



Microsoft[®] **Visual Basic** 6.0



'Une image vaut mille mots'

Bien débuter avec Visual Basic

1. Introduction
2. Comment démarrer avec Visual Basic ?
3. Les étapes de la réalisation d'une application
4. Sauvegarder une application
5. Notions de base (Le contrôle Form)
6. Notions de base (Le Label)
7. Notions de base (Le CommandButton)
8. Apprentissage par l'exemple



Introduction

*Visual Basic 6.0 est un logiciel de programmation créé par la société **Microsoft** pour permettre le développement des applications qui peuvent fonctionner de manière autonome sous Windows.*

*Le modèle de programmation utilisé par Visual Basic est de type **événementiel** c'est-à-dire que seulement une partie du code est exécuté suite à une action précise de l'utilisateur (appelée événement).*

*Visual Basic est considéré également comme un langage de programmation **orientée objet** : chaque partie du code est liée à un objet, et c'est une action sur cet objet qui déclenche son exécution.*

*Il existe sur le marché **trois** éditions de Visual Basic :*

- **L'édition Initiation** est la moins fournie des trois mais permet de créer de puissantes applications.*
- **L'édition Professionnelle** dispose de toutes les fonctionnalités de l'édition initiation et se voit enrichie de nombreux contrôles et outils de développement.*
- **L'édition Entreprise** est la plus complète.*

2. Les étapes de la réalisation d'une application

La création d'une application dans Visual Basic implique trois étapes principales :

- ✓ Créer l'interface.
- ✓ Définir les propriétés.
- ✓ Écrire le code.

3. Sauvegarder une application

Lorsqu'on sauvegarde une **application**, Visual Basic va créer :

1. Un ou plusieurs fichiers feuilles (**.frm**) : Un fichier mémorise tous les éléments attachés à une feuille : la description de la fenêtre, la description des contrôles inclus dans la fenêtre, les procédures et fonctions liées à la feuille.
2. Un fichier dit **projet** (**.vbp**) : Il mémorise la liste de tous les fichiers nécessaires au projet.

Quand le projet est terminé il peut être transformé (**compilé**) en un seul fichier **exécutable** « .EXE » qui s'exécutera sans lancer Visual Basic, mais il nécessitera toutefois d'avoir le logiciel installé.

2. Notions de base (le contrôle Form)

Par défaut Visual basic affiche une feuille vierge appelée **Form1**

Pour commencer, nous allons examiner deux propriétés :

- **La propriété Name** : il s'agit du nom de l'objet tel qu'il est géré par l'application. Par défaut, Visual Basic donne à tous les objets que vous créez de noms génériques, comme **Form1**, **Form2**, **Form3**...

- **La propriété Caption** : il s'agit du texte associé à l'objet sur l'écran.

Notes



Concernant la propriété **name**, il est important de changer les noms donnés par défaut aux contrôles utilisés et de choisir des noms plus évocateurs.

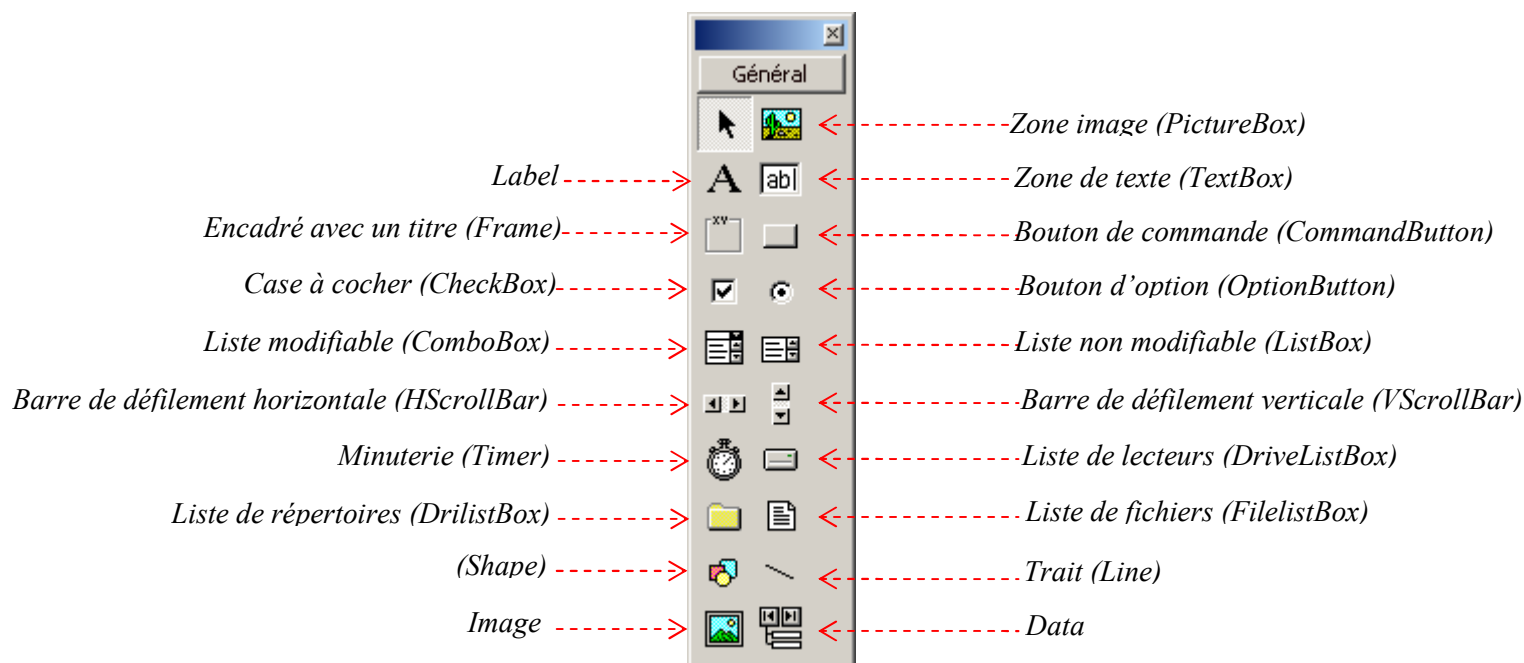
Par convention, on utilise un préfixe de 3 lettres spécifique à la famille de l'objet concerné, suivi d'un nom significatif du rôle rempli par le contrôle dans le projet.

Exemple : Au lieu de conserver **form1** comme **name**, on choisira **frmclients** pour une feuille qui servira à réunir les détails sur les clients.

Les préfixes souvent utilisés pour chaque objet :

<i>Objet</i>	<i>Préfixe</i>
<i>Bouton de commande (CommandButton)</i>	<i>cmd</i>
<i>Bouton d'option (OptionButton)</i>	<i>opt</i>
<i>Cadre (Frame)</i>	<i>fra</i>
<i>Case à cocher (CheckBox)</i>	<i>chk</i>
<i>Etiquette (Label)</i>	<i>lbl</i>
<i>Formulaire ou Feuille (Form)</i>	<i>frm</i>
<i>Image (Image)</i>	<i>img</i>
<i>Liste modifiable (ComboBox)</i>	<i>cbo</i>
<i>Menu (Menu)</i>	<i>mnu</i>
<i>Minuterie (Timer)</i>	<i>tmr</i>
<i>Photo (Picture)</i>	<i>pic</i>
<i>Zone de liste (ListBox)</i>	<i>lst</i>
<i>Zone de texte (TextBox)</i>	<i>txt</i>

La boîte à outils



Notes



Les principales propriétés d'un contrôle Label sont :

- **Alignment** : qui règle l'alignement du texte (gauche, centré, droite)
- **BorderStyle** : qui désigne le type de bordure du label
- **Autosize** : permet d'ajuster la taille du Label à son contenu.

Dans une zone de texte, il y a 3 possibilités :

- *Afficher une valeur*
- *Inscrire une valeur*
- *Modifier une valeur ;*

Dans un label, il y'a UNE possibilité :

- *Afficher une valeur*

4. Notions de base (Le `CommandButton`)



Il s'agit du bouton de commande du type `Ok`, `Annuler`, et permet a l'utilisateur de déclencher un événement avec la souris ou le clavier.

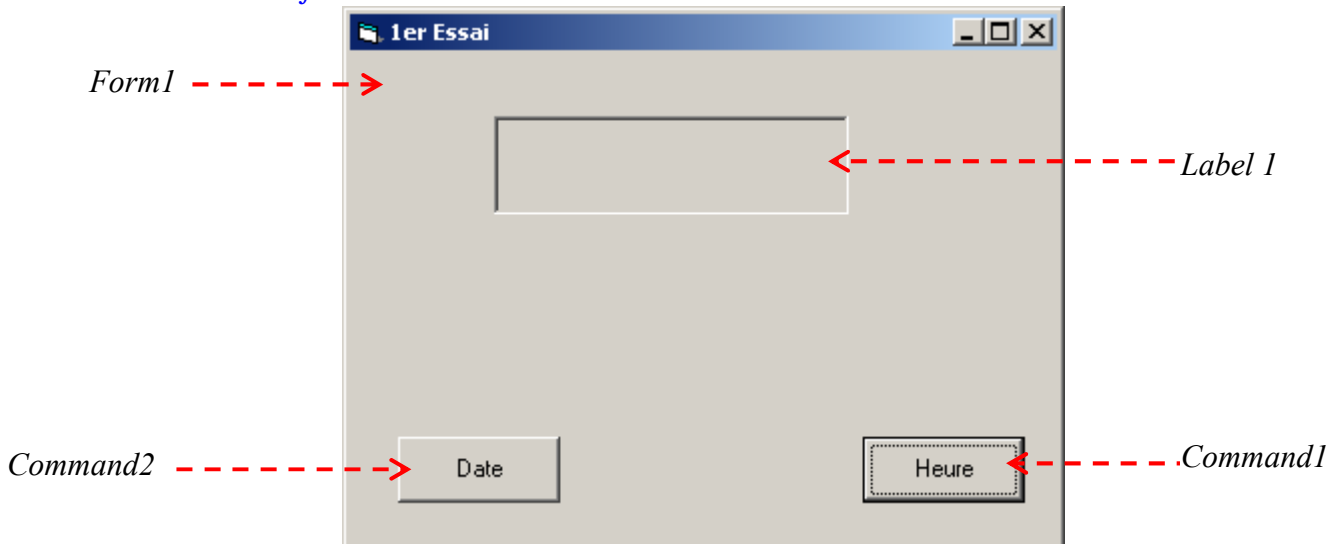
Quelques propriétés de la classe `CommandButton` :

- **Enabled** : cette propriété permet au bouton d'être actif (valeur **True**), c'est-à-dire de pouvoir recevoir des événements, et donc de déclencher des procédures. Inversement, elle interdit (valeur **False**) à un contrôle de recevoir quelque événement que ce soit de la part de l'utilisateur. Dans ce cas, le contrôle apparaît grisé à l'écran.*
- **STYLE** : si cette propriété possède comme valeur **graphical**, il devient alors possible d'ajouter une image sur le bouton, ou d'en modifier la couleur. dans le cas contraire (valeur standard, par défaut), le bouton conserve l'aspect par défaut des boutons de windows (le gris).*

5. Apprentissage par l'exemple

Réalisez votre premier programme qui permet de contrôler l'affichage de la date et l'heure à partir de deux boutons.

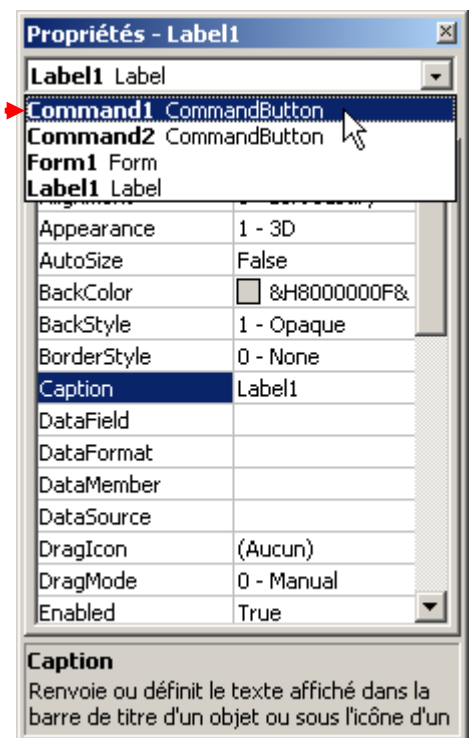
1. Créer l'interface suivante :



2. Définir les objets et leurs propriétés :

L'étape suivante consiste à définir pour chaque objet des **propriétés**.

1. Appuyez sur la touche **F4** pour ouvrir la Fenêtre Propriétés ;
(Si cette fenêtre est déjà ouverte passez à l'étape suivante)
2. Choisissez un objet de la liste :

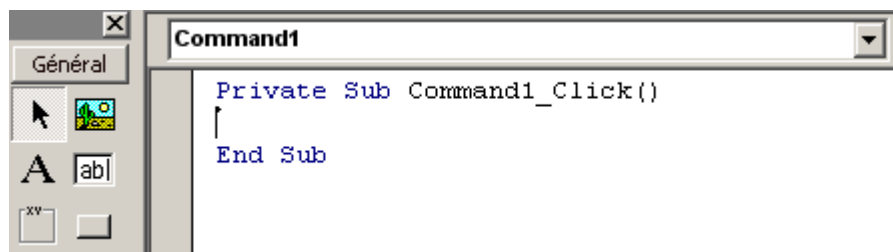


3. Affectez les propriétés suivantes :

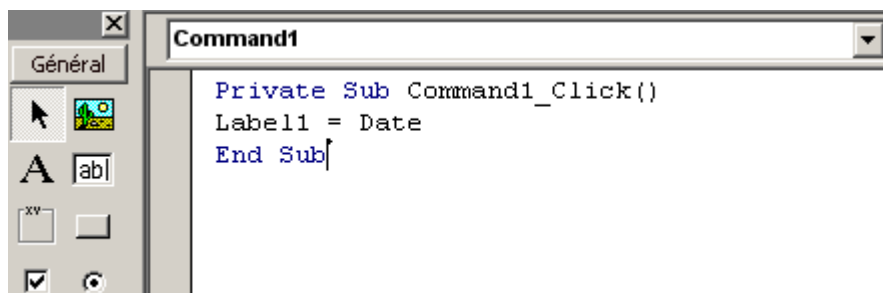
Objets	Propriétés	Valeurs
Form1	Caption	Essai
Command1	Caption	Date
Command2	Caption	Heure
Label1	Caption	
	Alignment	Center
	Border Style	Fixed Single

3. Ecrire le code

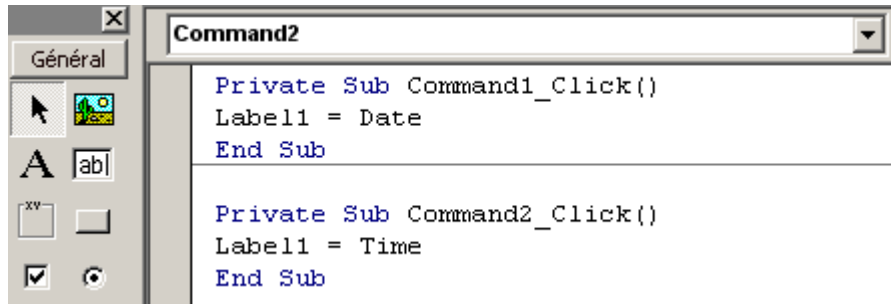
Pour ouvrir la fenêtre du Code, Double-cliquez sur la feuille ou sur le contrôle pour lequel vous souhaitez écrire le code (Dans notre exemple on a choisit le bouton de commande **Command1**).




- **Command1** est le **name** du bouton de commande choisi ;
- **Command1_Click** veut dire que le code qui sera écrit entre **Private Sub** et **End Sub** sera exécuté quand l'utilisateur va cliquer sur ce bouton ;
- **Private** veut dire que ce code sera utilisable **UNIQUEMENT** que dans cette feuille.



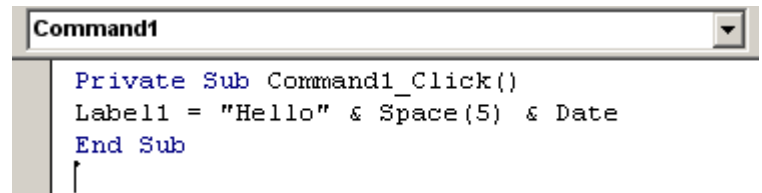
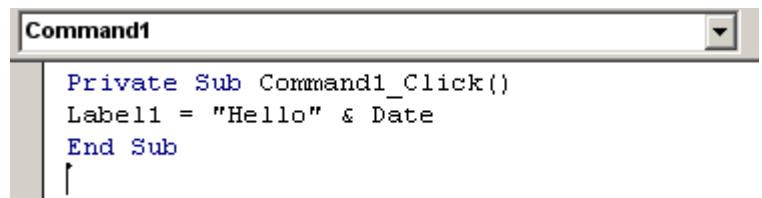
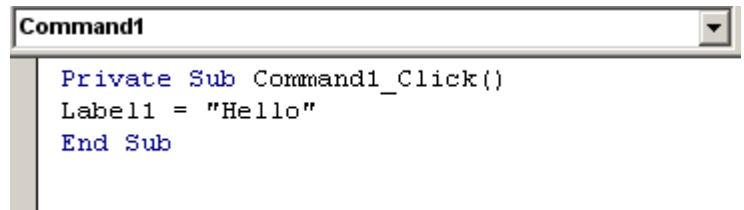
- **Label1=Date** : signifie que l'objet **Label1** va afficher la date système (quand l'utilisateur aura cliqué sur le bouton **Command1**).



Pour exécuter l'application, cliquez sur **Exécuter** dans le menu Exécution, ou sur le bouton  de la barre d'outils, ou encore appuyez sur la touche de fonction **F5**.

D'autres codes à tester:

- Le premier exemple montre qu'au contraire des fonctions Time et Date, pour afficher un texte dans un Label, il doit être placé entre " ;
- Le deuxième exemple montre l'utilité du caractère & : concaténer une fonction et un texte ;
- **Space(5)** permet d'insérer 5 espaces ;



D'autres propriétés valables pour de nombreux éléments :

Propriétés	Signification
BackColor	couleur de l'arrière-plan de l'objet (ex. : couleur de la fenêtre)
ForeColor	couleur du premier plan de l'objet (ex. : couleur du texte)
Font	style d'écriture
Autosize	Permet pour un Label d'adapter le texte à son bordure

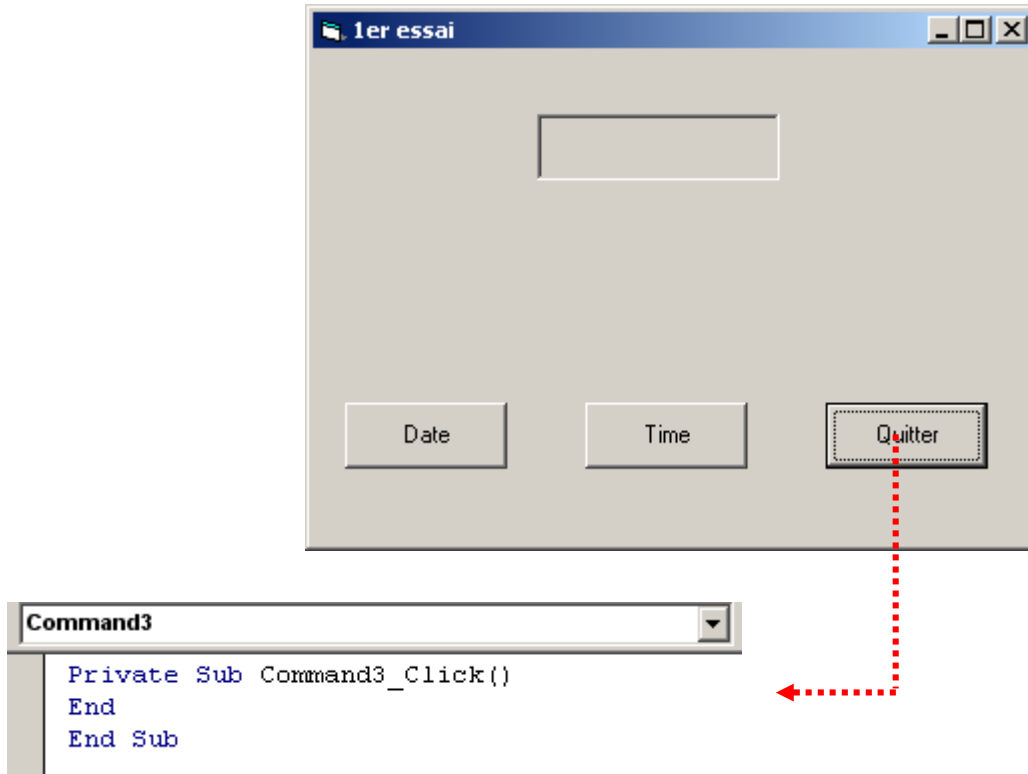
Quelques règles à respecter :

- Un **name** doit toujours commencer par une lettre, ne doit pas dépasser 255 caractères et ne doit pas comporter d'espaces.

