



FICHE TECHNIQUE N° III (1/2)

- Chapitre : ELEMENTS DU LANGAGE -

LES CONTROLES

Les contrôles sont des objets graphiques permettant à l'utilisateur d'interagir avec le programme. Ils permettent une utilisation dynamique du programme. Ce sont des zones d'entrées/sorties, des zones de commande ou autres zones diverses. Ces contrôles sont déposés sur une feuille de travail appelée "Form" qui est elle même un contrôle. L'ensemble des contrôles affichés à l'écran constitue donc l'interface graphique. Sachez qu'en terminologie Windows ces contrôles sont encore appelés WIDGETS (concaténation de WINDOWS et de GADGETS).

Les contrôles standards :

Ce sont les contrôles les plus souvent utilisés. Ils sont disponibles dans la boîte à outils de Visual Basic.

Voici un tableau récapitulatif des contrôles standards :

Nom	Libellé	Commentaires
Form	Feuille	Fenêtre Windows
TextBox	Zone de texte	
Label	Etiquette	
PictureBox	Zone d'image	
Image	Image	
Frame	Cadre	
CommandButton	Bouton de commande	
CheckBox	Case à cocher	Plusieurs choix possibles
OptionButton	Bouton d'option	Choix unique
ComboBox	Liste déroulante	ou liste modifiable
ListBox	Liste	
HScrollBar	Barre de défilement horizontale	
VScrollBar	Barre de défilement verticale	
Timer	Temporisateur	Instructions exécutées à intervalle régulier
DriveListBox	Liste de lecteurs	
DirListBox	Liste de répertoires	
FileListBox	Liste de fichiers	
Shape	Forme	Formes géométriques
Line	Ligne	Trait
Data Control	Contrôle d'accès aux données	Accès aux bases de données
OLE	Contrôle objet	OLE = Object Linking Embedding



Remarques : il existe trois types de feuilles :

La feuille classique (Form),
La feuille principale ou mère (MDIForm),
La feuille fille (MDIChild).

La feuille classique est indépendante. C'est la feuille par défaut. Elle n'a aucun lien avec une autre feuille. Pour ajouter une feuille, cliquez sur "Ajouter une feuille" du menu "Projet" de la barre de menu.

La feuille principale (MDIForm) est une feuille mère, c'est à dire qu'il ne peut en avoir qu'une par projet (ou programme). C'est sur cette feuille que va venir s'encapsuler toutes les autres feuilles du projet qui seront des feuilles MDIChild. Pour ajouter une feuille MDI, cliquez sur "Ajouter une feuille MDI" du menu "Projet" de la barre de menu.

La feuille MDIChild est une feuille fille, il peut en avoir plusieurs dans un projet (ou programme). Pour ajouter une telle feuille, il suffit d'ajouter une feuille classique et de renseigner la valeur "True" dans la propriété "MDIChild" de la fenêtre de propriété.

L'intérêt d'utiliser une feuille MDI est que toutes les feuilles filles se réduisent dans la feuille mère (et non dans la barre d'état), si il y a une barre de menu dans la feuille principale, elle remplacera celle des feuilles filles lorsque celles ci seront actives. Enfin, la feuille MDI ne peut contenir que des contrôles "Menu" et "PictureBox".

Les contrôles spéciaux :

Ces contrôles ne sont pas disponibles dans la boîte à outil par défaut. Pour les afficher, cliquez sur "composants" dans le menu contextuel "projet". La liste des composants optionnels disponibles apparaît alors. Sélectionnez le composant "Microsoft Windows Common Controls 6.0". Cliquez ensuite sur "appliquer", les contrôles optionnels apparaîtront dans la boîte à outils standard. De nombreux autres contrôles optionnels sont disponibles de cette manière dès lors que l'on sélectionne les composants de cette liste optionnelle, la difficulté étant de connaître les composants auxquels appartiennent ces contrôles. Exemple : Le contrôle MSFlexGrid (Grille de base de données) devient disponible dès lors que l'on sélectionne le composant "Microsoft FlexGrid Control 5.0". L'utilisation des composants est expliquée dans la section "composants".

