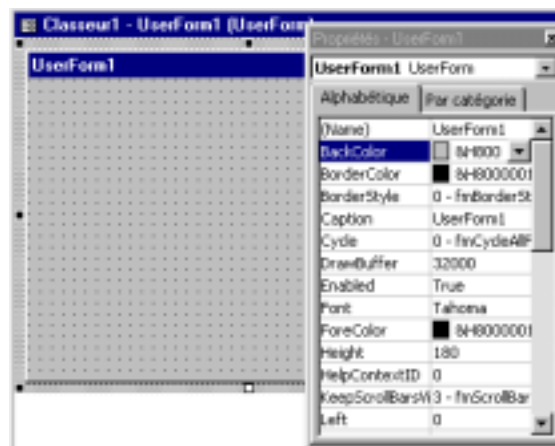


Une fois l'objet **UserForm** créé, utilisez les boutons de la boîte à outils pour dessiner des contrôles sur la feuille. Vous pouvez aligner automatiquement les contrôles sur la grille en choisissant l'option correspondante dans l'onglet Général de la boîte de dialogue Options.

Vous pouvez afficher la grille de feuille et définir l'espacement de ses lignes dans l'onglet Général de la boîte de dialogue Options.

3.1.2.- Modification du nom d'une UserForm.

- sélectionnez une UserForm ;
- dans la partie gauche de la fenêtre, face à « *Name* », modifiez le nom.



3.1.3.- Modification des propriétés d'une UserForm.

- sélectionnez une UserForm ;
- cliquez dans la partie gauche de la fenêtre, face à la propriété à modifier, sélectionnez la nouvelle valeur dans le menu déroulant.

Remarque : Pour connaître la signification d'une propriété, cliquez sur le libellé de la propriété et appuyez sur la touche F1.

3.2.- La boîte à outils

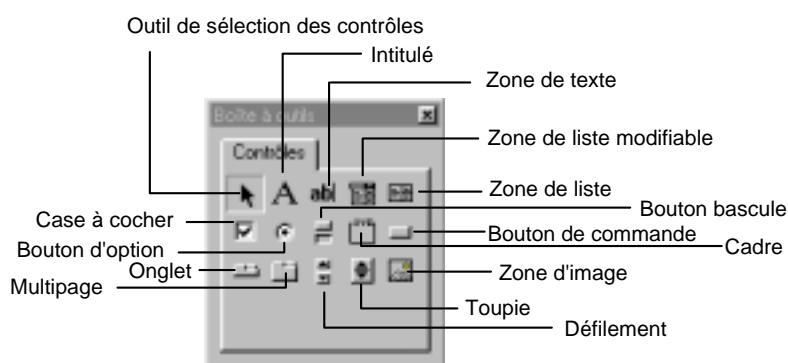
3.2.1.- Affichage ou masquage de la boîte à outils

Dans le menu Affichage, regardez si une coche s'affiche en regard de Boîte à outils. Si la coche est présente, la boîte à outils s'affiche, dans le cas contraire, elle est masquée.

Choisissez l'une des options suivantes :

- pour afficher la boîte à outils, vérifiez que la coche apparaît en regard de Boîte à outils. Si ce n'est pas le cas, sélectionnez Boîte à outils ;
- pour masquer la boîte à outils, vérifiez qu'aucune coche n'apparaît en regard de Boîte à outils. Si une coche apparaît, sélectionnez Boîte à outils pour la supprimer.

3.2.2.- Les contrôles de la boîte à outils



3.2.1.- Intitulé (Label)

Un contrôle Label dans une feuille affiche un texte descriptif tel que des titres, des légendes, des images ou de brèves instructions. Par exemple, les étiquettes d'un carnet d'adresses peuvent inclure un contrôle Label pour un nom, une rue ou une ville. Un contrôle Label n'affiche pas de valeurs issues de sources de données ou d'expressions ; il est toujours indépendant et ne change pas lorsque vous vous déplacez d'un enregistrement à l'autre.

La propriété par défaut d'un Label est la propriété **Caption**.

L'événement par défaut d'un Label est l'événement **Click**.

3.2.2.- Zone de texte (TextBox)

Le contrôle zone de texte est le contrôle le plus souvent utilisé pour afficher les informations entrées par un utilisateur. Il peut également afficher un ensemble de données, tel qu'une table, une requête, une feuille de programmation ou le résultat d'un calcul. Si un contrôle zone de texte est dépendant d'une source de données, les changements apportés au contenu d'un contrôle zone de texte modifient aussi la valeur de la source de données dont il dépend.

Le format appliqué à toute partie du texte du contrôle zone de texte se répercutera à la totalité du texte du contrôle. Par exemple, si vous changez la police ou la taille d'un caractère du contrôle, la modification affecte tous les caractères du contrôle.

La propriété par défaut d'un contrôle zone de texte est la propriété **Value**.

L'événement par défaut d'un contrôle zone de texte est l'événement **Change**.

3.2.3.- Zone de liste (ListBox)

Affiche une liste de valeurs et vous permet d'en sélectionner une ou plusieurs.

Si le contrôle Zone de liste est dépendant d'une source de données, il enregistre la valeur sélectionnée dans cette source de données.

Le contrôle Zone de liste peut apparaître sous la forme d'une liste, d'un groupe de contrôles OptionButton ou de contrôles CheckBox.

La propriété par défaut d'un contrôle Zone de liste est la propriété **Value**.

L'événement par défaut d'un contrôle Zone de liste est l'événement **Click**.

3.2.4.- Zone de liste modifiable (ComboBox)

Allie les caractéristiques d'un contrôle Zone de liste et d'un contrôle zone de texte. L'utilisateur peut entrer une nouvelle valeur, comme dans un contrôle zone de texte ou bien sélectionner une valeur existante comme dans un contrôle Zone de liste.

Si un contrôle Zone de liste modifiable est dépendant d'une source de données, il insère dans cette source de données la valeur que l'utilisateur entre ou sélectionne. Si une liste modifiable multicolonne est dépendante, la propriété BoundColumn détermine la valeur enregistrée dans la source de données dont dépend la liste modifiable.

La liste d'un contrôle Zone de liste modifiable se compose de lignes de données. Chacune d'elle peut contenir une ou plusieurs colonnes qui peuvent apparaître avec ou sans titre. Certaines applications n'utilisent pas les titres de colonne, d'autres ne les utilisent que partiellement.

La propriété par défaut d'un contrôle Zone de liste modifiable est la propriété **Value**.

L'événement par défaut d'un contrôle Zone de liste modifiable est l'événement **Change**.