- Les mots réservés par Visual Basic (*if, then, time, date...*) ne peuvent pas être utilisés pour la propriété name.

Ajout du bouton 'Quit' :

Objectif : ajouter un bouton de command qui permet de terminer l'application :



- Ce code permet d'arrêter la procédure du programme.

Ajout des touches de raccourcis :

Pour chaque bouton, ajouter un "&" pour que la première lettre devienne un raccourci ;

Exemple : pour le bouton 'Quit'

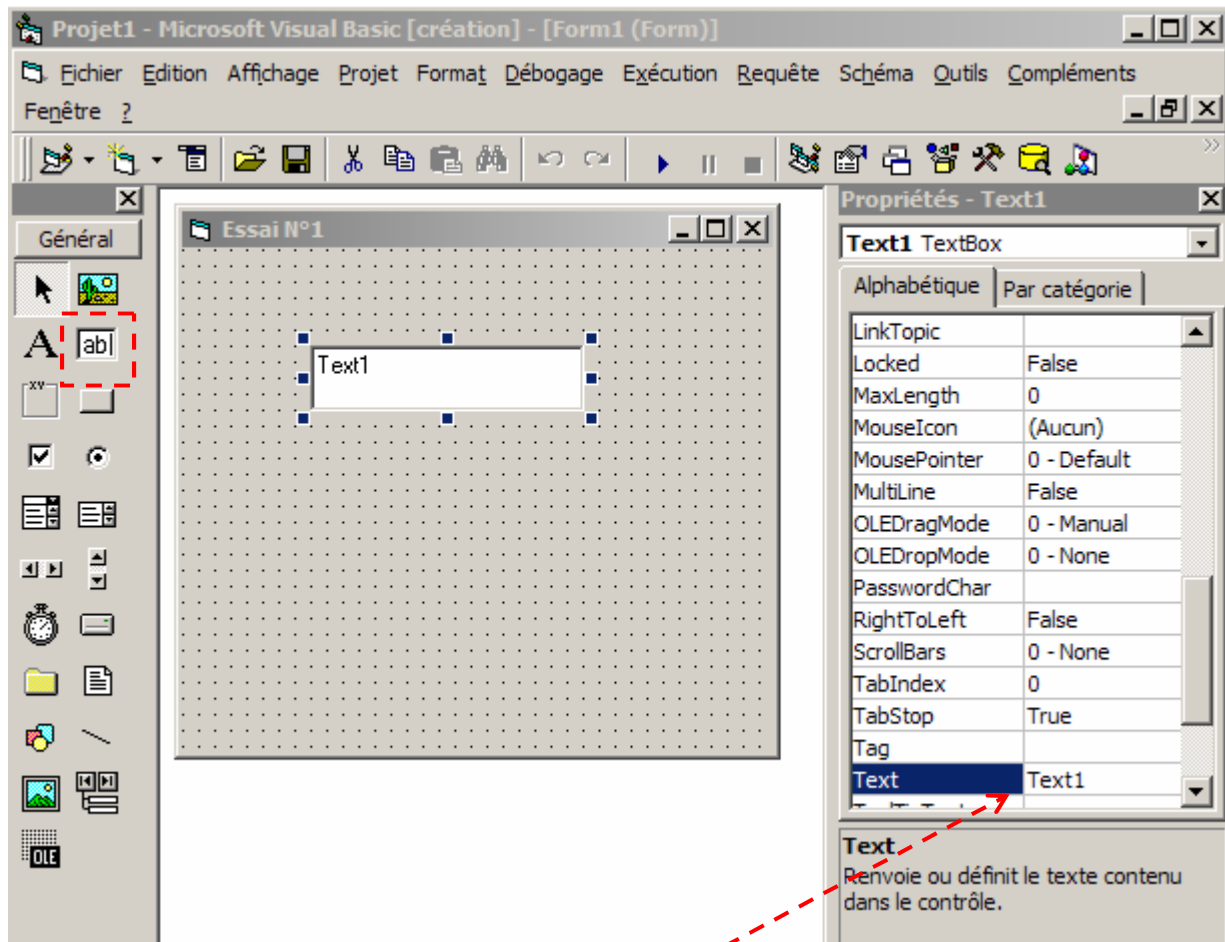


Ainsi en mode execution, il suffit d'utiliser la combinaison de touches **'ALT+Q'** pour terminer l'application.

6. Notions de base (La Zone de Texte (TextBox))



Contrairement à un Label, les "TextBox" servent généralement à saisir une information.

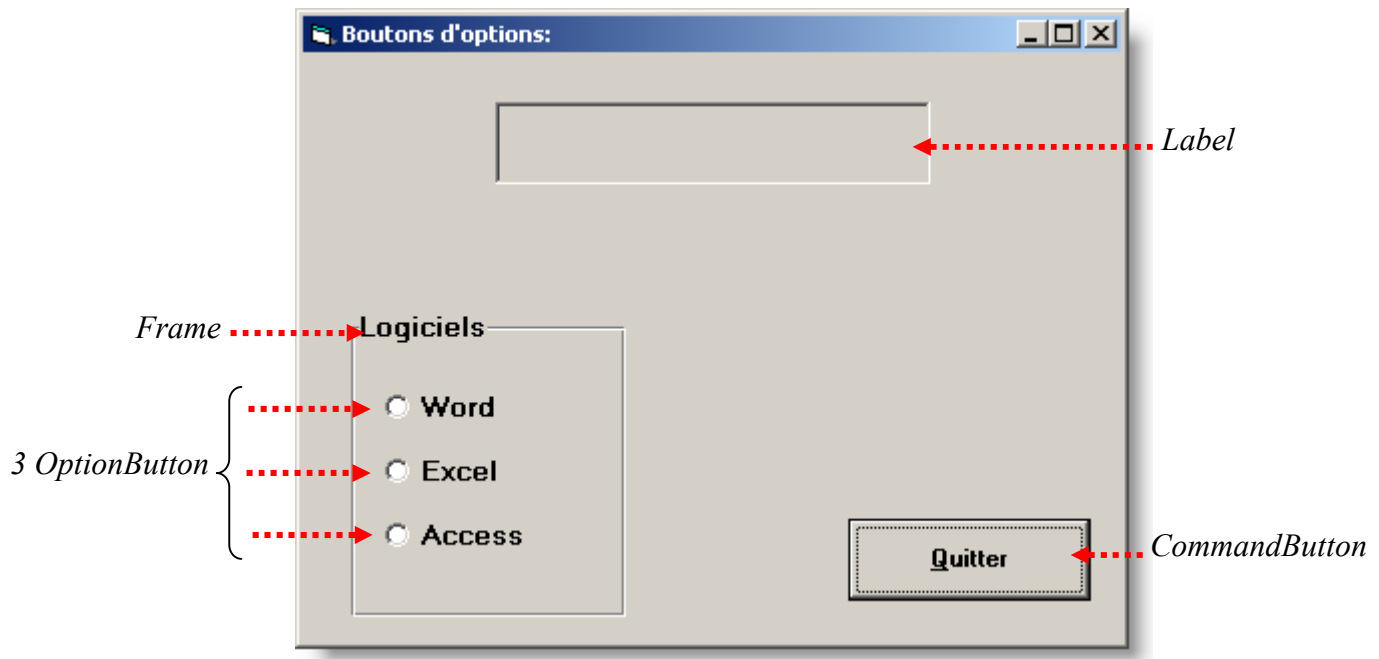


La propriété essentielle d'une Zone de Texte est **Text**. C'est la propriété qui désigne son contenu.

Il faut savoir que **toute information contenue dans une zone de texte est obligatoirement de type texte !** Donc, l'emploi de fonctions de conversion s'avèrera indispensable pour travailler avec des nombres.

Apprentissage par l'exemple :

1. Créez l'interface suivante :



Le but de cette application est de contrôler l'affichage dans Le **Label** en fonction du choix effectué :

- L'option Word pour afficher « Traitement de texte »
- L'option Excel pour obtenir « Tableur »
- L'option Access pour obtenir « Système Gestion bases de donnée »

Définir les propriétés :

Objets	Propriétés	Valeur de chaque propriété
Form 1	Caption	Boutons d'options:
Frame	Caption	Logiciels
	Font	Gras
Option 1	Caption	Word
	Name	optword
Option 2	Caption	Excel
	Name	Optexcel
Option 3	Caption	Access
	Name	Optaccess
Command 1	Caption	&Quitter
	Name	Cmdfin
Label 1	Caption	
	Name	lblaffichage
	Alignment	Center
	Border Style	Fixed Single
	Font	Gras, Taille 12



Ecrire le code pour *optword* :

```
Optword  
  
Private Sub Optword_Click()  
    lblaffichage = "Traitement de texte"  
End Sub  
|
```



Ecrire le code pour *optexcel* :

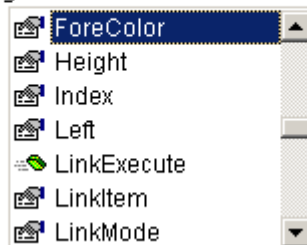
```
Optexcel  
  
Private Sub Optexcel_Click()  
    lblaffichage = "Tableur"  
End Sub
```



Ecrire le code pour *optaccess* :

```
optaccess  
  
Private Sub optaccess_Click()  
    lblaffichage = "SGBD"  
End Sub
```

```
Private Sub Optword_Click()  
    lblaffichage = "Traitement de texte"  
    lblaffichage.for  
End Sub
```



2ème exemple

On désire offrir à l'utilisateur de faire un choix entre trois possibilités :

- L'option **Word** pour afficher « Traitement de texte »
- L'option **Excel** pour obtenir « Tableur »
- L'option **Access** pour obtenir « Système Gestion bases de donnée »

Également, vous devez ajouter à votre application un bouton **Fin** pour quitter.



2. Définir les objets :

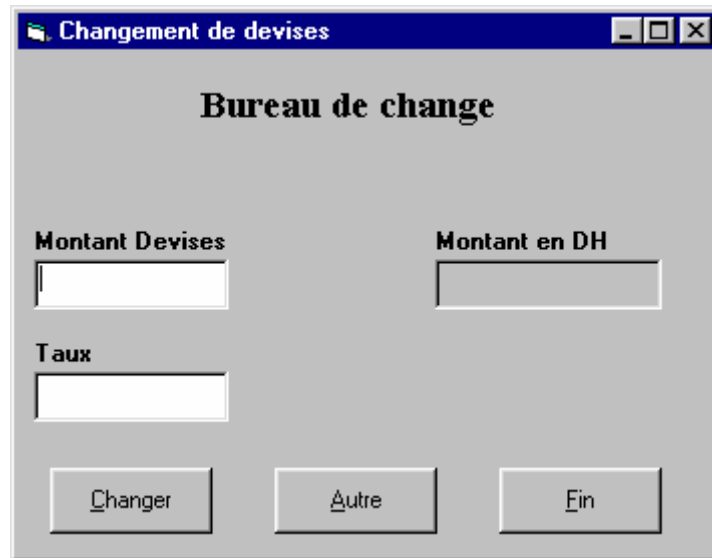
Objets	Propriétés	Valeurs de chaque propriété
Form1	Caption	Logiciels
Frame	Caption Font	Logiciels Gras
Option1	Caption Name	Word optword
Option2	Caption Name	Excel Optexcel
Option3	Caption Name	Access Optaccess
Command1	Caption Name	Fin Cmdfin
Label1	Caption Name Alignment Border Style Font	lblaffichage Center Fixed Single Gras, Taille 12

3. Définir les évènements

```
Private Sub optword_Click()
    lblaffichage.Caption = "Traitement de texte"
End Sub
Private Sub optexcel_Click()
    lblaffichage.Caption = "Tableur"
End Sub
Private Sub optaccess_Click()
    lblaffichage.Caption = "Système Gestion bases de données"
End Sub
Private Sub cmdfin_Click()
    End
End Sub
```

3ème exemple

Créer un projet qui permet de calculer la conversion d'une devise vers le dirham à partir d'un taux choisis par l'utilisateur.



1. Définir les objets :

Objets	Propriétés	Valeurs de chaque propriété
Form1	Caption	Changement de devises
Label1	Caption	Bureau de change
Label2	Caption Autosize	Montant Devises True
Label3	Caption Autosize	Taux True
Label4	Caption Autosize	Montant en DH True
Label5	Caption Name BorderStyle Font	Lblaffichage Fixedsingle Gras, Taille : 12
Text1	Text Name	Txtmontant
Text2	Text Name	Txtaux
Command1	Caption Name	&Changer cmdchanger
Command2	Caption Name	&Autre cmdautres
Command3	Caption Name	&Fin Cmdfin

NB :

La propriété **Autosize** permet d'ajuster le contrôle à son contenu ;

Les propriétés **Font** et **ForeColor** permettent de changer l'attribut de l'objet (la taille, la couleur...)

Le caractère **&** permet d'affecter une touche de raccourcis à un Bouton de commande.

La propriété **setfocus** permet d'insérer automatiquement le curseur dans une zone de texte (voir la procédure *cmdAutre*).

L'instruction **val** permet de convertir une chaîne de caractère en nombre (voir la procédure *cmdchanger*)

2. Définir les évènements


```
Private Sub cmdchanger_Click()  
    lblaffichage.Caption = Val(txtmontant.Text) * Val(txtaux.Text)  
End Sub  
  
Private Sub cmdautre_Click()  
    lblaffichage.Caption = ""  
    txtmontant.Text = ""  
    txtaux.Text = ""  
    txtmontant.SetFocus  
End Sub  
  
Private Sub cmdfin_Click()  
    End  
End Sub
```

4ème exemple

Bureau de change

Devises

☐ Euro

☐ \$Can

☐ \$Us

☐ £

Montant en devises

taux

Montant en dirhams

Changer

Annuler

Fermer

Chaque option donne un
taux :
1 Euro =
1 \$Can =
1 \$Usa =
1 £ =

1ère méthode

```
Private Sub cmdchanger_Click()  
    lblmontant.Caption = lbltaux.Caption * Val(txtmontant.Text)  
End Sub  
Private Sub optcan_Click()  
    lbltaux.Caption = 7.18  
End Sub  
Private Sub optfb_Click()  
    lbltaux.Caption = 0.268  
End Sub  
Private Sub optfr_Click()  
    lbltaux.Caption = 1.56  
End Sub  
Private Sub optusa_Click()  
    lbltaux.Caption = 10.89  
End Sub
```

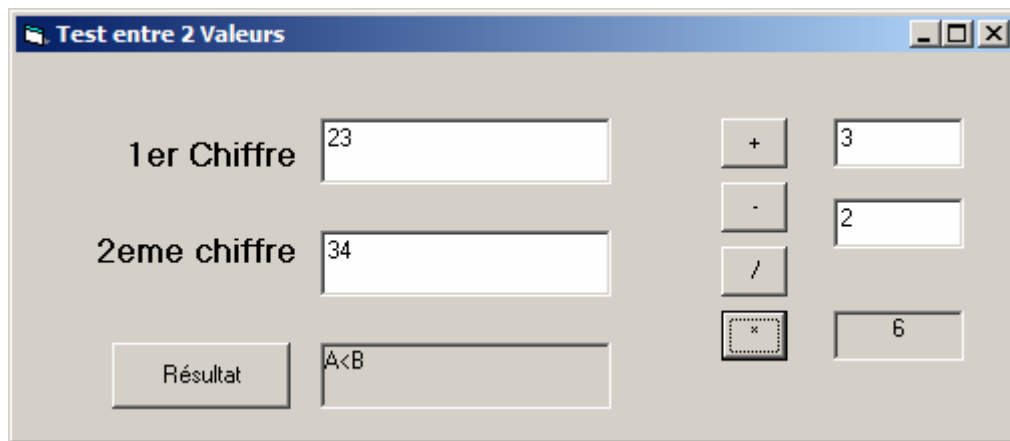
NB : Utiliser le point comme **séparateur décimal** dans le code.

2ème méthode

```
Dim taux as integer  
Private Sub cmdchanger_Click()  
    lblmontant.Caption = Val(txtmontant.Text) * taux  
End Sub  
Private Sub optcan_Click()  
    lbltaux.Caption = 7.18  
    taux = 7.18  
End Sub  
Private Sub optfb_Click()  
    lbltaux.Caption = 0.268  
    taux = 0.268  
End Sub  
Private Sub optfr_Click()  
    lbltaux.Caption = 1.56  
    taux = 1.56  
End Sub  
Private Sub optusa_Click()  
    lbltaux.Caption = 10.89  
    taux = 10.89  
End Sub
```

Cette méthode vous oblige à déclarer la variable **taux** comme variable général (Voir Page N°.....)

Test entre 2 valeurs



```
Private Sub cmddivis_Click()  
    If Val(Text4.Text) = 0 Then  
        MsgBox "Division impossible par Zéro"  
        Text4.SelStart = 0  
        Text4.SelLength = Len(Text4.Text)  
    Else  
        lblresultat.Caption = Val(Text3.Text) / Val(Text4.Text)  
        lblresultat.Caption = Format(lblresultat.Caption, "00.00")  
    End If  
End Sub
```

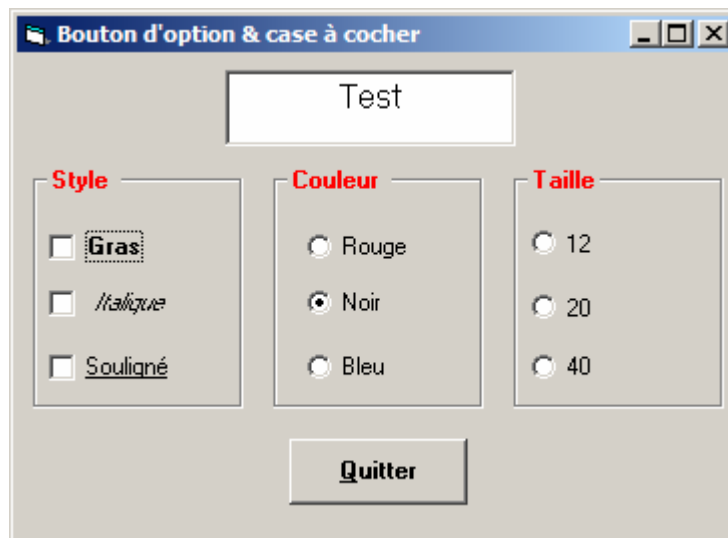
```
Private Sub cmdmoins_Click()  
    lblresultat.Caption = Val(Text3.Text) - Val(Text4.Text)  
End Sub
```

```
Private Sub cmdmult_Click()  
    lblresultat.Caption = Val(Text3.Text) * Val(Text4.Text)  
End Sub
```

```
Private Sub CMDPLUS_Click()  
    lblresultat.Caption = Val(Text3.Text) + Val(Text4.Text)  
End Sub
```

```
Private Sub Command1_Click()  
    If Val(Text1.Text) > Val(Text2.Text) Then  
        lblres.Caption = "A>B"  
    ElseIf Val(Text1.Text) = Val(Text2.Text) Then  
        lblres.Caption = "A=B"  
    Else  
        lblres.Caption = "A<B"  
    End If  
End Sub
```

Application N°4

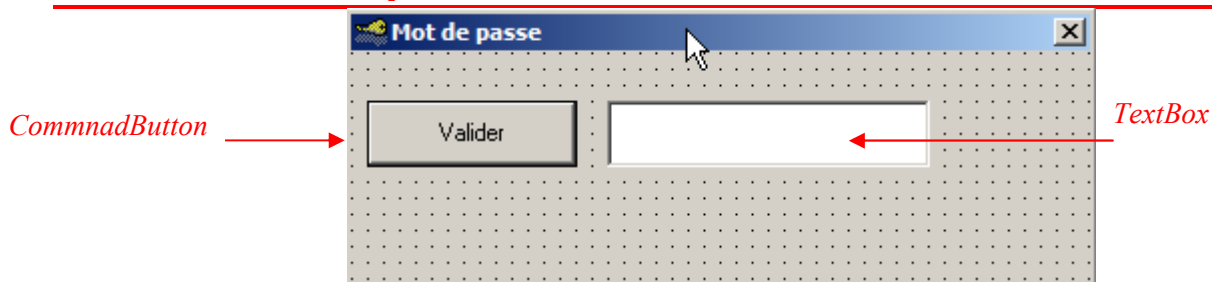


```
Private Sub chkgras_Click()  
    If (chkgras.Value = vbChecked) Then  
        Text1.Font.Bold = True  
    Else  
        Text1.Font.Bold = False  
    End If  
End Sub  
  
Private Sub chkitalique_Click()  
    If (chkitalique.Value = vbChecked) Then  
        Text1.Font.Italic = True  
    Else  
        Text1.Font.Italic = False  
    End If  
End Sub  
  
Private Sub chksou_Click()  
    If (chksou.Value = vbChecked) Then  
        Text1.Font.Underline = True  
    Else  
        Text1.Font.Underline = False  
    End If  
End Sub
```

```
Private Sub opt12_Click()  
    Text1.FontSize = 12  
End Sub  
  
Private Sub opt20_Click()  
    Text1.FontSize = 20  
End Sub  
  
Private Sub opt40_Click()  
    Text1.FontSize = 40  
End Sub
```

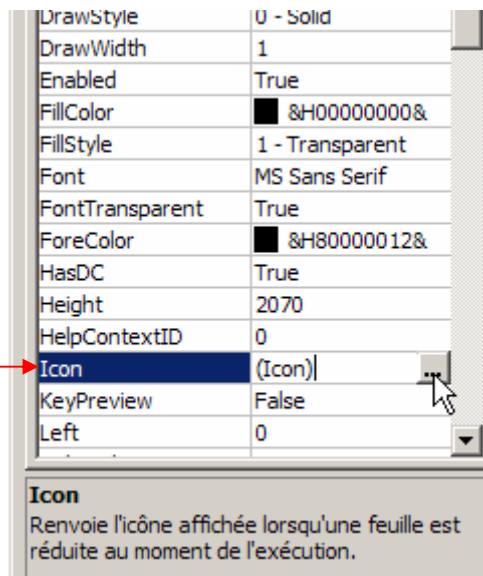
```
Private Sub optblue_Click()  
    Text1.ForeColor = vbBlue  
End Sub  
  
Private Sub optnoir_Click()  
    Text1.ForeColor = vbBlack  
End Sub  
  
Private Sub optrouge_Click()  
    Text1.ForeColor = vbRed  
End Sub
```

Gestion de mot de passe

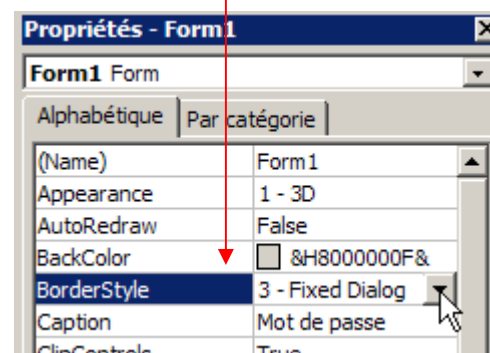


① Modifiez l'icône de l'application :

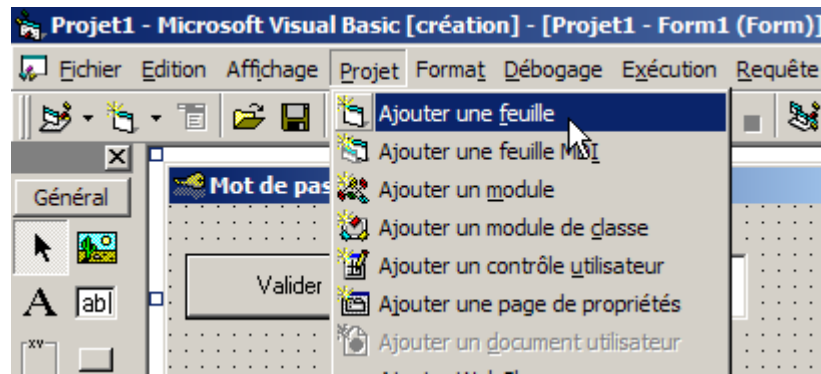
Des icônes supplémentaires sont installés avec Visual Basic dans le dossier : `C:\Program Files\Microsoft Visual Studio\COMMON\Graphics\Icons`.



