

Programmation Impérative Visual studio

INFO 2

Année universitaire 2022-2023

VB .NET (VB et premier programme)

- Ce document introduit l'outil de développement de programme Visual Studio .NET dans le contexte de Visual Basic.NET.
- Nous allons voir que les programmes sont fondés sur le framework .NET i.e. qu'ils utilisent essentiellement les classes du framework.
- En conséquence, nous devons donc apprendre à utiliser ces classes.
- L'avantage de cette approche est que ces mêmes classes peuvent être utilisées dans les autres langages de programmation: maintenant, une librairie de classes uniformes est disponible pour toute la plate-forme Windows.

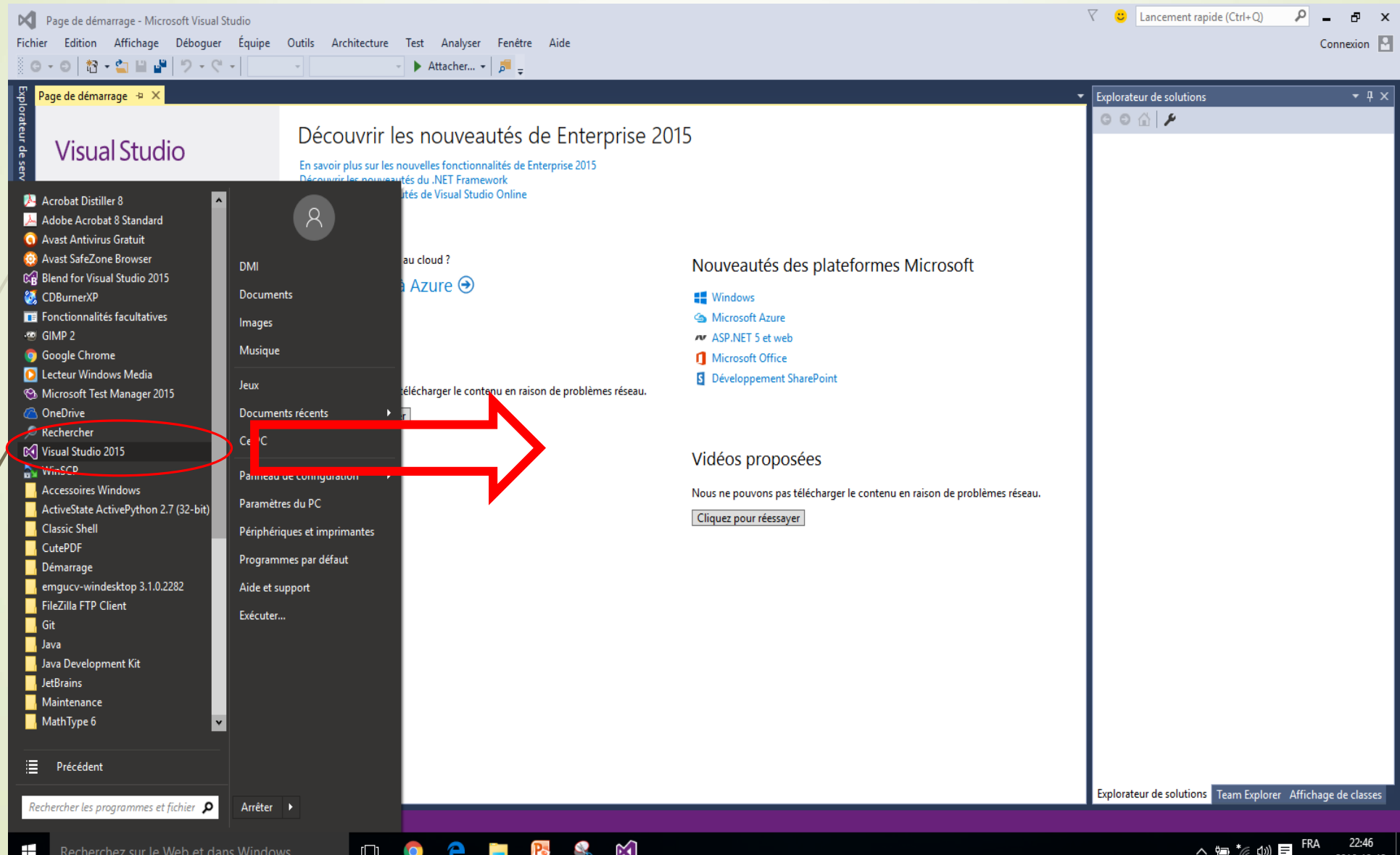
VB .NET (VB et premier programme)

■ Les Fenêtres

- Les fenêtres (Windows) sont à la base des interactions entre l'utilisateur et un programme d'application.
- Un programme qui interagit avec l'utilisateur possède toujours une fenêtre principale appelée "fenêtre principale de l'application".
- Cette section permet d'expliquer comment créer la fenêtre principale d'une application.

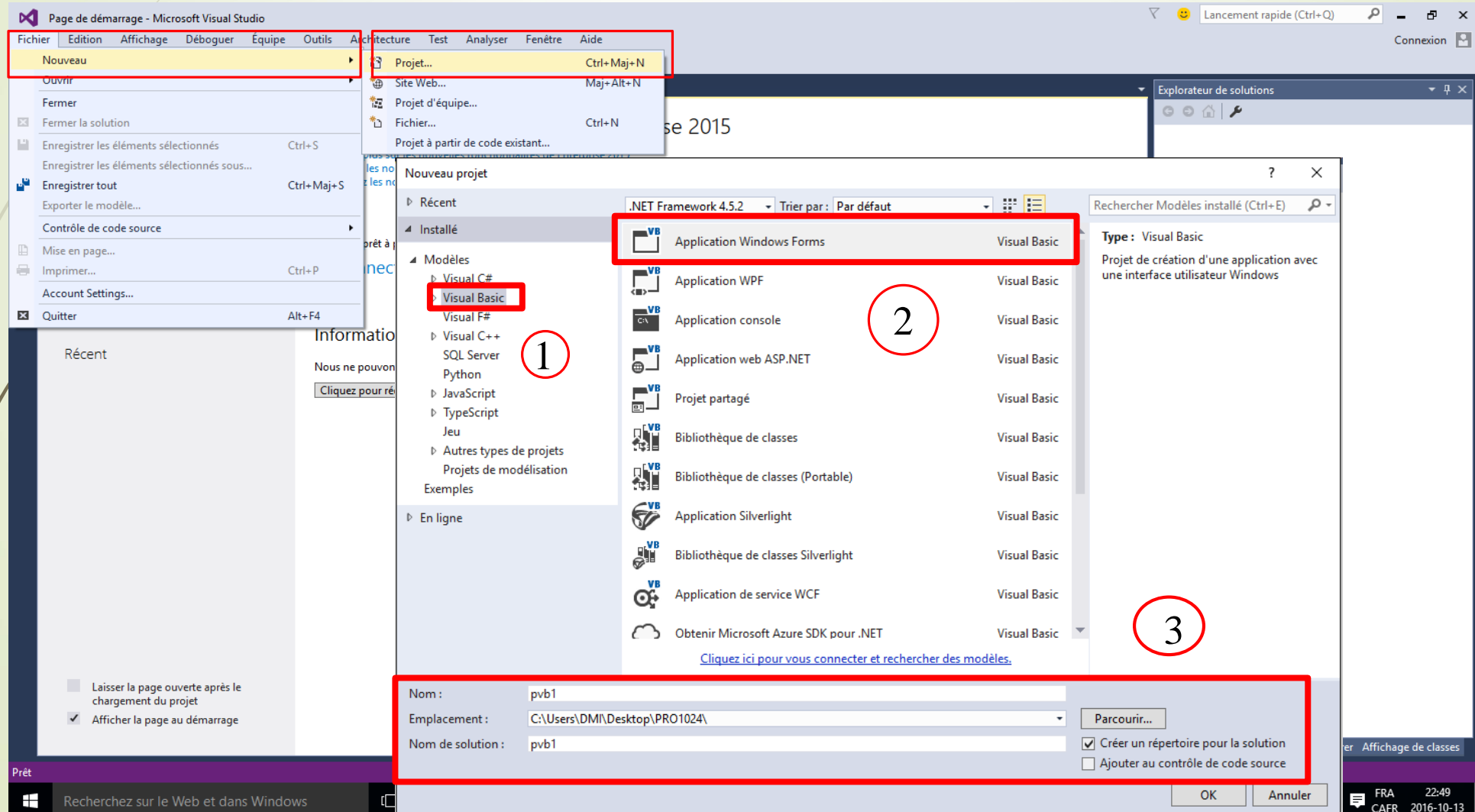
VB .NET (VB et premier programme)

➡ D'abord, démarrons Visual Studio comme suit:



VB .NET (VB et premier programme)

- Ensuite nous créons un nouveau projet en sélectionnant **FILE-NEW-PROJECT** à partir du menu ou **NEW PROJECT** à partir de la page de démarrage.

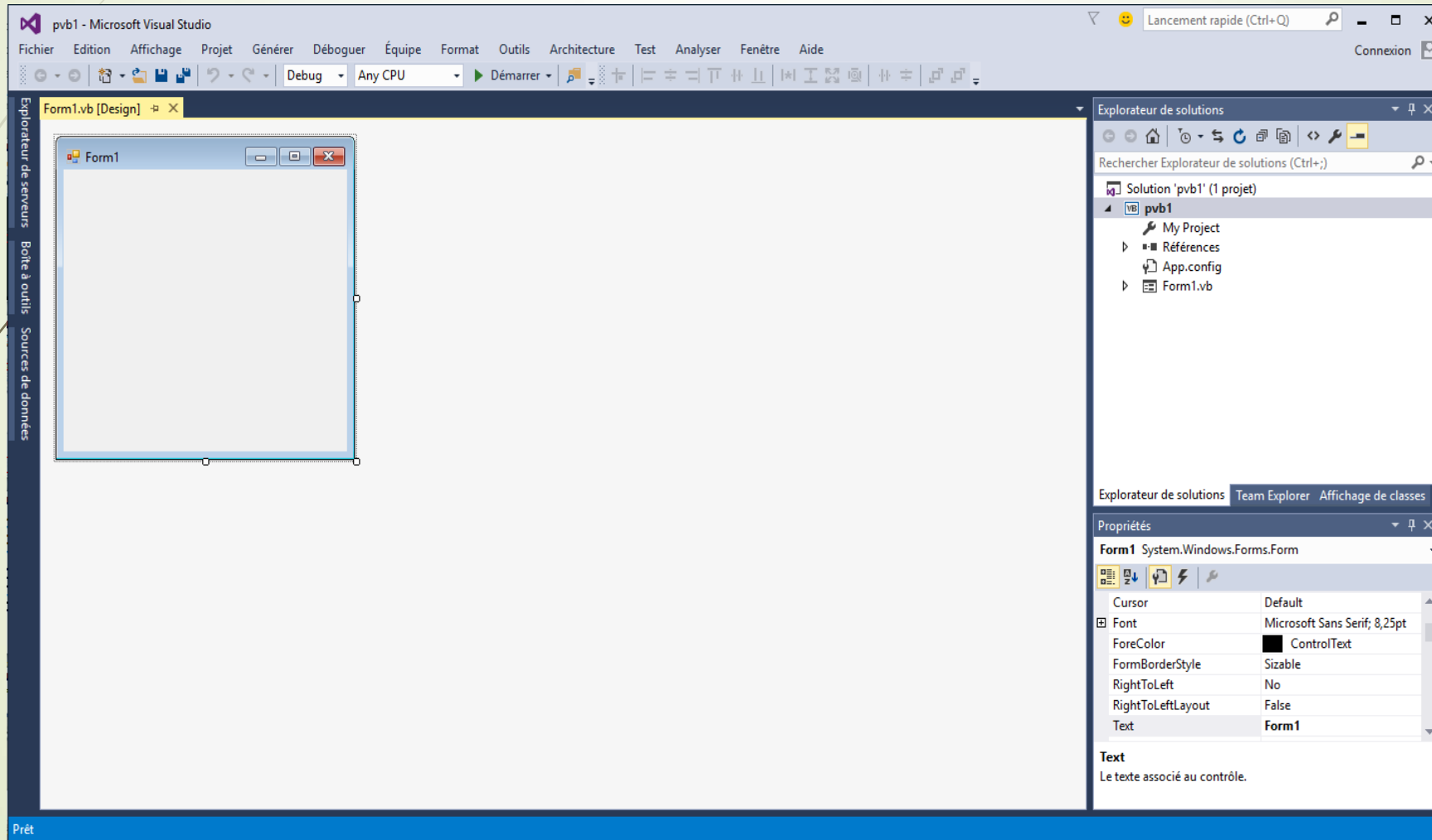


VB .NET (VB et premier programme)

- Nous pouvons remarquer que Visual Studio supporte les langages de programmation à partir du même outil de développement (Visual Studio).
- Nous pouvons donc créer des projets dans les langages: Visual Basic (VB), Visual C# (C sharp ou dièse) et Visual C++.
- Dans notre cas, nous sélectionnons:
 - Le langage VB
 - Le type d'application (Windows Application)
 - Le répertoire-racine (location)
 - Le nom du projet (pvb1).
- La ligne au bas de "Location:" indique que Visual Studio va créer un nouveau répertoire correspondant au nom du projet "pvb1". Dans ce nouveau répertoire Visual Studio va placer tous les éléments qui constituent ensemble notre projet de développement du programme (application).