3.2.5.- Case à cocher (CheckBox)

Affiche l'état de la sélection d'un élément.

Utilisez un contrôle Case à cocher pour permettre à l'utilisateur de choisir entre deux valeurs telles que Oui/Non, Vrai/Faux ou Actif/Inactif. Quand l'utilisateur sélectionne un contrôle Case à cocher, une marque spéciale (un X, par exemple) s'affiche et sa valeur courante est Oui, Vrai ou Actif ; si l'utilisateur ne sélectionne pas le contrôle Case à cocher, celui-ci est vide et sa valeur est Non, Faux ou Inactif. Selon la valeur de la propriété TripleState, un contrôle Case à cocher peut aussi avoir une valeur nulle.

Si un contrôle Case à cocher est dépendant d'une source de données, le changement de paramètre modifie la valeur de cette source. Un contrôle Case à cocher désactivé affiche sa valeur courante mais il apparaît ombré et ne permet pas à l'utilisateur de modifier sa valeur.

Vous pouvez aussi utiliser des cases à cocher à l'intérieur d'une zone de groupe pour sélectionner un ou plusieurs éléments connexes. Par exemple, vous pouvez créer un bon de commande contenant une liste des éléments disponibles, en faisant précéder chaque élément d'un contrôle Case à cocher. L'utilisateur peut sélectionner un ou plusieurs éléments particuliers en cochant le contrôle Case à cocher correspondant.

La propriété par défaut d'un contrôle Case à cocher est la propriété **Value**.

L'événement par défaut d'un contrôle Case à cocher est l'événement **Click**.

3.2.6.- Bouton d'options (OptionButton)

Affiche l'état de la sélection d'un élément faisant partie d'un groupe.

Utilisez un contrôle Bouton d'options pour afficher si un seul élément contenu dans un groupe est sélectionné. Tous les contrôles Bouton d'options d'un contrôle Frame s'excluent l'un l'autre.

Si un contrôle Bouton d'options est dépendant d'une source de données, il peut afficher la valeur de cette source de données comme étant Oui/Non, Vrai/Faux ou Actif/Inactif. Si l'utilisateur sélectionne le contrôle Bouton d'options, la valeur courante est Oui, Vrai ou Actif ; si l'utilisateur ne sélectionne pas le contrôle Bouton d'options, la valeur est Non, Faux ou Inactif. Par exemple, dans une application de suivi des stocks, un contrôle Bouton d'options peut faire apparaître si la production d'un élément est arrêtée. Si le contrôle Bouton d'options est dépendant d'une source de données, le changement de paramètre modifie la valeur de cette source. Un contrôle Bouton d'options désactivé est indisponible et il n'affiche aucune valeur.

Selon la valeur de la propriété TripleState, un contrôle Bouton d'options peut aussi avoir une valeur nulle.

Vous pouvez aussi utiliser un contrôle Bouton d'options à l'intérieur d'une zone de groupe pour sélectionner un ou plusieurs groupes d'éléments connexes. Par exemple, vous pouvez créer un bon de commande contenant une liste des éléments disponibles, en faisant précéder chaque élément d'un contrôle Bouton d'options. L'utilisateur peut sélectionner un élément particulier en cochant le contrôle Bouton d'options correspondant.

La propriété par défaut d'un contrôle Bouton d'options est la propriété **Value**.

L'événement par défaut d'un contrôle Bouton d'options est l'événement **Click**.

3.2.7.- Bouton bascule (ToggleButton)

Affiche l'état de la sélection d'un élément.

Utilisez un contrôle Bouton bascule pour afficher si un élément est sélectionné. Si un contrôle Bouton bascule est dépendant d'une source de données, il affiche la valeur courante de cette source de données comme étant Oui/Non, Vrai/Faux, Actif/Inactif ou tout autre choix de deux valeurs. Si l'utilisateur sélectionne le contrôle Bouton bascule, la valeur courante est Oui, Vrai ou Actif ; si l'utilisateur ne sélectionne pas le contrôle Bouton bascule, la valeur est Non, Faux ou Inactif. Si le contrôle Bouton bascule est dépendant d'une source de données, le changement de paramètre modifie la valeur de cette source. Un contrôle Bouton bascule désactivé affiche une valeur, mais il est indisponible et n'autorise pas les changements à partir de l'interface utilisateur.

Vous pouvez aussi utiliser un contrôle Bouton bascule à l'intérieur d'un contrôle Frame pour sélectionner un ou plusieurs groupes d'éléments connexes. Par exemple, vous pouvez créer un bon de commande contenant une liste des éléments disponibles, en faisant précéder chaque élément d'un contrôle Bouton bascule. L'utilisateur peut choisir un élément particulier en sélectionnant le Bouton bascule approprié.

La propriété par défaut d'un contrôle Bouton bascule est la propriété **Value**.

L'événement par défaut d'un contrôle Bouton bascule est l'événement **Click**.

3.2.8.- Cadre (Frame)

Crée un groupe de contrôles fonctionnel et visuel.

Tous les boutons d'option d'un contrôle Cadre s'excluent l'un l'autre, de sorte que vous pouvez utiliser le contrôle Cadre pour créer un groupe d'options. Vous pouvez aussi utiliser un contrôle **Cadre** pour regrouper des contrôles dont le contenu est étroitement associé. Par exemple, dans une application qui traite les commandes clients, vous pouvez utiliser un contrôle Cadre pour regrouper noms, adresses et numéros de comptes des clients.

Vous pouvez aussi utiliser un contrôle Cadre pour créer un groupe de boutons bascules, mais ceux-ci ne s'excluent pas l'un l'autre.

L'événement par défaut d'un contrôle Cadre est l'événement **Click**.

3.2.9.- Bouton de commande (CommandButton)

Lance, termine ou interrompt une action ou une série d'actions.

La macro ou la procédure d'événement affectée à l'événement Click du contrôle Bouton de commande détermine l'action de celui-ci. Par exemple, vous pouvez créer un contrôle Bouton de commande qui ouvre une autre feuille. Vous pouvez aussi afficher un texte, une image ou les deux sur un contrôle Bouton de commande.

La propriété par défaut d'un contrôle Bouton de commande est la propriété **Value**.

L'événement par défaut d'un contrôle Bouton de commande est l'événement **Click**.

3.2.10.- Onglet (tabStrip)

Présente un ensemble de contrôles connexes sous forme d'un groupe visuel.

