

Informations sur les composants principaux de Visual Basic

Généralités

Les composants de Visual Basic – feuille ou contrôles – sont identifiés à partir de leurs *noms*. Ces noms sont donnés au cours de la phase de conception de l'interface (lorsque la partie *Objet* est affichée), dans la première ligne de la barre de propriétés : partie **(Name)**.

Dans la partie *Code*, l'accès à une *propriété* du *composant* se fait de la manière suivante : **Composant.Propriété**

Par exemple, pour accéder au nombre d'éléments (valeur entière) d'un objet de type **ListBox** appelé **List1** (propriété **ListCount**), il faut faire :

List1.ListCount

Propriétés communes

Les propriétés communes à la plupart des composants Visual Basic sont les suivantes :

1. Disponibilité et visualisation du composant :

- **Enabled** : propriété booléenne qui détermine si une feuille ou un contrôle peuvent répondre aux événements générés par l'utilisateur.
- **Visible** : propriété booléenne qui indique si un objet est visible ou caché.

2. Emplacement du composant :

- **Left** : propriété entière qui indique l'endroit où s'affiche le composant à partir de la gauche de l'écran (s'il s'agit d'une feuille) ou à gauche de la feuille (s'il s'agit d'un contrôle de cette feuille) en nombre de twips¹ (unité définie par défaut).
- **Top** : propriété entière qui indique l'endroit où s'affiche le composant à partir du haut en nombre de twips (distance entre le bord interne supérieur d'un objet et le bord supérieur de son conteneur).

3. Dimension du composant :

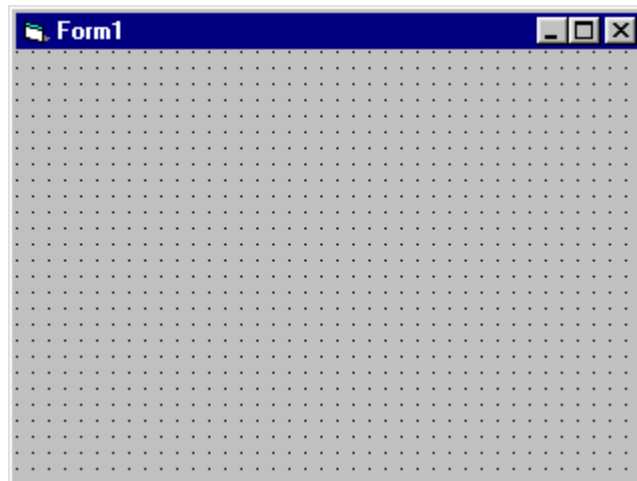
- **Height** : propriété entière qui indique la hauteur du composant en twips.
- **Width** : propriété entière qui indique la largeur du composant en twips.

¹ Un twip est une unité indépendante de l'écran, utilisée pour garantir que l'emplacement et les proportions de l'application soient les mêmes, quel que soit le système d'affichage. Le twip est une unité de mesure d'écran égale au 1/20^{ème} d'un point d'une imprimante, donc il y a environ 567 twips pour un centimètre logique.

La feuille (Form)

La feuille, ou **Form**, est le composant de base de Visual Basic.

C'est sur une feuille que vont s'afficher les contrôles.



Les principales propriétés d'une feuille sont :


- **Caption** : texte de la barre de titre de la feuille.
- **Moveable** : propriété booléenne qui indique si on peut déplacer la fenêtre de la feuille.
- **Picture** : image de fond de la feuille.
- **StartPosition** : valeur spécifiant la position de la feuille lors de sa première apparition.
- **WindowState** : valeur indiquant l'état visuel d'une fenêtre lors de l'exécution (0 pour la taille normale, 1 pour un état minimisé et 2 pour un état maximisé).

Parmi les autres propriétés d'une feuille, il y a :

- **BackColor** : couleur de fond du composant.
- **Font** : police de caractères.
- **ScaleMode** : pour passer de dimensions twips en centimètres ou en pouces.

L'étiquette (Label)

L'étiquette, ou **Label**, est un contrôle graphique qui permet d'afficher du texte que l'utilisateur ne peut pas modifier directement. Une étiquette est généralement utilisée dans le cas des titres d'autres contrôles ou pour afficher des informations résultant de traitements.

Une étiquette est sélectionnée, dans la partie *Objet*, à partir de l'icône suivante : 

L'étiquette ajoutée sur une feuille (ou un contrôle adapté) a cette forme :




La propriété essentielle d'une étiquette est le texte qu'elle affiche :

- **Caption** : texte de l'étiquette.

Une étiquette dispose de bien d'autres propriétés dont les plus intéressantes concernent les formats d'affichage (comme **Font** ou **BackColor**).

