

Tutoriel 8 : Utilisation primaire des contrôles

par Gilbert Miralles (gilmir.developpez.com)

Date de publication : Lundi 18 mars 2003

Dernière mise à jour : Lundi 28 janvier 2008



I - Introduction

II - Les contrôles Visual basic

II-A - Utilisation de l' objet "TextBox"

II-B - Utilisation de l' objet "CommandButton"

Le comportement du bouton doit répondre aux attentes de l' utilisateur.

IV - QCM

Prochain tutorial

I - Introduction

Dans les pages qui vont suivre nous allons apprendre à utiliser les contrôles avec pour chacun leur code correspondant afin de les rendre opérationnels.

Certains contrôles sont faciles à programmer, d' autres le sont moins surtout pour les débutants aussi nous accompagnerons les explications de ces modules avec des exemples à télécharger. Libre à vous d 'utiliser cette option.

II - Les contrôles Visual basic

Les contrôles Visual basic sont des outils graphiques qui servent à construire l'interface utilisateur d'un programme. Ils sont regroupés dans la boîte à outils de l'environnement de programmation.

Il suffit de quelques clics et de déplacement de souris pour créer des objets sur une feuille.

Vous apprendrez dans les prochaines leçons à afficher des informations dans une zone de texte, à parcourir les lecteurs et dossiers présents sur votre système, à traiter les données utilisateurs, à démarrer des applications Windows et à visualiser des enregistrements de bases de données.

Les exercices vous aideront à concevoir vos propres applications et vous feront étudier en détail les objets, propriétés et codes de programme.

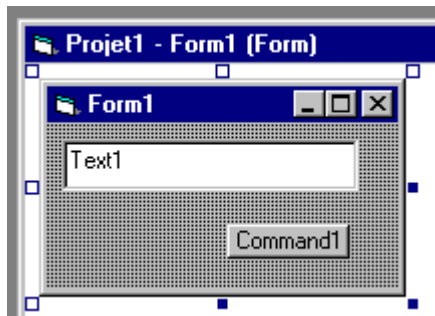
II-A - Utilisation de l'objet "TextBox"

Dans cet exercice nous utiliserons deux contrôles, le contrôle "TextBox" (zone de texte) et le contrôle "CommandButton" (bouton de commande), ils sont tout deux présents dans la boîte à outils.

- 1 Démarrez Visual Basic et créez une nouvelle application "EXE standard".
- 2 Positionnez sur la feuille(Form1) nouvellement créée un contrôle TextBox

Placez le pointeur de la souris au milieu de la feuille(il se transforme en croix), et tracez une zone de texte similaire au cliché ci-après.

- 3 Positionnez sur la feuille un contrôle "CommandButton".



Une zone de texte sert à afficher du texte sur une feuille ou de traiter les saisies de l'utilisateur pendant l'exécution d'un programme. L'utilisateur peut modifier le contenu de la saisie.

A la conception du programme ou en cours d'utilisation l'on peut modifier les propriétés de l'objet soit par la fenêtre de propriétés, soit en programmant les lignes de codes nécessaires à l'application de nouvelles instructions.

Dans notre exemple, l'objet zone de textes sera utilisé pour afficher un message lorsque l'utilisateur appuiera sur le bouton de commande.

II-B - Utilisation de l'objet "CommandButton"

Un bouton de commande permet de traiter l'événement utilisateur le plus courant.

Comme dans la vie courante lorsque vous appuyez sur un interrupteur dans votre salon, le lampadaire s'allume.

Vous avez provoqué un événement, vous attendiez une réponse à celui-ci, en l' occurrence que le lampadaire s' allume.

En programmation objet sous Visual Basic, il en va de même, l' utilisateur provoque un événement (lorsqu'il appuie sur le bouton) et attend en réponse l' exécution immédiate d' une action spécifique.

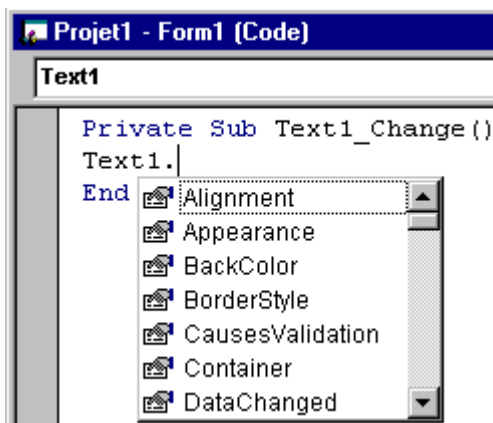
Le comportement du bouton doit répondre aux attentes de l'utilisateur.