Vous pouvez utiliser un contrôle Onglet pour visualiser différents ensembles d'informations. Par

exemple, les contrôles peuvent représenter des informations relatives à l'emploi du temps journalier d'un groupe de personnes, chaque ensemble d'informations correspondant à une personne différente du groupe. Définissez le titre de chaque onglet pour afficher le nom d'une seule personne. Vous pouvez ensuite écrire un code qui, quand vous cliquez sur l'onglet, met à jour les contrôles pour afficher les informations concernant la personne dont le nom figure sur l'onglet.

La propriété par défaut d'un contrôle Onglet est la propriété **SelectedItem**.

L'événement par défaut d'un contrôle Onglet est l'événement **Change**.

3.2.11.- MultiPage (MultiPage)

Le contrôle MultiPage est un conteneur d'une collection d'objets Pages dans laquelle chaque page contient un ou plusieurs objets Page.

Un contrôle MultiPage est utile lorsque vous travaillez avec un gros volume d'informations qui peuvent être triées en plusieurs catégories. Vous pouvez notamment utiliser un contrôle MultiPage pour afficher les informations contenues dans une candidature et, par exemple, réserver une page aux renseignements personnels tels que le nom et l'adresse, une autre à la liste des précédents employeurs et une troisième à la liste des références. Le contrôle MultiPage vous permet de combiner visuellement des informations connexes tout en conservant un enregistrement complet facilement accessible.

Les nouvelles pages sont ajoutées à droite de la page sélectionnée.

La propriété par défaut d'un contrôle MultiPage est la propriété **Value** qui renvoie l'index de la Page couramment active dans la collection de Pages du contrôle MultiPage.

L'événement par défaut d'un contrôle MultiPage est l'événement **Change**.

3.2.12.- Barre de défilement (ScrollBar)

Retourne ou définit la valeur d'un autre contrôle en fonction de la position du curseur de défilement.

Le contrôle Barre de défilement est un contrôle autonome que vous pouvez placer sur une feuille. Son aspect visuel est identique à celui de la barre de défilement présente dans certains objets tels qu'un contrôle Zone de liste ou dans la partie déroulante d'un contrôle Zone de liste modifiable. Toutefois, à l'inverse des barres de défilement citées dans ces exemples, le contrôle Barre de défilement autonome ne fait pas partie intégrante d'un autre contrôle.

Pour utiliser le contrôle Barre de défilement afin de définir ou de lire la valeur d'un autre contrôle, vous devez écrire un code pour les événements et les méthodes du contrôle Barre de défilement. Par exemple, pour utiliser le contrôle Barre de défilement en vue de mettre à jour la valeur d'un contrôle Zone de texte, vous pouvez écrire un code lisant la propriété Value du contrôle Barre de défilement, puis définir la propriété Value du contrôle Zone de texte.

La propriété par défaut d'un contrôle Barre de défilement est la propriété Value.

L'événement par défaut d'un contrôle Barre de défilement est l'événement Change.

Pour créer un contrôle Barre de défilement vertical ou horizontal, faites glisser les poignées de redimensionnement du contrôle Barre de défilement horizontalement ou verticalement sur la feuille.

3.2.13.- Toupie (SpinButton)

Incrémente et décrémente des nombres.

Cliquer sur un contrôle Toupie ne modifie que la valeur de celui-ci. Vous pouvez écrire un code qui utilise le contrôle Toupie pour mettre à jour la valeur affichée d'un autre contrôle. Par exemple, vous pouvez utiliser un contrôle Toupie pour changer le mois, le jour ou l'année dans une date. Vous pouvez aussi utiliser un contrôle Toupie pour faire défiler une série de valeurs ou une liste d'éléments, ou bien pour modifier la valeur affichée dans une zone de texte.

Pour afficher une valeur mise à jour à l'aide d'un contrôle Toupie, vous devez affecter la valeur du contrôle Toupie à la partie affichée d'un contrôle telle que la propriété Caption d'un contrôle Label ou la propriété Text d'un contrôle Zone de texte. Pour créer un contrôle Toupie horizontal ou vertical, faites glisser les poignées de redimensionnement du contrôle Toupie horizontalement ou verticalement sur la feuille.

La propriété par défaut d'un contrôle Toupie est la propriété **Value**.

L'événement par défaut d'un contrôle Toupie est l'événement **Change**.

3.2.14.- Zone d'Image (Image)

Le contrôle Image vous permet d'afficher une image comme partie des données contenues dans une feuille. Par exemple, vous pouvez utiliser un contrôle Image pour afficher la photographie des employés dans une feuille du personnel.

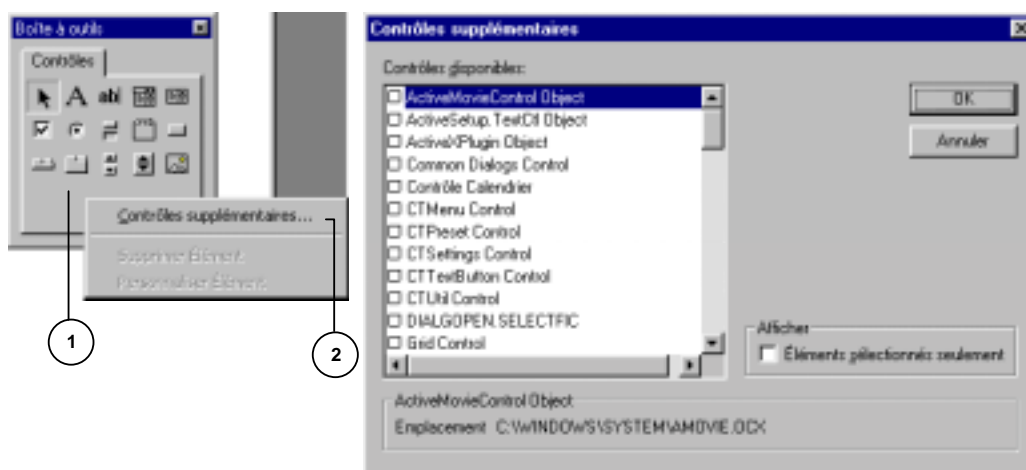
Le contrôle Image vous permet de couper, de redimensionner ou d'effectuer un zoom avant ou arrière sur l'image, mais ne vous permet pas d'en modifier le contenu. Par exemple, vous ne pouvez utiliser le contrôle Image pour modifier les couleurs d'une image, pour la rendre transparente ou pour affiner la vue de l'image. Pour effectuer ces opérations, il vous faut un logiciel de traitement d'image.

