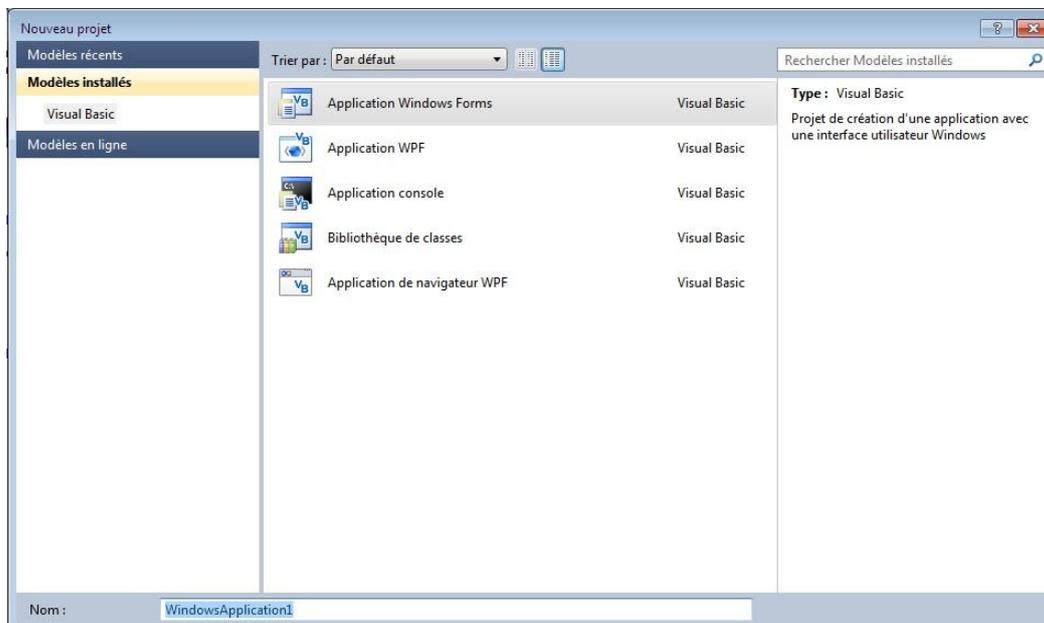


## PRISE EN MAINS DE VB.NET

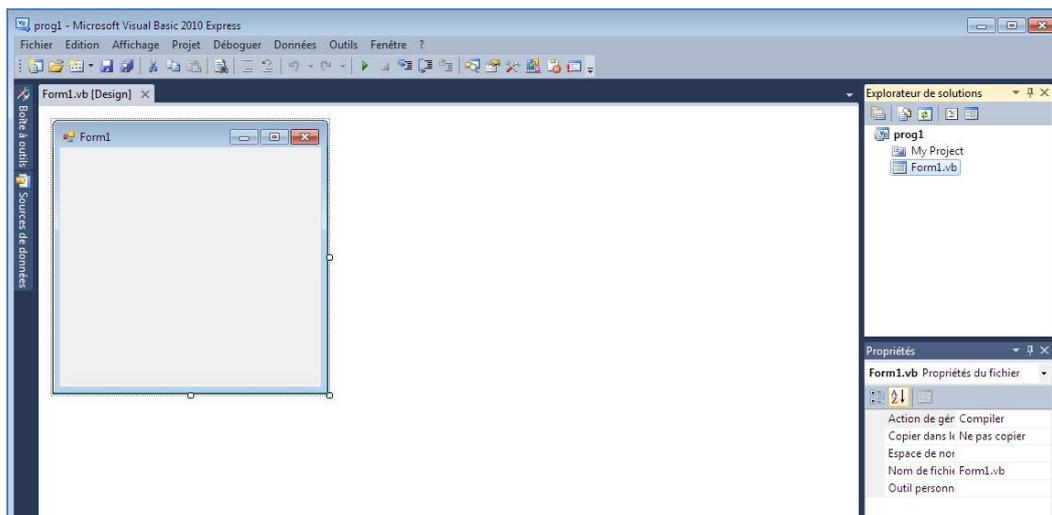
H. TSOUNGUI, ISTV – UVHC, jan. 2018

Ce document va vous permettre de créer votre première application en VB.net. Il est constitué de captures vous menant de la conception des interfaces à l'exécution du projet final.



Capt1 : création du nouveau projet

Le projet est de type **WindowsForms**, c'est-à-dire un programme windows classique avec des fenêtres, des boutons, des menus déroulants, par opposition à un programme en console DOS. Dans cette première étape, vous devez changer le nom du projet. J'ai choisi de l'appeler **prog1**.