VB .NET (VB et premier programme)

➡ Ensuite, en pressant **ok** nous obtenons la fenêtre suivante:

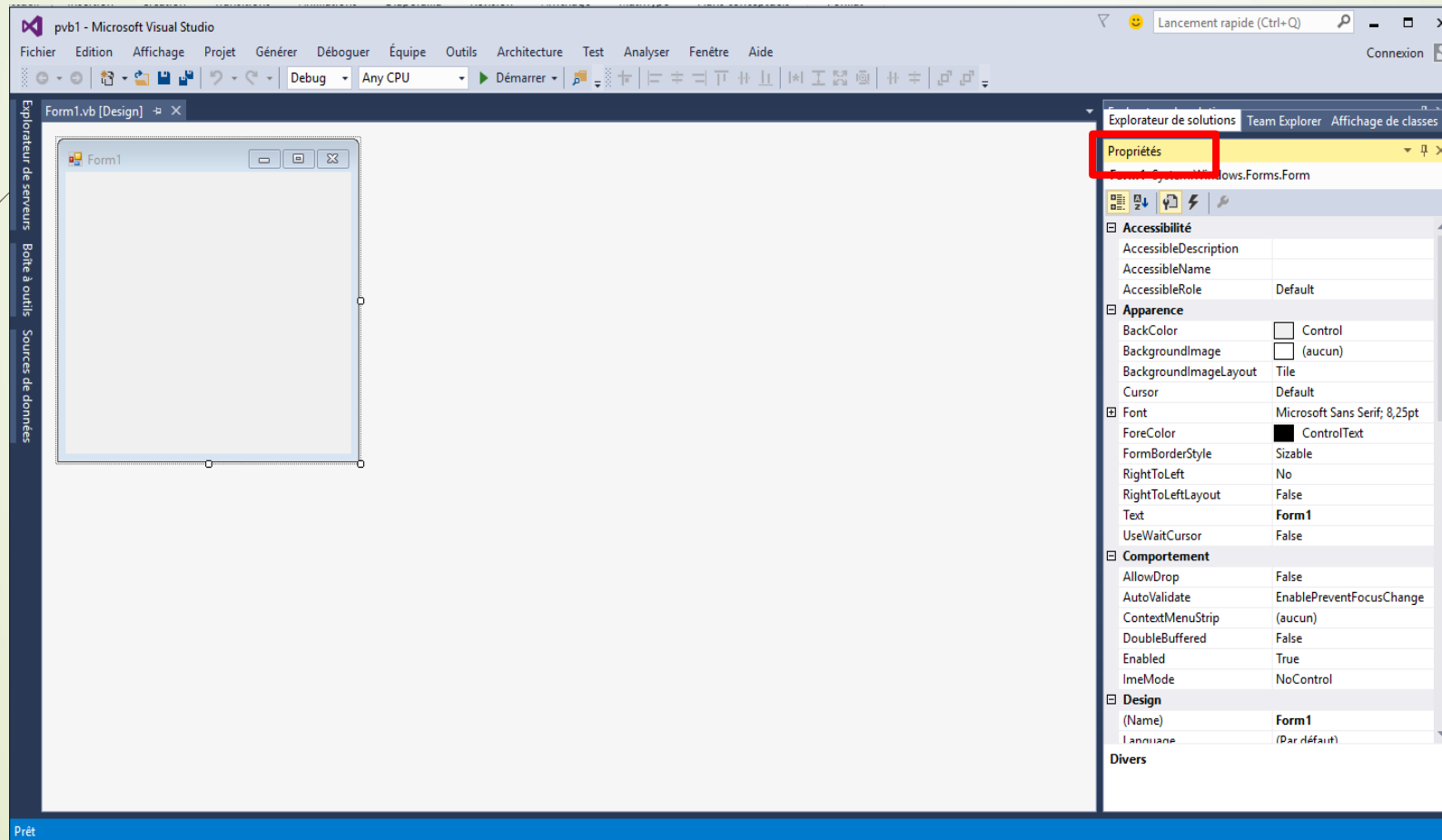


VB .NET (VB et premier programme)

- En haut de la figure précédente, nous pouvons retrouver le menu principal.
- Au milieu à gauche, nous trouvons la section qui permet de développer (Design) la fenêtre principale (Form1.vb) de notre programme.
- Du côté droit, nous disposons de plusieurs onglets qui montrent des éléments du projet. Actuellement, nous avons choisi l'option **Solution Explorer** (voir le menu View).
- L'objet Form1 est stocké dans le fichier **Form1.vb**. **Form1.vb** est un élément du projet "pvb1" qui lui-même est un élément de la solution (une solution peut contenir plusieurs projets).

VB .NET (VB et premier programme)

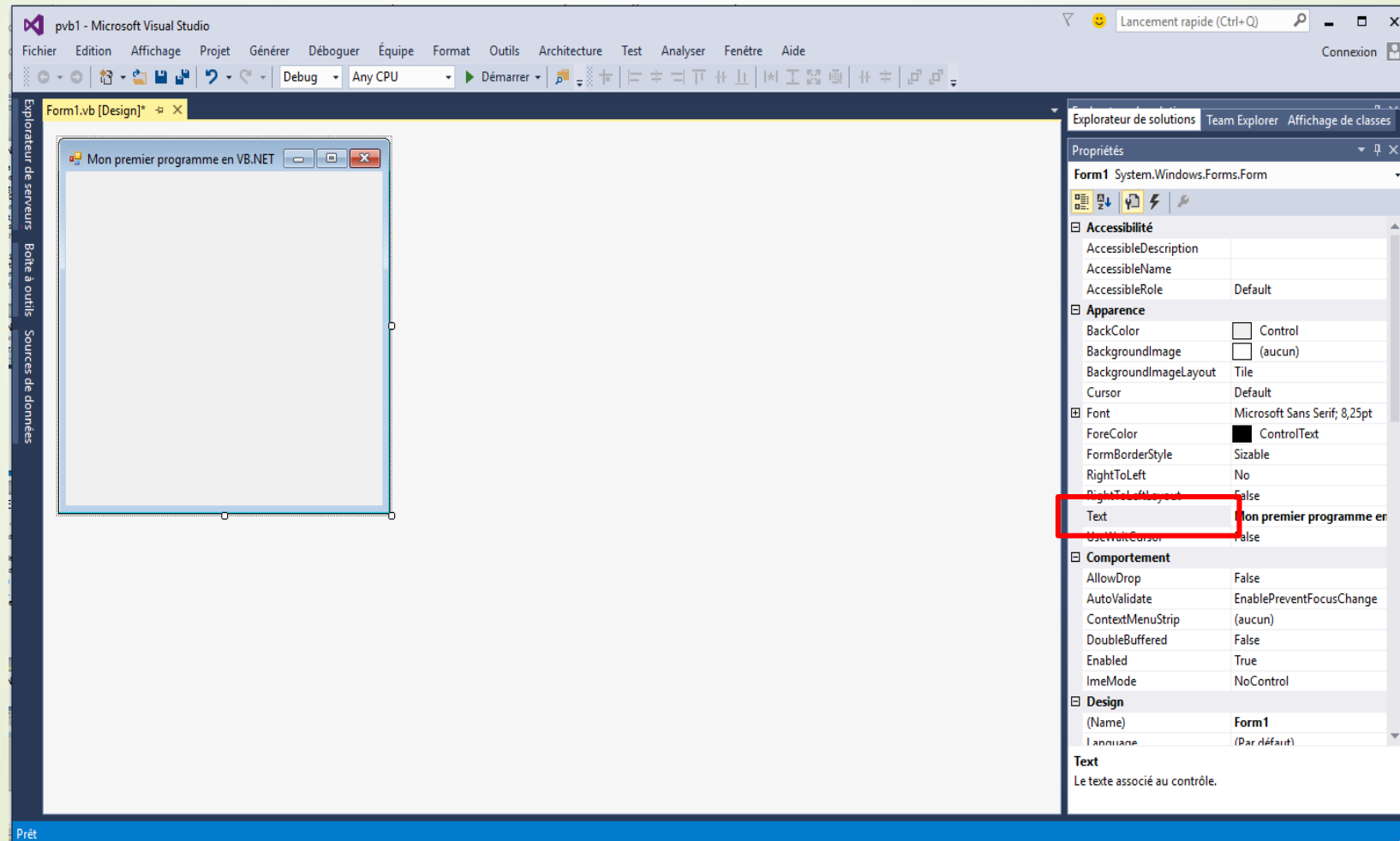
- ➡ La figure suivante montre les propriétés de la forme Form1 (onglet-propriétés):



VB .NET (VB et premier programme)

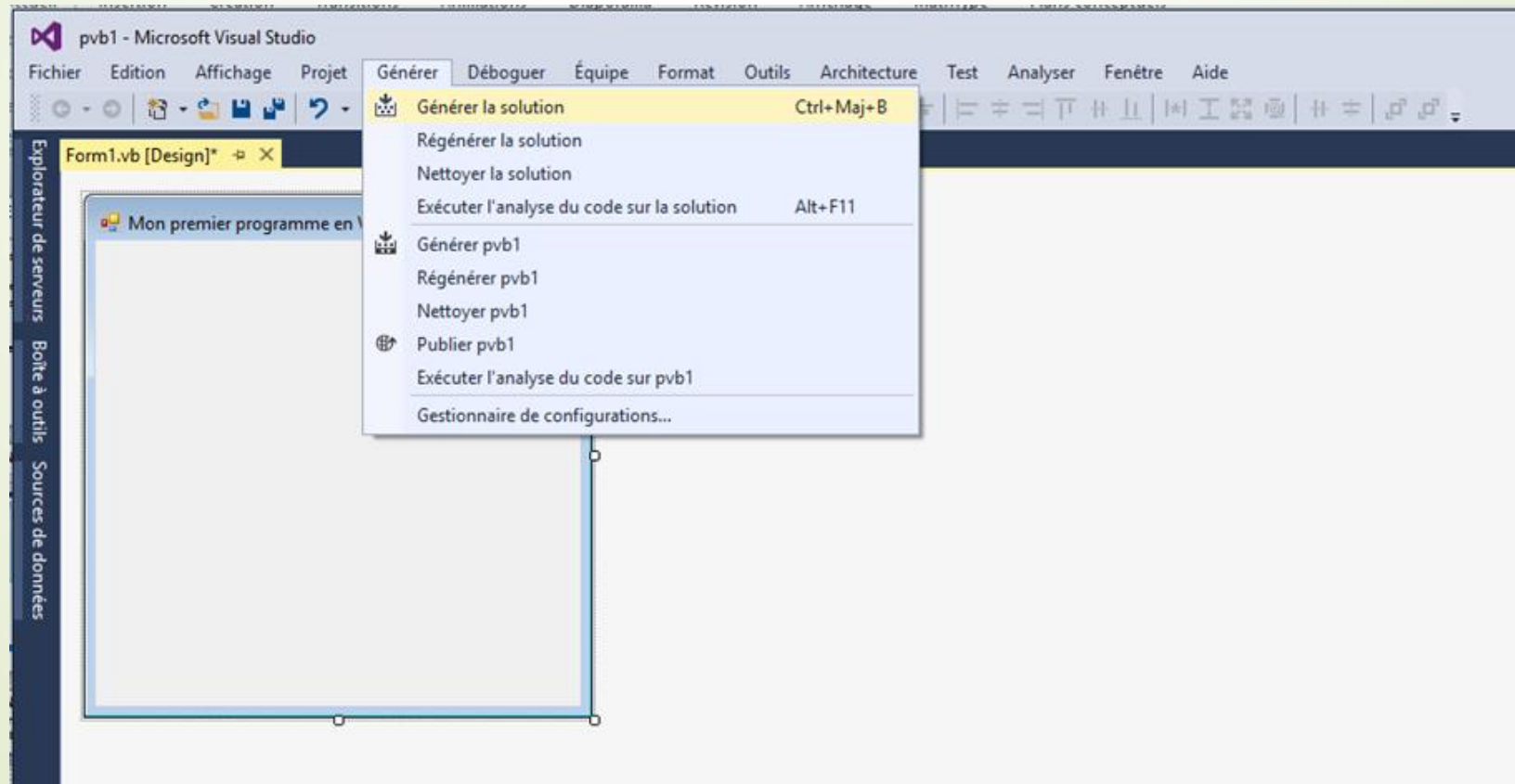
- La propriété "**Name**" d'un objet est son nom. En général, nous devrions donner un nom significatif à chaque élément.
- Le nom par défaut de la forme est **Form1**.
- Nous pouvons changer la valeur (l'état) d'une propriété en changeant la partie droite (la valeur) de la propriété.
- Par exemple, si nous changeons la propriété **Text** de la forme, c'est le nom au haut de la forme qui sera alors modifié.