Le contrôle Image supporte les formats de fichiers suivants : *.bmp, *.cur, *.gif, *.ico, *.jpg, *.wmf

3.3.- Les contrôles supplémentaires

Pour afficher ou supprimer des contrôles de la boîte à outils :

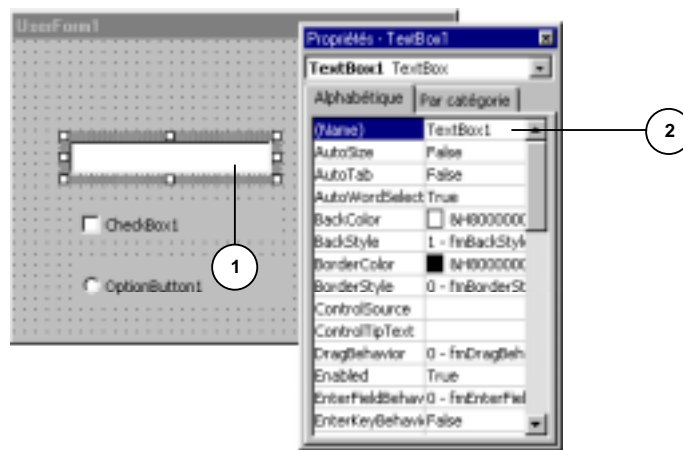
1. Cliquez avec le bouton droit dans la boîte à outils.
2. Sélectionnez la commande « Contrôles supplémentaires ».



3.4.- Modification d'un contrôle

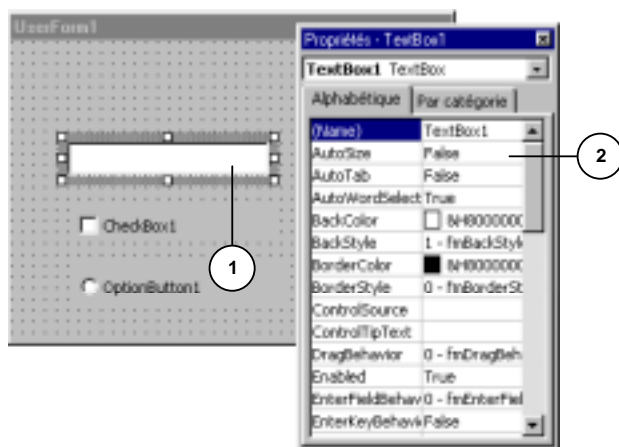
3.4.1.- Modification du nom

1. sélectionnez le contrôle ;
2. dans la partie gauche de la fenêtre, face à « Name » ,modifiez le nom.



3.4.2.- Modification d'une propriété

1. sélectionnez le contrôle ;
2. cliquez dans la partie gauche de la fenêtre, face à la propriété à modifier, sélectionnez la nouvelle valeur dans le menu déroulant.



Remarque : Pour connaître la signification d'une propriété, cliquez sur le libellé de la propriété et appuyez sur la touche F1.

3.5.- La superposition des contrôles

On utilise la notion de hiérarchie pour superposer les contrôles.

Pour avancer ou pour reculer un contrôle dans la hiérarchie:

- sélectionnez le contrôle à repositionner ;
- dans le menu Format, sélectionnez « *Ordre* » ;
- dans le menu en cascade, sélectionnez « *Amener au premier plan* » ou « *Envoyer à l'arrière-plan* ».

Pour déplacer un contrôle d'une position dans la hiérarchie :

- sélectionnez le contrôle à repositionner.
- dans le menu Format, sélectionnez « *Ordre* ».
- dans le menu en cascade, sélectionnez « *Amener vers l'avant* » ou « *Envoyer vers l'arrière* ».

4.- Les menus

4.1.- Le « menu barre »

Pour créer une nouvelle barre de menu dans votre application, il faut utiliser l'outil « personnalisé » d'Excel.



Une fois le menu créé (Cf. documentation sur Excel), il est possible dans votre code Visual Basic » de le rendre « visible » ou non en utilisant la fonction « CommandBars ». Cette fonction peut également être utilisée pour les menus prédéfinis du logiciel Excel (« standard », « mise en forme »...). Ci-dessous un exemple de code lancé au démarrage d'une application (dans la méthode auto_open – Cf. chapitre 7.1). Ce code permet de rendre « invisible » les menus « standard » et « mise en forme » et de rendre visible un menu créé par le programmeur appelé « Saisie ».

```
Sub auto_open ()
    Application.CommandBars("Standard").Visible = False
    Application.CommandBars("Formatting").Visible = False
    Application.CommandBars("Saisie").Visible = True
End Sub
```

4.2.- Le « menu contextuel »

C'est le menu qui s'active lorsque vous cliquez sur le bouton droit de la souris. Il est possible de supprimer certaines lignes de ce menus et d'en créer de nouvelles. En sélectionnant une ligne de ce menu vous pouvez déclencher un module (et donc un traitement). Ci-dessous le code permettant d'activer le menu contextuel sur le fond d'une feuille Excel en cliquant sur le bouton droit de la souris. Dans ce cas, les lignes 4 et 5 du menu existant sont remplacées par deux lignes permettant d'activer les modules « Moyenne par élève » et « Moyenne par Matière ». L'objet « menu contextuel » est un objet de la classe « CommandBars ».

```

Private cbBar As CommandBar ' CommandBar sans « s » !

Sub Active_Menu_contextuel()

    Set cbBar = CommandBars("Cell") 'Cell est le menu contextuel des cellules

    cbBar.Controls(4).Delete 'Détruit le 4eme choix (en partant d'en
                                'haut) du menu contextuel existant

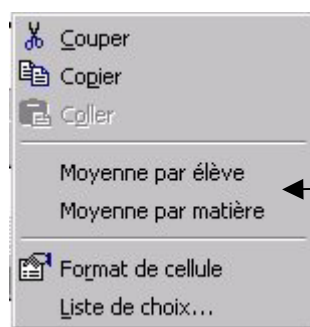
    cbBar.Controls(5).Delete

    cbBar.Controls(4).FaceId = 0 'Efface le bouton du menu
    cbBar.Controls(4).Caption = "Moyenne par élève" 'libellé dans le menu
    cbBar.Controls(4).OnAction = "MoyParEleve" 'Active la macro MoyParEleve

    cbBar.Controls(5).FaceId = 0 'Efface le bouton du menu
    cbBar.Controls(5).Caption = "Moyenne par matière" 'libellé dans le menu
    cbBar.Controls(5).OnAction = "MoyParMat" 'Active la macro MoyParMat

End Sub

```



Insertion de « Moyenne par élève » en ligne 4 et de « Moyenne par Matière » en ligne 5.