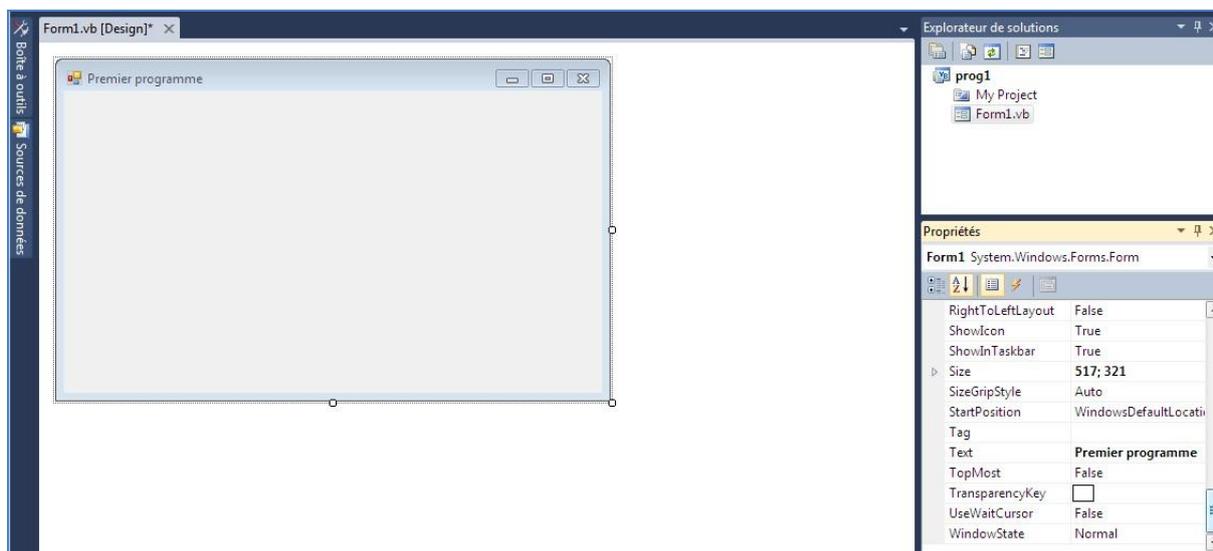


Capt2 : affichage de la première feuille (form1), interface que nous allons modifier

Le programme **prog1** fait partie d'un ensemble d'éléments constituant le nouveau « projet », d'où le nom générique **Myproject** affiché à droite et en dessous du nom du programme.

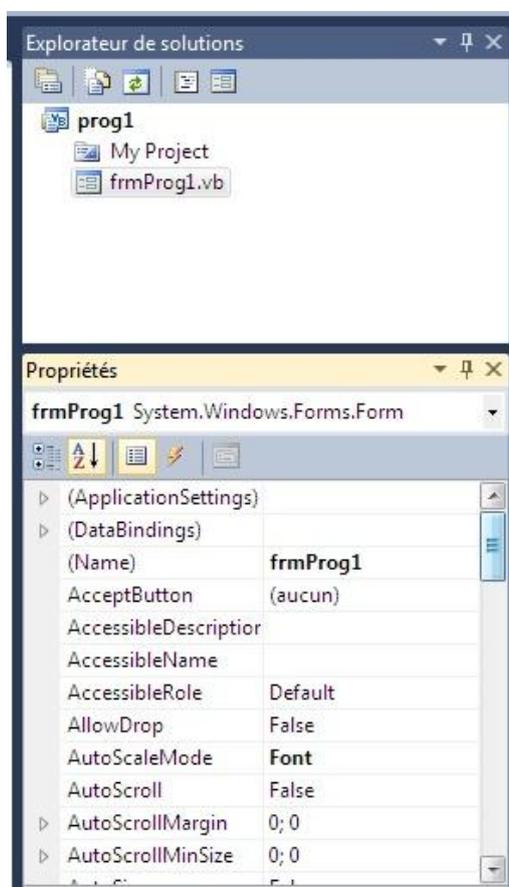
La première feuille appelée form1 est affichée au centre et c'est elle qui constitue la base de notre application. Pour l'instant, elle est vide et n'affiche que le titre « **form1** » et en l'activant, c'est-à-dire en cliquant légèrement dessus, une fenêtre de « **Propriétés** » s'ouvre en bas et à droite.

Vous remarquez que pour le moment la première feuille ne comporte que le titre par défaut « form1 » que vous retrouverez pour chaque début de programme et les trois boutons de réduction, agrandissement et fermeture habituels des applications windows, en haut et à droite de la fenêtre.