Vous pouvez comme pour la feuille Form1 étudiée précédemment, modifier les caractéristiques des boutons de commande (comme celles de tous les objets) à l'aide des propriétés du contrôle utilisé. Vérifiez les possibilités offertes dans la feuille de propriétés. Lorsque l'on a positionné les objets sur la feuille comme indiqué sur la figure ci-dessus, on va apporter les modifications souhaitées, à savoir :

| Contrôle | Propriété | Paramètre |
|----------|-----------|------------|
| Text1 | Text | Vide(rien) |
| Command1 | Caption | &Ok |

Une particularité de la version 6.0 de Visual basic qui n'est pas à négliger pour les débutants, c'est la zone de liste qui affiche toutes les propriétés valides pour l'objet que vous utilisez.

(Le caractère (perluette) écrit devant le "O" de "OK" sert simplement à souligner le premier caractère de chaque mot afin de créer un lien devant lancer la procédure par l'appui de ce caractère depuis le clavier.)



Cette zone de liste s'affiche lorsque vous saisissez dans la feuille de code le nom de l'objet, suivi du point.

Dans le cas présent il s'agit du contrôle "TextBox" qui, lorsque l'on écrira Text1 suivi du point (Text1.) déroulera la zone de liste.

Celle-ci vous indique toutes les propriétés disponibles valides pour ce contrôle.

Ecriture du code devant générer l'événement souhaité :

Effectuez un double-clic sur le bouton de commande "OK".

Une fenêtre s'ouvre, c'est la feuille des procédures (écriture du code) qui correspond à l'objet à traiter. Dans cette feuille nous trouvons déjà les instructions par défaut de l'objet.

```
Private Sub Command1_Click( )
'
End Sub
```

Si vous n'écrivez rien, il ne se passe rien.

Comme nous souhaitons générer un événement, nous allons écrire : Votre prénom suivi de votre nom! Ce qui donne :

```
Private Sub Command1_Click( )
'Commentaires (écrire ici l'idée de l'événement à générer) je mettrais par Ex : 'Ecrire dans la
  TextBox
  Text1.Text = "Gaston Lagaffe"
End Sub
```

Explications : `Private sub` est le début de la procédure "Command1_Click" Suivi par les commentaires (je précise que ceux-ci ne sont pas pris en compte par le compilateur ,donc ne vous en privez pas) qui doivent obligatoirement commencer par une apostrophe.

Viennent ensuite les instructions qui doivent être prises en compte et exécutées par le programme.

Je traduirais cette instruction plus simplement pour les néophytes par :

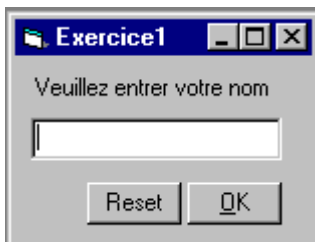
Dans l' objet intitulé Text1 et particulièrement dans la procédure Text, écrit la chaîne de caractères suivante "Gaston Lagaffe" (qui s' écrira obligatoirement entre guillemets).

Et `End Sub` qui indique que la procédure est terminée.

Vous pouvez également pour donner du cachet à votre réalisation créer un second bouton qui effacera le texte déjà écrit dans la TextBox comme par exemple :

| Contrôle | Propriété | Paramètre |
|----------|-----------|-----------|
| Command2 | Caption | Reset |

```
Private Sub Command2_Click( )
'Effacer la zone de texte
Text1.Text = ""
End Sub
```



Nous pouvons également à titre pédagogique utiliser le contrôle "Label" Nous l' utiliserons particulièrement pour créer des étiquettes car contrairement au contrôle TextBox, le "Label" ne peut pas être modifié par l' utilisateur.


Exemple : Vous placez un "Label"(étiquette) au dessus de la "TextBox" et vous écrivez dans la propriété "Caption" de celui-ci : "Veuillez entrer votre nom " Ce qui modifiera la présentation de notre réalisation par la figure suivante.

Nous terminons notre réalisation en lui apportant les modifications décrites en aparté de la figure jointe et nous procédons aux essais en passant du mode "**Conception**" au mode "**Exécution**" comme expliqué dans les leçons précédentes.

IV - QCM

Vous pouvez si vous le souhaitez répondre à votre premier QCM (questions à choix multiples) réalisé à l'intention de ceux qui suivent ces tutoriels. **QCM1 - Visual Basic 6.0**

Prochain tutorial

 **Prochain tutoriel** : *Utiliser des objets systèmes de fichiers*