VB .NET (VB et premier programme)



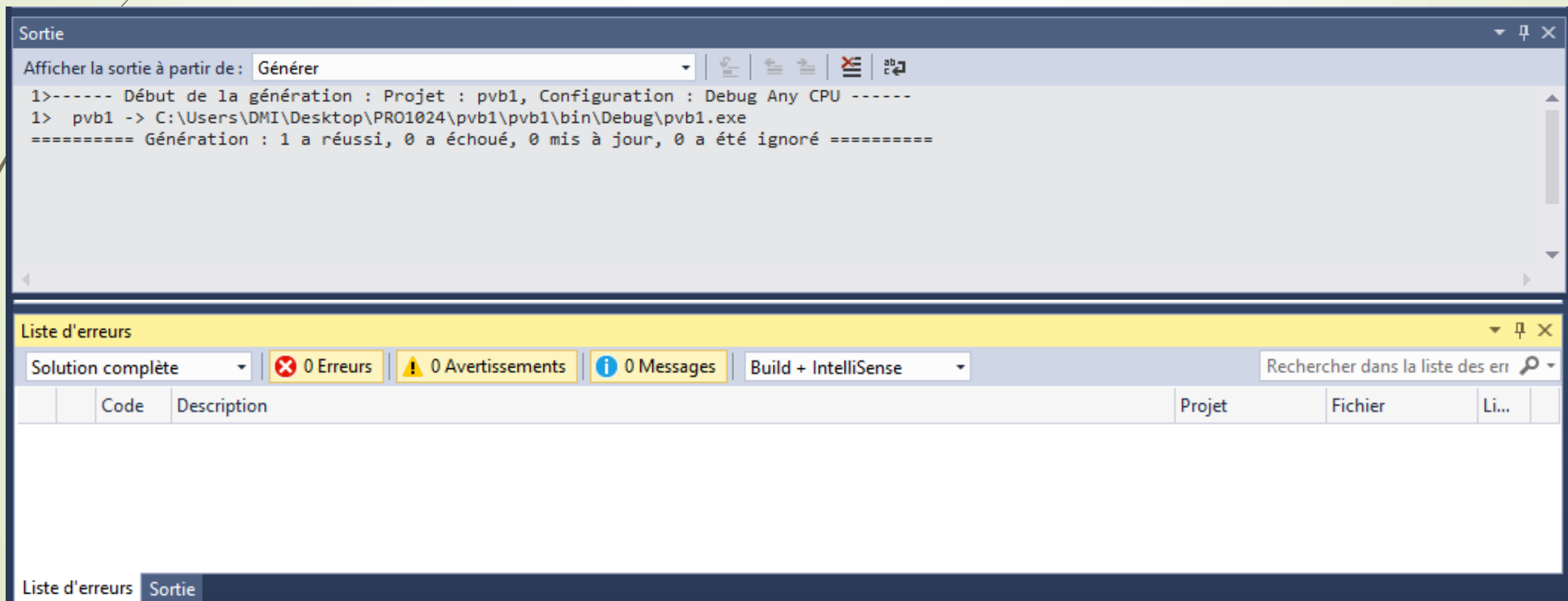
VB .NET (VB et premier programme)

- Puis, nous construisons (compiler) le programme avec le menu "**Build/Build Solution**":

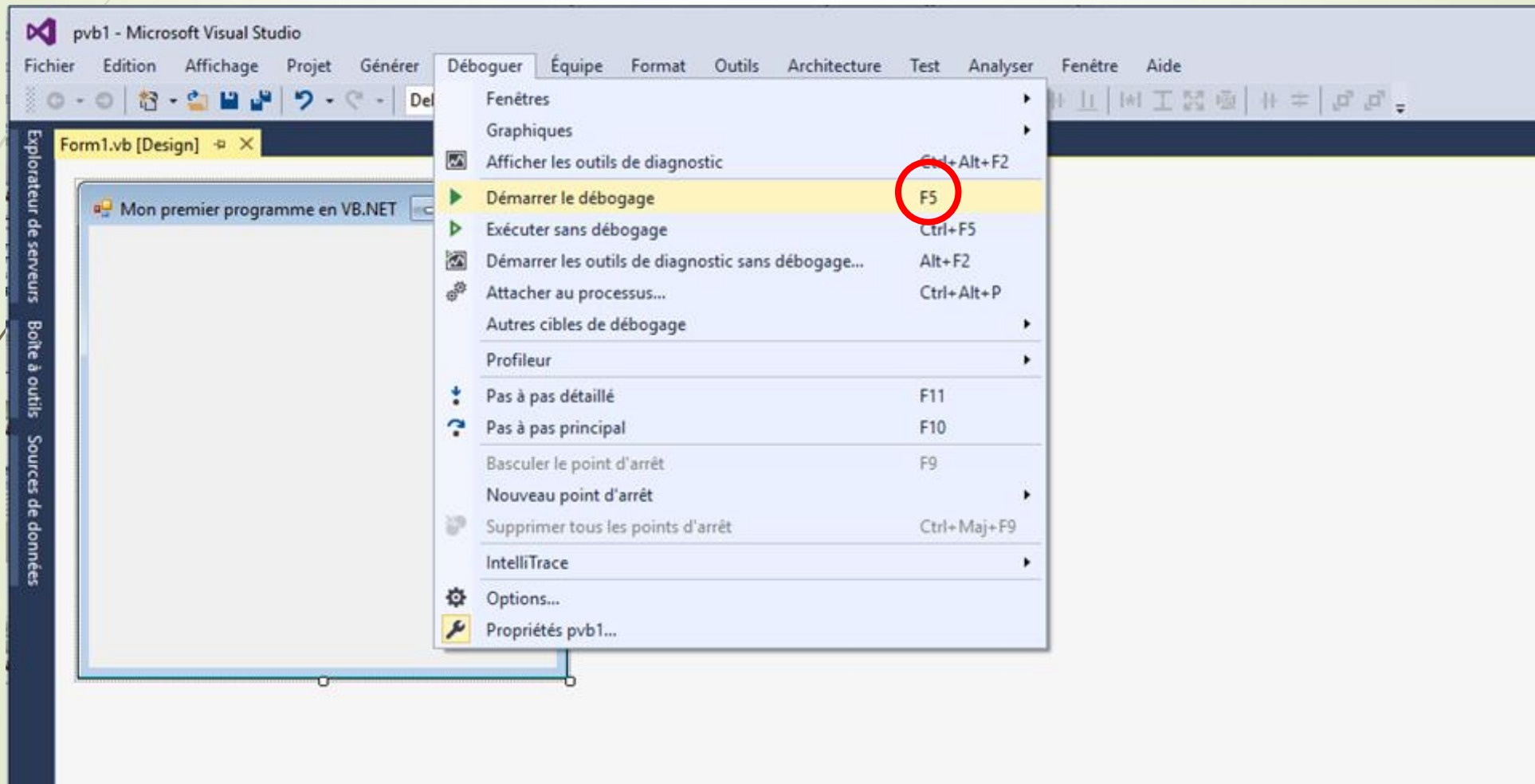


VB .NET (VB et premier programme)

- Le résultat du **BUILD** devrait apparaître au bas de la figure précédente: "**Build: 1 succeeded, 0 failed**".
- Si des erreurs surviennent, alors l'onglet "**Task List**" donne la liste des erreurs et des commentaires sur ces erreurs.
- Ensuite, nous pouvons exécuter le programme avec le menu "**Debug/Start**":

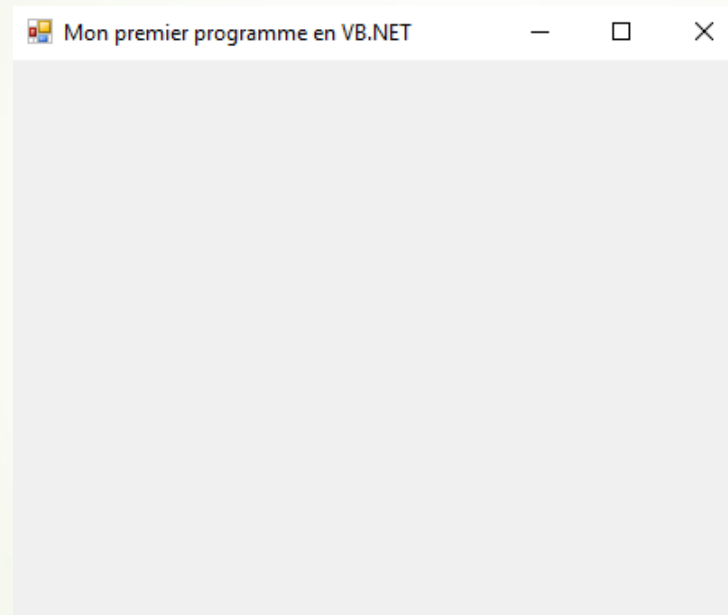


VB .NET (VB et premier programme)



VB .NET (VB et premier programme)

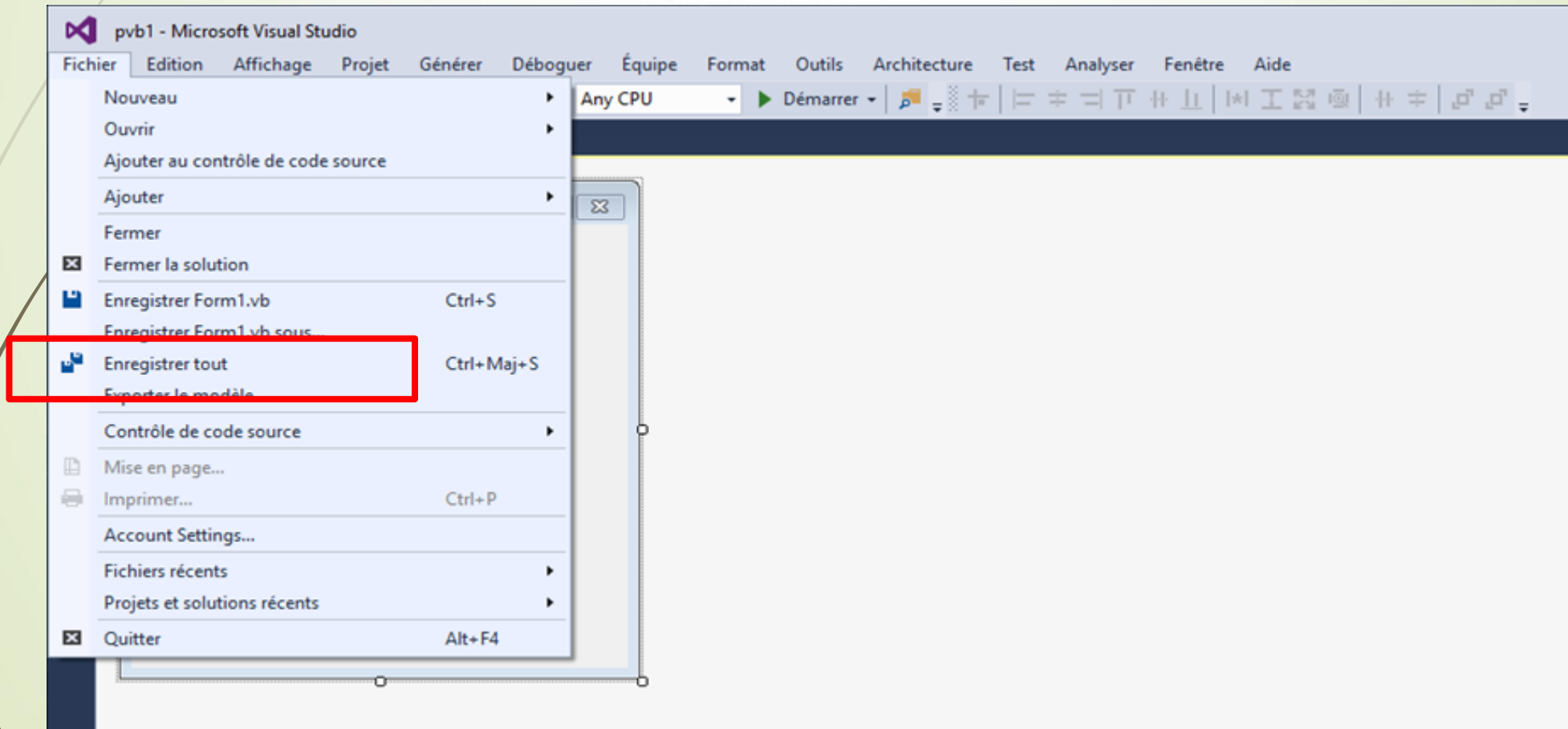
- Le résultat de l'exécution de notre programme est une fenêtre vide



- Dans sa forme actuelle, notre programme ne fait rien d'autre que d'afficher sa fenêtre principale.
- Nous pouvons stopper le programme en pressant le bouton "X" en haut à droite de la fenêtre (ou avec le menu "**Debug/Stop Debugging**").