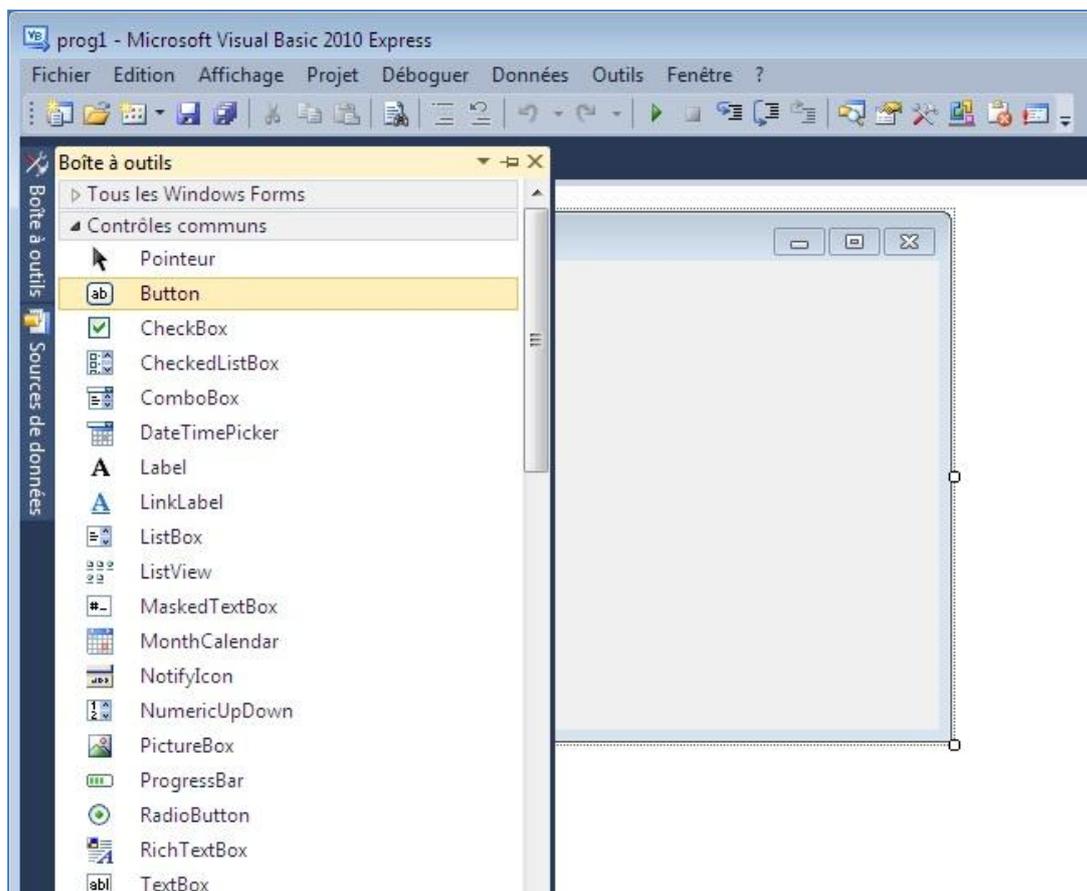
Capt3 : modification du titre de la form dans la propriété « Text »

On peut modifier le titre affiché dans la fenêtre en tapant « **Premier programme** » dans la propriété « Text » de l'objet fenêtre qui s'appelle désormais **frmProg1**. Il faut que l'objet soit « activé » pour modifier ses propriétés. Quand il l'est, on voit les petits points d'activation qui l'entourent.



Capt4 : modification du nom de la feuille en frmProg1

Par la suite, nous appellerons les fenêtres par des noms reconnaissables, pour aider ceux qui vont poursuivre vos développements à se retrouver très vite et à comprendre votre code. Ainsi toutes les feuilles devront avoir un nom (dans la propriété « Name ») commençant par **frm**, tous les boutons de commande devront avoir un nom commençant par **btn**, les labels par **lbl**, les zones de texte modifiables par **txt**, ainsi de suite. Il est très facile de s'habituer à cette convention et je vous recommande de la respecter scrupuleusement.

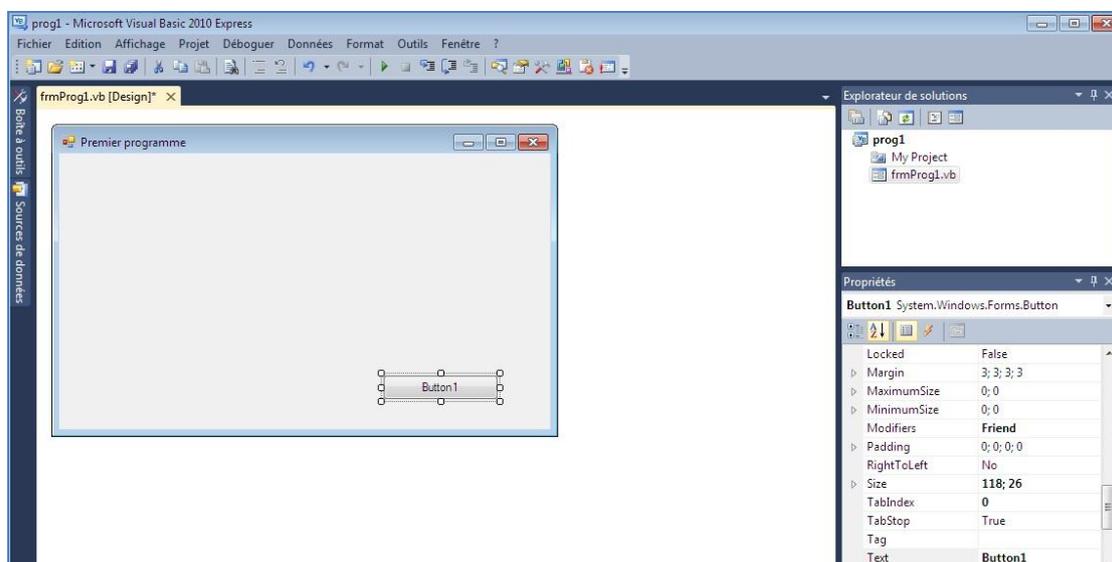


Capt5 : la boîte à outils et sélection du « contrôle » appelé « Button »