Attention, il est indispensable de désactiver le menu contextuel pour qu'il ne soit plus visible à l'écran.

```

Sub Desactive_Menu_contextuel()
    CommandBars("cell").Reset
End Sub

```

Si vous voulez ajouter plus de lignes qu'il y en avait dans le menu d'origine (9 dans l'exemple ci-dessous), il faut écrire le code suivant avant d'ajouter de nouvelles lignes dans le menu.

```

Private cbctl As CommandBarControl ' a ajouter aux declarations

lIndex = 9 ' on ajoute une 9eme ligne
Set cbctl = cbBar.Controls.Add(Type:=msoControlButton, Before:=lIndex)

```

5.- Réponses aux événements

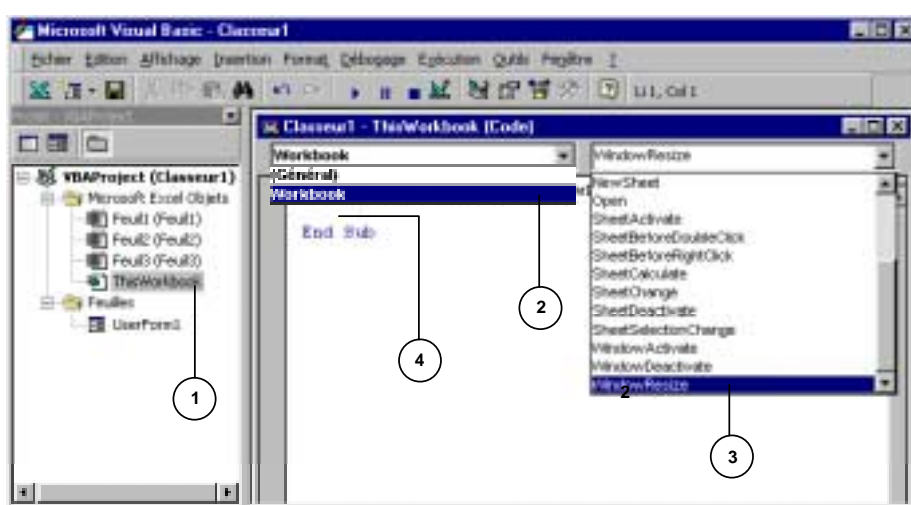
Votre application doit pouvoir réagir aux divers événements, par exemple exécuter une procédure lorsqu'une feuille est activée, ou lorsque la valeur d'une cellule est modifiée.

Dans la fenêtre de code, pour chaque objet, ou chaque contrôle vous avez une liste de procédures événementielles dans lesquelles vous pouvez ajouter du code qui s'exécutera lors de la détection de l'événement correspondant.

Exemple 1 :

Supposons que vous désirez exécuter une tâche lorsque la fenêtre d'un classeur est redimensionnée :

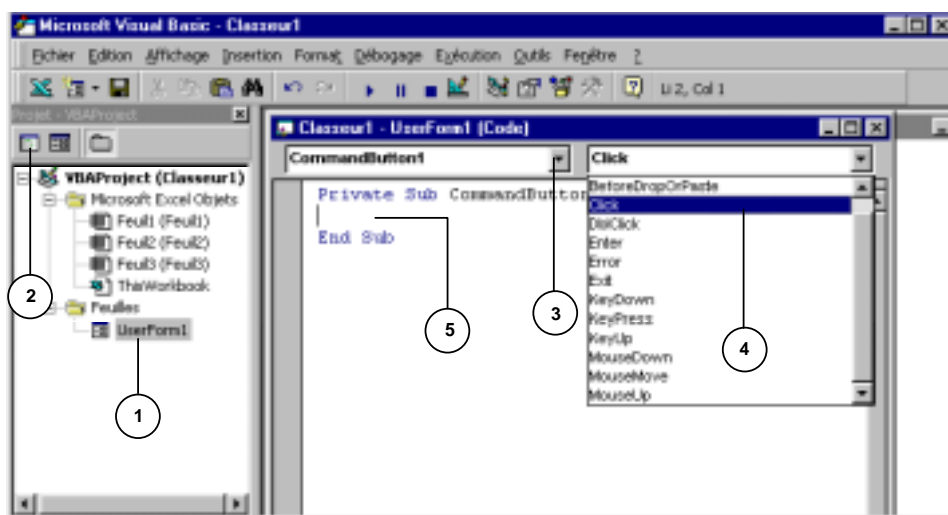
1. dans la fenêtre du projet, vous sélectionnez l'objet « Classeur » ;
2. dans la fenêtre de code du classeur, vous sélectionnez « Workbook » ;
3. dans la liste des procédures événementielles vous sélectionnez « WindowResize » ;
4. il ne vous reste plus qu'à écrire le code à l'intérieur de la procédure créée par Visual Basic.



Exemple 2 :

Supposons que vous désirez exécuter une tâche lorsque l'on cliquera sur un contrôle.

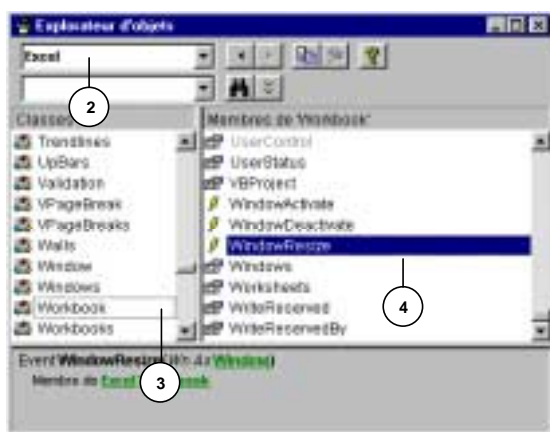
1. Dans la fenêtre du projet, sélectionnez la feuille ou la boîte de dialogue contenant le contrôle ;
2. Cliquez sur l'icône pour afficher la fenêtre de code ;
3. Sélectionnez le contrôle ;
4. Sélectionnez l'événement ;
5. Il ne vous reste plus qu'à écrire le code à l'intérieur de la procédure créée par Visual Basic.



6.- L'aide en ligne sur les procédures événementielles

Pour revenir aux exemples de la page précédente, supposons que nous voulions avoir de l'aide sur la procédure événementielle « *WindowResize* ».

1. affichez l'explorateur d'objets ;
2. sélectionnez la bibliothèque « *Excel* » ;
3. sélectionnez « *Workbook* » ;



4. sélectionnez « *WindowResize* » ;
5. appuyer sur la touche F1 pour obtenir l'aide ci-dessous.