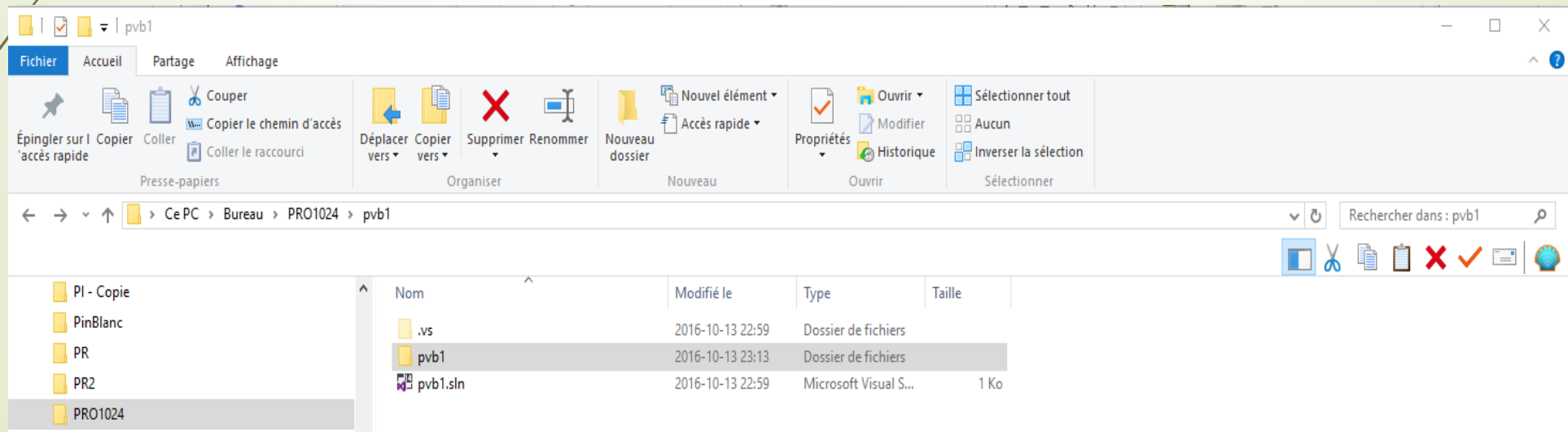
VB .NET (VB et premier programme)

- **Sauvegarde du projet:** Le menu "**File/Save All**" permet de sauvegarder sur disque tous les éléments du projet:



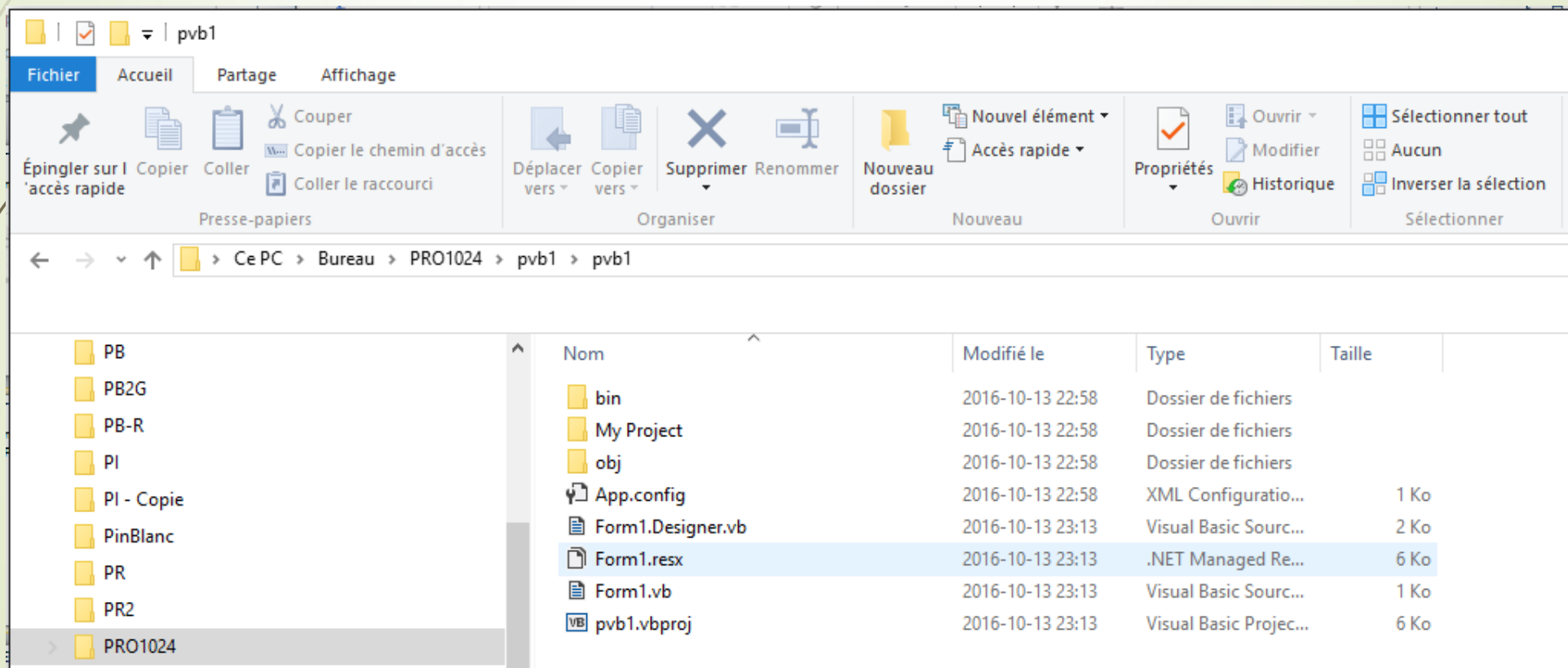
VB .NET (VB et premier programme)

- La figure suivante montre le contenu du répertoire du projet:



VB .NET (VB et premier programme)

- La figure suivante montre le contenu du répertoire du projet (sous-répertoire pvb1):

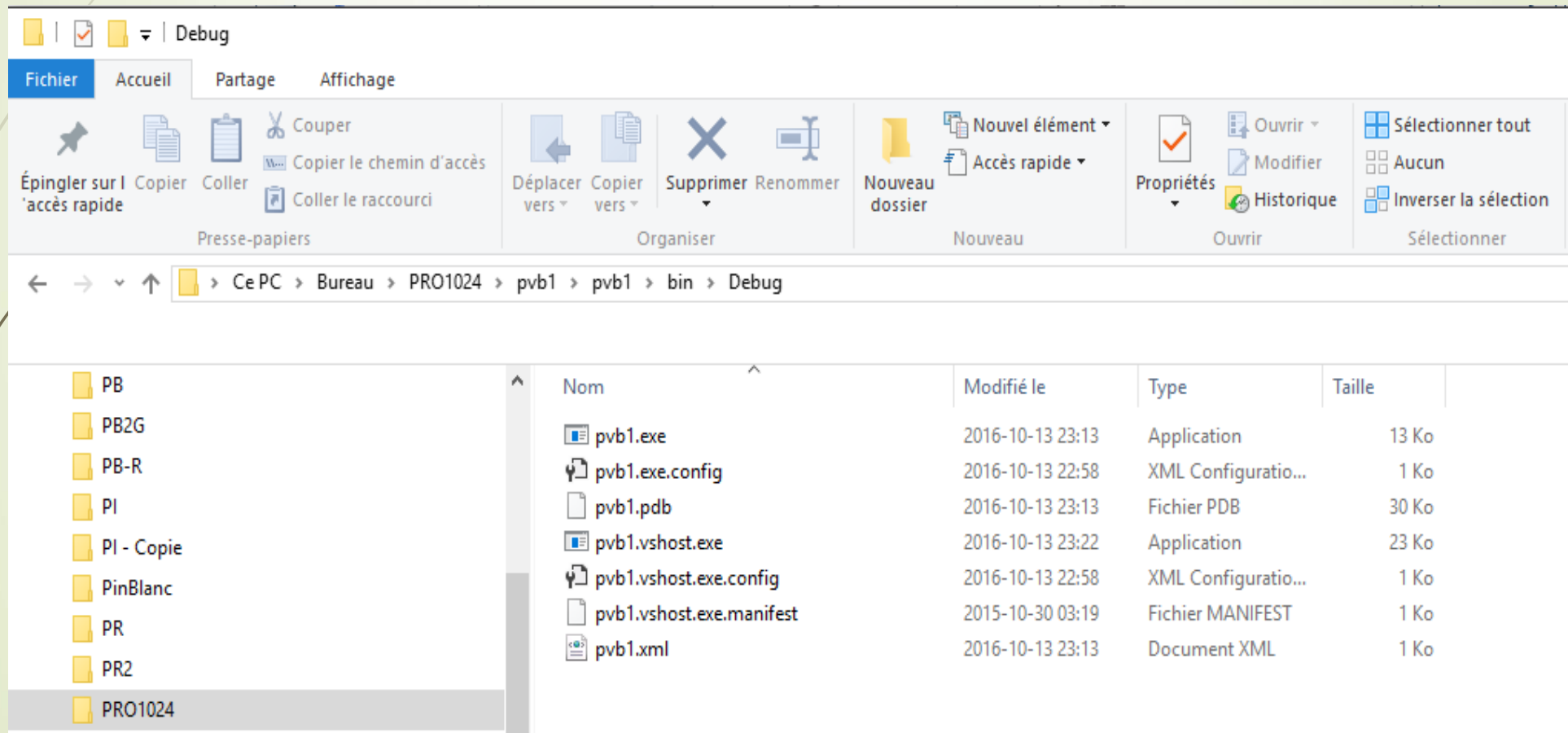


VB .NET (VB et premier programme)

- Le fichier "**pvb1.sln**" est le fichier qui contrôle toute la "solution".
- Une solution contient normalement un projet, mais elle peut aussi contenir plusieurs projets qui sont inter-reliés.
- Si nous effectuons un double click sur ce fichier dans Explorer, Visual Studio est alors démarré et la solution est chargée.

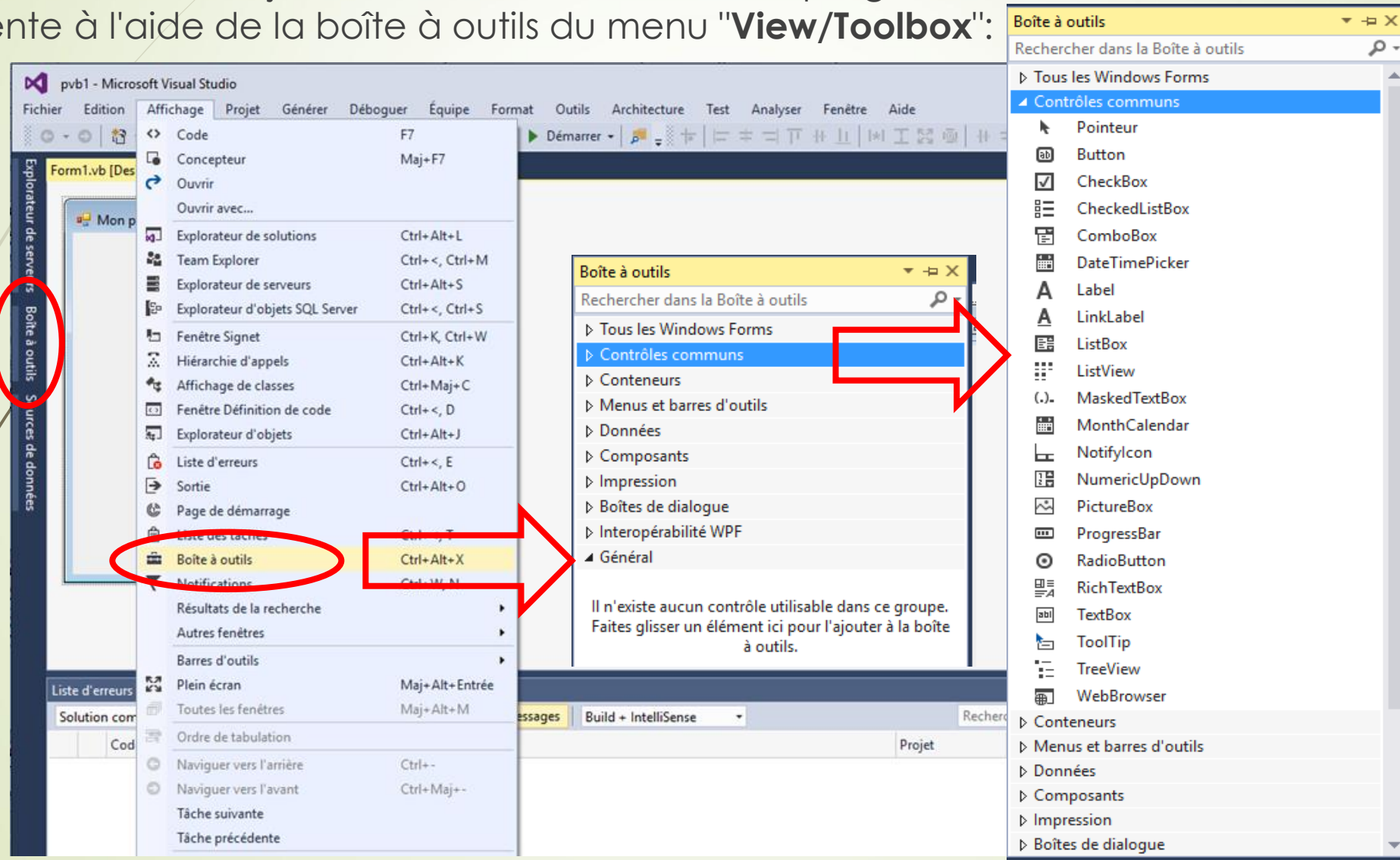
Le sous-répertoire "**Debug>bin**" (binaire) contient le programme (exécutable) de notre application "**pvb1.exe**".