### Premier contrôle Visual Basic : « button »

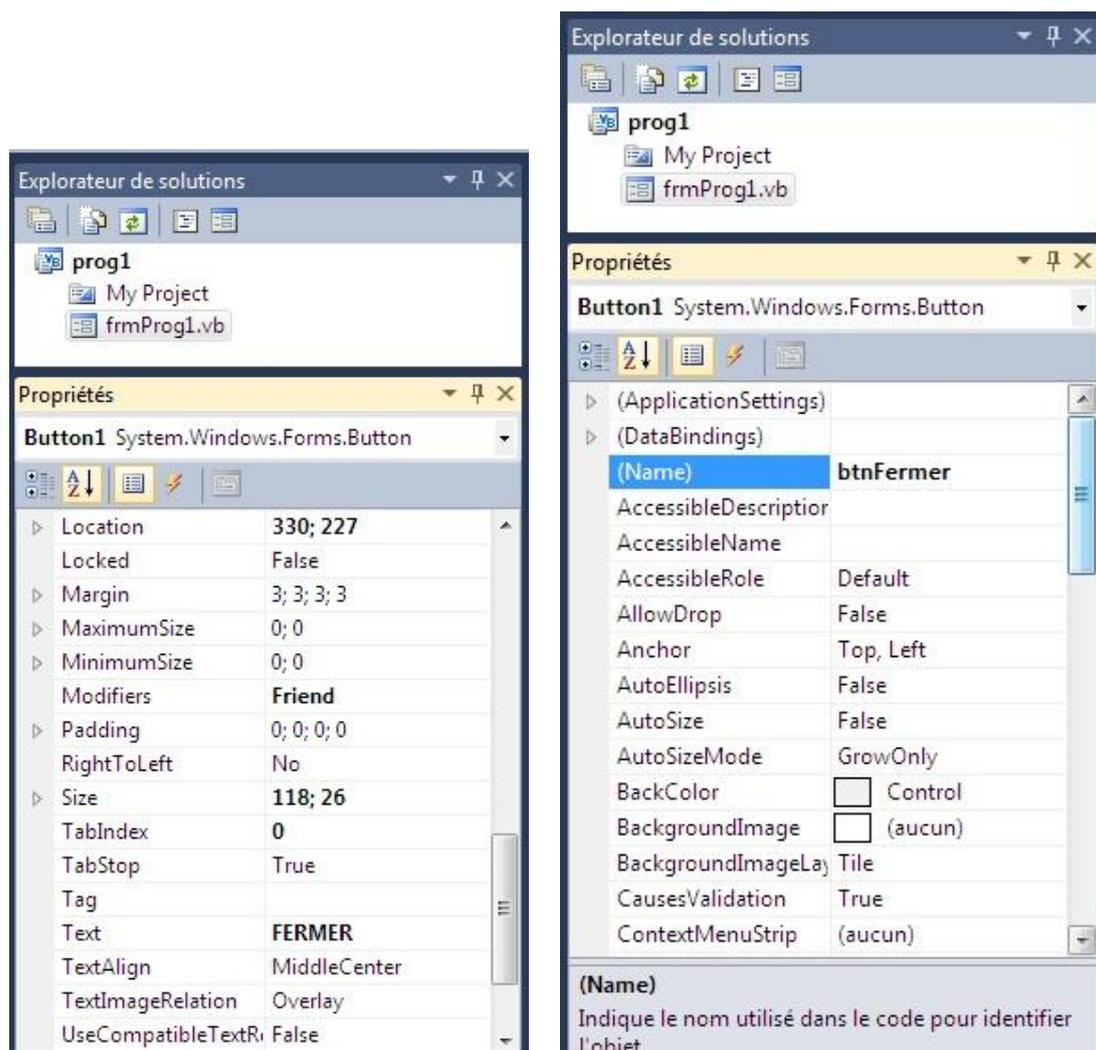
Pour continuer la conception visuelle de notre première feuille, nous faisons maintenant appel à la partie incontournable du langage, la boîte à outils, placée en haut et à gauche de la fenêtre de conception. En-dessous d'elle se trouve la boîte des sources de données qui sert à faciliter les connections à des bases de données.

Après avoir sélectionné le contrôle bouton (capt5 ci-dessus), il faut le dessiner en traçant un petit rectangle sur la feuille. Il faudra procéder ainsi pour dessiner la plupart de contrôles visuels.