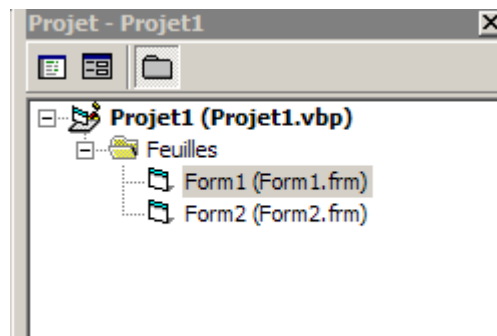
② Désactivez l'affichage des boutons   :



③ Ajoutez une deuxième feuille à votre projet :



Normalement, dans, *l'Explorateur de Projets* (Ctrl + r) vous devez avoir un seul projet contenant deux feuilles :



4 Ajoutez les propriétés suivantes :

Objets	Propriété	Valeur
Command1	Name	cmdvalid
	Caption	Valider
Text1	Name	txtpassword
	Text	<i>Cette propriété doit rester vide</i>
	PasswordChar	*
	Maxlenght	10

NB :

- La propriété PasswordChar permet de masquer les caractères tapés par l'utilisateur par celui défini (Dans notre cas la zone de texte n'affichera que des étoiles).
- La propriété Maxlenght impose une longueur maximale pour les caractères saisis.

5 : Construisez le code de votre application

```

Private Sub cmdValid_Click()
    If txtpassword.Text = "" Then
        MsgBox "Veuillez saisir un mot de passe", vbOKOnly, "Réessayer"
    ElseIf UCase(txtpassword.Text) = "BEST" Then
        Form2.Show
        Unload Form1
    ElseIf MsgBox("Mot de passe Incorrect", vbRetryCancel + vbCritical, "Accès refusé") = vbRetry Then
        txtpassword.SetFocus
        txtpassword.SelStart = 0
        txtpassword.SelLength = Len(txtpassword.Text)
    Else
        End
    End If
End Sub

```



La propriété UCase permet de convertir toutes les lettres minuscules en majuscules.

Application N°8 : Les variables

Réalisez le code de la facture suivante :

Facture

le:

Reference:

Designation:

Prix unitaire: Quantité:

T.V.A. ☐ 7% ☐ 14% ☐ 20%

Remise ☐ 0% ☐ 5% ☐ 10%

MT HT Mt Remise Net Com MT TVA

MT TTC

Calculer Autres Fermer

Règles commerciales :

Le montant Hors taxe = quantité * Prix

Le montant de la remise = Le montant Hors taxe * Taux de remise

Le net commercial = Le montant Hors taxe - Le montant de la remise

Le montant TVA = Le net commercial * taux TVA

Le montant TTC = Le net commercial + Le montant TVA

Dim remise, tva

Private Sub calculer_Click()

labmht = txtQ * txtprixuni

labmtremise = labmht * remise

labnetcom = labmht - labmtremise

labmttva = labnetcom * tva

labmttc = labnetcom + labmttva

End Sub

Private Sub tauxR0_Click()

remise = 0

End Sub

Private Sub tauxR10_Click()

remise = 0.1

End Sub

Private Sub tauxR5_Click()

remise = 0.05

End Sub

Private Sub taux TVA14_Click()

tva = 0.14

End Sub

Private Sub taux TVA20_Click()

tva = 0.2

End Sub

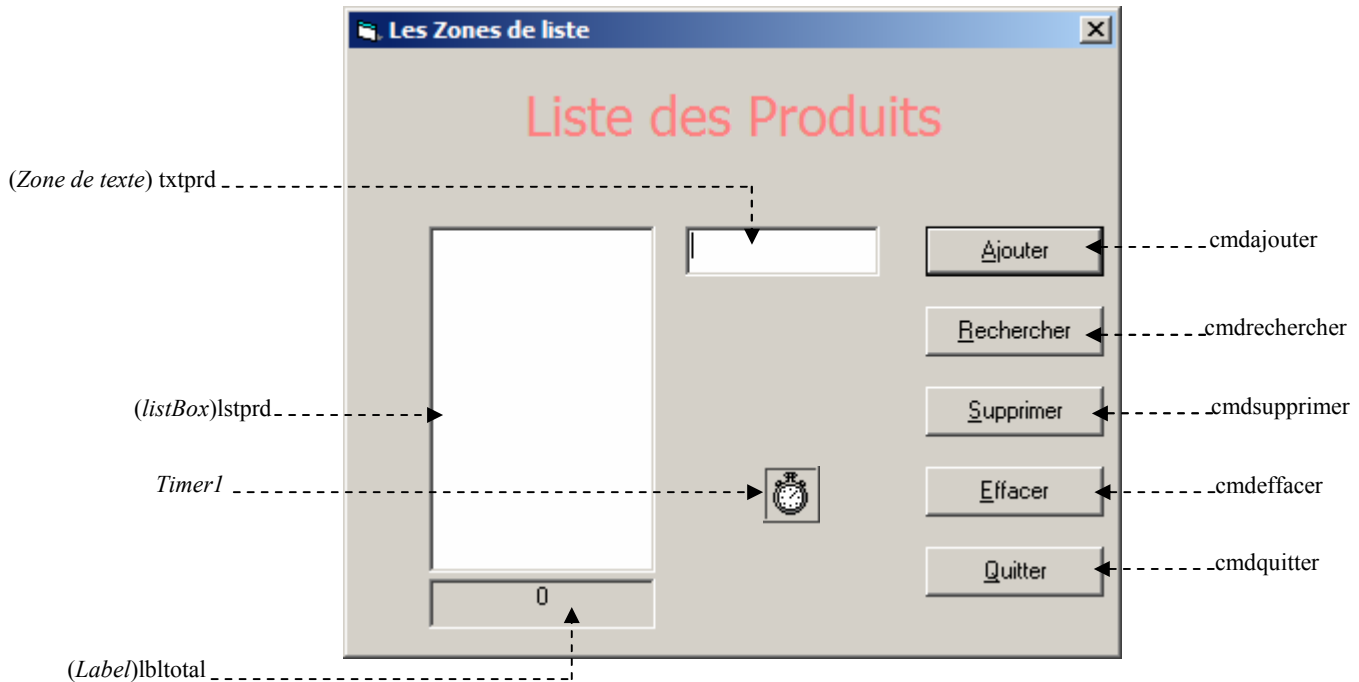
Private Sub taux TVA7_Click()

tva = 0.07

End Sub

5ème exemple (Les ListBox)

Etablir un programme qui permet de contrôler l'ajout vers une zone de Liste



```
Private Sub cmdajout_Click()  
Dim i, nbv As Integer  
nbv = lstprd.ListCount  
If nbv > 0 Then  
    For i = 0 To nbv - 1  
        lstprd.Selected(i) = True  
        If UCase(lstprd.Text) = UCase(txtprd.Text) Then  
            MsgBox "Produits existe déjà !!!", 48, "attention"  
            txtprd.Text = ""  
            txtprd.SetFocus  
            Exit Sub  
        End If  
    Next  
End If  
lstprd.AddItem Trim(txtprd.Text)  
txtprd.Text = ""  
txtprd.SetFocus  
End Sub
```

```
Private Sub cmdeffacer_Click()  
    If MsgBox("Etes vous sur de vouloir tout Effacer ", vbYesNo,"Confirmer")= _  
vbYes Then  
        lstprd.Clear  
        End If  
        txtprd.SetFocus  
End Sub
```

```
Private Sub cmdsupprimer_Click()  
    If lstprd.ListCount = 0 Then  
        MsgBox "la list est vide", vbOKOnly, "Supprimer"  
    ElseIf lstprd.ListIndex = -1 Then  
        MsgBox "vous devez selectionner un element de la liste"  
    Else  
        lstprd.RemoveItem lstprd.ListIndex  
    End If  
End Sub
```

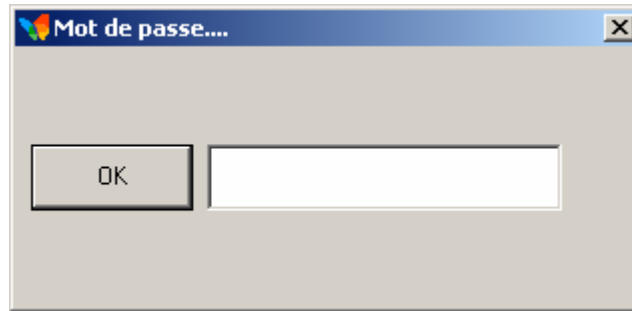
```
Private Sub cmdfin_Click()  
    If MsgBox("Etes vous sur de vouloir fermer", vbYesNo, "Quitter")= vbYes Then  
        End  
    End If  
End Sub
```

```
Private Sub Timer1_Timer()  
    lbltotal.Caption = lstprd.ListCount  
End Sub
```

```
Private Sub txtprd_Change()  
    If Len(Trim(txtprd.Text)) = 0 Then  
        cmdajout.Enabled = False  
    Else  
        cmdajout.Enabled = True  
    End If  
End Sub
```

```
Private Sub cmdrechercher_Click()  
    Dim produit As String  
    Dim i, nbprd As Integer  
    produit = InputBox("Entrez le nom du produit", "Rechercher")  
    nbprd = lstprd.ListCount  
    For i = 0 To nbprd - 1  
        lstprd.Selected(i) = True  
        If UCase(lstprd.Text) = UCase(produit) Then  
            MsgBox "Le produit a été trouvé", 64, "Fin de la recherche"  
            Exit Sub  
        End If  
    Next  
    MsgBox "Produit non trouvé", 64, "Fin de la recherche"  
End Sub
```

6ème exemple (Application mot de passe)



```
Private Sub cmdValid_Click()  
    If txtpassword.Text = "" Then  
        MsgBox "Veuillez saisir un mot de passe", vbOKOnly, "Reéssayer"  
    ElseIf UCase(txtpassword.Text) = "BEST" Then  
        Form2.Show  
        Unload Form1  
    ElseIf MsgBox("Mot de passe Incorrect", vbRetryCancel + vbCritical, "Accès  
refusé") = vbRetry Then  
        txtpassword.SelStart = 0  
        txtpassword.SelLength = Len(txtpassword.Text)  
    Else  
        End  
    End If  
End Sub  
Private Sub Form_Activate()  
    txtpassword.SetFocus  
End Sub
```

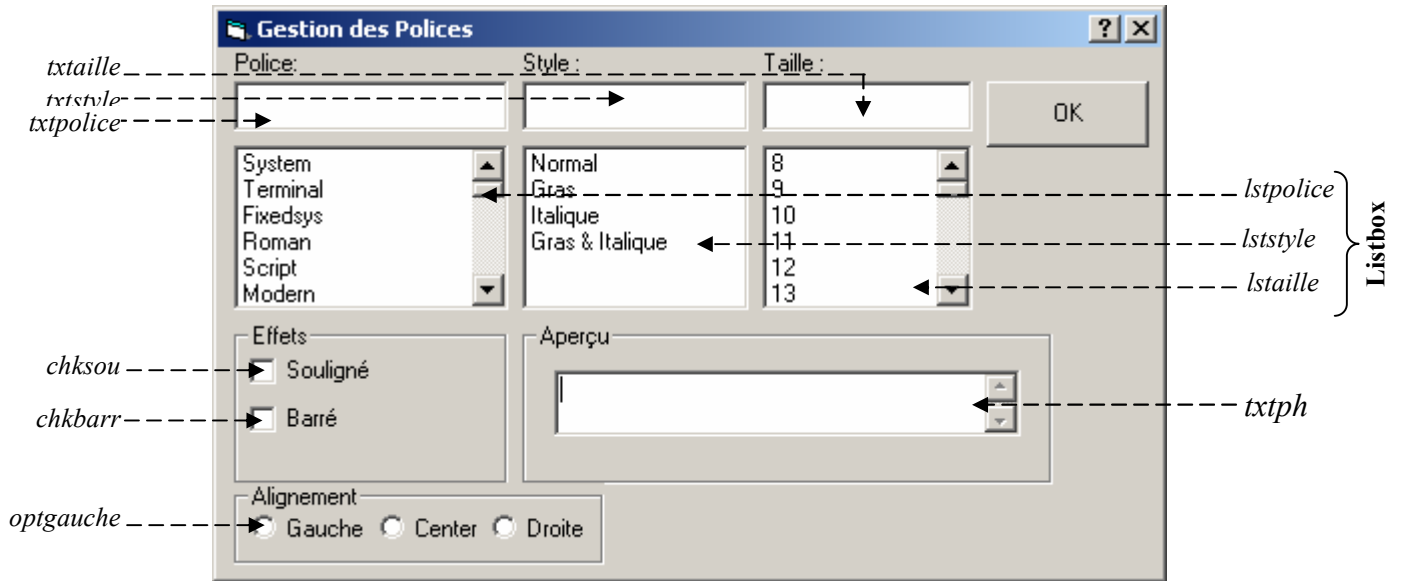
2^{ème} méthode (Limiter le nombre d'essai a 3)

```
Dim i  
Private Sub Command1_Click()  
    Do While i < 3  
        If txtpassword.Text <> "123" Then  
            i = i + 1  
            txtpassword.SetFocus  
            txtpassword.Text = ""  
            MsgBox "essai N° " & i  
            Exit Sub  
        Else  
            MDIForm1.Show  
            pass.Hide  
            Exit Sub  
        End If  
    Loop  
    MsgBox "Dernier essai .....!!!,"  
    MsgBox "Au revoir"  
End  
End Sub
```



```
Private Sub cmdresultat_Click()  
If optoperation(0).Value Then  
    lblresultat.Caption = Val(txttop1) + Val(txttop2)  
ElseIf optoperation(1).Value Then  
    lblresultat.Caption = Val(txttop1) - Val(txttop2)  
ElseIf optoperation(2).Value Then  
    lblresultat.Caption = txttop1 * txttop2  
ElseIf txttop2.Text = 0 Then  
    MsgBox "Division par 0 impossible", 16, "Erreur"  
    lblresultat.Caption = ""  
Else  
    lblresultat.Caption = txttop1.Text / txttop2.Text  
End If  
End Sub  
Private Function Controle_saisie(KeyAscii As Integer) As Integer  
If (KeyAscii < 48 Or KeyAscii > 57) And KeyAscii <> 8 And KeyAscii <> 46 Then  
    Controle_saisie = 0  
Else  
    Controle_saisie = KeyAscii  
End If  
End Function  
  
Private Sub txttop1_KeyPress(KeyAscii As Integer)  
KeyAscii = Controle_saisie(KeyAscii)  
End Sub  
  
Private Sub txttop2_KeyPress(KeyAscii As Integer)  
KeyAscii = Controle_saisie(KeyAscii)  
  
End Sub
```

7ème exemple (Gestion des polices)



```

Private Sub chkbarre_Click()
    If txtph.Font.Strikethrough = True Then
        txtph.Font.Strikethrough = False
    Else
        txtph.Font.Strikethrough = True
    End If
End Sub

Private Sub chksou_Click()
    If txtph.Font.Underline = True Then
        txtph.Font.Underline = False
    Else
        txtph.Font.Underline = True
    End If
End Sub

Private Sub cmdclose_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    Dim i, j
    For i = 0 To Screen.FontCount - 1
        lstpolice.AddItem Screen.Fonts(i)
    Next
    For j = 8 To 100
        lstaille.AddItem j
    Next
End Sub

Private Sub lstaille_Click()
    txtph.Font.Size = Val(lstaille.Text)
    txttaille.Text = lstaille.Text
End Sub

Private Sub lstpolice_Click()
    txtpolice.Text = lstpolice.Text
    txtph.Font.Name = lstpolice.Text
End Sub

Private Sub lststyle_Click()
    Dim style
    style = lststyle.ListIndex
    Select Case style
    Case 0
        txtph.Font.Bold = False

```

```

        txtph.Font.Italic = False
    Case 1
        txtph.Font.Bold = True
        txtph.Font.Italic = False
    Case 2
        txtph.Font.Bold = False
        txtph.Font.Italic = True
    Case Else
        txtph.Font.Bold = True
        txtph.Font.Italic = True
    End Select
    txtstyle.Text = lststyle.Text
End Sub

Private Sub optcenter_Click()
txtph.Alignment = vbCenter
End Sub

Private Sub optdroite_Click()
txtph.Alignment = vbRightJustify
End Sub

Private Sub optgauche_Click()
txtph.Alignment = vbLeftJustify
End Sub

```

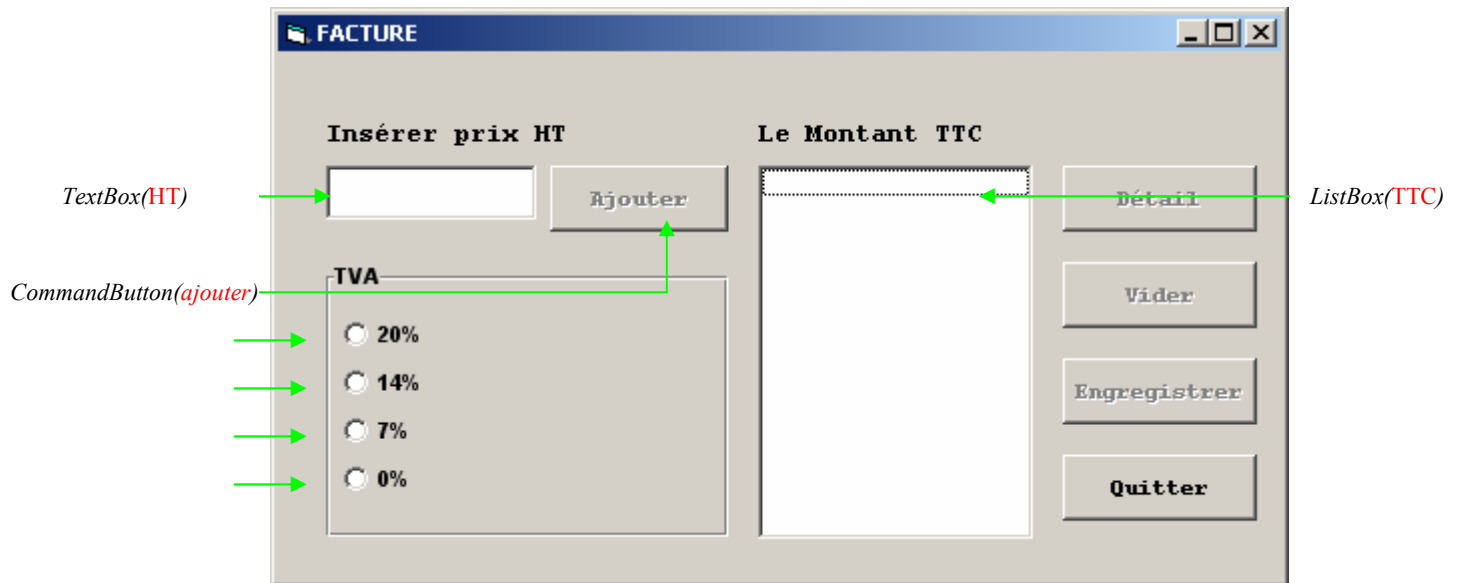
NB:

Affectez à la zone de texte txtph les propriétés suivantes :

Multiline =True

Scrollbars = 2-Vertical

Facture pour comptabiliser



```

Private Sub ajouter_Click()
    If ht.Text = "" Then
        MsgBox ("Attention, vous devrez insérer un prix")
    Else
        initialiser.Enabled = True 'activer le bouton Initialiser
        If tva0.Value = True Then
            detail.Enabled = True
            TTC.AddItem (Str(Val(ht.Text)))
        ElseIf tva7.Value = True Then
            detail.Enabled = True
            TTC.AddItem (Str(Val(ht.Text) * 1.07))
        ElseIf tva20.Value = True Then
            detail.Enabled = True
            TTC.AddItem (Str(Val(ht.Text) * 1.2))
        ElseIf tva14.Value = True Then
            detail.Enabled = True
            TTC.AddItem (Str(Val(ht.Text) * 1.14))
        Else 'si aucun bouton n'est activé
            MsgBox ("Attention, vous devez choisir le taux de TVA")
        End If
        ht.Text = ""
        ht.SetFocus
    End If
End Sub

```

```

Private Sub detail_Click()
    Dim i As Integer
    Dim sum As Single
    Dim moyenne
    sum = 0
    For i = 0 To TTC.ListCount - 1 'parcourir les éléments de la liste

```



```
sum = sum + Val(TTC.List(i))
Next i
moyenne = sum / TTC.ListCount
MsgBox ("Le nombre d'articles est " & TTC.ListCount & "; la montant total à payer est " & sum)
sauvegarder.Enabled = True
End Sub
```

```
Private Sub ht_Change()
ajouter.Enabled = True
End Sub
```

```
Private Sub initialiser_Click()
TTC.Clear
detail.Enabled = False
sauvegarder.Enabled = False
End Sub
```

```
Private Sub quitter_Click()
End
End Sub
```

```
Private Sub sauvegarder_Click()
Dim i As Integer
Open "c:\facture" For Output As #1
For i = 0 To TTC.ListCount - 1
Print #1, TTC.List(i)
Next i
Close
End Sub
```

Contrôler la saisie dans une zone de texte

1.

```
Private Sub Text1_LostFocus()  
    If IsNumeric(Text1.Text) = False Then  
        MsgBox "Erreur de saisie" & vbCrLf & "Seulement numérique"  
        Text1.SetFocus  
        Text1.Text = ""  
    End If  
End Sub
```

NB :

VbLf Ou Chr (10) : Caractère de saut de ligne

2.

```
Private Sub Text1_LostFocus()  
    If IsDate(Text1.Text) = False Then  
        MsgBox "Erreur de saisie" & vbCrLf & "Seulement Date"  
        Text1.SetFocus  
        Text1.Text = ""  
    End If  
End Sub
```

NB :

La fonction IsDate renvoie la valeur True si l'expression est une date ou peut être reconnue en tant que date ; sinon elle renvoie la valeur False.

3.

```
Private Sub Text1_KeyPress(KeyAscii As Integer)  
    If (KeyAscii < 48 Or KeyAscii > 57) And (KeyAscii <> 8) Then  
        KeyAscii = 0  
    End If  
End Sub
```

NB :

Chaque caractère est associé à un code compris entre 0 et 255 (on appelle code ascii)

Exercice :

Ecrire un programme qui permet de limiter la saisie dans une zone de texte aux touches alphabétiques.

Exécuter une application Windows a partir du VB

Utiliser la fonction Shell...

Exemple :

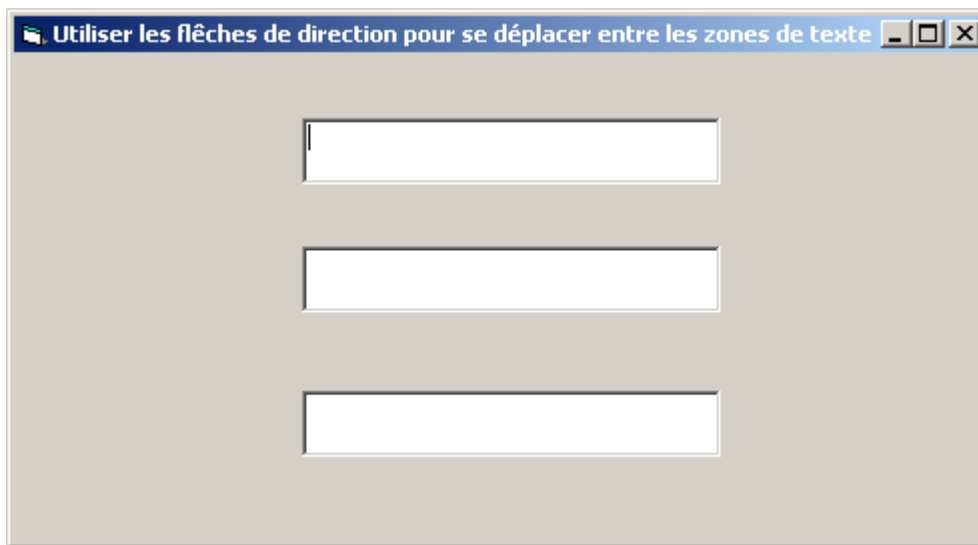
```
Shell "c:\Program files\Microsoft office\office\winword.exe", vbNormalFocus
```

Exercice :

Ecrire un programme qui permet d'ouvrir les programmes suivants :

- Paint
- Calculatrice
- Le Bloc note
- Les jeux de Windows

Se déplacer avec les touches de direction

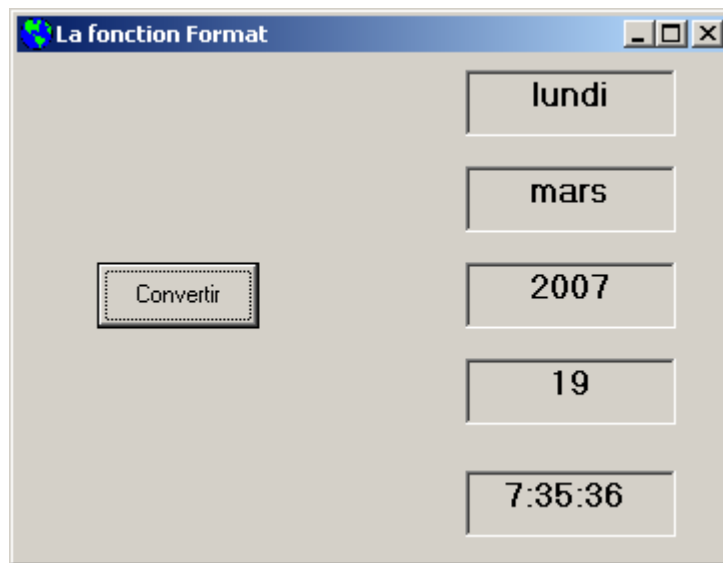


```
Private Sub Text1_KeyDown(KeyCode As Integer, Shift As Integer)  
    If KeyCode = vbKeyDown Then  
        Text2.SetFocus  
    End If  
End Sub
```

```
Private Sub Text2_KeyDown(KeyCode As Integer, Shift As Integer)  
    If KeyCode = vbKeyUp Then  
        Text1.SetFocus  
    Elseif KeyCode = vbKeyDown Then  
        Text3.SetFocus  
    End If  
End Sub
```

```
Private Sub Text3_KeyDown(KeyCode As Integer, Shift As Integer)  
    If KeyCode = vbKeyUp Then  
        Text2.SetFocus  
    End If  
End Sub
```

Utilisation de la fonction Format



Private Sub Command1_Click()

Dim Today As Variant

Today = Now

lblDay.Caption = Format(Today, "dddd")

lblMonth.Caption = Format(Today, "mmm")

lblYear.Caption = Format(Today, "yyyy")

lblnumber.Caption = Format(Today, "d")

lblTime.Caption = Format(Today, "h:mm:ss ampm")

End Sub



09/02/2006 10:19:36

Traitement Salaire

Nombre heure :

Taux :

Salaire :

Une entreprise qui emploi des salariés à l'heure, les paie chaque semaine suivant un taux horaire auquel on applique un coefficient K donnée par :

39 premières heures	$K = 1$
de la 40 ^{ème} à la 44 heure	$K = 1,2$
De la 45 ^{ème} à la 49 heure	$K = 1,5$
Après la 49 ^{ème} heure	$K = 1,8$

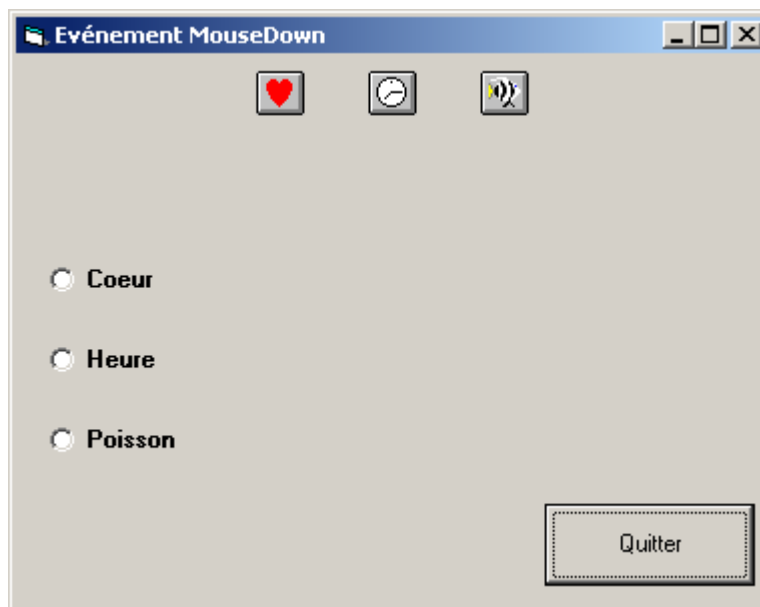
Travail à faire :

Déterminer à partir d'un volume horaire, le salaire de la semaine.

Utilisation de l'événementMouseDown

Exemple 1 :

Ce programme illustre la façon de concevoir des programmes permettant à l'utilisateur de déplacer des objets dans la fenêtre du programme à l'aide de la souris.

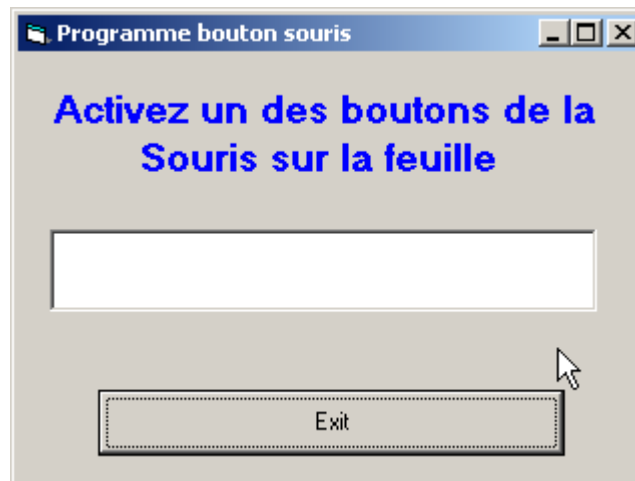


```
Form
MouseDown
Private Sub Command1_Click()
    End
End Sub

Private Sub Form_MouseDown(Button As Integer, Shift As Integer)
    If optcoeur.Value = True Then
        imgcoeur.Move X, Y
    ElseIf optheure.Value = True Then
        imgheure.Move X, Y
    Else
        imgpoisson.Move X, Y
    End If
End Sub
```

Exemple 2 :

Ce programme utilise l'événement MouseDown .



```
Form
Private Sub cmdexit_Click()
    End
End Sub

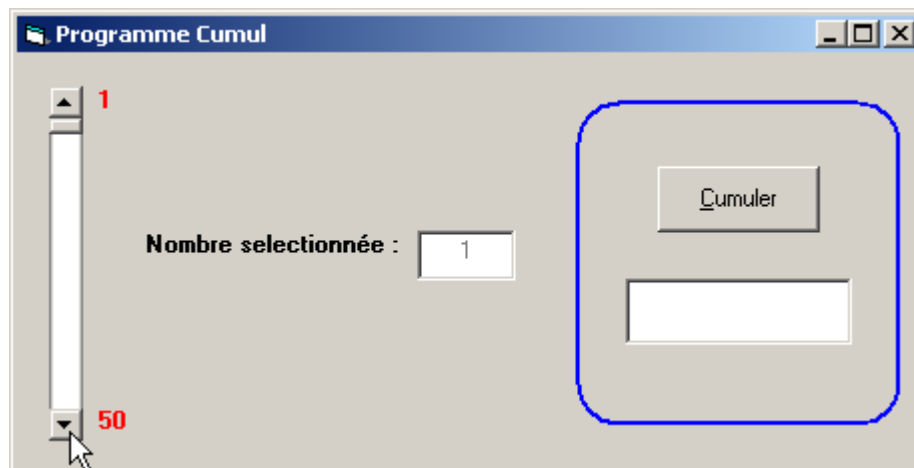
Private Sub Form_MouseDown(Button As Integer, Shift As Integer)
    If Button = 1 Then
        txtdisplay = "Le click GAUCHE a été actif"
    ElseIf Button = 2 Then
        txtdisplay = "Le click DROIT a été est actif"
    End If
End Sub

Private Sub Form_MouseUp(Button As Integer, Shift As Integer)
    txtdisplay = ""
End Sub
```

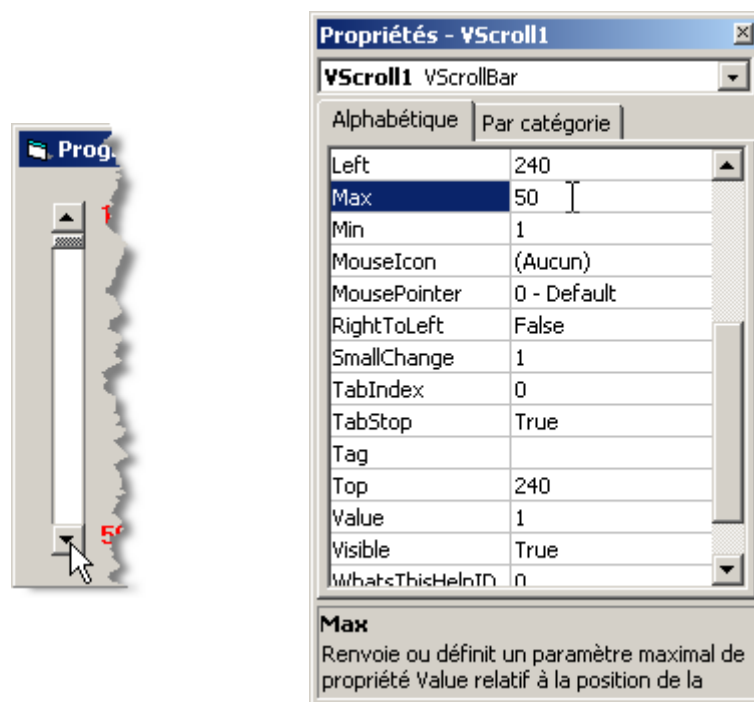


La procédure Form_MouseDown est exécutée dès que pressez le bouton de la souris, avant même que vous ne le relâchiez.

CUMUL de Chiffres 1 + 2 + 3 + ...n



Etape 1




```
cmdcumul

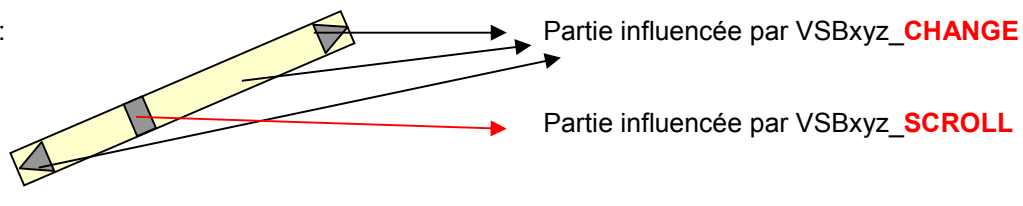
Private Sub cmdcumul_Click()
    Dim I, R
    For I = 1 To VScroll11.Value Step 1
        R = R + I
    Next
    txtresultat.Text = Str(R)
End Sub

Private Sub Form_Load()
    txtnombre.Text = VScroll11.Min
End Sub

Private Sub VScroll11_Change()
    txtnombre.Text = Str(VScroll11.Value)
End Sub

Private Sub VScroll11_Scroll()
    VScroll11_Change
End Sub
```

ScrollBar :



Hôtel Suisse 3*

Hôtel Suisse Facture du: 19/03/2007

Nom & Prénom:

N° Chambre:

Type

☐ Simple

☐ Double

☐ Duplex

De:

A:

Divers:

Nombre de jours:

Montant Du:

Calculer

Autre

Fermer

Dim prix As Integer

```
Private Sub cmdcalculer_Click()  
    Dim nbj, divers, a, de, mtdu  
    a = CDate(txta.Text)  
    de = CDate(txtde.Text)  
    nbj = a - de  
    divers = Val(txtdivers.Text)  
    mtdu = (nbj * prix) + divers  
    lblnbj.Caption = nbj  
    lblmtdu.Caption = mtdu  
End Sub
```

```
Private Sub Form_Load()  
    lbldate.Caption = "Facture du: " & Date  
End Sub
```

```
Private Sub optdouble_Click()  
    prix = 220  
End Sub
```

```
Private Sub optdup_Click()  
    prix = 350  
End Sub
```

```
Private Sub optsimple_Click()
```

```

        prix = 180
End Sub

Private Sub txta_LostFocus()
    If Not IsDate(txta.Text) Then
        txta.Text = ""
        txta.SetFocus
    Else
        If CDate(txta.Text) < CDate(txtde.Text) Then
            MsgBox "La date saisie est inferieur à la date d'entrée", vbExclamation, "Date incorrect"
            txta.Text = ""
            txta.SetFocus
        End If
    End If
End Sub

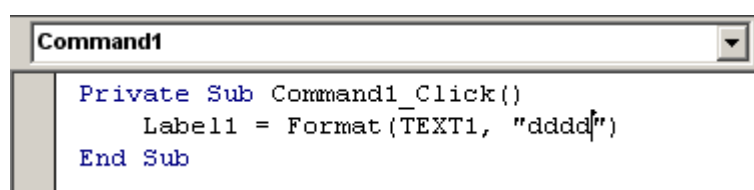
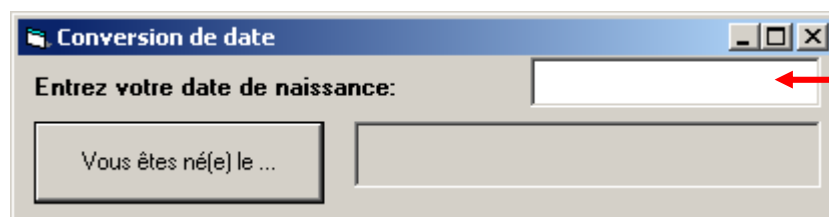
Private Sub txtde_GotFocus()
    Dim debut
    debut = InputBox("Entrez la date debut", "Saisie des données")
    txtde.Text = debut
End Sub

Private Sub txtde_LostFocus()
    If Not IsDate(txtde.Text) Then
        txtde.Text = ""
        txtde.SetFocus
    End If
End Sub

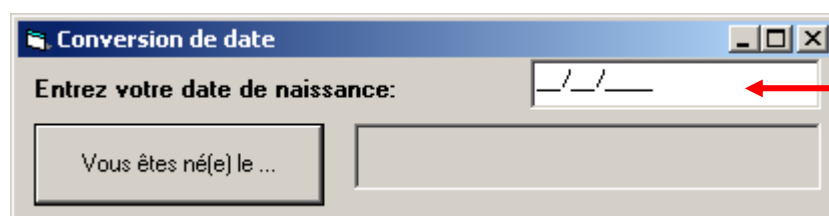
Private Sub txtdivers_KeyPress(KeyAscii As Integer)
    If Not IsNumeric(KeyPress) Then
        txtdivers.Text = ""
        txtdivers.SetFocus
    End If
End Sub