VB .NET (VB et premier programme)



VB .NET (VB et premier programme)

- La boîte à outils: Nous ajoutons des fonctionnalités au programme de la section précédente à l'aide de la boîte à outils du menu "**View/Toolbox**":

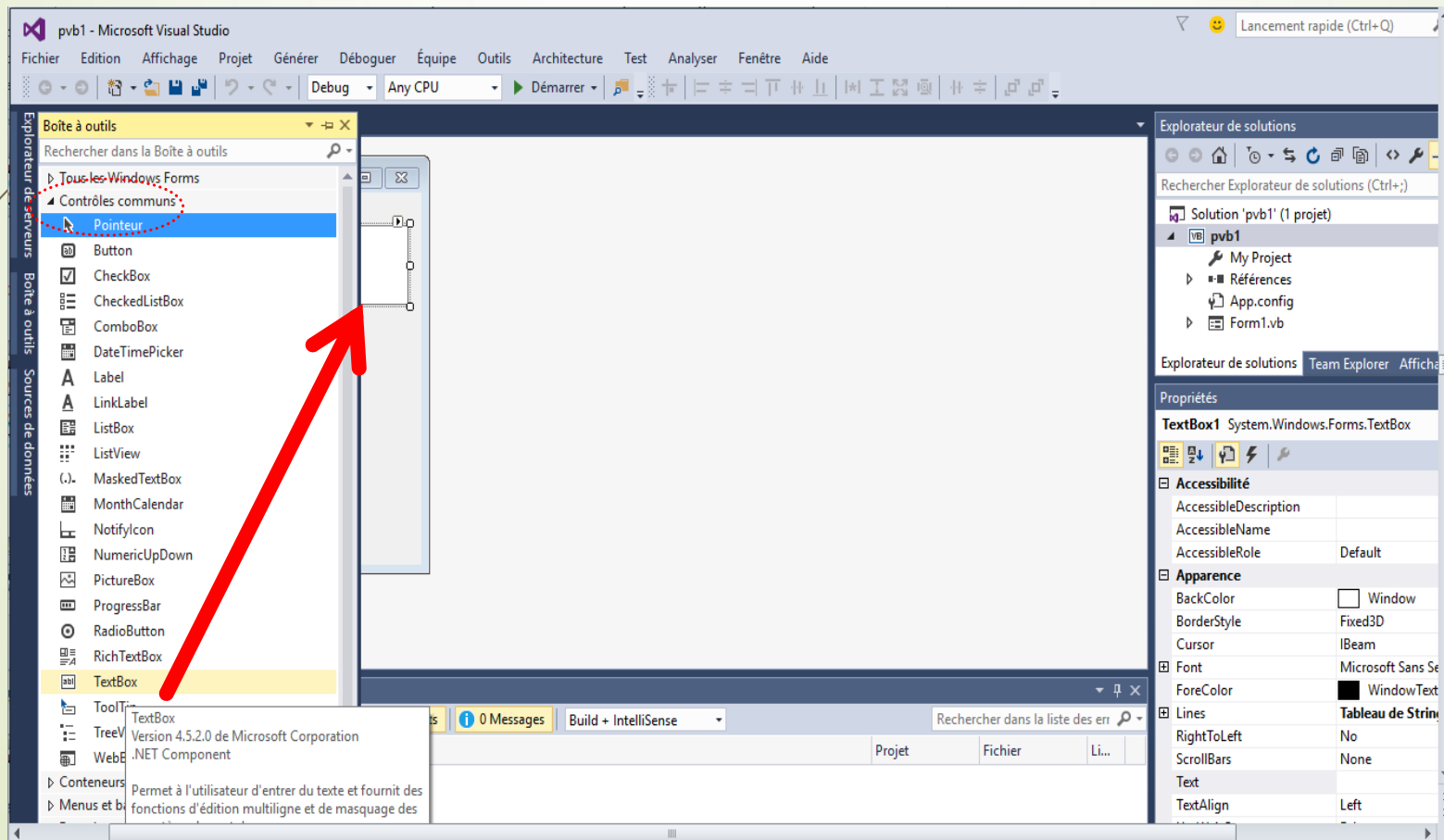


VB .NET (VB et premier programme)

- Nous pouvons par exemple, sélectionner le contrôle (click) **TextBox** de la boîte à outils.
- Ensuite, pour positionner le contrôle dans la forme, nous plaçons d'abord le curseur dans la forme à la position correspondant au coin supérieur gauche de la boîte de texte.
- Nous enfonçons alors le bouton gauche de la souris, et déplaçons la souris jusqu'au coin inférieur droit, où nous relâchons le bouton de la souris.

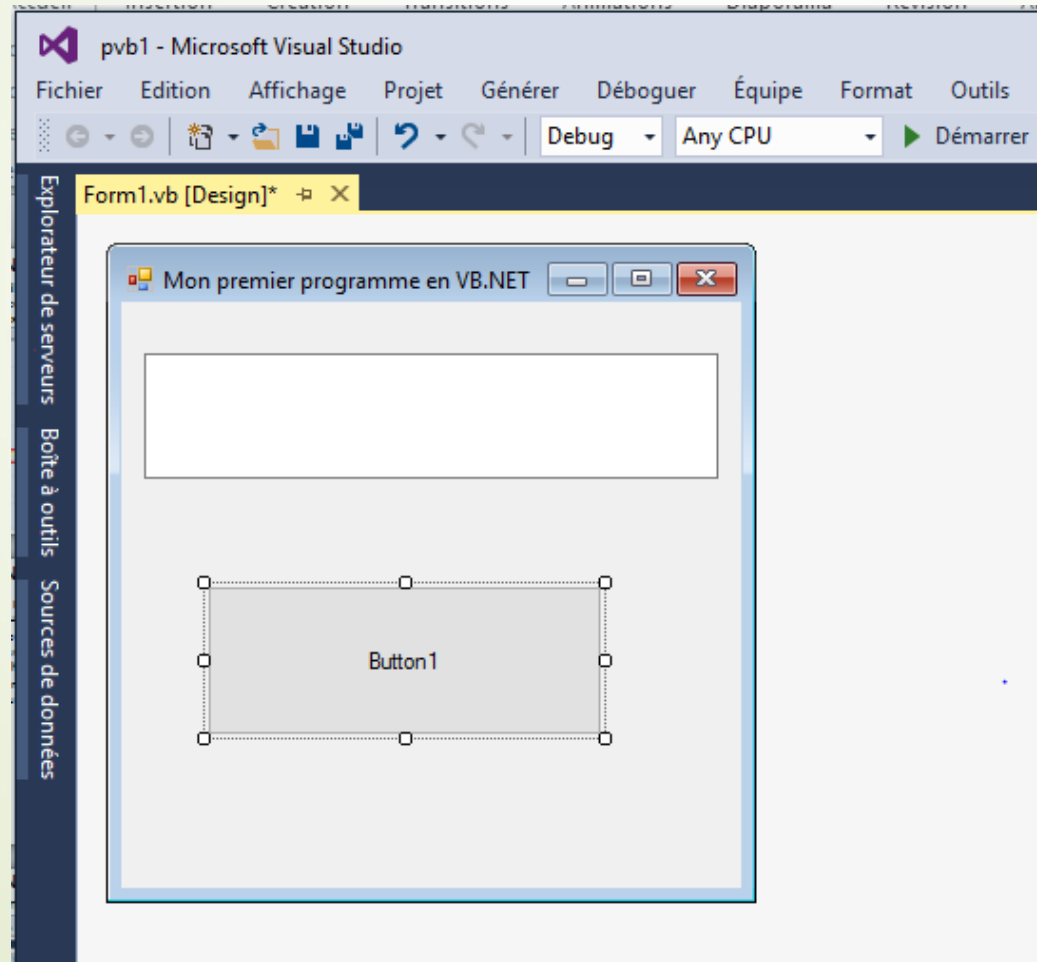
VB .NET (VB et premier programme)

➡ Insertion d'un contrôle **TextBox**



VB .NET (VB et premier programme)

- Plaçons de la même façon un élément de type "**Button**":

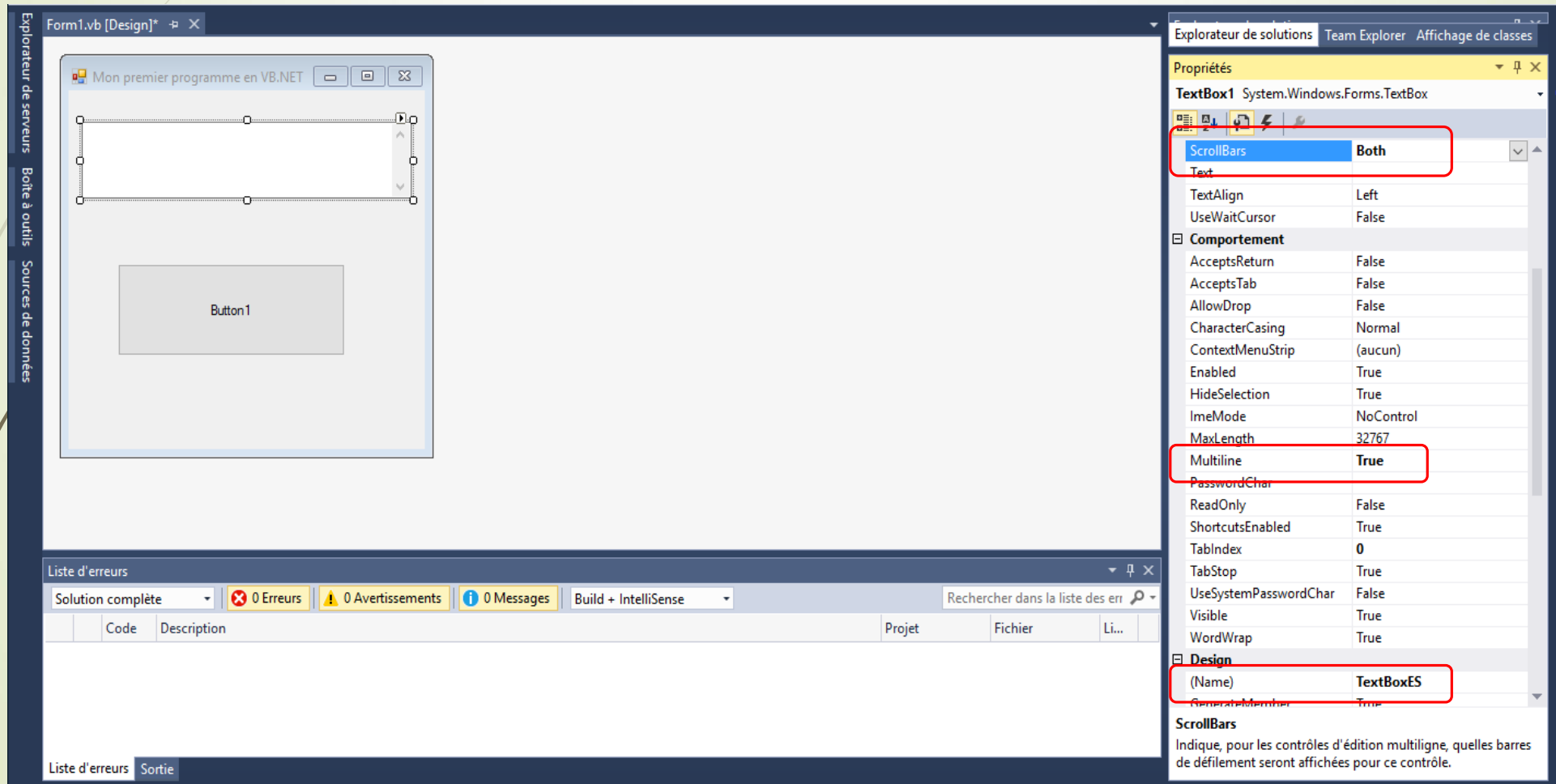


VB .NET (VB et premier programme)

➤ Modifications des propriétés du contrôle TextBox:

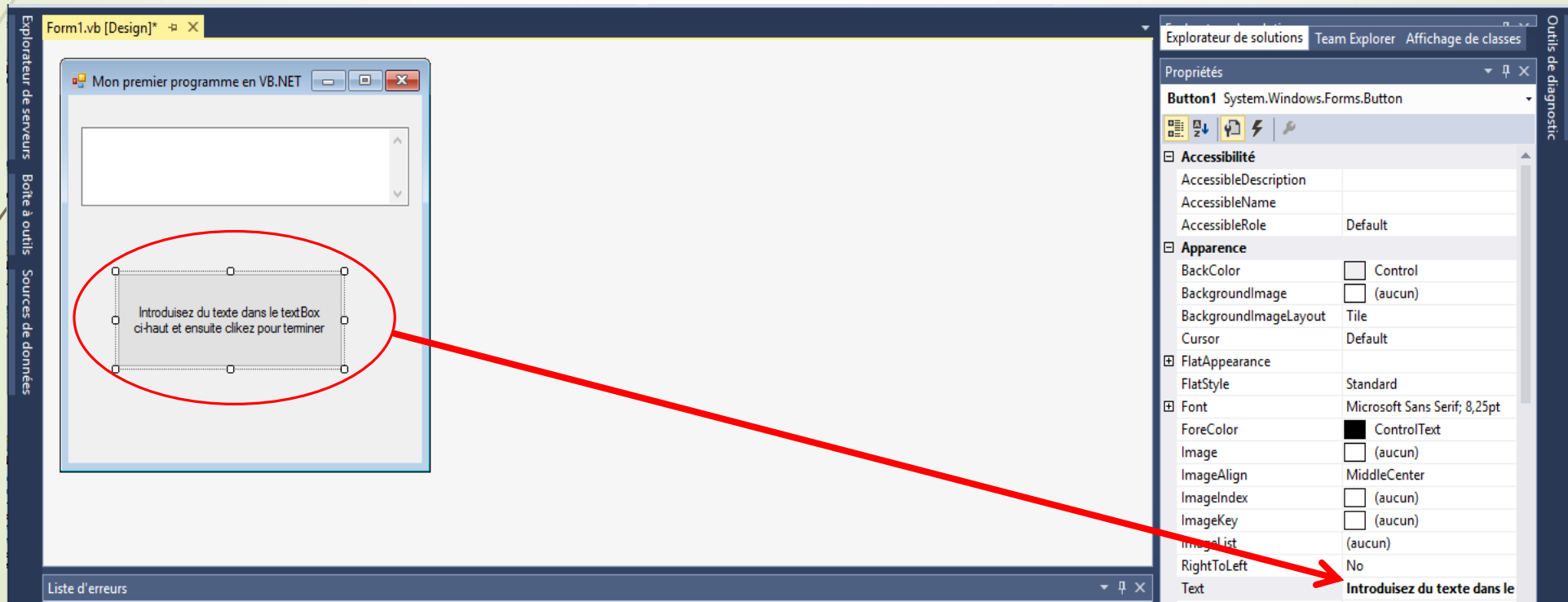
- La propriété **Name** devrait dans notre exemple contenir **TextBoxES**
- La propriété **Multiline** doit contenir **True** puisque notre contrôle **TextBox** pourrait recevoir plus d'une ligne de texte
- La propriété **Text** contient aucun caractère
- La propriété **Scrollbars** doit contenir **Both** permet les déplacements horizontal et vertical de la fenêtre de texte (défilement)

VB .NET (VB et premier programme)



VB .NET (VB et premier programme)

- Changeons aussi la propriété **Text** du bouton comme suit:

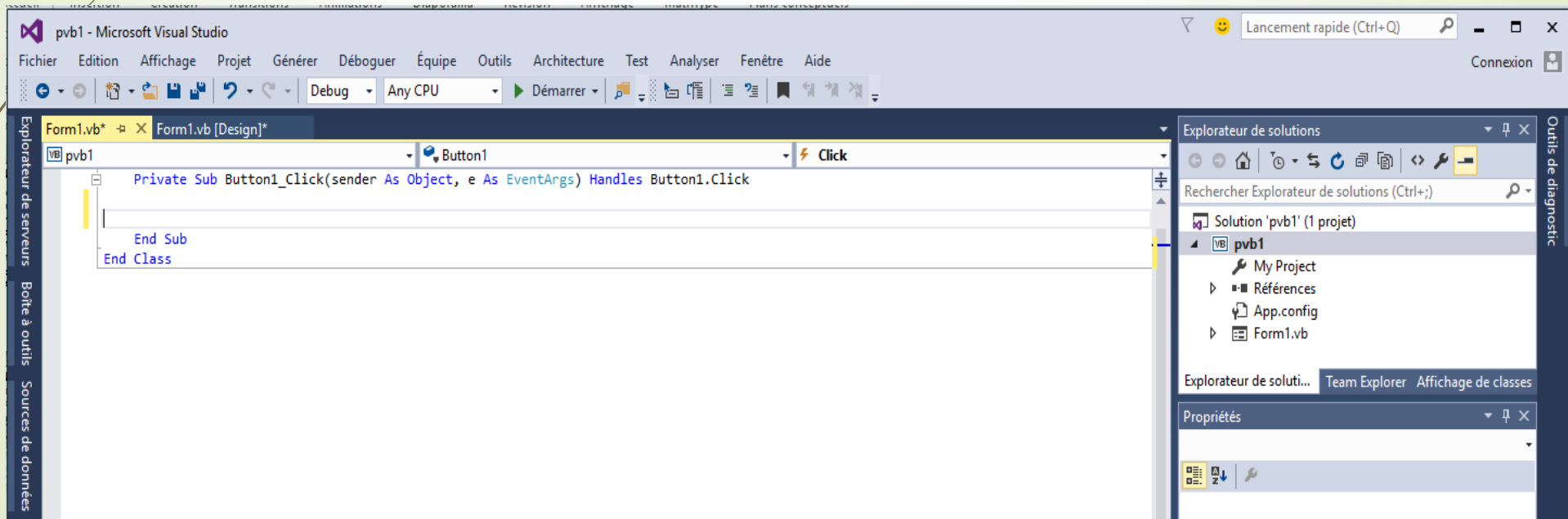


VB .NET (VB et premier programme)

- **Associer un traitement à un événement comme un click de souris:**
 - Pour générer le code associé à l'événement **click** sur le contrôle **Button**, nous devons double cliquer le bouton dans la forme
 - Nous voyons alors la fenêtre du code de la forme
 - Nous pouvons alors y insérer des énoncés de programme en VB

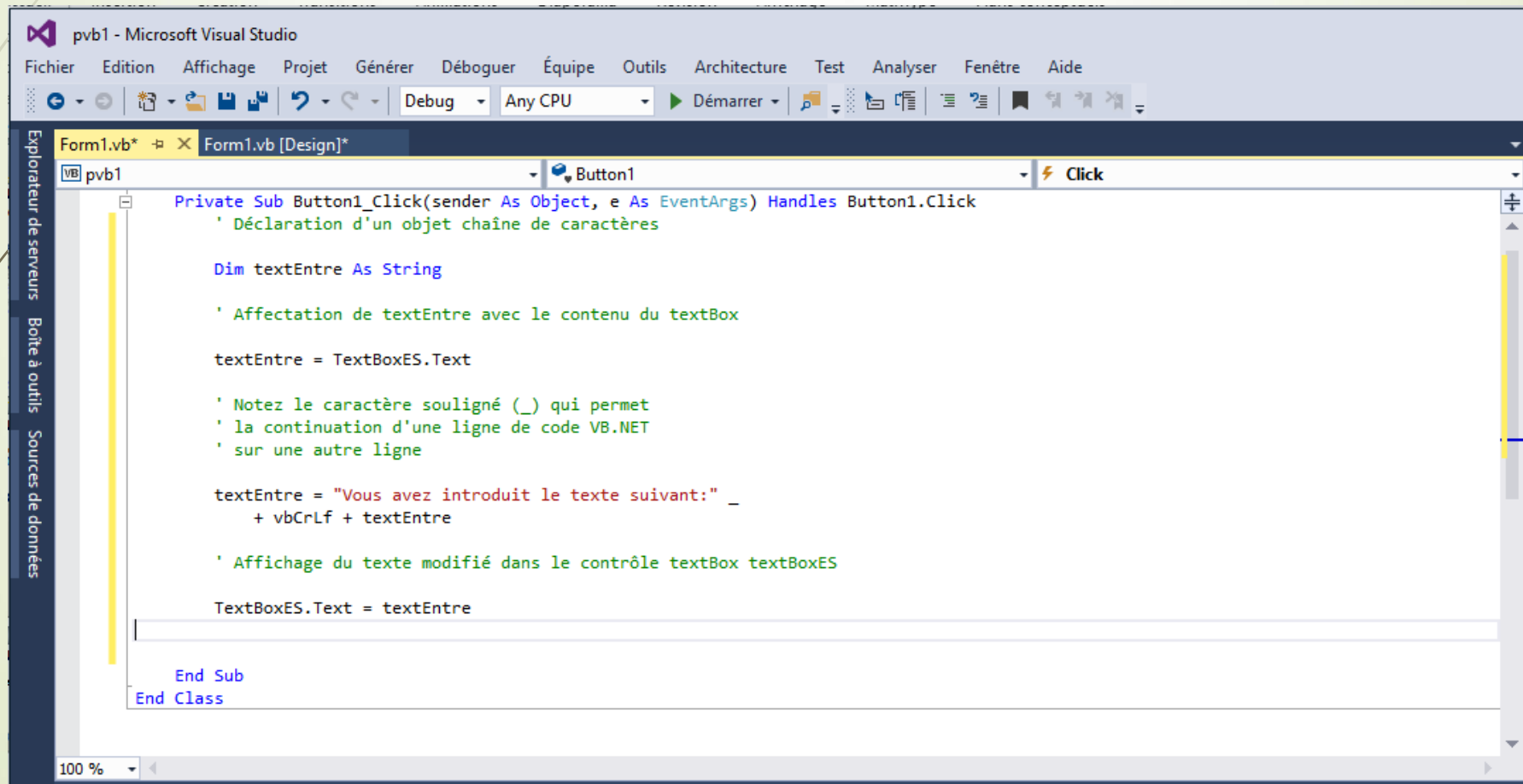
VB .NET (VB et premier programme)

- Après le double click, nous pouvons observer le schéma de la fonction associé à cet événement de souris (**click**)



VB .NET (VB et premier programme)

➤ Insertion du code



The screenshot shows the Microsoft Visual Studio IDE with a project named 'pvb1'. The 'Form1.vb' file is open in the 'Design' view. The 'Click' event for 'Button1' is selected, and the following VB.NET code is displayed in the code editor:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    ' Déclaration d'un objet chaîne de caractères

    Dim textEntree As String

    ' Affectation de textEntree avec le contenu du textBox

    textEntree = TextBoxES.Text

    ' Notez le caractère souligné (_) qui permet
    ' la continuation d'une ligne de code VB.NET
    ' sur une autre ligne

    textEntree = "Vous avez introduit le texte suivant:" _
        + vbCrLf + textEntree

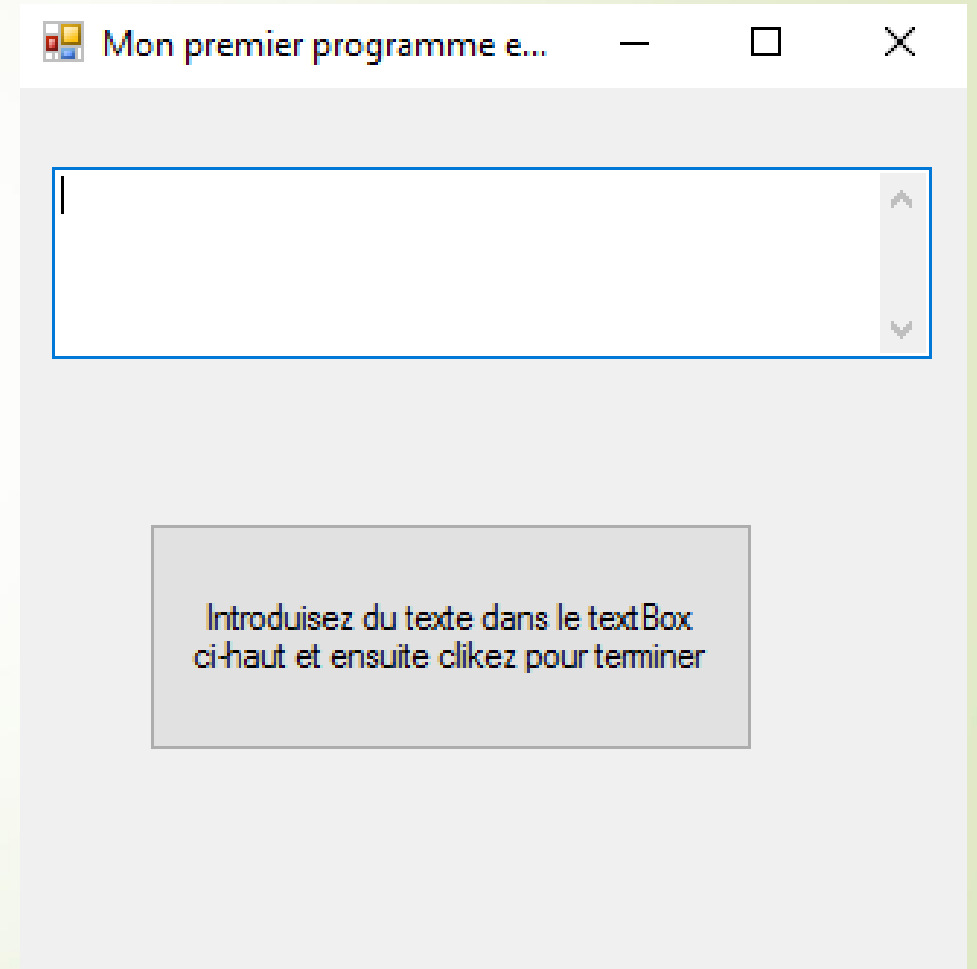
    ' Affichage du texte modifié dans le contrôle textBox textBoxES