Capt6 : bouton de commande créé avec le texte par défaut « Button1 »

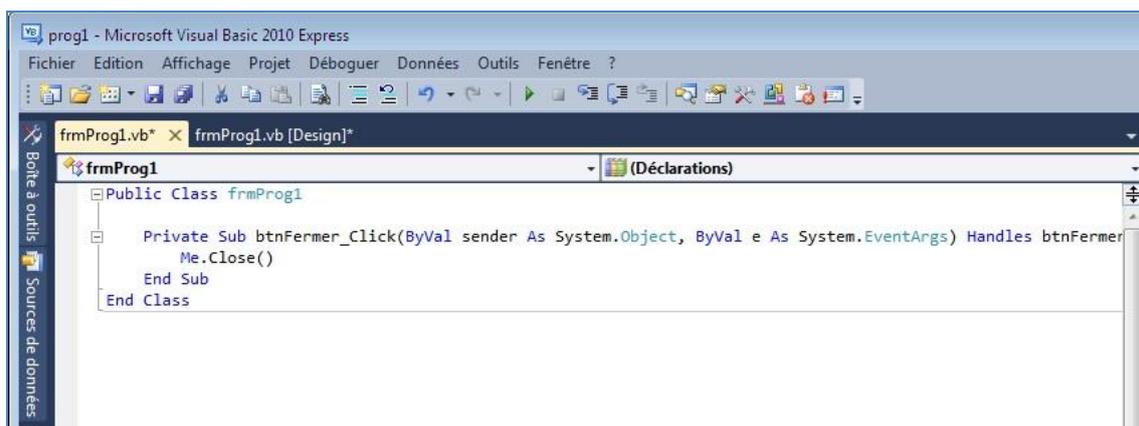
Il convient de modifier deux propriétés : le nom (Name) et le texte affiché sur le bouton (Text).



Capt7 : modification des propriétés de l'objet bouton de commande (Text et Name)

Afin de rendre notre application dynamique, la faire inter-réagir avec son utilisateur, nous utiliserons le bouton de commande précédemment créé pour fermer la fenêtre courante, donc la form qui le porte.

Sur le bouton appelé désormais « **btnFermer** », nous affichons le texte « FERMER » pour faire comprendre son utilité. Ce n'est pas tout. Nous allons gérer à travers ce bouton, notre premier **événement**, le « **click** » sur le bouton qui doit déclencher la fermeture de la fenêtre courante.