Voici un tableau récapitulatif de ces contrôles :

Nom	Libellé	Commentaires
ToolBar	Barre d'outils	
StatusBar	Barre d'état	
ProgressBar	Barre de progression	Représentation de l'état d'avancement d'une tâche
Slider	Contrôle curseur	Sélection d'une valeur par glissement

Il est possible de créer des menus contextuels pour vos applications Visual Basic grâce à des contrôles "Menu". Pour cela, il faut lancer le créateur de menu placé dans le menu contextuel "Outils". Le principe d'utilisation des menu est expliqué dans la section "Menu".

Exemple d'une fenêtre recensant les contrôles les plus fréquemment utilisés. Les noms qui sont affichés sur les contrôles correspondent au type de ces contrôles.

Exemple : Le contrôle "Label" est un contrôle de type "Label".

Remarques : La fenêtre "Form" est un contrôle "Form". Chaque fenêtre créée en VB est un contrôle.

La barre bleue est une ProgressBar, l'image est affichée dans un contrôle "PictureBox" et la barre verticale à droite de la fenêtre est une "VScrollBar".



Lors de l'édition du programme, quand vous tapez le nom de votre contrôle suivi du point, VB affiche une liste des propriétés et méthodes disponibles pour le type de votre contrôle. Les propriétés sont préfixées d'un dessin représentant une main sur une feuille et les méthodes sont préfixées d'un cylindre vert.



LES PROPRIETES

Les propriétés permettent de définir les caractéristiques (comme les dimensions, les couleurs ...) et les états (comme actif, visible ...) des contrôles. Lors de la création d'un objet, les propriétés sont renseignées avec des valeurs par défaut. Ces propriétés sont modifiables lors de l'élaboration du programme à l'aide de la fenêtre de propriété ou (pour la plupart) lors de l'exécution à l'aide d'instructions inscrites dans les procédures événementielles. Selon les types de contrôles, il existe un certain nombre de propriétés disponibles. Lors de l'édition du programme, quand vous tapez le nom de votre contrôle suivi du point, VB affiche une liste des propriétés et méthodes disponibles pour le type de votre contrôle. Les propriétés sont préfixées d'un dessin représentant une main sur une feuille et les méthodes sont préfixées d'un cylindre vert.

Voici une liste des propriétés les plus souvent rencontrées :

Nom	Libellé	Valeur
Name	Nom	Alphanumérique
Caption	Etiquette	Alphanumérique
Text	Texte	Alphanumérique
Enabled	Disponibilité	True or false
Font	Style de police	Sélection police et couleur
ForeColor	Couleur d'avant plan	couleur en hexadécimale
BackColor	Couleur d'arrière plan	couleur en hexadécimale
Visible	Visibilité	True or false
Autosize	Autoredimensionné	True or false
Value	Valeur	Alphanumérique (ex:zone texte) ou booléen (bouton d'options)
BorderStyle	Style de bordure	Liste
ToolTipText	Infos bulles	Alphanumérique
TabStop	Arrêt tabulation	True or false
TabIndex	Index de tabulation	Numérique
Picture	Nom de l'image	Alphanumérique

Attention : la propriété "Name" doit être obligatoirement renseignée en mode création et est unique dans un programme car elle sert d'identifiant au contrôle. Deux contrôles ne peuvent pas avoir le même nom (sauf si il y a création d'un groupe de contrôle, dans ce cas chaque contrôle du groupe à le même nom suivi d'un numéro unique qui est son identifiant dans le groupe). Cette propriété n'est pas modifiable en mode exécution.

Remarque : il est conseillé de nommer clairement vos contrôles afin de simplifier la clarté de votre programme. Il existe d'ailleurs une convention de dénomination standard qui consiste à préfixer le nom de votre contrôle des trois lettres correspondantes à son type.

Ex : Le préfixe du "CommandButton" est "Cmd", on le nommera donc "CmdXXXX" où "XXXX" est le nom que vous aurez choisi (nom qui sera de préférence en rapport avec son rôle (exemple : CmdQuitter pour un bouton permettant de quitter l'application).