    TextBoxES.Text = textEntree

End Sub
End Class
```

VB .NET (VB et premier programme)

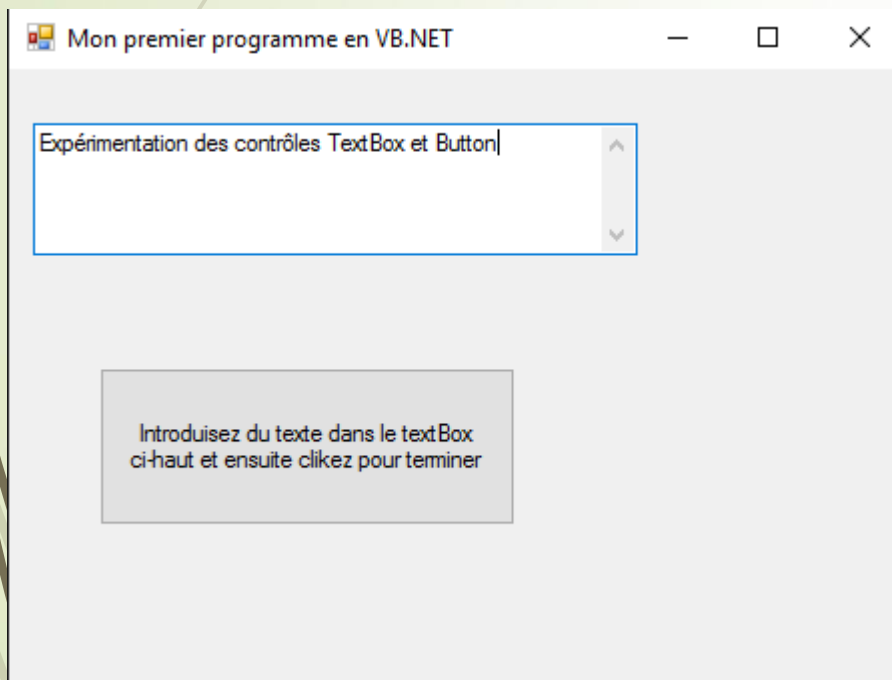
- Après avoir compiler le projet pvb1 (**Build/Build pvb1**) et démarrer le programme (**Debug/Start**), nous obtenons le résultat d'exécution suivant:



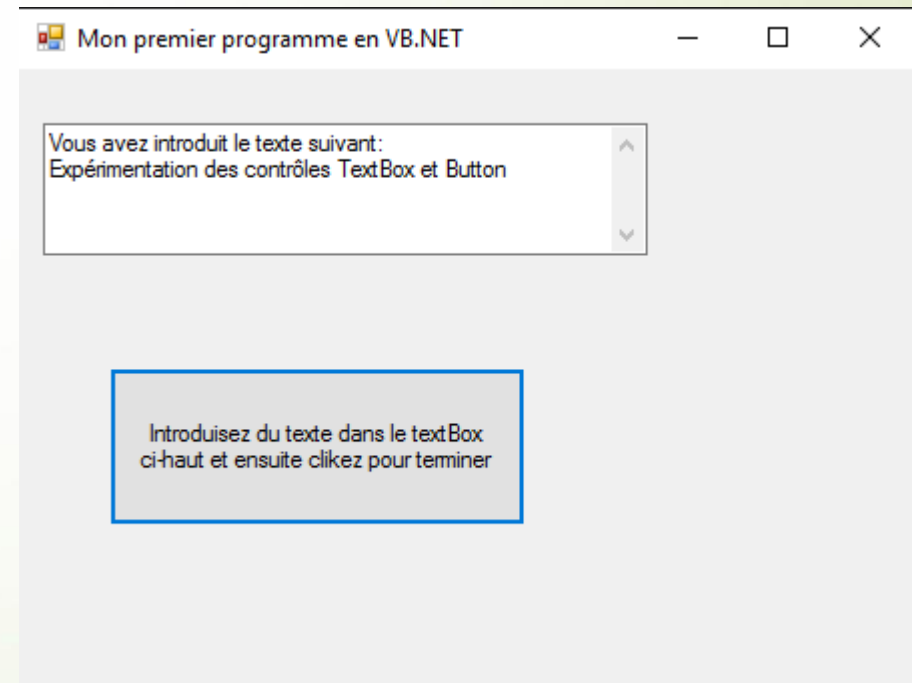
VB .NET (VB et premier programme)

Utilisation de l'application

Introduction du texte



Après avoir enfoncé le bouton



VB .NET (VB et premier programme)

- Le programme précédent, bien que fort simple, décrit le fonctionnement des applications modernes.
- En général, les programmes possèdent une fenêtre principale qui permet d'échanger des informations entre l'utilisateur du programme et le code du programme.
- Nous reviendrons sur ce programme lorsque nous discuterons des événements



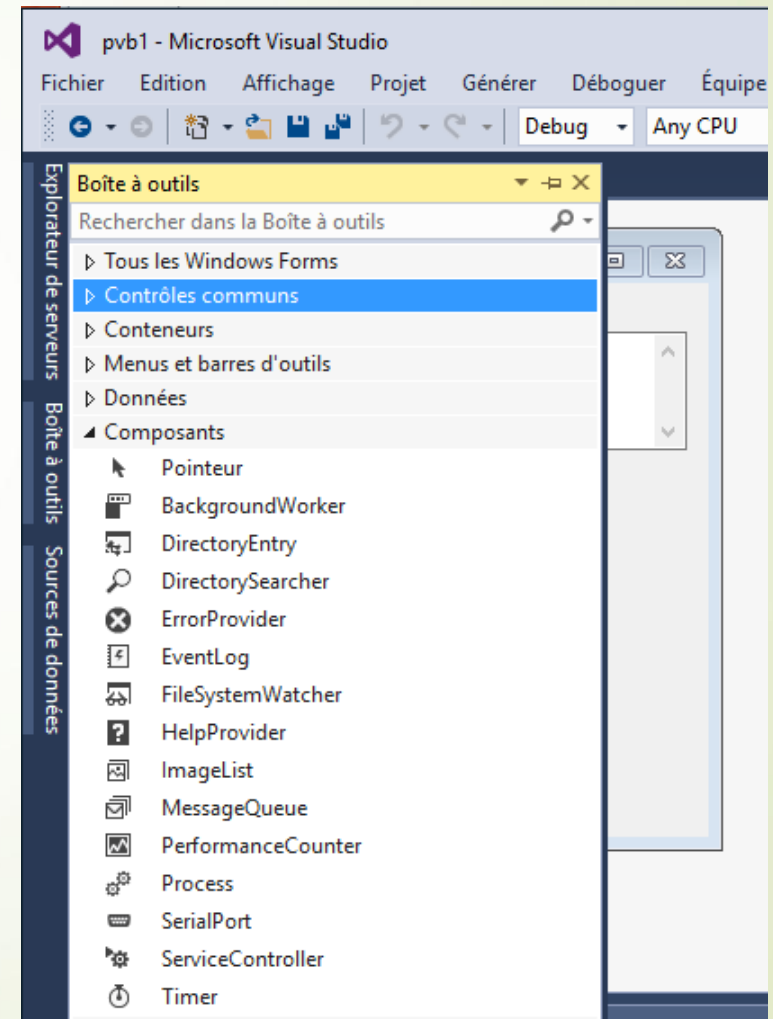
VB .NET (VB et premier programme)

➤ Autres Éléments de la boîte à outils

- La boîte à outils (**Toolbox**) contient des boutons qui permettent d'ajouter d'autres éléments à un projet.
- La section "**Components**" permet d'ajouter des composantes logicielles (COM ou .NET) qui rendent des services spécialisés comme Excel, Access, Word etc.

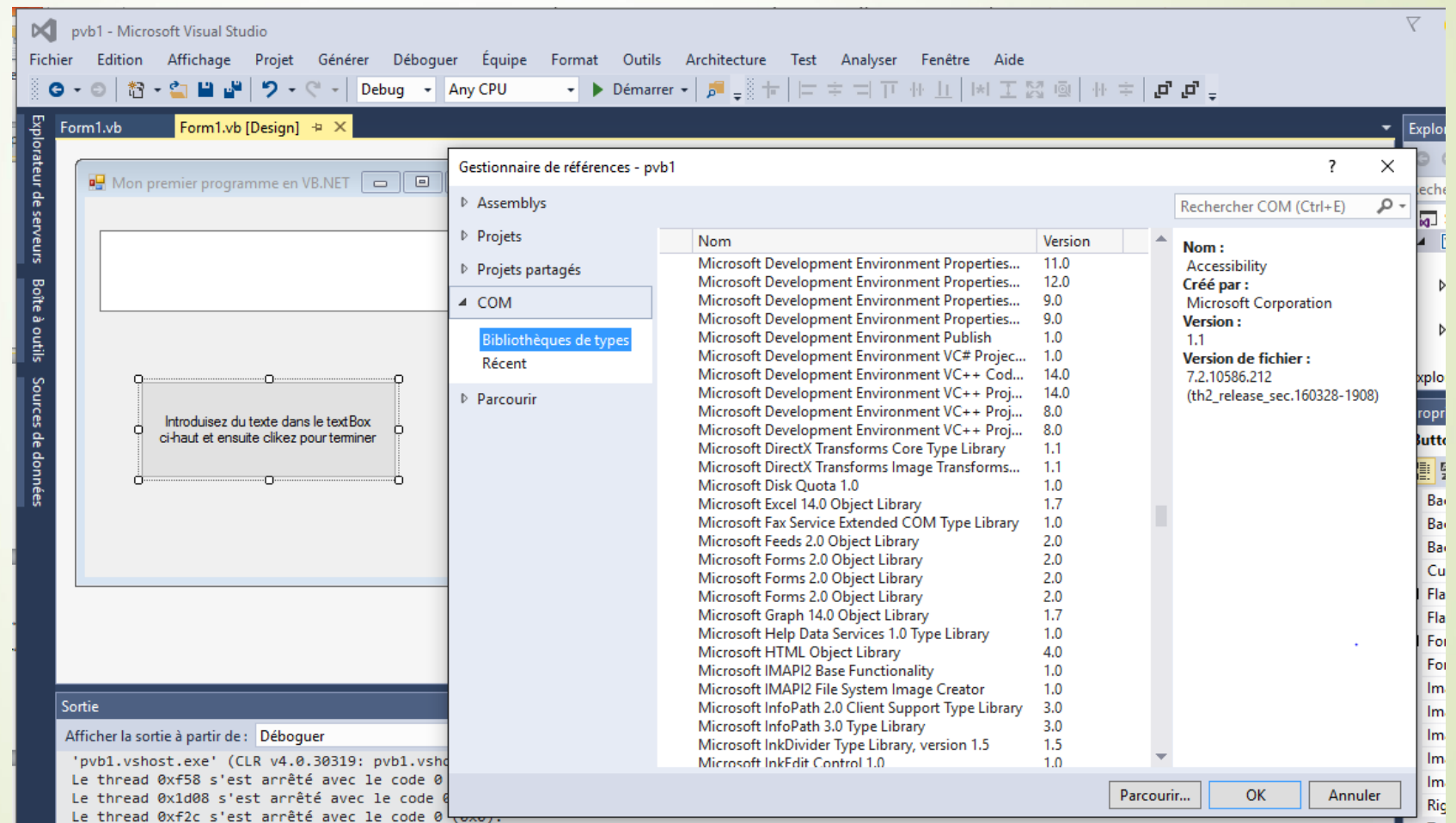
VB .NET (VB et premier programme)

➤ Composants disponibles



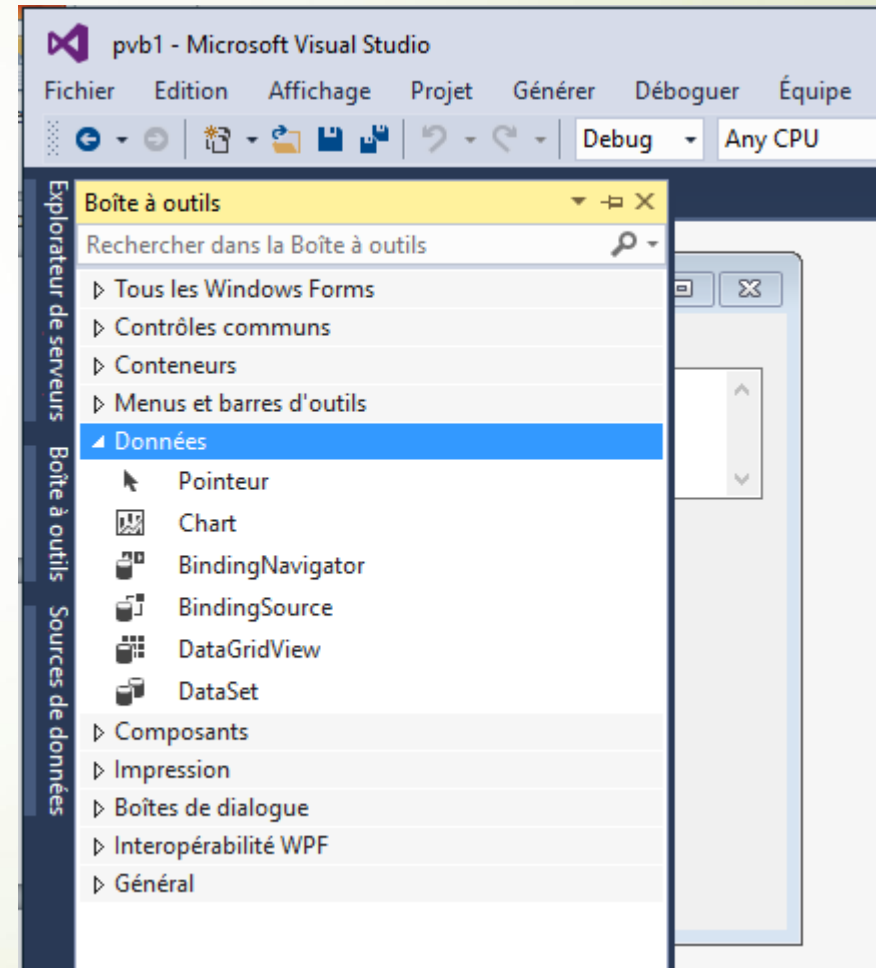
VB .NET (VB et premier programme)

- Notez que nous pouvons ajouter d'autres composants logiciels à Visual Studio (voir le menu-contextuel de la boîte à outils ou le menu "**Project/Add References**":



VB .NET (VB et premier programme)

- De même, la section "**Data**" permet d'ajouter des composants liés aux accès aux bases de données:



VB .NET (VB et premier programme)

➤ Les événements

- Quand nous avons double-clické sur le bouton à la section précédente, le système a ajouté automatiquement le code suivant (dans le code de la forme):

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As  
    System.EventArgs) Handles Button1.Click
```

```
End Sub
```

- L'énoncé **Sub** marque le début d'une procédure dans un programme VB et **End Sub** marque la fin du code de la procédure.
- La procédure s'appelle **Button1_Click**, et nous retrouvons les paramètres passés à la procédure entre les parenthèses.
- **Button1** est le nom du bouton. **Click** est le nom de l'événement qui est activé lorsque l'utilisateur click le bouton durant l'exécution du programme.
- Lorsque l'événement est activé, le code de la procédure est exécuté.