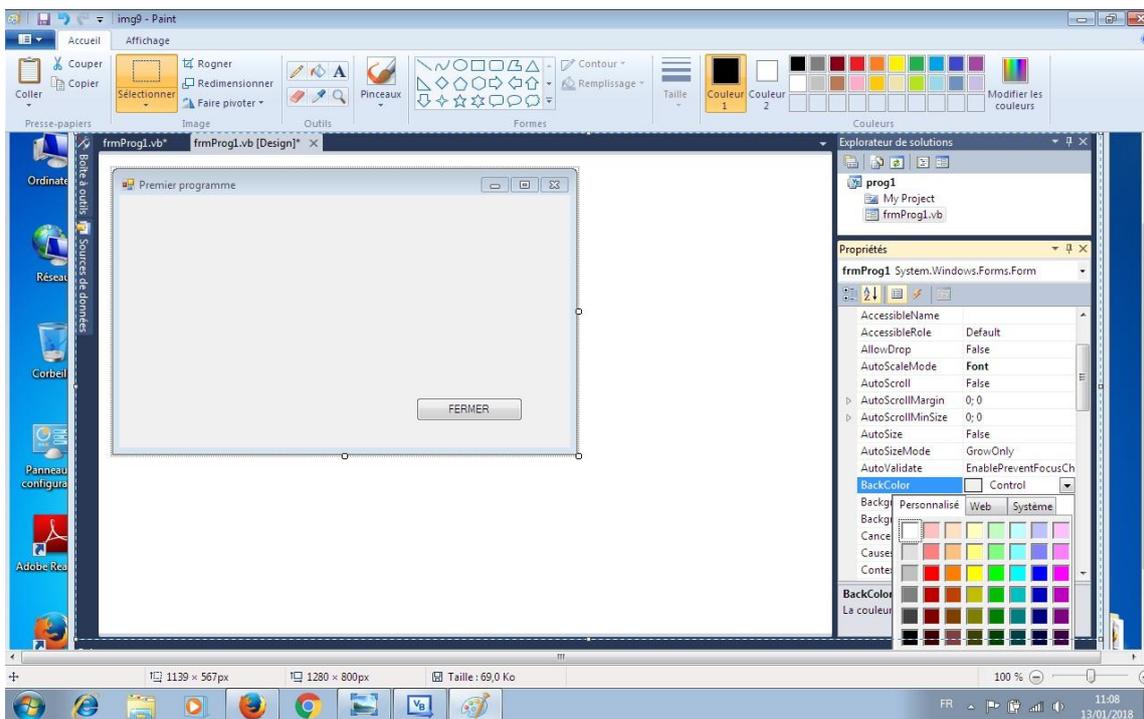


Capt8 : code de gestion de l'évènement click sur le bouton de commande

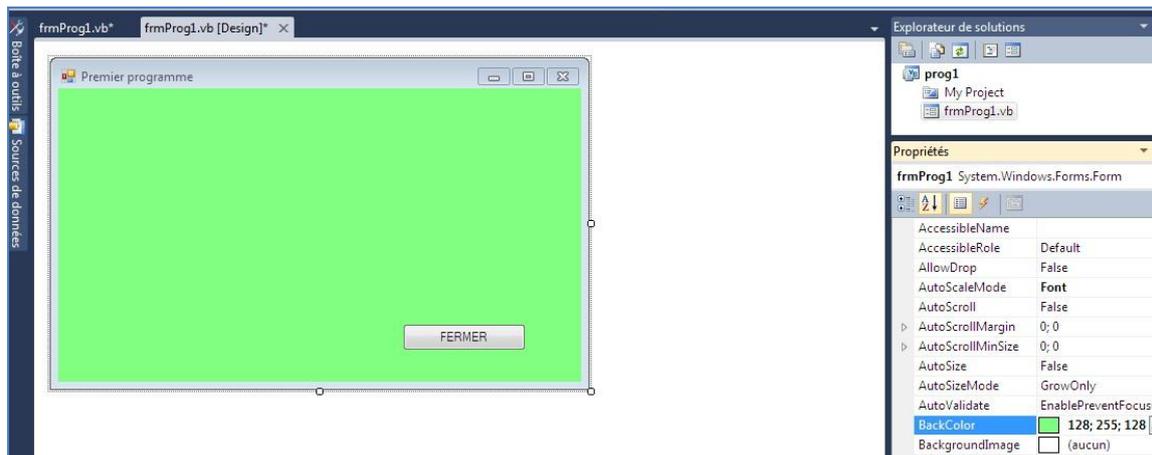
Le double-click sur le bouton FERMER ouvre une **procédure événementielle** que nous allons compléter en tapant les lignes de code des actions à effectuer. On tape **Me.close()** qui fermera la form ouverte.

#### Utilisation des propriétés pour modifier l'interface

On se propose de modifier la couleur du fond de la fenêtre en la faisant passer au vert.



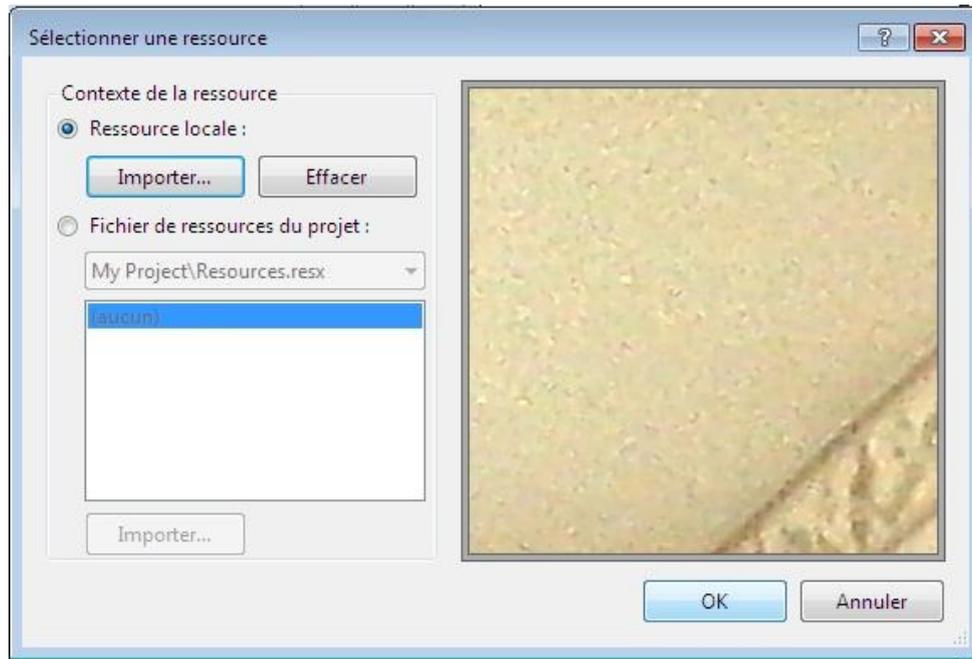
Capt9 : sélection de la couleur pour la propriété « BackColor »