La zone de texte (TextBox)

Une zone de texte, ou **TextBox**, est un contrôle qui affiche des informations entrées au moment de la création, tapées par l'utilisateur ou affectées au contrôle au moment de l'exécution. Une zone de texte est aussi parfois appelée champ d'édition ou contrôle d'édition.

Une zone de texte est définie à partir de l'icône suivante : 

La zone de texte ajoutée sur une feuille (ou un contrôle adapté) a cette forme :



La propriété essentielle d'une zone de texte est son texte :


- **Text** : texte de la zone de texte.

Autres propriétés utiles d'une zone de texte :

- **Locked** : propriété booléenne qui permet de bloquer l'écriture dans la zone de texte (une zone de texte devient alors équivalente à une étiquette).
- **MultiLine** : propriété booléenne qui indique si une zone de texte peut accepter des lignes multiples.

La liste de texte (ListBox)

Un contrôle **ListBox** affiche une liste simple dans laquelle l'utilisateur peut sélectionner un ou plusieurs éléments. Si le nombre d'éléments est trop grand pour tenir dans la zone affichée, une barre de défilement lui est automatiquement ajoutée.

La liste de texte simple est créée à partir de l'icône suivante : 

Une liste de texte simple se présente sous cette forme :




La liste de texte est un tableau de chaînes de caractères dont le premier élément a l'indice 0. L'accès aux éléments du tableau se fait à partir de la propriété **List**. Ainsi, l'instruction **List1.List(3)** fera appel au 4^{ème} élément de la liste de texte appelée **List1**.

Les propriétés principales d'une liste de texte sont :

- **List** : élément du tableau.
- **ListIndex** : numéro de l'élément de la liste sélectionné. Si aucun élément n'est sélectionné, la propriété **ListIndex** a la valeur -1. La propriété **ListIndex** du premier élément de la liste a la valeur 0.
- **ListCount** : nombre d'éléments de la liste. La valeur de la propriété **ListCount** est toujours supérieure d'une unité à la valeur la plus élevée de **ListIndex** puisque le comptage des éléments de la liste commence à partir du numéro 0.
- **AddItem** : méthode qui permet d'ajouter une chaîne de caractères dans la liste de texte simple (à la fin de la liste). Exemple : **List1.AddItem("Bonjour")**.
- **RemoveItem** : méthode qui permet de supprimer l'élément *i* de la liste de texte (avec *i* = 0 pour le premier élément, *i* = 1 pour le second élément, etc.).
- **Clear** : méthode qui efface tous les éléments de la liste.
- **Sorted** : propriété booléenne qui, si elle est mise à vrai (True), permet d'ajouter des éléments en les triant par ordre alphabétique.

La liste combinée (ComboBox)

Un contrôle **ComboBox** est la combinaison d'une liste de texte simple et d'une zone de texte. L'icône qui permet d'ajouter une liste combinée est : .


La liste combinée a la forme suivante :



En plus des propriétés d'une liste simple, la liste combinée a les propriétés suivantes :

- **Style** : propriété qui indique le comportement de la liste : 0 (*DropDown Combo*) pour une liste combinée déroulante (saisie de la zone de texte ou choix de l'item dans la liste) ; 1 (*Simple Combo*) pour une liste combinée simple (zone de texte de saisie et liste non déroulante) et 2 (*Dropdown List*) pour une liste déroulante sans possibilité de saisie.
- **Text** : texte de la zone de texte de la liste combinée.

Le bouton de commande (CommandButton)

Un bouton de commande est un contrôle utilisé spécifiquement pour lancer les traitements lorsque l'on clique sur celui-ci. L'icône qui permet d'ajouter un bouton de commande est : .


Un bouton de commande a la forme suivante :



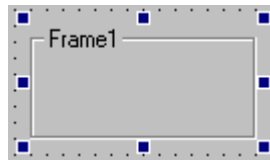
La propriété essentielle d'un bouton de commande est le texte que celui-ci affiche et qui précise le type de traitement effectué lorsqu'on clique dessus :

- **Caption** : texte du bouton de commande.

Le cadre (Frame)

Un cadre est un contrôle qui permet de regrouper d'autres contrôles, en particulier des boutons d'option (**OptionButton**) ou des cases à cocher (**CheckBox**). L'icône qui permet d'ajouter un cadre est : .

Un cadre a la forme suivante :



La propriété importante d'un cadre est son texte de titre (affichage d'informations concernant les contrôles qui s'y trouvent regroupés) :

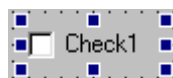
- **Caption** : texte du cadre.