VB .NET (VB et premier programme)

- Voici le code que nous avons inséré dans la procédure:

```
Dim textEntre As String
```

```
'Affectation de textEntre avec le texte du TextBox
```

```
textEntre = TextBoxES.Text
```

```
'Notez le caractère souligné ( _ ) qui permet la
```

```
'continuation d'une ligne de code VB
```

```
'sur une autre ligne
```

```
textEntre = "Vous avez introduit le texte suivant:" _  
            + vbCrLf + textEntre
```

```
'Affichage du texte modifié dans le contrôle TextBox textBoxES
```

```
TextBoxES.Text = textEntre
```

VB .NET (VB et premier programme)

- Dans le code précédent, chaque ligne est une spécification du travail que doit effectuer notre programme lorsque le bouton est enfoncé.
- La ligne "**Dim textEntre As String**" spécifie que "**textEntre**" est le nom d'une variable qui va contenir une chaîne de caractères (String).
- Une variable est le nom associé à une zone de mémoire servant à contenir des valeurs intermédiaires.
- "**TextBoxES.Text**" est la valeur de la propriété **Text** de l'objet **TextBoxES** associé au contrôle **TextBox**. Cette propriété contient le texte qui apparaît dans le **TextBox**.
- Donc, la ligne "**textEntre = TextBoxES.Text**" signifie de mettre le texte contenu dans le contrôle **TextBox** dans la variable **textEntre**.

VB .NET (VB et premier programme)

- Les lignes suivantes:

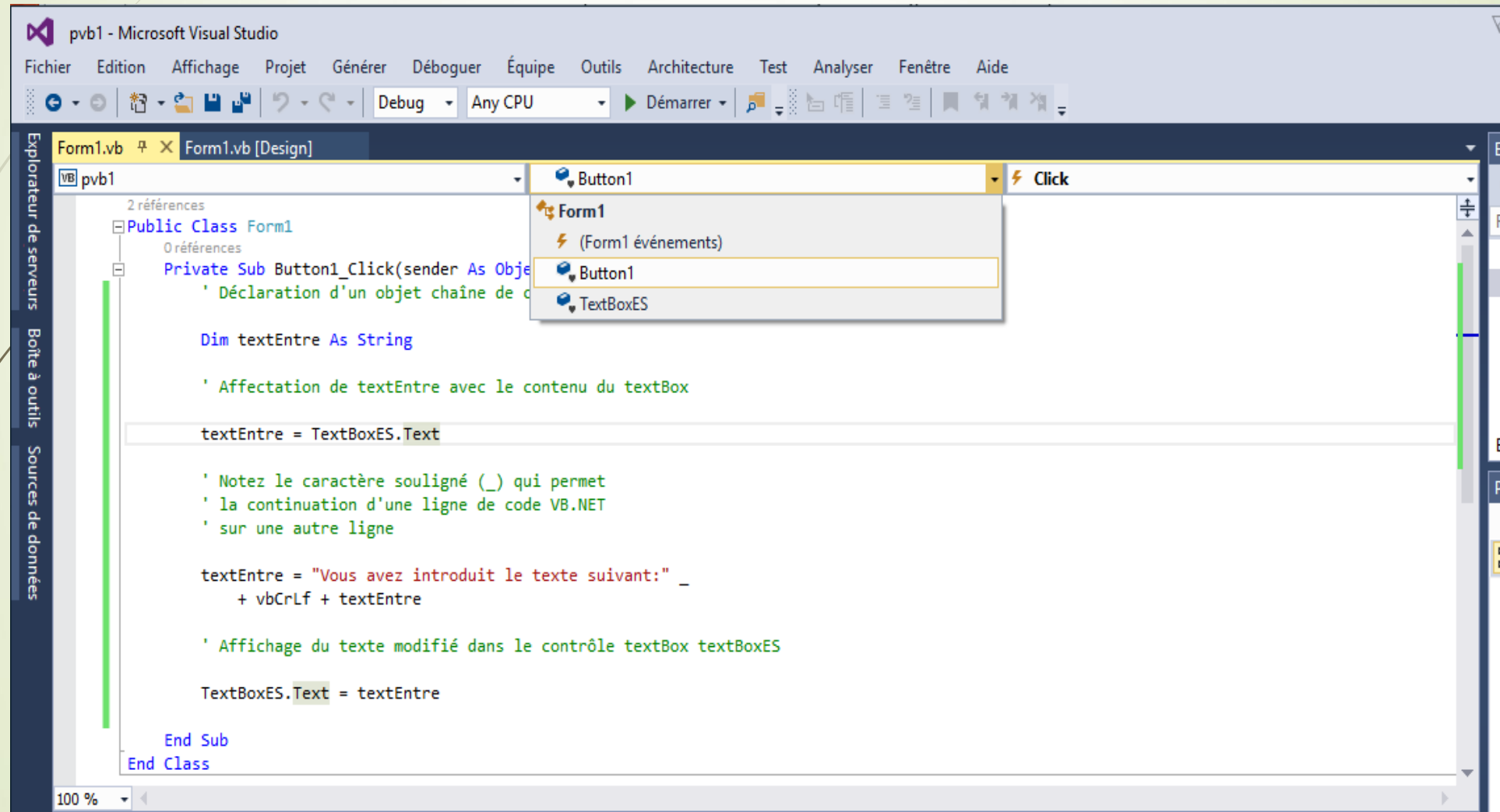
```
textEntree = "Vous avez introduit le texte suivant:" _  
+ vbCrLf + textEntree
```

- Servent à modifier le contenu de la variable "**textEntree**". Le signe "+" sert à concaténer des chaînes de caractères.
- "**vbCrLf**" est une constante pré-définie de VB qui représente les caractères ASCII qui servent à changer de ligne (Carriage-return, LineFeed).
- Finalement, la ligne:
TextBoxES.Text = textEntree
- Sert à mettre le contenu de la variable "**textEntree**" dans la propriété **Text** du contrôle **textBox** (et donc d'afficher le résultat).

VB .NET (VB et premier programme)

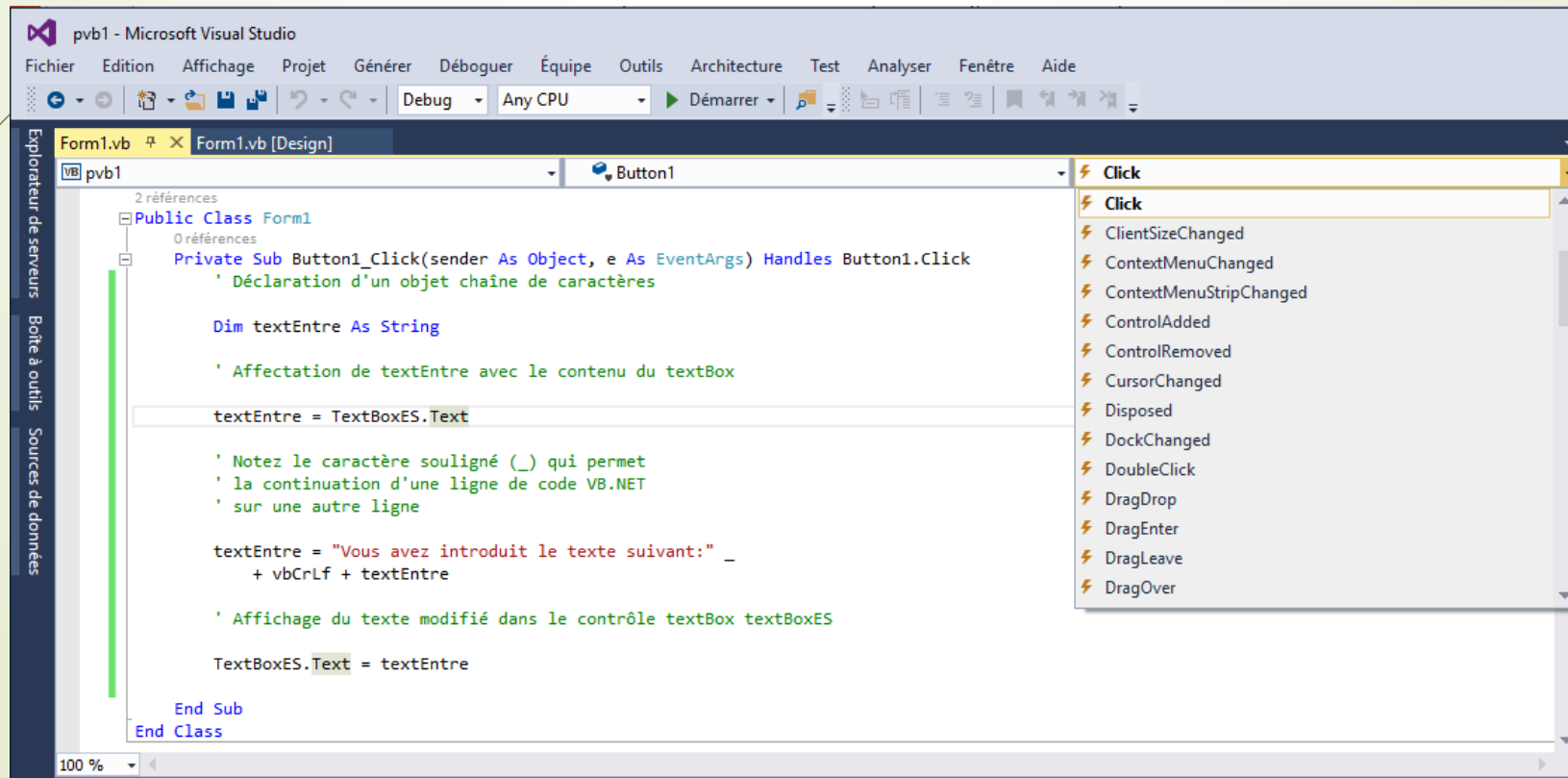
- En résumé, la fenêtre de "Code" sert à définir les instructions des programmes.
- Une procédure (Sub) est un groupe d'instructions qui a un nom et qui peut être exécutée en utilisant le nom de la procédure.
- En fait, le bouton n'est rien d'autre qu'une fenêtre de Windows (du système d'exploitation Windows). Ce type de fenêtre s'appelle un contrôle.
- Tous les éléments qui apparaissent dans la boîte à outils sont tous des contrôles.
- Windows répond au click sur le bouton en exécutant la procédure associée à l'événement "**Click**" sur le bouton.
- En haut de la fenêtre de code, se trouvent deux listes déroulantes. Celle de gauche donne la liste des objets de notre programme: Button1 (le bouton), Form1 (la forme) et TextBoxES (le TextBox).

VB .NET (VB et premier programme)



VB .NET (VB et premier programme)

- La liste de droite donne la liste des événements disponibles pour le bouton "Button1": Notez que nous utilisons uniquement l'événement "**Click**" qui est en caractères gras.



VB .NET (VB et premier programme)

- La classe **Form1**

- Les lignes suivantes apparaissent au début de la fenêtre de code de la forme:

Public Class Form1

Inherits System.Windows.Forms.Form

- Ces lignes indiquent qu'une classe appelée "**Form1**" a été définie par Visual Studio et cette classe représente notre fenêtre principale.
- Cette classe hérite (**inherits**) de toutes les propriétés de la classe Form du framework .NET.
- La procédure (méthode) **Button1_Click** a été ajoutée à la classe **Form1**.
- En plus, Visual Studio a ajouté du code qui représente les propriétés des éléments à l'intérieur de la forme dans la section "**Windows Form Designer generated code**".
- Normalement, on ne doit pas modifier cette section.

VB .NET (VB et premier programme)

- Une bonne méthode pour apprécier un langage de programmation est d'essayer les programmes-exemples (Samples).
- Souvent nous pouvons trouver des "Tutorials" qui sont des exemples expliqués.
- Pour trouver de l'information, on peut utiliser l'aide (help) ou sélectionner un item quelconque et faire F1

VB .NET (VB et premier programme)