```

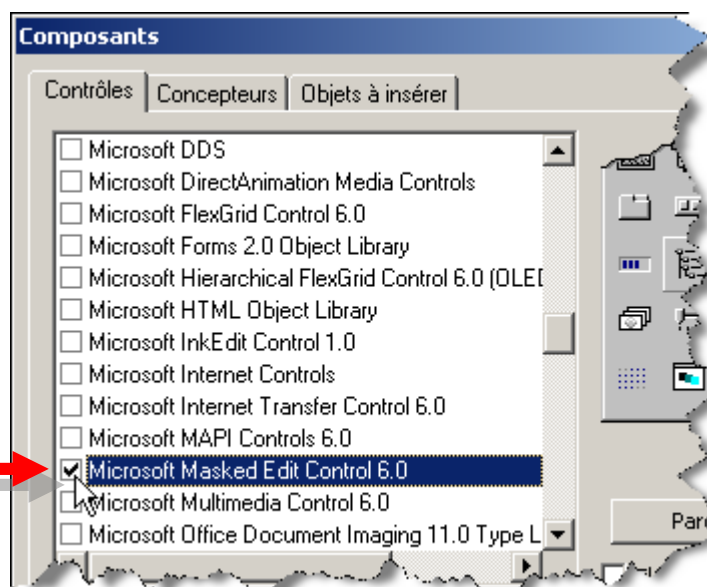
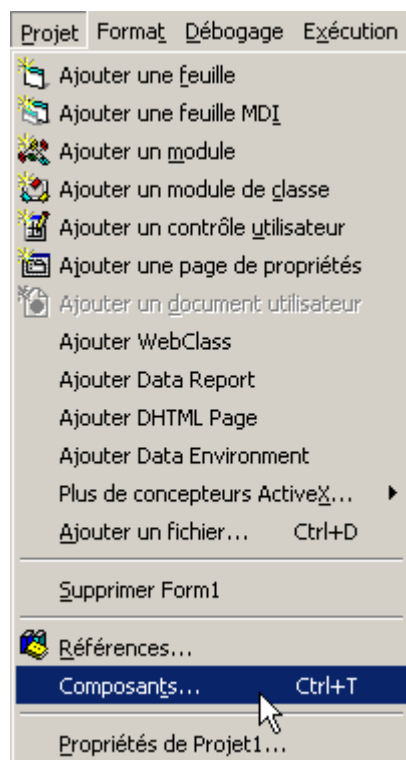
Utiliser le contrôle MaskEdit



On va changer la zone de texte par un nouveau composant MaskEdit



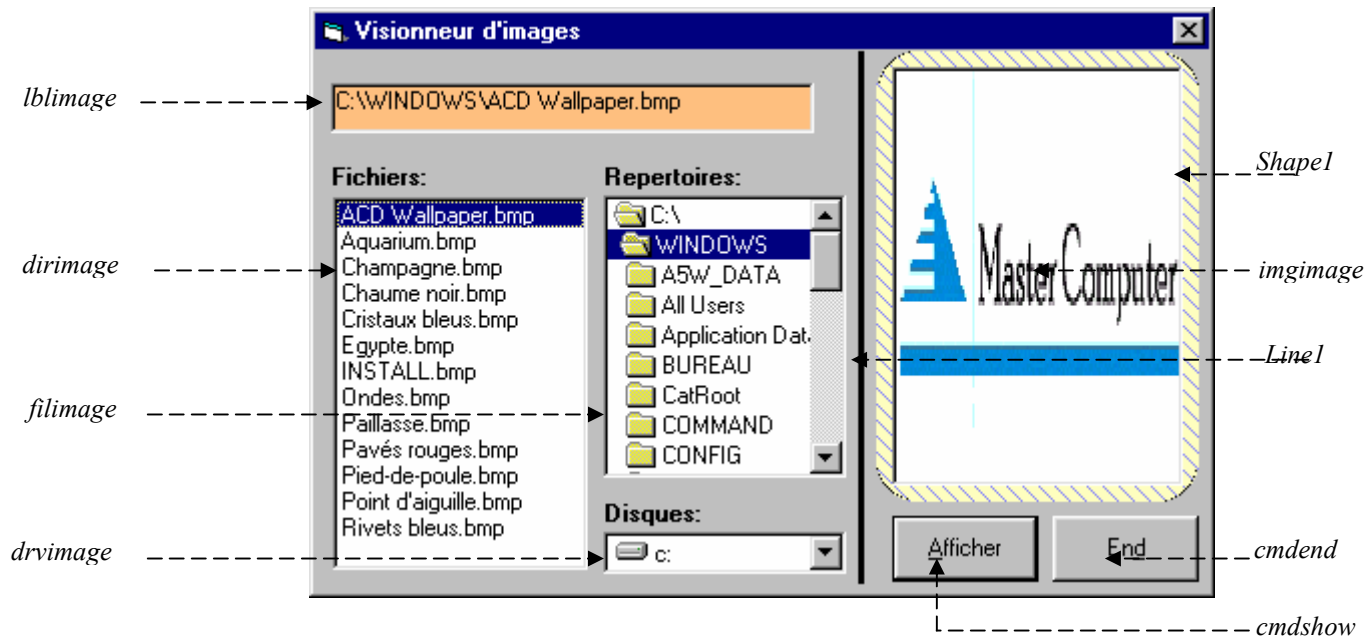
Pour l'insérer



Le contrôle d'édition Masked (MaskedEdit) permet d'inviter les utilisateurs à entrer des données à l'aide d'un modèle de masque.

Vous pouvez également l'utiliser pour demander l'entrée de dates, de devises et d'heure, ou pour convertir des données entrées en majuscules ou minuscules.

9ème exemple en VB(Le visionneur d'image)



```
Private Sub drvImage_Change()
    On Error GoTo testerr
    dirimage.Path = drvimage.Drive
    Exit Sub
testerr:
    MsgBox "Le lecteur n'est pas prêt"
    drvimage.Drive = "c:"
    Resume Next
End Sub

Private Sub dirImage_Change()
    filImage.Path = dirimage.Path
End Sub

Private Sub cmdShow_Click()
    Dim ImageName As String
    If Right(filImage.Path, 1) = "\" Then
        ImageName = filImage.Path + filImage.FileName
    Else
        ImageName = filImage.Path + "\" + filImage.FileName
    End If
    lblimage.Caption = ImageName
    imgimage.Picture = LoadPicture(ImageName)
End Sub

Private Sub filImage_DblClick()
    Call cmdShow_Click
End Sub
```

Shape1:

BackColor :Cyan
 BackStyle :1-Opaque
 FillColor :Blue
 FillStyle :4-Upward Diagonal
 Shape :4-Rounded Rectangle

File1:

Name :filImage
 Pattern :*.bmp;*.ico;*.wmf;*.gif;*.jpg (Sans espaces)

Line1:

BorderWidth 3

Exemple N°12 (CommonDialog; ScrollBar)

Formulaire Stagiaire...

Nom & Prénom: []

Sexe:

☐ Masculin

☐ Féminin

Filière:

☐ Technicien Spécialisé

☐ Secrétaire de Direction

☐ Développement

☐ Réseaux

Date de naissance:

[1] [Janvier] [1970]

Charger Image...

Afficher Info... Nouveau Exit

VScrollBar

Option Explicit

Dim Mois(12) As String

Dim filiere As String

```
Private Sub cmdinfo_Click()  
    Dim Msg As String, Age As Integer, prenom As String  
    Dim M As Integer, D As Integer, Y As Integer  
  
    M = Val(Format(Now, "mm"))  
    D = Val(Format(Now, "dd"))  
    Y = Val(Format(Now, "yyyy"))  
  
    Age = Y - vsbyear  
    If vsbmois.Value > M Or (vsbmois.Value = M And vsbjour > D) Then Age = Age - 1  
  
    'tester si la date est valide  
    If Age < 0 Then  
        MsgBox "La Date de naissance doit être avant la date actuelle .", vbOKOnly +  
vbCritical, "Invalide Date"  
        Exit Sub  
    End If  
  
    'tester si le nom a été rempli  
    If txtname.Text = "" Then  
        MsgBox "Vous devez saisir un nom.", vbOKOnly + vbCritical, "Entrez un nom"  
        Exit Sub  
    End If  
  
    Msg = txtname.Text + " poursuit ses études dans la filière " + filiere + vbCrLf  
    If optsex(0).Value = True Then prenom = "Il " Else prenom = "Elle "  
    Msg = Msg + prenom + " a" + Str(Age) + " ans." + vbCrLf  
    MsgBox Msg, vbOKOnly, "Profile Etudiant"  
End Sub
```

```
Private Sub Form_Load()  
    Mois(1) = "Janvier"  
    Mois(2) = "Février"  
    Mois(3) = "Mars"  
    Mois(4) = "Avril"  
    Mois(5) = "Mai"  
    Mois(6) = "Juin"
```

```

Mois(7) = "Juillet"
Mois(8) = "Août"
Mois(9) = "Septembre"
Mois(10) = "Octobre"
Mois(11) = "Novembre"
Mois(12) = "Decembre"
lbljour.Caption = Str(vsbjour.Value)
lblmois.Caption = Mois(vsbmois.Value)
lblyear.Caption = Str(vsbyear.Value)
End Sub

```

```

Private Sub optfil_Click(Index As Integer)
    Select Case Index
    Case 0
        filiere = "Technicien Spécialisé"
    Case 1
        filiere = "Secrétaire de Direction"
    Case 2
        filiere = "Développement"
    Case 3
        filiere = "Réseaux"
    End Select
End Sub

```

Définissez les propriétés suivantes pour les différentes barres de défilement:

<u>VScrollBar vsbYear:</u>	<u>VScrollBar vsbjour:</u>	<u>VScrollBar vsbMois:</u>
Max = 1900	Max = 1	Max = 1
Min = 2100	Min = 31	Min = 12
Value = 1960	Value = 1	Value = 1

```

Private Sub vsbjour_Change()
    lbljour.Caption = Str(vsbjour.Value)
End Sub

Private Sub vsbmois_Change()
    lblmois.Caption = Mois(vsbmois.Value)
End Sub

Private Sub vsbYear_Change()
    lblyear.Caption = Str(vsbyear.Value)
End Sub

```

```

Private Sub cmdnouveau_Click()
    txtname.Text = ""
    imgeleve.Picture = LoadPicture("")
End Sub

```

Il faut ajouter le contrôle **CommonDialog** dans votre boîte à outils. Pour ce faire, placer votre souris sur la boîte à outils et cliquez sur le bouton droit. Ensuite, choisissez la commande "Composants". Une liste de tous les contrôles apparaît alors. Celui qui nous intéresse ici, c'est le contrôle **CommonDialog**. Cochez la case du contrôle **Microsoft Common Dialog Control 6.0** et appuyez sur le bouton Appliquer. Le contrôle **CommonDialog** va alors apparaître sur votre boîte à outils. A présent, vous pouvez l'utiliser dans vos applications.

CommonDialog **cdlBox**:
 Filter = Bitmap(*.bmp)|*.bmp|Images Gif(*.gif)|*.gif|Images Jpeg(*.jpg)|*.jpg

```

Private Sub cmdcharger_Click()
    cdlbox.ShowOpen
    imgeleve.Picture = LoadPicture(cdlbox.FileName)
End Sub

```

```

Private Sub cmdend_Click()
    End
End Sub

```


Cours 9 : Accès aux fichiers

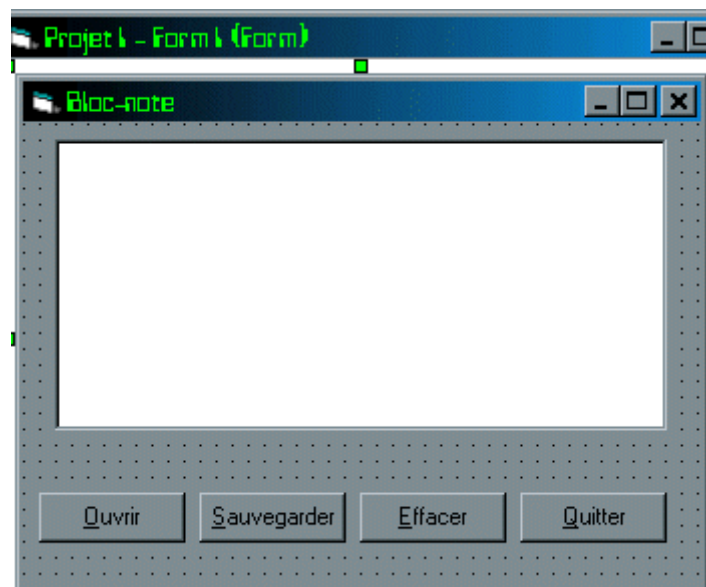
Il existe trois modes d'accès aux fichiers dans Visual Basic 6 :

1. Les fichiers à accès séquentiel ne peuvent ouvrir que des fichiers avec un contenu textuel.
2. Les fichiers à accès direct sont formés en général de plusieurs enregistrements de longueur fixe.
3. Les fichiers binaires sont tous les fichiers avec un contenu graphique.

1. Les fichiers à accès séquentiel

Il existe trois manières d'ouvrir un fichier texte :

1. La 1^{ère} manière est l'ouverture d'un fichier uniquement en lecture à l'aide de l'instruction *Input*.
2. La 2^{ème} manière est l'ouverture d'un fichier uniquement en écriture à l'aide de l'instruction *Output*.
3. La dernière manière est l'ouverture d'un fichier en écriture aussi mais à la seule différence de la précédente manière, les nouvelles données sont entrées à la suite de l'enregistrement précédent. Pour cela, on utilise l'instruction *Append*.



Multiline True

```
Private Sub ouvrir_Click()  
On Error Resume Next  
Open "C:\note.TXT" For Input As #1  
retour = Chr$(13) + Chr$(10)  
Line Input #1, texte  
tout = texte  
If Len(tout) <> 0 Then  
While Not EOF(1)  
Line Input #1, texte  
tout = tout + retour + texte  
Wend  
End If  
Text1.Text=tout  
Close #1  
End Sub
```

Explication :

- L'instruction *On error Resume Next* permet d'ignorer toute erreur que peut entraîner cette procédure.
- L'instruction *Open "C:\note.TXT" For Input As #1* permet d'ouvrir le fichier "note" situé sur le bureau. Lorsque l'application est exécutée pour la première fois, une erreur 53 devrait apparaître car le fichier "note" n'existe pas encore. Mais grâce à l'instruction *On Error Resume Next*, cette erreur sera tout simplement ignorée.
- L'instruction *retour = Chr(13)\$ + Chr(10)\$* permet à la ligne suivante. Le code ASCII 13 correspond au "retour chariot", c'est-à-dire de revenir à gauche du texte et 10 au "passage à la ligne", c'est-à-dire d'aller à la ligne suivante.
- L'instruction *Line Input* permet de placer dans la variable "texte" le contenu de la première ligne du fichier "note.TXT".
- Ensuite, le contenu de la variable "texte" est à son tour, affecté à la variable "tout".
- Le test avec l'instruction *If* permet de vérifier que le fichier "note.TXT" n'est pas vide à l'aide de la fonction *Len* qui compte le nombre de lettres ou tout autre symbole contenu dans la variable "tout". Si le nombre de lettres est différent de zéro alors, les instructions à l'intérieur du test sont exécutés.
- La boucle avec *While* permet de parcourir tout le fichier à la recherche d'éventuelles autres lignes que la première. La fonction *EOF* permet de savoir si l'on arrive à la fin du fichier ouvert. Cette boucle est donc exécutée jusqu'à ce que tout le contenu du fichier soit placé dans la variable tout. Les chaînes de caractères affectées à la variable "tout" sont concaténées (concaténées = ajoutées) aux précédentes contenues dans la variable. La variable "retour" provoque un retour à la ligne à chaque fois que cela est nécessaire.
- Lorsque tout le contenu du fichier sera affecté à la variable "tout", cette dernière sera à son tour affectée à la propriété *Text* du contrôle *TextBox*. Ce qui provoquera l'affichage du contenu du fichier dans le contrôle *TextBox*.
- L'instruction *Close* ferme le fichier ouvert en lecture.

Écrivons maintenant la portion de code correspondant au bouton "Sauvegarder" :

```
Private Sub Sauvegarder_Click()
Open "C:\Windows\Bureau\note.TXT" For Output As #1
Print #1, Text1.Text
Close #1
End Sub
```

Explication :

- L'instruction *Open "C:\Windows\Bureau\note.TXT" For Output As #1* ouvre le fichier "note.TXT" en écriture.
- Le contenu de la zone de texte "Text1" est ensuite copié dans le fichier à l'aide de l'instruction *Print*.
- Enfin, le fichier est fermé à l'aide de l'instruction *Close*.

Passons à l'écriture de la procédure concernant le bouton "Effacer" :

```
Private Sub Effacer_Click()
Text1.Text = ""
Text1.SetFocus
End Sub
```

24 .Utiliser le contrôle DATA

Le programme *DATA* permet à votre programme d'avoir accès à des bases de données externes, de type *Microsoft Access*, *bBase*, *FoxPro*...

ETAPE 1

Dans *Access*, créer la base de données suivante : *essai.mdb* qui contient la table *Clients* :

Clients : Table		
Nom du champ	Type de données	
Nom	Texte	
Prénom	Texte	
date de naissance	Date/Heure	

Clients : Table			
	Nom	Prénom	date de naissar
	Mountasir	Rachid	10/11/1972
	Abounassr	rami	30/04/1966
*			

ETAPE 2

Convertir cette BDD vers le format *Access 97* : cette étape est nécessaire

Choisissez Le menu

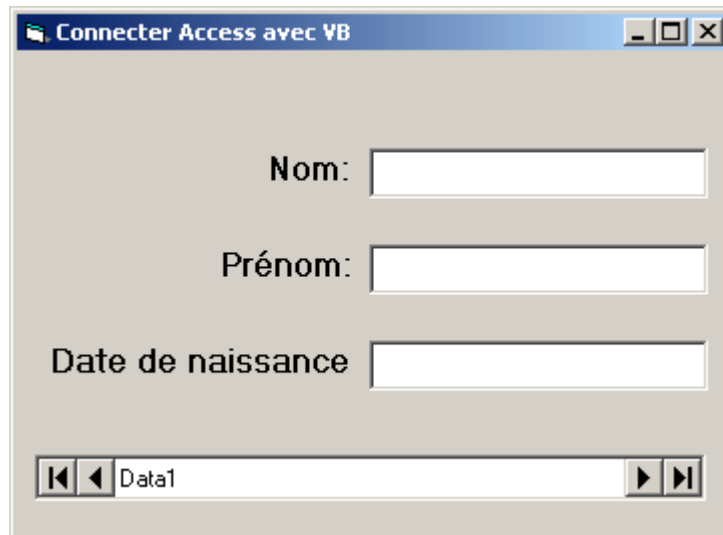
- Outils
- Utilitaire de base de données
- Convertir une base de données
- Vers le format de fichiers *Access 97*
- Sauvegarder la BDD sous le nom : *essai97*

ETAPE 3

Dans *Visual Basic*, Insérez le contrôle *DATA* :



Construisez l'interface suivante :



Connecter Access avec VB

Nom:

Prénom:

Date de naissance

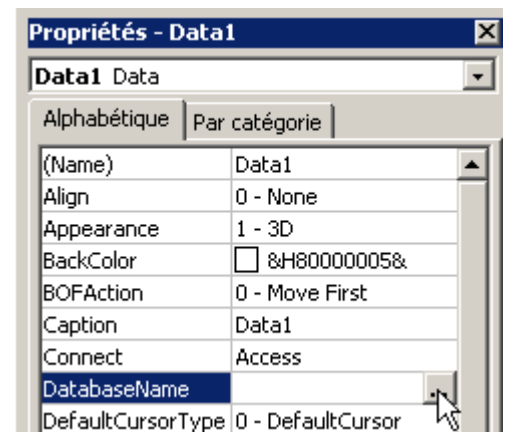
Navigation buttons: << Data1 >>

ETAPE 4

Spécifiez la propriété *Connect vers*
Access

Choisissez ensuite pour la propriété
DatabaseName la BDD créer en haut :
essai97

Pour la propriété *RecordSource*,
Choisissez la table **Clients**

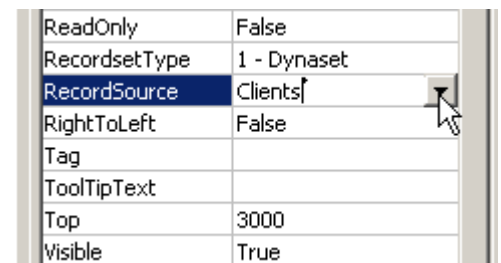


Propriétés - Data1

Data1 Data

Alphabétique | Par catégorie

(Name)	Data1
Align	0 - None
Appearance	1 - 3D
BackColor	<input type="checkbox"/> &H80000005&
BOFAction	0 - Move First
Caption	Data1
Connect	Access
DatabaseName	
DefaultCursorType	0 - DefaultCursor