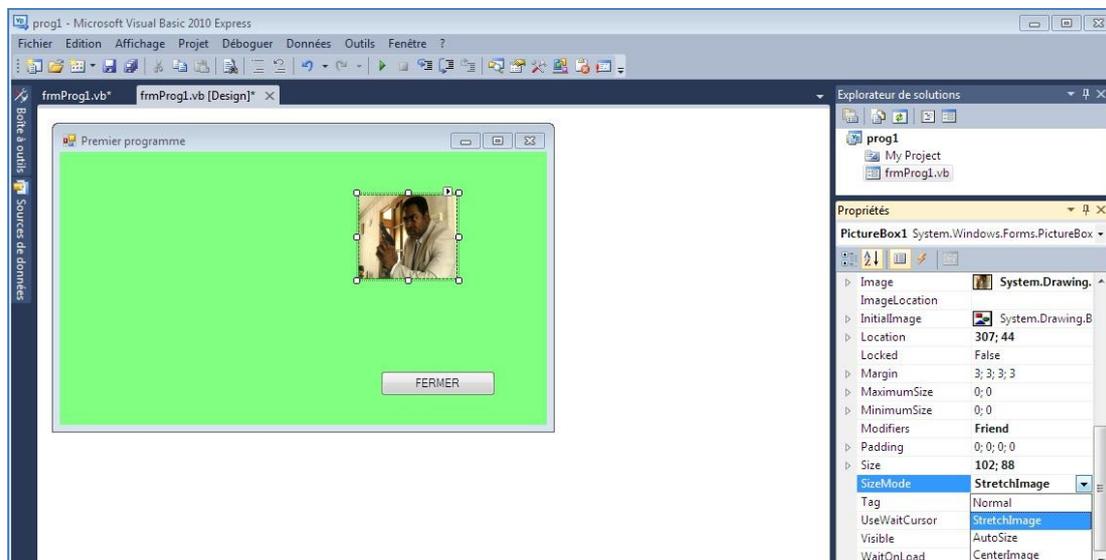
Capt10 : la couleur du fond change dès le choix dans la grille des couleurs

#### Utilisation du contrôle « Image » (Picture)

Pour ajouter une image sur notre feuille frmProg1, on sélectionne le contrôle **Picture** que l'on dessine sur la feuille. On va ensuite dans la propriété **Image** pour aller choisir/importer une image dont nous connaissons le chemin pour l'atteindre. L'image importée remplit le contrôle de façon anarchique. Elle est illisible.