WindowResize, événement

Voir aussi Exemple S'applique à
Cet événement survient dès que la fenêtre d'un classeur est redimensionnée.

Syntaxe

Private Sub objet_WindowResize (**ByVal** Wb **As** Excel.Workbook, **ByVal** Wn **As** Excel.Window)

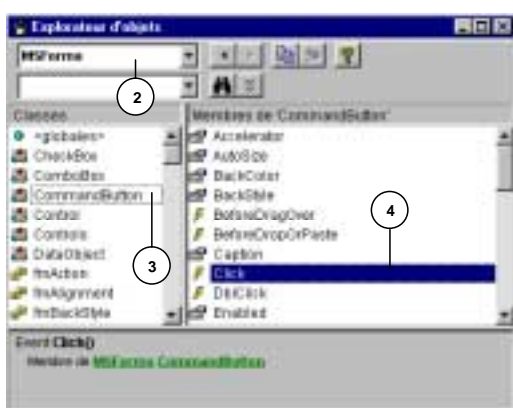
objet Application ou Workbook. Pour plus d'informations sur l'utilisation d'événements avec l'objet Application, consultez Utilisation d'événements avec l'objet Application.

Wb Utilisé uniquement avec l'objet Application. Classeur affiché dans la fenêtre redimensionnée.

Wn Fenêtre redimensionnée.

On ferait de même pour l'exemple 2 de la page précédente :

1. affichez l'explorateur d'objet ;
2. sélectionnez la bibliothèque « *MSForms* » ;
3. sélectionnez « *CommandButton* ».



4. sélectionnez « *Click* » ;
5. appuyer sur la touche F1 pour obtenir l'aide en ligne.

7.- Actions possibles sur un document Excel

7.1.- Ouverture et fermeture d'un document Excel

Si vous voulez effectuer un traitement au moment de l'ouverture d'un document Excel, vous devez écrire votre code dans un module particulier ayant pour nom : `auto_open()`. Il est intéressant d'effectuer la phase d'initialisation dans ce module.

```
Sub auto_open ()  
    Call Initialisation 'appel d'un module Initialisation'  
End Sub
```

De même, si vous voulez effectuer un traitement au moment de la fermeture d'un document Excel, vous devez écrire votre code dans un module particulier ayant pour nom : `auto_close()`.

```
Sub auto_close ()  
    ...  
End Sub
```

7.2.- Actions sur l'objet « feuille Excel »

Masquer une feuille de calcul

```
If Sheets("NomOnglet").Visible Then Sheets("NonOnglet").Visible = False
```

Démasquer une feuille de calcul

```
If Not Sheets("NomOnglet").Visible Then Sheets("NonOnglet").Visible = True
```

Oter la protection d'une feuille de calcul

```
Sheets("NomOnglet").Select  
ActiveSheet.Unprotect
```

Protéger une feuille de calcul

```
Sheets("NomOnglet").Select  
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
```

Lancer l'impression d'une feuille

```
Sheets("NomOnglet").Select  
ActiveWindow.SelectedSheets.PrintOut Copies:=2, Collate:=True
```

MCours.com

7.3.- Actions sur l'objet « cellule Excel »

Affecter la valeur d'une cellule à une variable.

```
MaVariable = Sheets("NomOnglet").Cells (N° de ligne, N° de colonne).Value
```

Affecter la valeur d'une variable à une cellule.

```
Sheets("NomOnglet").Cells(N° de ligne, N° de colonne).Value = MaVariable
```

Obtenir l'adresse de la cellule active

```
'Renvoi sous la forme R3C4  
Dim RefCel As String  
RéfCel = ActiveCell.Address(ReferenceStyle:=xlR1C1)
```

```
'Pour obtenir la ligne et la colonne  
Dim Ligne As Integer, Colonne As Integer  
Ligne = CInt(Mid$(RefCel, 2, InStr(RefCel, "C") - 2))  
Colonne = CInt(Mid$(RefCel, InStr(RefCel, "C") + 1))  
'Obtenir l'adresse de la plage de cellules sélectionnées  
ReferenceCelluleActive = Selection.Address(ReferenceStyle:=xlR1C1)  
'Renvoi R3C4:R10C5
```

Sélectionner une cellule

```
Sheets("NomOnglet").Cells(2,2).Select
```

Activer une cellule

```
Sheets("NomOnglet").Cells(2,2).Activate
```

Déplacer la cellule active d'une colonne

```
ActiveCell.Offset(0,1).Activate
```

Déplacer la cellule active d'une ligne

```
ActiveCell.Offset(1,0).Activate
```

Sélectionner une plage de cellules

```
With Sheets("NomOnglet ")  
    Range(.Cells(1,1), .Cells(5,5)).Select  
End With
```

Sélectionner plusieurs plages de cellules

```
Sub SélectionMultiple()  
    Dim r1, r2, plusieursPlages As Range  
    Set r1 = Sheets("NomOnglet ").Range(Cells(1,1), Cells(5,2))  
    Set r2 = Sheets("NomOnglet ").Range(Cells(1,6), Cells(5,8))  
    Set plusieursPlages = Union(r1,r2)  
    plusieursPlages.Select  
End Sub
```

Affecter une formule à une cellule

```
Cells(7, 1).Select  
ActiveCell.FormulaR1C1 = "=SUM(R[-6]C:R[-1]C)"
```

7.4.- Autres actions sur des objets Excel

Supprimer la mise à jour de l'écran

```
Application.ScreenUpdating = False
```

Autoriser la mise à jour de l'écran

```
Application.ScreenUpdating = True
```

Demander une information à l'utilisateur

```
MaVariable = InputBox("Entrez le n° de la commande à supprimer", "")
```

Envoyer un message à l'utilisateur

```
'Dim message as string , rep as integer  
'Chr(13) Pour aller à la ligne  
message = "ATTENTION : Vous avez oublié d'indiquer le code comptable" +  
Chr$(13) + Chr$(13) + "Opération annulée!..."  
rep = MsgBox(message, 48, "")
```

Supprimer une ligne

```
Rows(N° de la ligne).Select  
Selection.Delete Shift:=xlUp
```

Insérer une ligne

```
Rows("N° de la ligne ").Select  
Selection.Insert Shift:=xlDown
```

Charger en mémoire une boîte de dialogue UserForm

```
Load UserForm1     'Charge un objet sans l'afficher
```

Décharger une boîte de dialogue UserForm

```
Unload UserForm1
```

Affichage d'une boîte de dialogue UserForm

```
UserForm1.Show     'Si l'objet spécifié n'est pas chargé lors de l'appel de  
                  'la méthode Show, Visual Basic le charge automatiquement.
```

Remarque :

Un objet UserForm est toujours modal, il en résulte que l'utilisateur doit toujours répondre avant d'utiliser une autre partie de l'application. Aucun autre code ne s'exécutera tant que l'objet UserForm ne sera pas masqué ou déchargé. Bien que les autres feuilles de l'application soient désactivées pendant l'affichage d'un objet UserForm, les autres applications ne le sont pas.