La case à cocher (CheckBox)

Une case à cocher est un contrôle qui peut changer de valeur par un simple click de souris. Les cases à cocher peuvent être cochées, non cochées ou cochées sans changement possible (les cases apparaissent cochées et grisées). Il est possible de cocher plusieurs cases, ou aucune, parmi le groupe où elles appartiennent (la fenêtre entière ou un cadre donné).

L'icône qui permet d'ajouter une case à cocher est : .

Une case à cocher prend la forme suivante :



Les propriétés principales d'une case à cocher sont :

- **Caption** : texte de la case à cocher.
- **Value** : valeur de la case, soit 0 si la case n'est pas cochée (*Unchecked*), soit 1 si la case est cochée (*Checked*), soit 2 si la case est cochée et grisée (*Grayed*).

Le bouton d'option (OptionButton)

Un bouton d'option est un contrôle qui change de valeur par un simple click de souris. Les boutons d'option peuvent être cochés ou non cochés. Contrairement aux cases à cocher, il ne peut y avoir qu'un seul bouton d'option activé parmi le groupe de boutons d'option auquel il appartient (que ce soit la fenêtre entière ou un cadre).

L'icône qui permet d'ajouter un bouton d'option est : .

Un bouton d'option, sur une fenêtre ou un cadre, prend la forme suivante :




Les propriétés principales d'un bouton d'option sont :

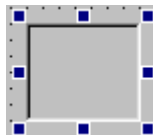
- **Caption** : texte du bouton d'option.
- **Value** : valeur booléenne du bouton d'option (vrai – *True* – si le bouton est activé, faux – *False* – dans le cas contraire).

L'image (PictureBox)

Un contrôle **PictureBox** peut afficher un élément graphique issu d'une image bitmap (BMP ou DIB), d'une icône (ICO) ou d'un métafichier (WMF), ainsi que d'un métafichier étendu ou de fichiers JPEG ou GIF. L'élément graphique est découpé si le contrôle n'est pas assez grand pour afficher l'image complète.

L'icône qui permet d'ajouter une image est : .

Un contrôle **PictureBox** – lorsque l'image n'est pas encore chargée – prend la forme suivante :




La propriété principale d'une image est :

- **Picture** : image de format divers (BMP, DIB, ICO, WMF, GIF, JPG, etc.) qui va être chargée à l'emplacement du contrôle **PictureBox**.

La grille (MSFlexGrid)

La grille est un contrôle qui permet d'afficher des tableaux à deux dimensions (données en ligne et en colonne).

Par défaut, la grille n'apparaît pas sur la barre d'outils des composants standards. Pour l'ajouter, il faut faire un click droit sur cette barre, choisir l'option "**Composants...**", choisir le composant "**Microsoft FlexGrid Control 5.0**" dans la liste des composants en cliquant sur sa case à cocher pour l'activer, valider cette sélection en appuyant sur le bouton de commande "**Appliquer**" et revenir au menu de conception de l'interface en cliquant sur le bouton de commande "**OK**".

L'icône qui permet d'ajouter une grille est : .

Une grille a la forme suivante :



Les propriétés principales d'une grille sont :

- **Cols** : nombre de colonnes de la grille.
- **Rows** : nombre de lignes de la grille.
- **FixedCols** : nombre de colonnes grisées (pour le texte de titre).
- **FixedRows** : nombre de lignes grisées.

Comme dans le cas d'une liste, les données ne sont pas entrées directement dans une grille. Les données d'une grille peuvent être entrées de deux manières.

La première manière consiste à indiquer quelle est la colonne courante, quelle est la ligne courante, et quelle est la valeur texte de cette cellule de la grille.

- **Col** : valeur entière de la colonne courante de la grille.
- **Row** : valeur entière de la ligne courante de la grille.
- **Text** : valeur texte de la cellule courant de la grille.

Par exemple :

```
MSFlexGrid1.Col = 1  
MSFlexGrid1.Row = 3  
MSFlexGrid1.Text = "Bonjour"
```


A partir de la grille suivante :

Les instructions précédentes ajouteront du texte dans la colonne n° 1, au niveau de la ligne n° 3. Attention, tout comme pour une liste, les indices commencent à partir de 0. Ainsi, la colonne n° 1 est la deuxième colonne et la ligne n° 3 est la quatrième ligne (les cases grisées font partie du compte de l'ensemble des lignes et des colonnes).

Les instructions précédentes auront ainsi pour résultat la modification suivante :

	Bonjour			

Une seconde manière de procéder consiste à utiliser la seule instruction **TextMatrix** qui fonctionne de la manière suivante :

```
MSFlexGrid1.TextMatrix(3, 1) = "Bonjour"
```

Le texte est directement ajouté dans la grille à partir de deux paramètres : l'indice de la ligne (3) et l'indice de la colonne (1).