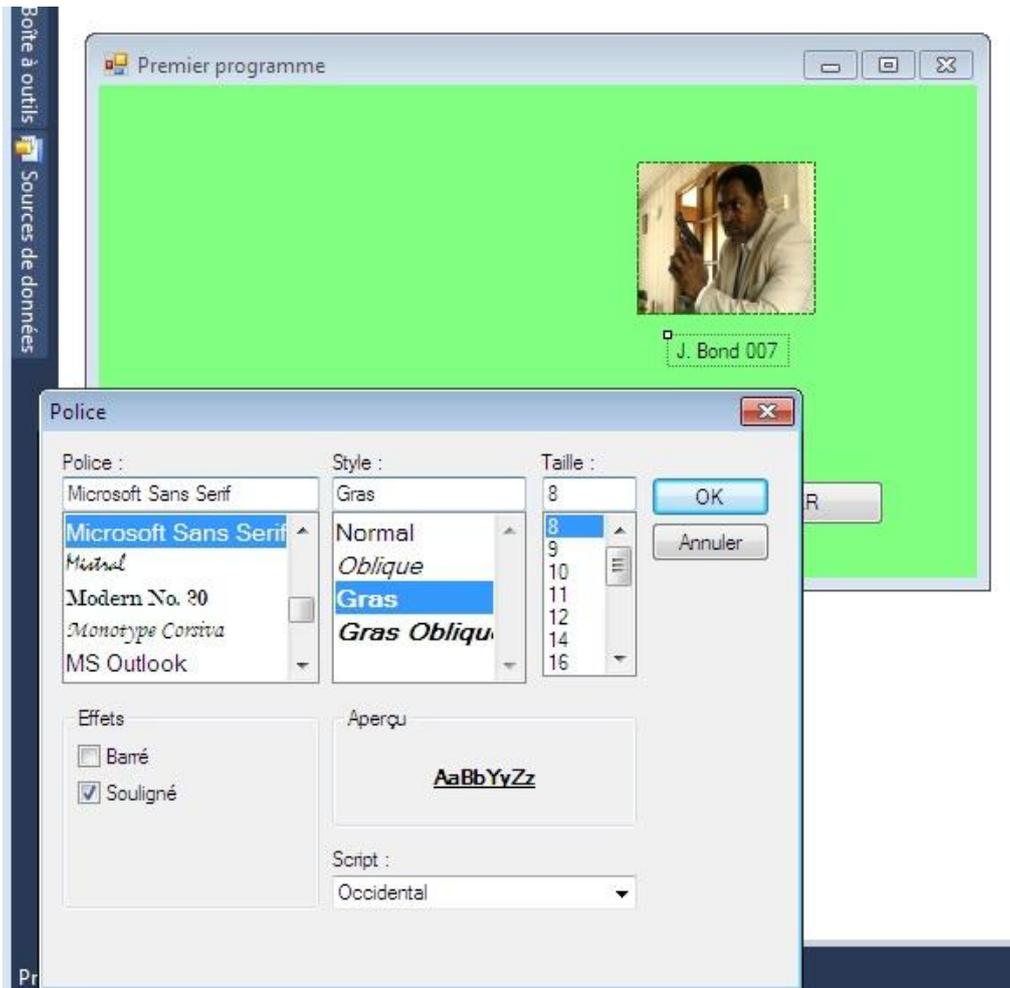


Capt11 : sélection et importation d'une image

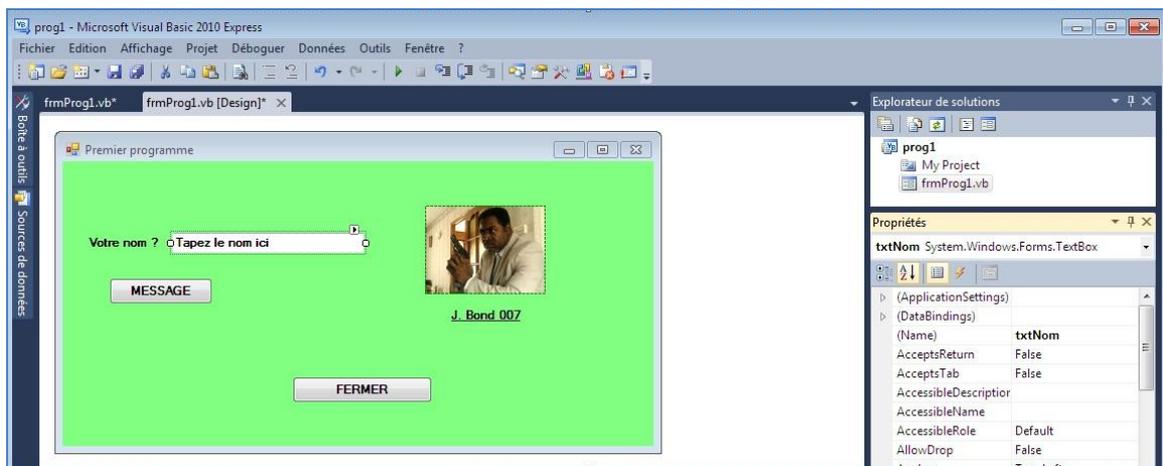


Capt12 : réorganisation de l'affichage de l'image

La propriété **SizeMode** à « stretchImage » de l'objet **Picture** permet de l'afficher correctement dans le contrôle image.

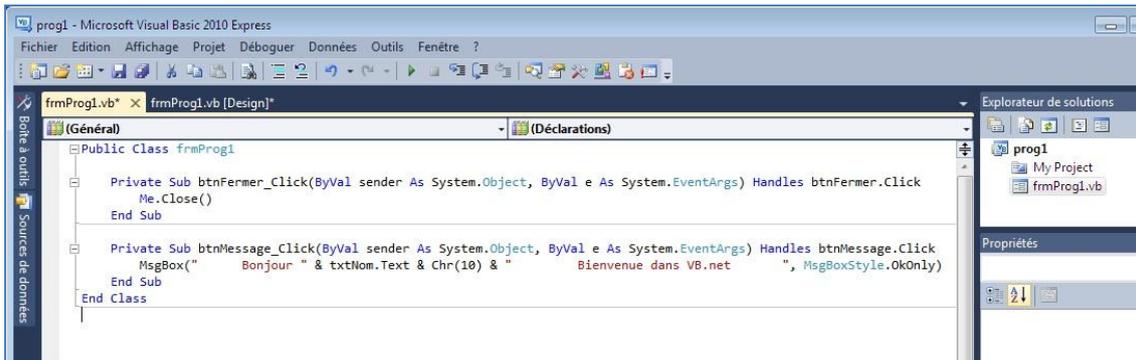


Capt13 : ajout d'un label sous l'image et modification de sa police



Capt14 : ajout d'une zone de texte et d'un deuxième contrôle bouton de commande

Dans ces dernières captures, on a ajouté une **zone de texte modifiable** (objet `TextBox`), c-à-d qu'on peut saisir un texte variable. Puis au click sur le bouton appelé `btnMessage` on va afficher une **boîte de dialogue** reprenant le texte `txtNom` qui a été saisi. Les boîtes de dialogue permettent le dialogue entre l'utilisateur et son programme. Il existe plusieurs types de boîtes de dialogues prédéfinies que le programmeur, vous, n'avez plus qu'à modifier à son goût. La boîte basique est générée par la fonction `MsgBox("Texte du message à afficher")`. Dans notre projet, on a utilisé le code `MsgBox(" Bonjour " & txtNom.Text & Chr(10) & " Bienvenue dans VB.net ")`. Le caractère `Chr(10)` permet de faire un *saut de ligne* et le `&` permet la concaténation des textes (comme une addition de chaînes de caractères). `txtNom.Text` représente le texte de la propriété `Text` de l'objet zone de texte, en fait, ce que l'on tape dans la zone de texte.



```
Public Class frmProg1
    Private Sub btnFermer_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnFermer.Click
        Me.Close()
    End Sub

    Private Sub btnMessage_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnMessage.Click
        MsgBox("  Bonjour " & txtNom.Text & Chr(10) & "      Bienvenue dans VB.net      ", MsgBoxStyle.OkOnly)
    End Sub
End Class
```

Capt15 : code du bouton btnMessage



Capt16 : test du bouton de message



Capt17 : re-test du bouton avec un autre message