Masquer d'une boîte de dialogue UserForm

```
UserForm1.Hide     'Masque un objet sans le décharger.
```

Récupérer l'information contenue dans une zone de texte

```
MonTexte = NomUserForm.NomZoneDeTexte.Text
```

Test d'une case à cocher

```
If NomUserForm.NomCaseàcocher Then
    'La case est cochée
Else
    'La case n'est pas cochée
End If
```

Test d'une case d'option

```
If NomUserForm.NomCaseOptionThen
    'L'option est cochée
Else
    'L'option n'est pas cochée
End If
```

Récupérer l'information sélectionnée dans une zone de liste

```
NomSélectionné = NomUserForm.NomListe.value
```

7.5.- Quelques exemples d'actions plus complexes

Ajout et suppression dans une listBox

L'exemple suivant ajoute et supprime le contenu d'un contrôle ListBox . un contrôle ListBox nommé ListBox1, et 2 contrôles CommandButton nommés CommandButton1 et CommandButton2.

```
Dim EntryCount As Single
Private Sub CommandButton1_Click()
    EntryCount = EntryCount + 1
    UserForm1.ListBox1.AddItem (EntryCount & " - Sélection")
End Sub

Private Sub CommandButton2_Click()
    With UserForm1
        ListBox1.SetFocus
        'Vérifiez que le contrôle ListBox contient des éléments de liste
        If ListBox1.ListCount >= 1 Then    'Si aucune sélection, choisissez le
                                            'dernier élément de la liste.
            If ListBox1.ListIndex = -1 Then
                ListBox1.ListIndex = ListBox1.ListCount-1
            End If
            ListBox1.RemoveItem (ListBox1.ListIndex)
        End If
    End With
End Sub

Private Sub UserForm_Initialize()
    EntryCount = 0
    CommandButton1.Caption = "Ajoutez l'élément"
    CommandButton2.Caption = "Supprimez l'élément"
End Sub
```

Transfert de données vers le Presse-papiers

L'exemple suivant illustre les mouvements de données d'un contrôle TextBox vers le Presse-papiers, du Presse-papiers vers un objet DataObject et d'un objet DataObject vers un autre contrôle TextBox. La méthode GetFromClipboard transfère les données du Presse-papiers vers un objet DataObject. Les méthodes Copy et GetText sont aussi utilisées. Deux contrôles TextBox (TextBox1, TextBox2) et un contrôle CommandButton (CommandButton1) sont créés.

```

Dim MyData As DataObject
Private Sub CommandButton1_Click()      'Sélectionnez un texte avant de le
                                         'copier dans le Presse-papiers
    TextBox1.SelStart = 0
    TextBox1.SelLength = TextBox1.TextLength
    TextBox1.Copy
    MyData.GetFromClipboard
    TextBox2.Text = MyData.GetText(1)
End Sub

Private Sub UserForm_Initialize()
    Set MyData = New DataObject
    TextBox1.Text = "Placez ces données dans " & _
        "le Presse-papiers, dans un objet " & _
        "DataObject, puis dans le contrôle " & "TextBox2 !"
End Sub

```

L'exemple suivant illustre les mouvements de données d'un contrôle TextBox vers un objet DataObject, d'un objet DataObject vers le Presse-papiers et du Presse-papiers vers un autre contrôle TextBox. La méthode PutInClipboard transfère les données d'un objet DataObject vers le Presse-papiers. Les méthodes SetText et Paste sont aussi utilisées. Deux contrôles TextBox (TextBox1 et TextBox2) et un contrôle CommandButton (CommandButton1) sont créés.

```

Dim MyData As DataObject
Private Sub CommandButton1_Click()
    Set MyData = New DataObject
    MyData.SetText TextBox1.Text
    MyData.PutInClipboard
    TextBox2.Paste
End Sub

Private Sub UserForm_Initialize()
    TextBox1.Text = "Placez ces données dans un " & _
        "objet DataObject, dans le " & _
        " Presse-papiers, puis dans " & _
        "le contrôle TextBox2 !"
End Sub

```

Annexes





A1.- Constantes, Boutons et icônes des boîtes de messages

Un message dans une boîte de dialogue attend que l'utilisateur clique sur un bouton puis renvoie une valeur de type Integer qui indique le bouton choisi par l'utilisateur.

```
MsgBox(prompt[, buttons] [, title] [, helpfile, context])
```

Facultatif : expression numérique qui représente la somme des valeurs indiquant le nombre et le type de boutons à afficher, le style d'icône à utiliser, l'identité du bouton par défaut, ainsi que la modalité du message. Si l'argument buttons est omis, sa valeur par défaut est 0.

L'argument buttons peut prendre les valeurs suivantes :

Constante	Valeur	Description
vbOKOnly	0	Affiche le bouton OK uniquement.
vbOKCancel	1	Affiche les boutons OK et Annuler.
vbAbortRetryIgnore	2	Affiche le bouton Abandonner, Réessayer et Ignorer.
vbYesNoCancel	3	Affiche les boutons Oui, Non et Annuler.
vbYesNo	4	Affiche les boutons Oui et Non.
vbRetryCancel	5	Affiche les boutons Réessayer et Annuler.
<hr/>		
vbCritical	16	Affiche l'icône Message critique. 
vbQuestion	32	Affiche l'icône Requête d'avertissement. 
vbExclamation	48	Affiche l'icône Message d'avertissement. 
vbInformation	64	Affiche l'icône Message d'information. 
<hr/>		
vbDefaultButton	0	Le premier bouton est le bouton par défaut.
1		
vbDefaultButton	256	Le deuxième bouton est le bouton par défaut.
2		
vbDefaultButton	512	Le troisième bouton est le bouton par défaut.
3		
vbDefaultButton	768	Le quatrième bouton est le bouton par défaut.
4		
<hr/>		
vbApplicationModal	0	Boîte de dialogue modale. L'utilisateur doit répondre au message affiché dans la zone de message avant de pouvoir continuer de travailler dans l'application en cours.
vbSystemModal	4096	Modal système. Toutes les applications sont interrompues jusqu'à ce que l'utilisateur réponde au message affiché dans la zone de message.