Voici la convention de dénomination pour les contrôles les plus utilisés :

Contrôle	Préfixe
Check Box	Chk
Combo Box	Cbo
Command Button	Cmd
Data	Dat
Form	Frm
Frame	Fra
Horizontal Scroll Bar	Hsb
Image	Img
Label	Lbl
List Box	Lst
Menu	Mnu
OLE	Ole
Option Button	Opt
Picture Box	Pic
Text Box	Txt
Timer	Tmr
Vertical Scroll Bar	Vsb

La convention de dénomination complète est disponible dans l'aide en ligne du logiciel Visual Basic".



Exemple d'utilisation d'une propriété dans le code :

On a un bouton de commande (commandbutton) dont la propriété "name" est "CmdQuit".
On veut modifier son texte (ou étiquette). On modifie donc la propriété "caption" ce qui se traduit par l'instruction suivante en mode exécution :

```
CmdQuit.Caption = "Quitter"
```



LES METHODES

Dans Visual Basic, les objets peuvent avoir des comportements grâce à des fonctions qui leurs sont propres. Ces fonctions associées aux objets sont appelées "Méthode". Comme pour les propriétés, chaque type de contrôle à ses méthodes spécifiques. Contrairement aux propriétés, les méthodes ne sont évidemment disponibles qu'en mode exécution.

La syntaxe est la suivante : NomContrôle.NomMéthode

Lors de l'édition du programme, quand vous tapez le nom de votre contrôle suivi du point, VB affiche une liste des propriétés et méthodes disponibles pour le type de votre contrôle. Les propriétés sont préfixées d'un dessin représentant une main sur une feuille et les méthodes sont préfixées d'un cylindre vert.

Voici quelques exemples de méthodes :

Nom	Libellé	Commentaires
Additem	Ajouter Ligne	Dans une liste ou liste déroulante
Clear	Effacer toutes les lignes	Dans une liste ou liste déroulante
Refresh	Actualiser	



LES EVENEMENTS

Visual Basic est un langage de programmation événementiel. C'est à dire que l'exécution des instructions est déclenchée lorsqu'un événement se produit. Cet événement est soit déclenché par l'utilisateur (clique sur un bouton, sélection d'une valeur dans une liste, déplacement de la souris ...) ou par le programme lui même (au chargement d'une feuille, temporisation ...).

Le principe de cette programmation est donc relativement simple. On doit se poser les deux questions suivantes :

- par quel objet (contrôle) est exécuté la ou les instructions ?
- Sur quel(s) événement(s) est exécutée l'instruction ?

Une fois la réponse aux deux questions trouvée. Il suffit d'écrire la suite d'instructions dans le bon événement du bon contrôle.

Voici une liste des principaux événements :

Nom	Action exécutée
Load	au chargement de l'objet
Unload	au déchargement de l'objet
Click	sur clique
DbClick	sur double clique
KeyPress	sur pression d'une touche
GotFocus	sur prise de focus
LostFocus	sur perte de focus
Change	sur changement de valeur

Dans Visual Basic, tous le programme repose sur l'utilisation des procédures (sub). Les instructions déclenchées par un événement sont donc enregistrées dans une procédure dont le nom a la structure suivante :

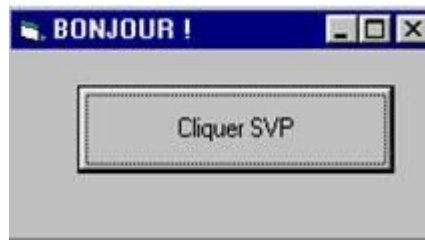
NomDuContrôle_NomDeL'Événement.

Ce type de procédure est appelée "procédure événementielle". Pour l'exemple qui va suivre, les instructions sont inscrites dans la procédure Sub BUTTON1_Click.

Exemple : Un clique sur un bouton permet d'afficher le message "Bonjour".

On place l'instruction : MSGBOX ("Bonjour") dans l'événement "CLICK" du bouton "BUTTON".

Interface :



Code source :

NOM DU CONTROLE	Type d'événement
BUTTON1	Click


```
Private Sub BUTTON1_Click()  
MsgBox ("BONJOUR")  
End Sub
```

Procédure événementielle "BUTTON1_Click"