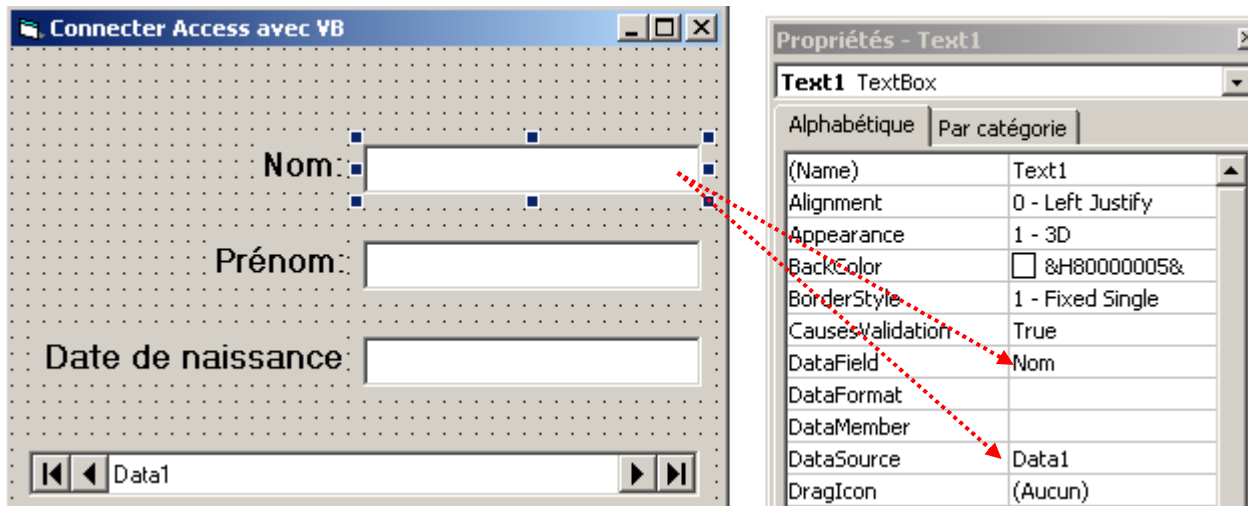


ReadOnly	False
RecordsetType	1 - Dynaset
RecordSource	Clients
RightToLeft	False
Tag	
ToolTipText	
Top	3000
Visible	True

ETAPE 5

On va maintenant afficher les données de la table *Clients* dans les zones de texte déjà crée :

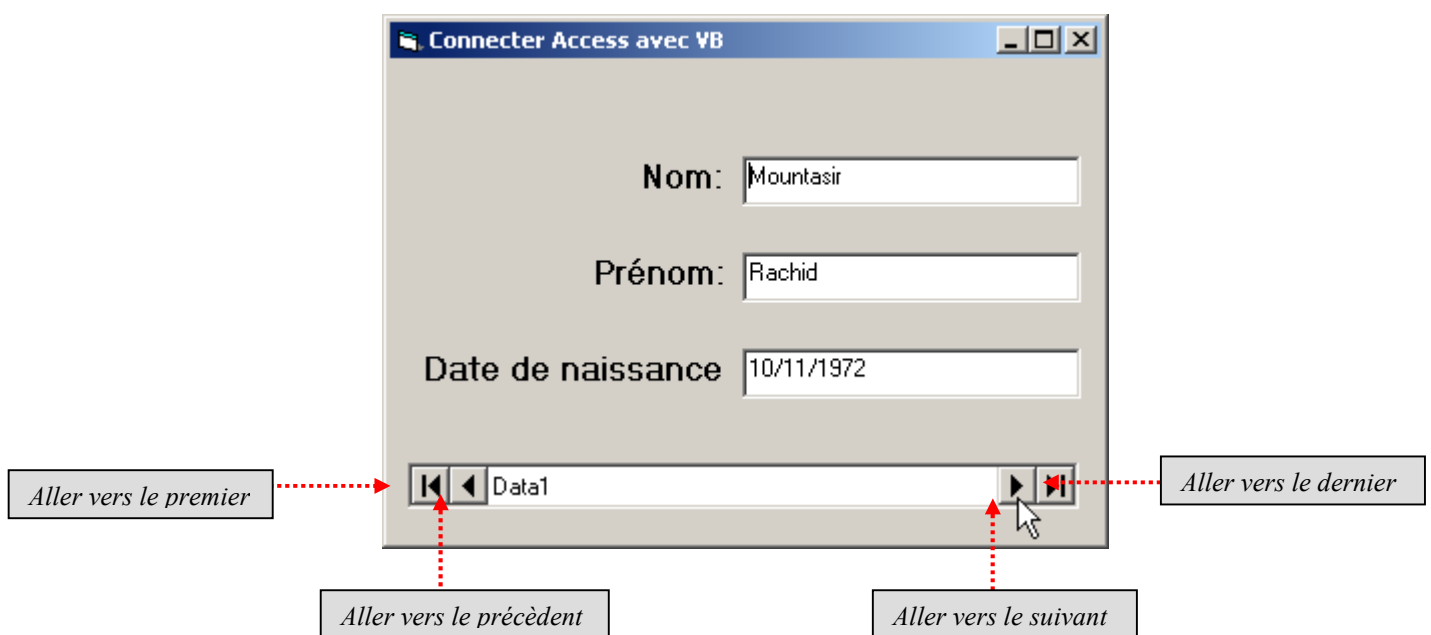
Pour chaque zone de texte vous devez spécifier ces deux propriétés :



DataField permet de choisir quel champ de la table clients sera affiché dans chaque zone de texte.

ETAPE 6

Exécutez votre application :

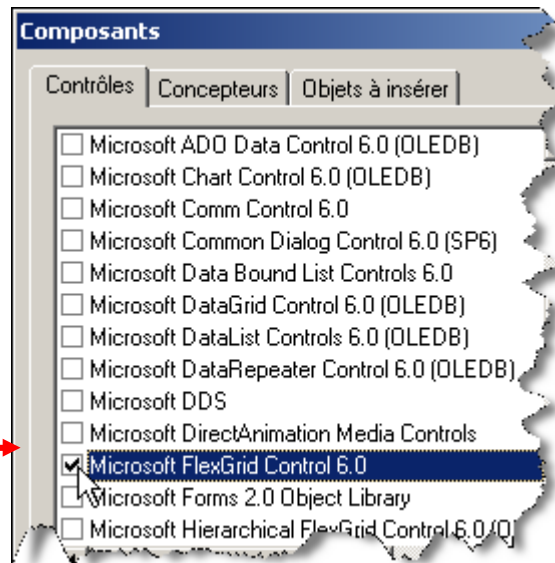
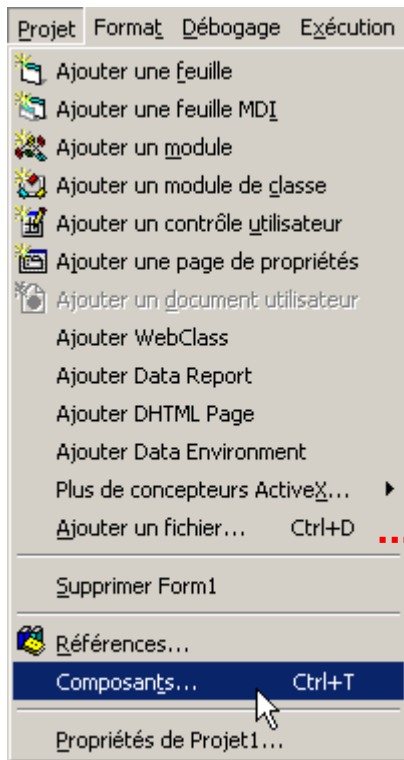


ETAPE 7

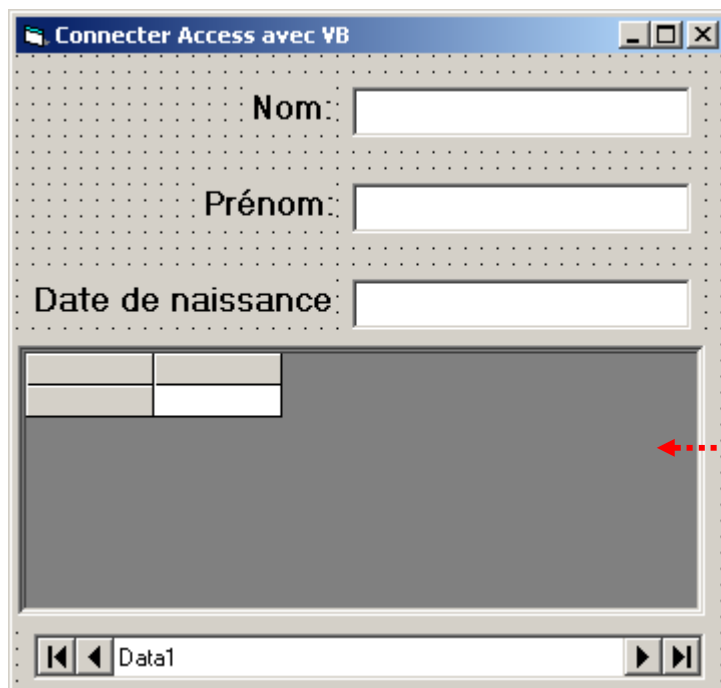
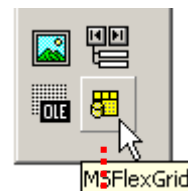
Le contrôle Microsoft FlexGrid (**MSFlexGrid**) affiche et opère sur des données placées dans des tableaux.

Lorsque ce contrôle est lié à un contrôle Data, il affiche les données en **lecture seule**.

Si le texte placé dans une cellule est trop long pour y être affiché entièrement et que la propriété **WordWrap** est définie sur **True**, le texte passe à la ligne suivante dans la même cellule.



Dans la boîte à outils vous verrez le composant suivant :



Insérez le contrôle MSFlexGrid dans la feuille et ajustez sa propriété :

DataSource Data1

A l'exécution vous obtiendrez ceci :

Connecter Access avec VB

Nom: Mountasir

Prénom: Rachid

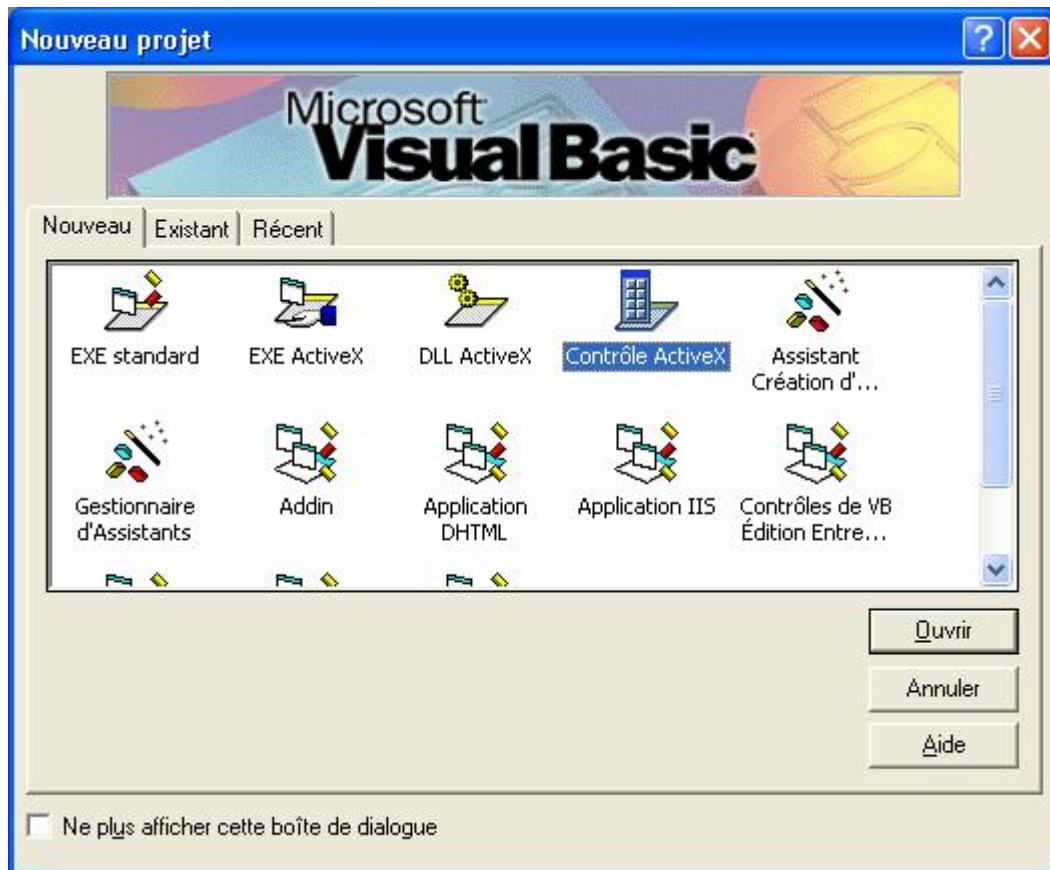
Date de naissance: 10/11/1972

	Nom	Prénom	date de
	Mountasir	Rachid	10/11/1972
	Abounassr	rami	30/04/1966

« « Data1 » »

Création de contrôles ActiveX

La première chose qui nous faut réaliser c'est donc de créer le projet qui va contenir notre (ou nos) contrôle(s). Par défaut, quand vous lancez Visual Basic, une boîte de dialogue apparaît et vous propose plusieurs modèles de projets. Si ce n'est pas le cas, cliquez dans le menu "Fichier" sur "Nouveau Projet". La capture d'écran suivante montre cette boîte de dialogue.



Le modèle de projet qui nous intéresse est donc le projet de contrôle ActiveX. Cliquez sur ouvrir après l'avoir choisi. Dans une certaine mesure, ce type de projet ressemble à un projet de type "Exe Standard". En effet, vous allez pouvoir éditer votre contrôle via deux outils : le concepteur visuel et l'éditeur de code. Mais une différence apparaît malgré tout : dans le concepteur graphique, votre contrôle n'a pas de détournage autour de sa zone visuelle comme l'aurait une feuille. Cela est normal : l'objectif du contrôle c'est d'être utilisé sur une feuille.

Vous pouvez ensuite remplir l'espace de votre contrôle avec des sous-contrôles. La partie visuelle de notre contrôle va contenir : un contrôle de type Frame , et Label (placé dans le Frame) pour afficher l'heure et un contrôle Timer pour actualiser régulièrement l'heure dans le Label. Nommez respectivement vos contrôles fraClock, lblClock et tmrClock. Affectez aussi à la propriété Interval du Timer la valeur 1000 (l'unité étant la milli-seconde, nous avons donc 1 seconde d'intervalle). Au final, votre contrôle (en mode conception) devrait ressembler à la capture d'écran suivante.



Une chose importante est à ne pas oublier : il faut renommer le projet et le contrôle
Le nom du projet est "Infini" et l'unique contrôle de ce projet est nommé "Clock".
Changer la propriété **ToolboxBitmap** : elle permettra d'afficher l'icône souhaitée lorsque votre contrôle devra être insérer dans la boîte à outils.

Option Explicit

```
'-Mise à jour du label d'affichage de l'heure-----
Private Sub tmrSeconds_Timer()
    lblClock.Caption = Time
End Sub

'-Code de retaillage de l'horloge-----
Private Sub UserControl_Resize()
    fraClock.Width = UserControl.Width
    fraClock.Height = UserControl.Height

    lblClock.Width = UserControl.Width
    lblClock.Top = (UserControl.Height - lblClock.Height) / 2 + 100
End Sub

'-Initialisation de la valeur du label au premier affichage-----
Private Sub UserControl_Show()
    tmrSeconds_Timer
End Sub
```

Ajout du projet de test

Nous allons maintenant ajouter un projet secondaire. En effet, Visual Basic gère la notion de groupe de projet. Pour ce faire, allez dans le menu "Fichier" et cliquez sur "Ajouter un projet". Le type de projet à ajouter est "Exe standard". La capture d'écran suivante montre l'explorateur de projets après l'ajout. Notez aussi que, comme les deux projets font parties du même groupe de projet, l'icône du contrôle ActiveX apparaît lorsque que vous éditez la feuille du projet de test.



A partir de maintenant, vous pouvez poser vos contrôles sur la feuille de test



Pour l'heure, votre contrôle ActiveX n'existe pas réellement. Certes, il apparaît dans la boîte à outils, mais cela est fait car les deux projets sont intégrés dans le même groupe de projets. Pour pouvoir utiliser votre contrôle dans un autre projet, sans forcément disposer du code source, il vous faut compiler votre code pour générer le fichier ".ocx". Ensuite, il vous faut enregistrer votre (ou vos) composant(s) dans la base de registre Windows.

Tant que vous êtes sur le poste de développement les choses sont plus simples qu'il n'y paraît : en effet, le fait de compiler le projet via l'environnement Visual Basic, enregistre automatiquement le contrôle dans la base. Si par contre, vous êtes sur une autre machine et que vous n'avez que le fichier ".ocx" à votre disposition il va falloir réaliser l'enregistrement à la main, via l'outil "regsvr32.exe". Pour ce faire, prenez une console, placez-vous dans le répertoire contenant votre fichier ".ocx" et tapez la commande "regsvr32 file.ocx" (ou file.ocx représente le nom de votre fichier).

L'outil d'emballage et de déploiement fourni avec Visual Basic permet de gérer un programme d'installation, qui saura installer les contrôles automatiquement sur le poste client. Il ne sera donc plus utile de lancer la commande "regsvr32.exe" manuellement.

Utiliser le contrôle Data

Base de données Clients

Code client: BSBEV

Société: B's Beverages

Contact: Victoria Ashworth

Fonction: Représentant(e)

Adresse: Faulteroy Circus

Ville: London

Région:

Code postal: EC2 5NT

Pays: Royaume-Uni

Téléphone: (71) 555-1212

Fax:

Nouveau

Enregistrer

Rechercher

Déplacement

Suppression

Enregistrement N° 1 sur 79

```
Private Sub cmdAdd_Click()  
    Data1.Recordset.AddNew  
End Sub  
  
Private Sub cmdUpdate_Click()  
    Data1.UpdateRecord  
    Data1.Recordset.Bookmark = Data1.Recordset.LastModified  
End Sub  
  
Private Sub cmdClose_Click()  
    Unload Me  
End  
End Sub  
  
Private Sub cmdDelete_Click()  
    Dim rep  
    rep = MsgBox("Suppression", vbYesNo, "Attention")  
    If rep = vbYes Then  
        Data1.Recordset.Delete  
        precedent  
    End If  
End Sub  
  
Private Sub cmdfind_Click()  
    Dim rep  
    rep = InputBox("Entrer le Code Client recherché")  
    If rep = "" Then Exit Sub  
    Data1.Recordset.MoveFirst  
    Data1.Recordset.FindFirst "[code client]=" & "'" & rep & "'"  
    If Data1.Recordset.NoMatch = True Then  
        MsgBox "Le code client " & rep & "n'a pas été trouvé"  
    End If  
End Sub  
  
Private Sub cmdfirst_Click()
```

```

        Data1.Recordset.MoveFirst
End Sub
Private Sub cmdlast_Click()
    Data1.Recordset.MoveLast
End Sub
Private Sub cmdnext_Click()
    Data1.Recordset.MoveNext
    If Data1.Recordset.EOF = True Then
        Data1.Recordset.MoveLast
    End If
End Sub
Private Sub cmdprevious_Click()
    precedent
End Sub
Private Sub Data1_Reposition()
    Data1.Caption = "Enregistrement N° " & Data1.Recordset.AbsolutePosition + 1 & " sur " &
    Data1.Recordset.RecordCount
End Sub
Public Sub precedent()
    Data1.Recordset.MovePrevious
    If Data1.Recordset.BOF = True Then
        Data1.Recordset.MoveFirst
    End If
End Sub
Private Sub Form_Activate()
    Data1.Recordset.MoveLast
    Data1.Recordset.MoveFirst
End Sub
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
    Dim rep
    rep = MsgBox("Etês vous sur de bien vouloir quitter...!!!", vbYesNo, "Quitter")
    If rep = vbYes Then
        Cancel = False
    Else
        Cancel = True
    End If
End Sub

```

Accéder à une BDD Access en utilisant DataEnvironnement

Etape 1 : Créer la table **CDs** dans la BDD CDCollection.mdb :

CDs : Table			
	Nom du champ	Type de données	
	Artist	Texte	
	Album	Texte	
	Tracks	Numérique	

CDs : Table			
	Artist	Album	Tracks
▶	Amr Diab	Tamaly maak	12
	Charles Aznavour	La Bohème	14
	Cheb Khaled	Aicha	14
	Cheb Khaled	Nsi Nsi	12
	Youness Migri	Laily Taouil	11
*			0

Enr : 1 sur 5

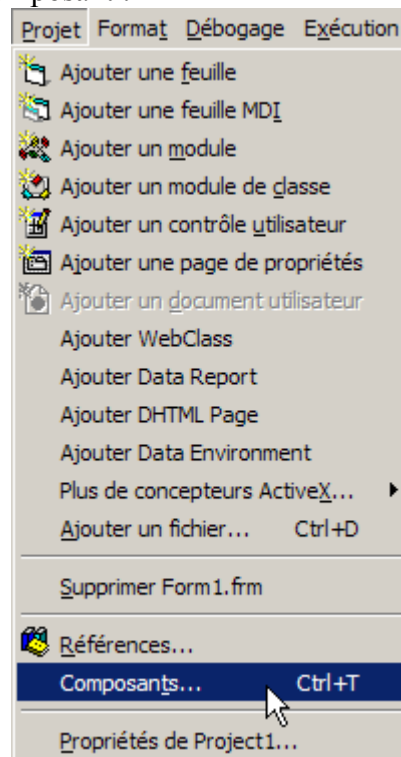
Cette BDD ainsi que le projet Visual Basic doivent être sauvegardé dans le même répertoire.