Le premier groupe de valeurs (0 à 5) décrit le nombre et le type de boutons affichés dans la boîte de dialogue. Le deuxième groupe (16, 32, 48 et 64) décrit le style d'icône. Le troisième groupe

(0, 256 et 512) définit le bouton par défaut.

Enfin, le quatrième groupe (0 et 4 096) détermine la modalité de la zone de message. Au moment d'additionner ces nombres pour obtenir la valeur finale de l'argument buttons, ne sélectionnez qu'un seul nombre dans chaque groupe.

Note Ces constantes sont indiquées par **Visual Basic pour Applications**. Vous pouvez par conséquent les utiliser n'importe où dans votre code à la place des valeurs réelles correspondantes.

Valeurs renvoyées

Constante	Valeur	Bouton
vbOK	1	OK
vbCancel	2	Annuler
vbAbort	3	Abandonner
vbRetry	4	Réessayer
vbIgnore	5	Ignorer
vbYes	6	Oui
vbNo	7	Non

A2.- Constantes de code de touches

Les constantes suivantes peuvent remplacer les valeurs réelles partout dans votre code :

Constante	Valeur	Description
vbKeyLButton	0x1	Bouton gauche de la souris
vbKeyRButton	0x2	Bouton droit de la souris
vbKeyCancel	0x3	Touche ANNULER
vbKeyMButton	0x4	Bouton secondaire de la souris
vbKeyBack	0x8	Touche RET.ARR
vbKeyTab	0x9	Touche TAB
vbKeyClear	0xC	Touche EFFACER
vbKeyReturn	0xD	Touche ENTRÉE
vbKeyShift	0x10	Touche MAJ
vbKeyControl	0x11	Touche CTRL
vbKeyMenu	0x12	Touche MENU
vbKeyPause	0x13	Touche PAUSE
vbKeyCapital	0x14	Touche VERR.MAJ
vbKeyEscape	0x1B	Touche ÉCHAP
vbKeySpace	0x20	Touche ESPACE
vbKeyPageUp	0x21	Touche PG.PRÉC
vbKeyPageDown	0x22	Touche PG.SUIV
vbKeyEnd	0x23	Touche FIN
vbKeyHome	0x24	Touche ORIGINE
vbKeyLeft	0x25	Touche FLÈCHE GAUCHE
vbKeyUp	0x26	Touche FLÈCHE HAUT
vbKeyRight	0x27	Touche FLÈCHE DROITE
vbKeyDown	0x28	Touche FLÈCHE BAS
vbKeySelect	0x29	Touche SÉLECTION
vbKeyPrint	0x2A	Touche IMPR.ÉCRAN
vbKeyExecute	0x2B	Touche EXÉCUTER
vbKeySnapshot	0x2C	Touche SNAPSHOT
vbKeyInsert	0x2D	Touche INSER
vbKeyDelete	0x2E	Touche SUPPR
vbKeyHelp	0x2F	Touche AIDE
vbKeyNumlock	0x90	Touche VERR.NUM

Les touches A à Z sont les mêmes que leurs équivalents ASCII :

Constante	Valeur ascii	Description
vbKeyA	65	Touche A
vbKeyB	66	Touche B
vbKeyC	67	Touche C
vbKeyD	68	Touche D
vbKeyE	69	Touche E
vbKeyF	70	Touche F
vbKeyG	71	Touche G
vbKeyH	72	Touche H
vbKeyI	73	Touche I

vbKeyJ	74	Touche J
vbKeyK	75	Touche K
vbKeyL	76	Touche L
vbKeyM	77	Touche M
vbKeyN	78	Touche N
vbKeyO	79	Touche O
vbKeyP	80	Touche P
vbKeyQ	81	Touche Q
vbKeyR	82	Touche R
vbKeyS	83	Touche S
vbKeyT	84	Touche T
vbKeyU	85	Touche U
vbKeyV	86	Touche V
vbKeyW	87	Touche W
vbKeyX	88	Touche X
vbKeyY	89	Touche Y
vbKeyZ	90	Touche Z

Les touches 0 à 9 sont les mêmes que leurs équivalents ASCII :

Constante	Valeur	Description
vbKey0	48	Touche 0
vbKey1	49	Touche 1
vbKey2	50	Touche 2
vbKey3	51	Touche 3
vbKey4	52	Touche 4
vbKey5	53	Touche 5
vbKey6	54	Touche 6
vbKey7	55	Touche 7
vbKey8	56	Touche 8
vbKey9	57	Touche 9

Les constantes suivantes représentent les touches du pavé numérique :

Constante	Valeur	Description
vbKeyNumpad0	0x60	Touche 0
vbKeyNumpad1	0x61	Touche 1
vbKeyNumpad2	0x62	Touche 2
vbKeyNumpad3	0x63	Touche 3
vbKeyNumpad4	0x64	Touche 4
vbKeyNumpad5	0x65	Touche 5
vbKeyNumpad6	0x66	Touche 6
vbKeyNumpad7	0x67	Touche 7
vbKeyNumpad8	0x68	Touche 8
vbKeyNumpad9	0x69	Touche 9
vbKeyMultiply	0x6A	Touche MULTIPLICATION (*)
vbKeyAdd	0x6B	Touche PLUS (+)
vbKeySeparator	0x6C	Touche ENTRÉE
vbKeySubtract	0x6D	Touche MOINS (-)
vbKeyDecimal	0x6E	Touche POINT DÉCIMAL (.)

vbKeyDivide 0x6F Touche DIVISION (/)

Les constantes suivantes représentent les touches de fonction :

Constante	Valeur	Description
vbKeyF1	0x70	Touche F1
vbKeyF2	0x71	Touche F2
vbKeyF3	0x72	Touche F3
vbKeyF4	0x73	Touche F4
vbKeyF5	0x74	Touche F5
vbKeyF6	0x75	Touche F6
vbKeyF7	0x76	Touche F7
vbKeyF8	0x77	Touche F8
vbKeyF9	0x78	Touche F9
vbKeyF10	0x79	Touche F10
vbKeyF11	0x7A	Touche F11
vbKeyF12	0x7B	Touche F12
vbKeyF13	0x7C	Touche F13
vbKeyF14	0x7D	Touche F14
vbKeyF15	0x7E	Touche F15
vbKeyF16	0x7F	Touche F16