Etapes 2 : Passer à Visual Basic et réalisez l'interface suivante :

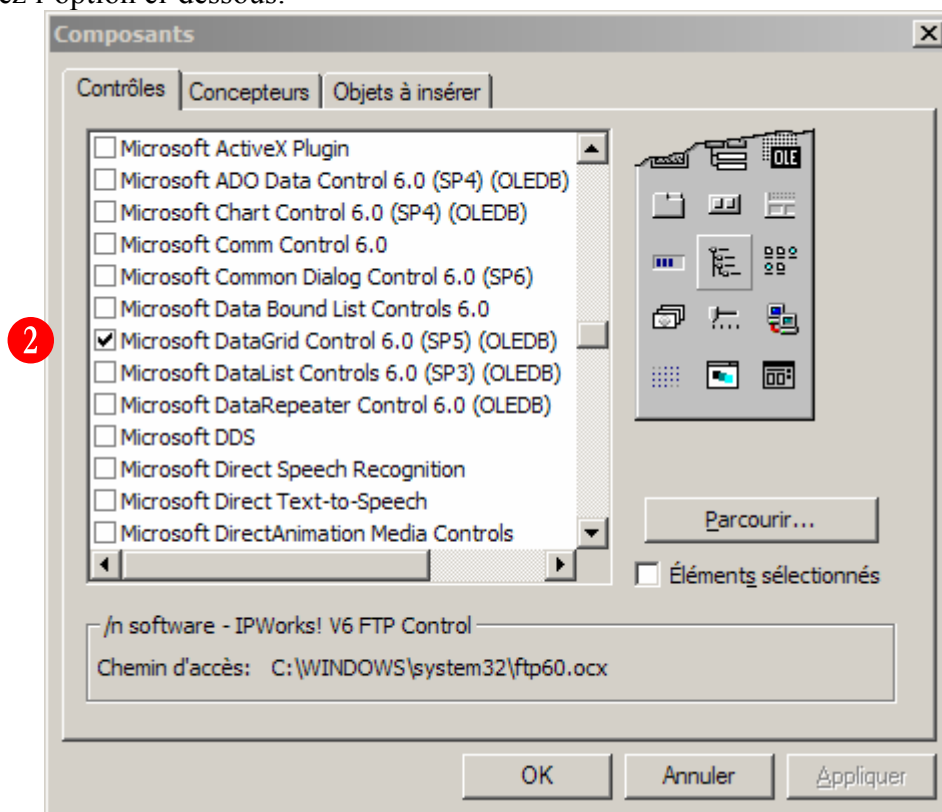
Le contrôle
DataGridView

Le contrôle **DataGrid** n'apparaît pas par défaut dans la boîte à Outils et il faudra l'insérer manuellement :

1. Affichez la boîte à outils Composant :



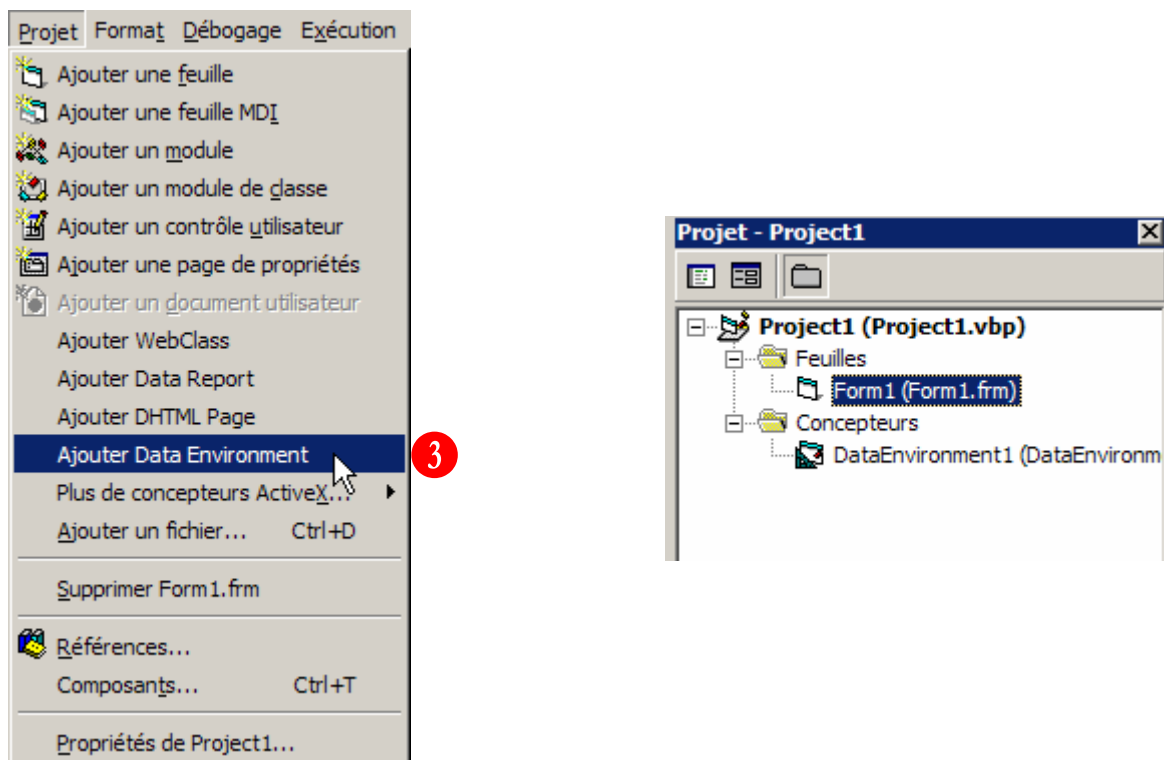
2. Cochez l'option ci-dessous:



Etape 3 : Paramétrez les différentes propriétés des contrôles que contient votre feuille.

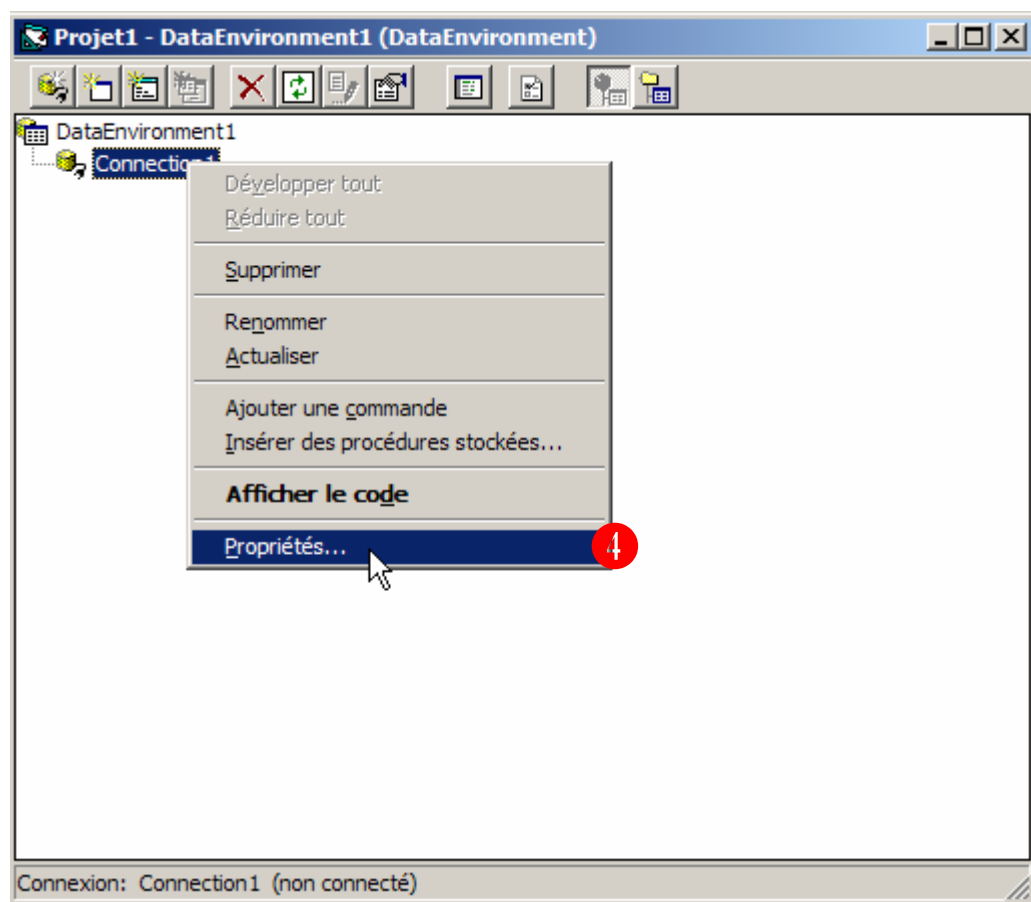
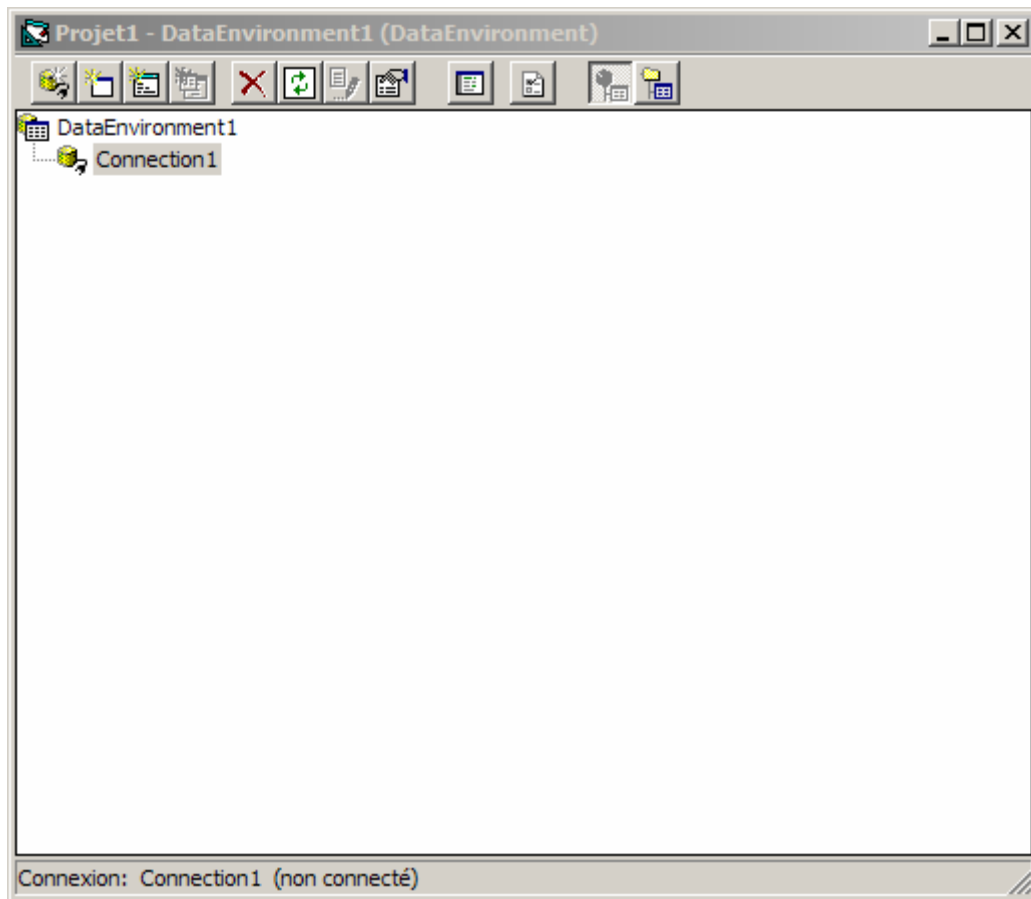
Contrôles	Names
DataGrid	DataGrid1
Ajouter	cmdAjouter
Supprimer	cmdsupprimer
Nom	txtArtist
Titre de l'album	txtAlbum
N° de Track	txtTrack

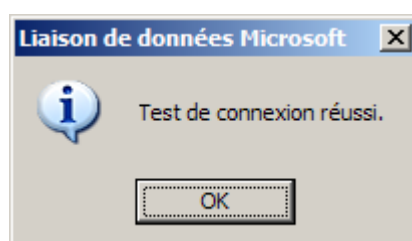
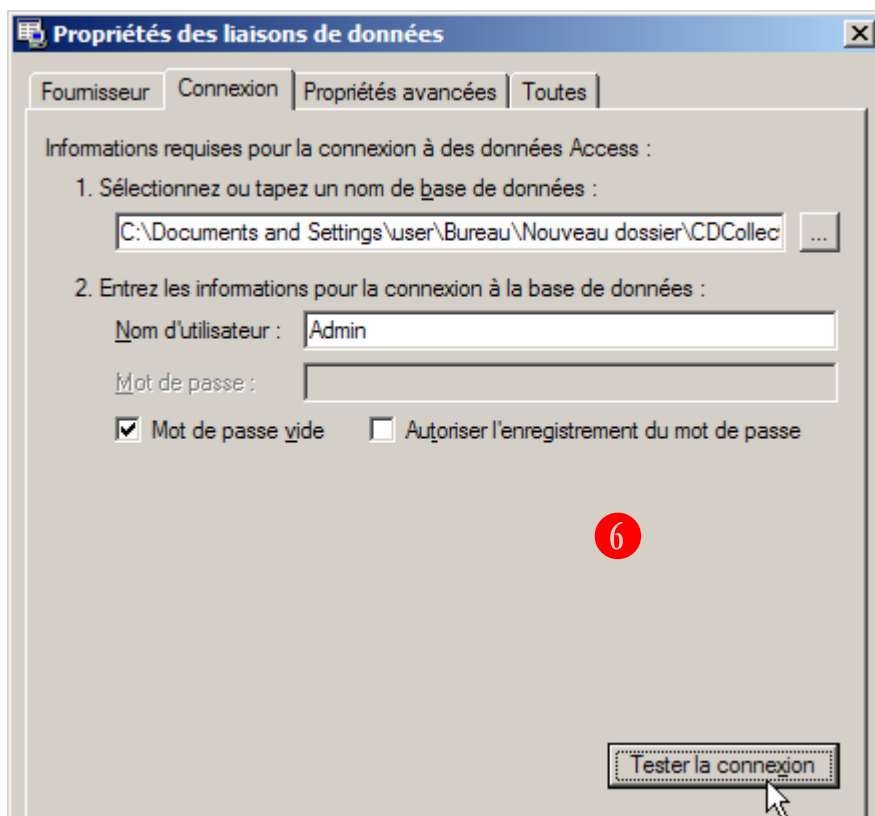
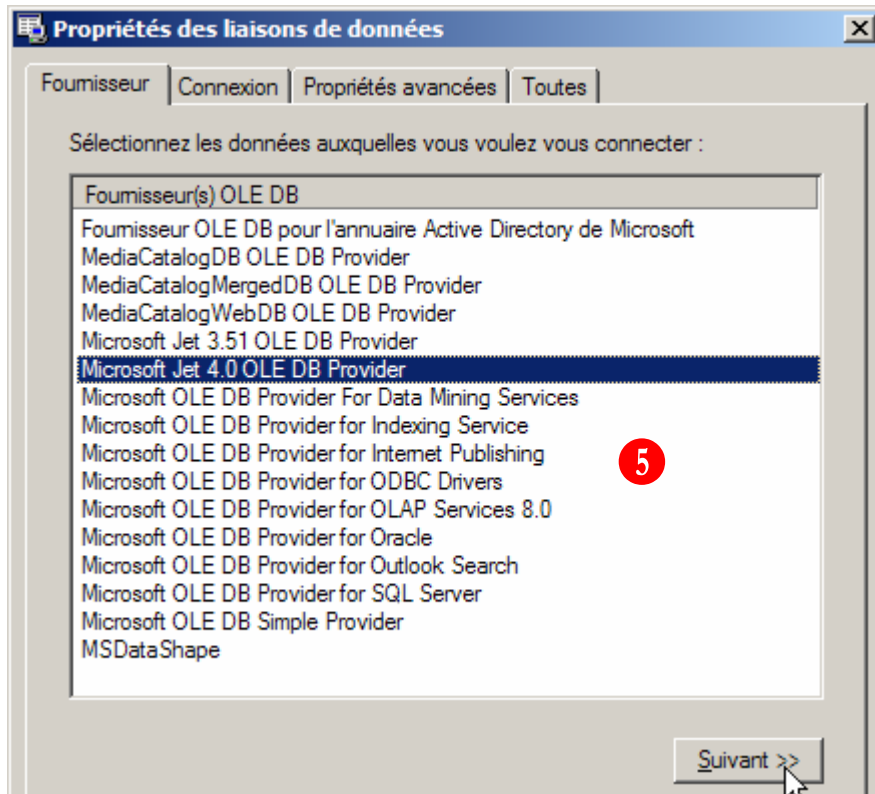
Etape 4 : Ajouter DataEnvironnement :

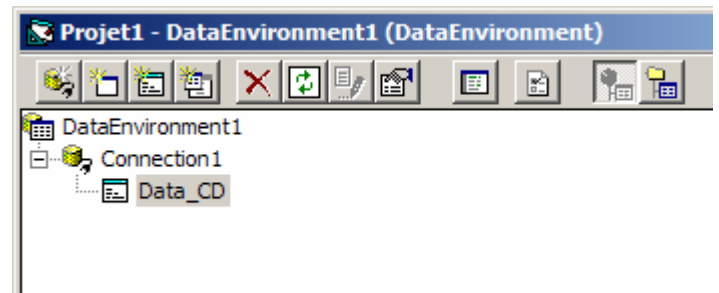
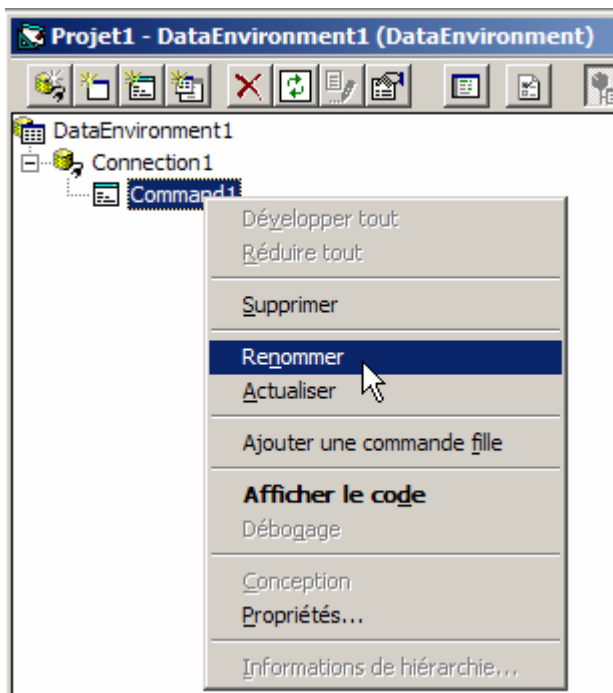
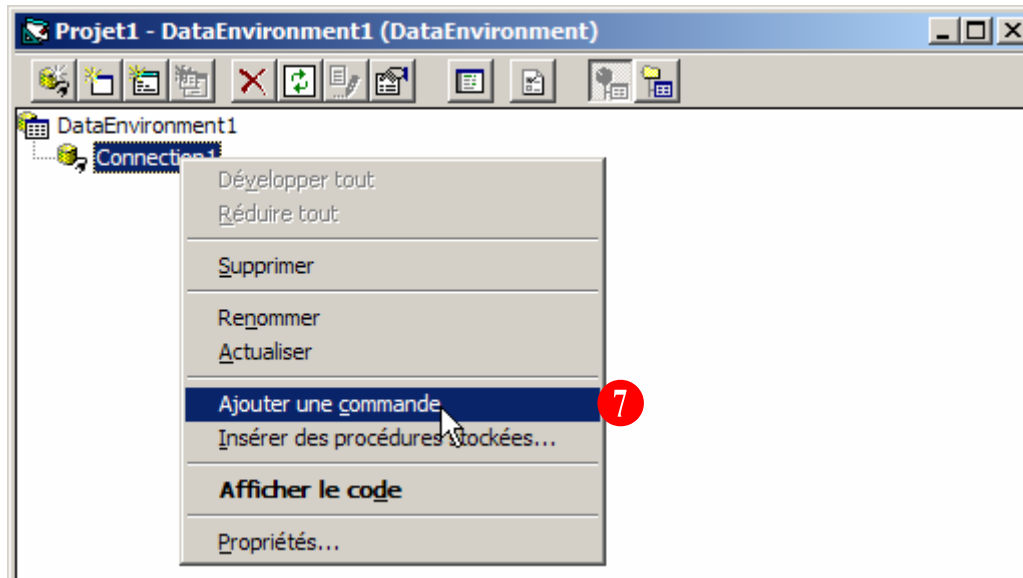


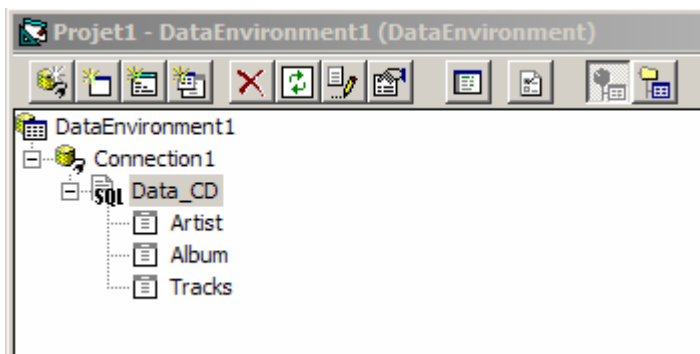
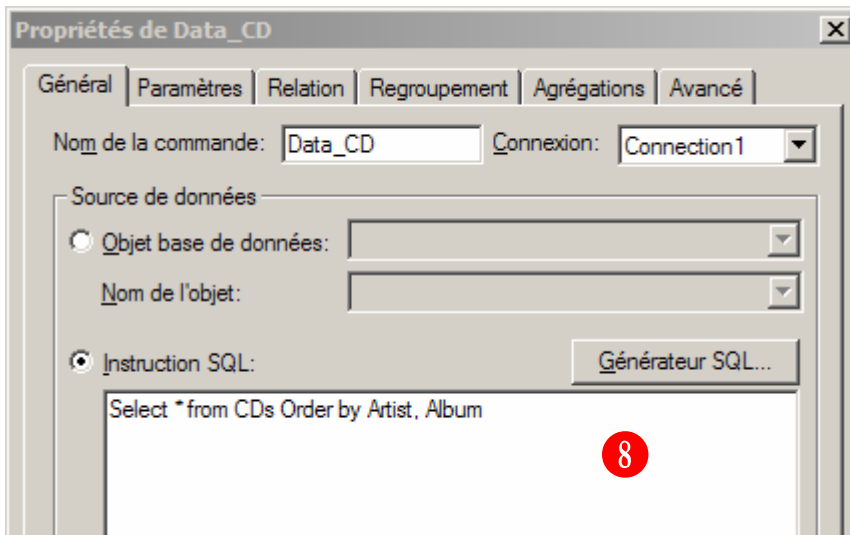
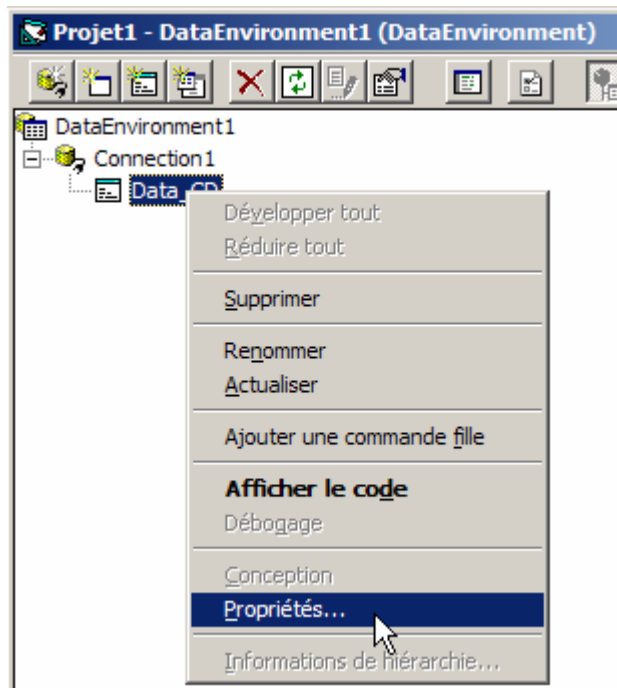
Le paramétrage de DataEnvironnement s'effectue en suivant les étapes suivantes :

Paramétrer la connexion à la BDD **CDCollection.mdb** :

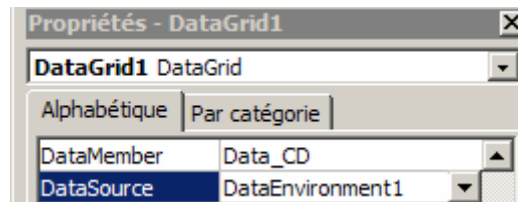








Sélectionnez DataGrid1 (Dans Form1) et Attribuez le les propriétés suivantes :



Artist	Album	Tracks
Amr Diab	Tamaly maak	12
Charles Aznavour	La Bohème	14
Cheb Khaled	Aïcha	14
Cheb Khaled	Nsi Nsi	12
Youness Migri	Laily Taouil	11

Remarquez le tri

.....

```
Private Sub cmdajouter_Click()
    With DataEnvironment1.rsDataTable
        .AddNew
        !Artist = txtArtist
        !Album = txtAlbum
        !Tracks = txtTrack
        .Update
    End With

    DataEnvironment1.rsDataTable.Requery
    Set DataGrid1.DataSource = DataEnvironment1

    txtArtist = ""
    txtAlbum = ""
    txtTrack = ""
    txtArtist.SetFocus
End Sub
```

```
Private Sub cmdsupprimer_Click()
    DataEnvironment1.rsDataTable.Delete adAffectCurrent
End Sub
```

```
Private Sub Form_Load()  
    cmdajouter.Enabled = False  
End Sub
```

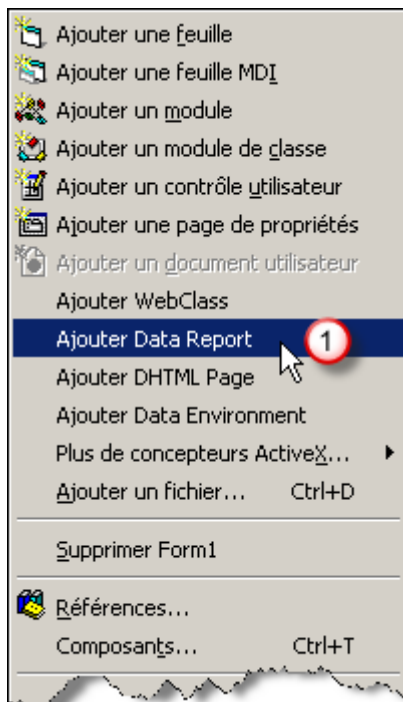
```
Private Sub txtArtist_Change()  
    If txtArtist.Text <> "" And _  
    txtAlbum.Text <> "" And _  
    txtTrack.Text <> "" Then  
        cmdajouter.Enabled = True  
    Else  
        cmdajouter.Enabled = False  
    End If  
End Sub
```

```
Private Sub txtAlbum_Change()  
    Call txtArtist_Change  
End Sub
```

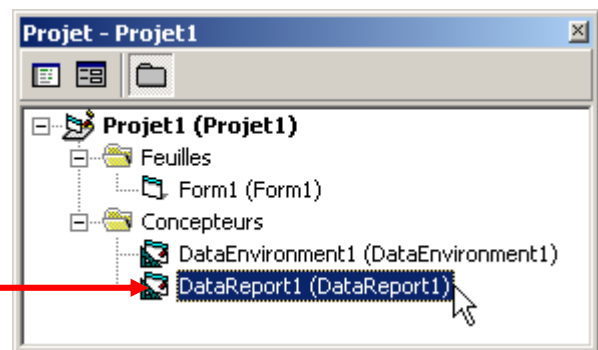
```
Private Sub txtTrack_Change()  
    Call txtArtist_Change  
End Sub
```

```
Private Sub txtTrack_KeyPress(KeyAscii As Integer)  
    Dim TrackKey As String  
    TrackKey = Chr(KeyAscii)  
  
    If (Not IsNumeric(TrackKey) And Not (KeyAscii = vbKeyBack)) Then  
        KeyAscii = 0  
    End If  
End Sub
```

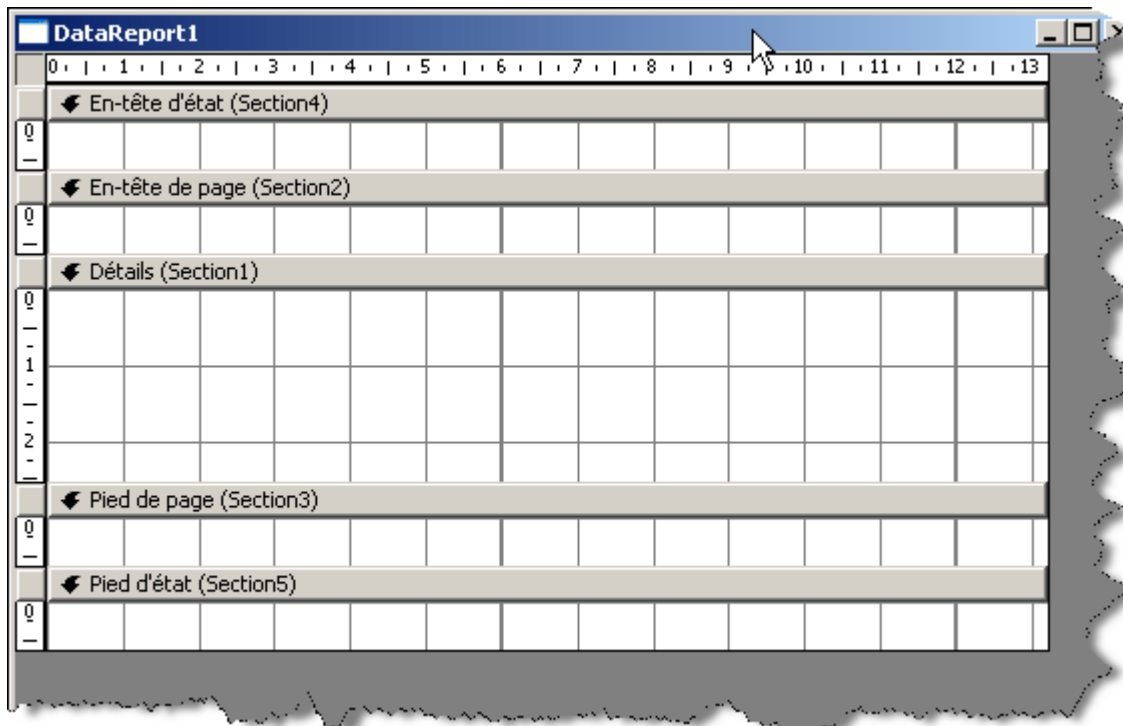
Créer un Data Report



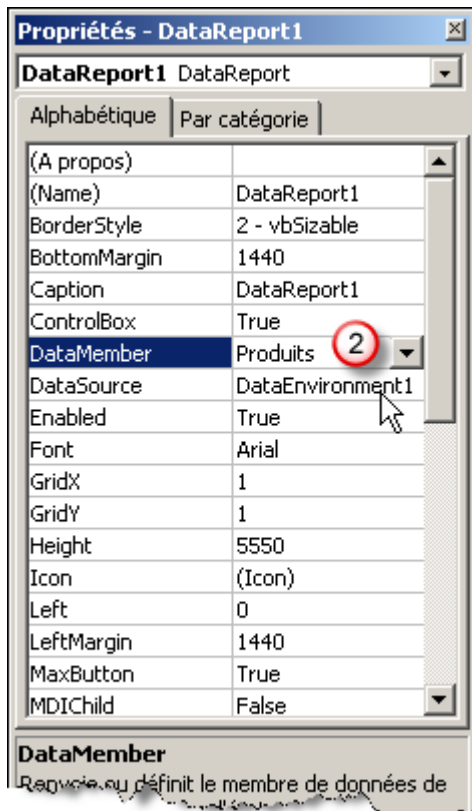
Insérer le contrôle Data Report



L'état suivant apparaîtra :

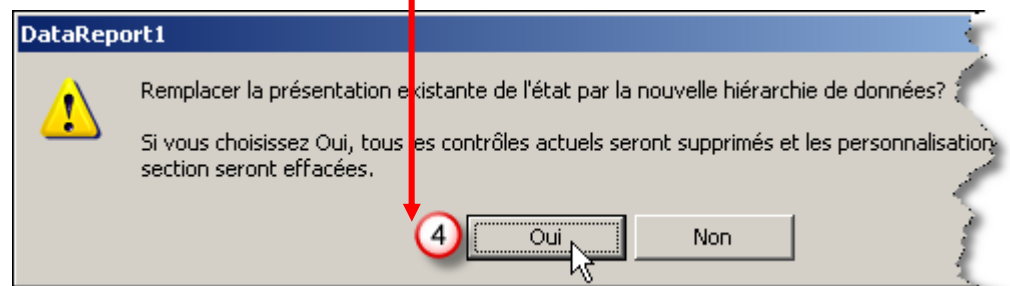
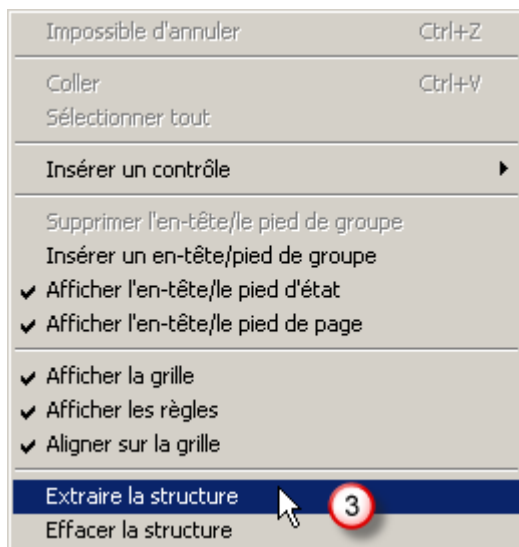


Définir les propriétés de cet état :

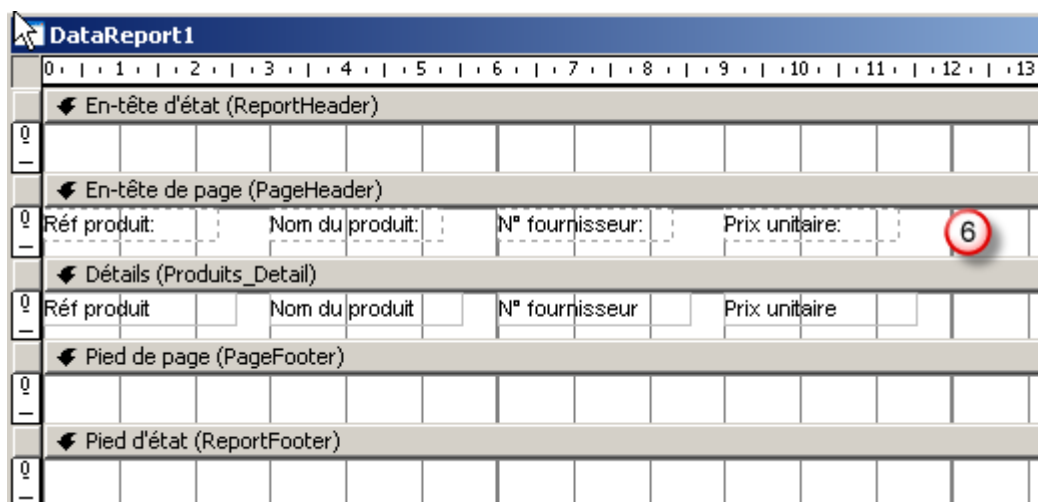
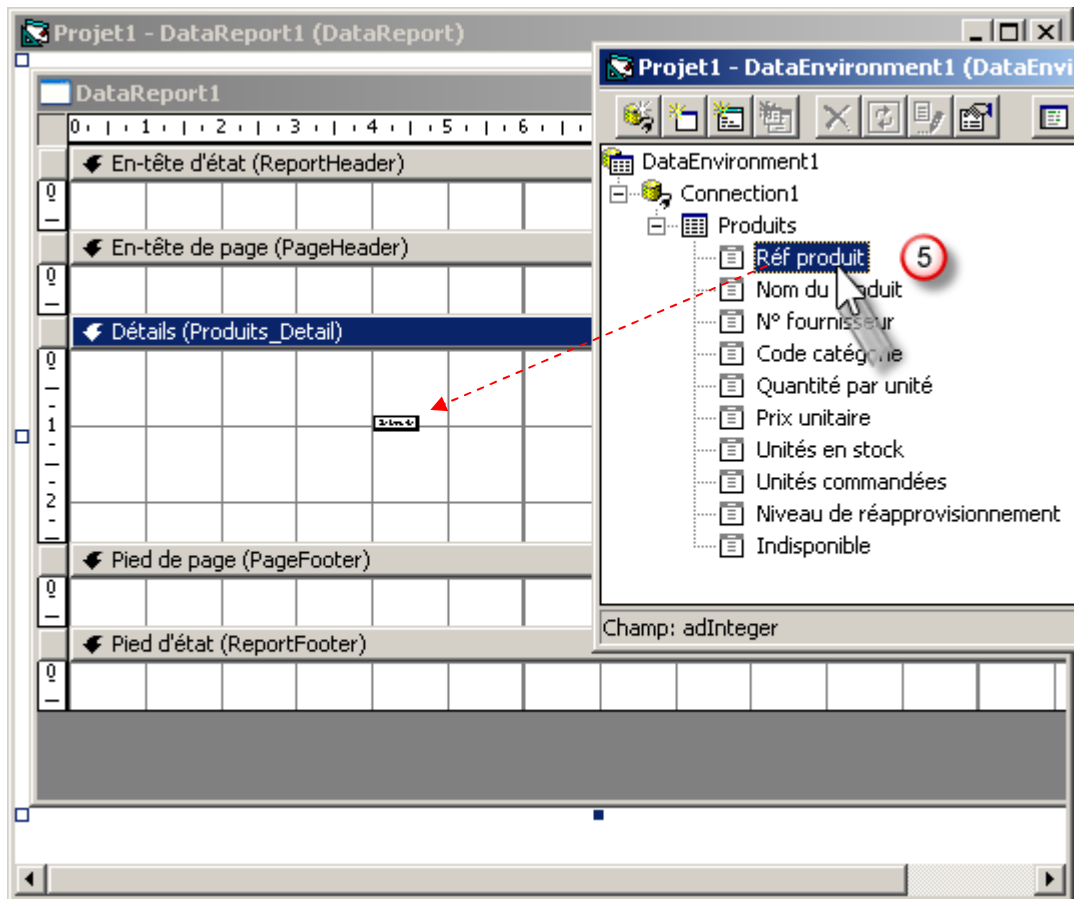


DataSource : Correspond à l'outil choisi pour se connecter à votre BDD.

DataMember : Correspond à la table définie dans dataenvironnement.










Glissez les champs que vous voulez afficher vers la section détails :






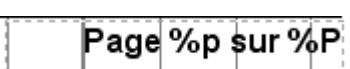









Ensuite vous devez :

DataReport1

En-tête d'état (ReportHeader)															
<i>Master Computer S.a.r.l</i>															
En-tête de page (PageHeader)															
TOTAL DES PRODUITS VENDUES															
Réf produit	Nom du produit				N°	Prix unitaire				Unités en stock					
Détails (Produits_Detail)															
Réf produit	Nom du produit				N° fournisseur	Prix unitaire				Unités en stock					
Pied de page (PageFooter)															
														Page %p sur %P	
Pied d'état (ReportFooter)															
														Total : SUM(Prix unitaire)	

Toolbar: Général, DataReport, , , , , , 

	 Ce contrôle ressemble au contrôle Label . Insérez le et modifiez ses propriétés Caption & fonts.
	Idem que pour le précédent
	 Insérez ce contrôle et modifiez sa propriété : Picture
	 Caption : Page %p sur %P
	 DataMember : Produits DataField : Prix unitaire
	
	

Il ne vous reste plus que créer un bouton de commande sur une feuille(Form) et d'y ajouter le code suivant :

```
Private Sub Command1_Click()  
DataReport1.Show  
End Sub
```

Le résultat final :

Master Computer S.a.r.l



TOTAL DES PRODUITS VENDUES

<i>Réf produit:</i>	<i>Nom du produit:</i>	<i>N°</i>	<i>Prix unitaire:</i>	<i>Unités en stock:</i>
1	Chai	1	90	39
2	Chang	1	95	17
3	Aniseed Syrup	1	50	13
4	Chef Anton's	2	110	53
5	Chef Anton's	2	106,75	0
6	Grandma's	3	125	120
7	Uncle Bob's	3	150	15
8	Northwoods	3	200	6
9	Mishi Kobe Niku	4	485	29
10	Ikura	4	155	31
11	Queso Cabrales	5	105	22
12	Queso Manchego	5	190	86
13	Konbu	6	30	24
14	Tofu	6	116,25	35
15	Genen Shouyu	6	77,5	39

En bas de chaque page :

27	Schoggi	11	219,5	49
----	---------	----	-------	----

Page 1 sur 3

En bas de l'Etat :

75	Rh.nbr ou	12	38,75	125
76	Lakkalik_ri	23	90	57
77	Original	12	65	32

Total : 11113,55