

## 3<sup>ème</sup> partie – Ajout et Utilisation d'un Menu Strip

Je vais développer les chapitres précédents, et montrer comment ajouter une barre de menu et l'utiliser en VFP. Le [code source téléchargeable](#) fourni dans le chapitre précédent vous sera nécessaire.

Mike S l'avait demandé, alors allons-y pour Mike S...

Ouvrez votre projet dans VB Express, et vous pouvez commencer.

L'ajout d'une « barre de menu » est très facile. Double-cliquez sur le fichier « MyToolstrip.vb » pour l'ouvrir dans le concepteur. Dans la boîte à outils à gauche, trouvez le contrôle MenuStrip et sélectionnez-le ; cliquez alors sur votre form pour l'y ajouter.

Nous utiliserons pour notre exemple les valeurs par défaut proposées, ainsi que les boutons. Il nous faut donc cliquer-droit sur le MenuStrip et choisir « Insérer des éléments standard ». un ensemble d'éléments de menu est alors ajouté, comme vous le voyez ci-dessous.



Dans la fenêtre de Code (double-click sur le contrôle, ou bien depuis le menu Afficher le Code), ajoutez cet évènement après le dernier que vous avez publié :

**'Please enter any new code here, below the Interop code**

**' these events will be published**

**Public Event NewDocument()**

**Public Event Open()**

**Public Event Save()**

**Public Event Print()**

**Public Event Cut()**

**Public Event Copy()**

**Public Event Paste()**

**Public Event Help()**

**Public Event ChkOnOffChanged()**

' **Gestion de tous les évènements du menu** <<<--- C'est ici que vous devez ajouter votre code

**Public Event MenuItemSelected(ByVal Menuname As String)**

Il nous reste maintenant à créer les évènements pour chaque MenuClick.

Pour cela, double-cliquez sur chaque élément de menu dans la fenêtre de conception : une vue sur le code sera alors ouverte, avec le squelette de la méthode Click. Il ne vous reste qu'à coller la ligne suivante (celle qui est en gras) dans chaque méthode.

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

```
    RaiseEvent MenuItemSelected(sender.name)
```

```
End Sub
```

Je ne l'ai fait ici que pour le menu Fichier, mais vous devez le faire pour **tous** les menus.

*Note du traducteur :*

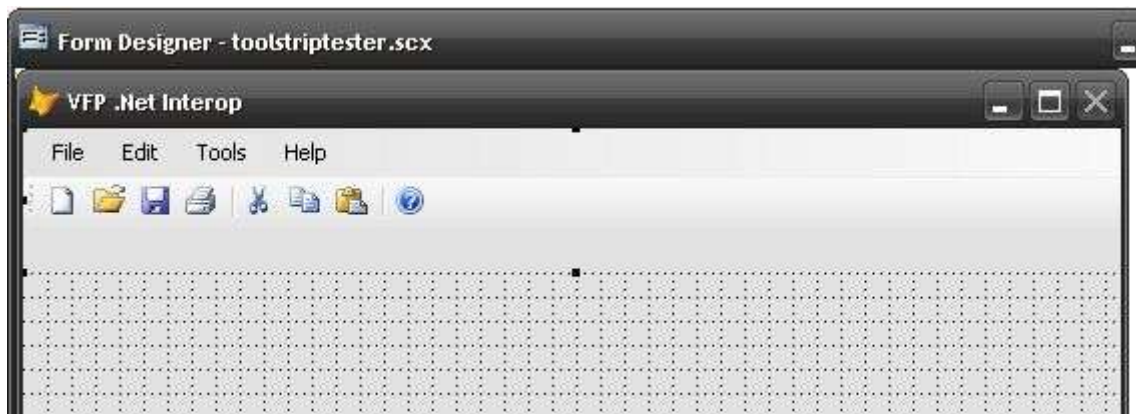
Cette syntaxe employée par B.Bout ne fonctionne que si Option Strict est OFF dans VB.Net, ce qui est fortement déconseillé. Si vous voulez vous conformer aux « Best Practices » en .Net, vous devez mettre Option Strict à ON, et cela interdit les liaisons tardives (l'évaluation du type d'objet sender reçu en paramètre au moment de l'exécution seulement).

Il convient d'évaluer Sender à la compilation ; la syntaxe correcte est donc :

```
    RaiseEvent MenuItemSelected(CType(sender, System.Windows.Forms.ToolStripItem).Name)
```

Recompilez ensuite le contrôle ; nous en avons fini du côté de VB.Net.

Dans VFP, ouvrez votre Form de test, et constatez que maintenant, le contrôle contient bien les éléments de menu.



Il nous faut suffir alors de nous raccorder au **Menu handler**. Dans VFP, ouvrez la fenêtre de code du **.MenuItemSelected**. C'est le nouvel évènement que nous venons d'exposer. La fenêtre de code s'ouvre bien en nous présentant le nouveau paramètre, et ceci parce que nous avons spécifié que ce paramètre devait être passé par l'évènement, quand nous avons exposé cet évènement en VB.

Ajoutez alors l'extrait de code suivant :

**\*\*\* ActiveX Control Event \*\*\***

**LPARAMETERS menuname**

**LOCAL cMenu as String**

**cMenu = UPPER(STRTRAN(menuname,"ToolStripMenuItem",""))**

**IF cMenu = "EXIT"**

**ThisForm.Release**

**ELSE**

**WAIT WINDOW NOWAIT "You selected menu item :"+cMenu**

**ENDIF**

Il vous faut gérer les clics de menu à un seul et même endroit. Ce sera très simple à faire en utilisant une structure CASE...ENDCASE, avec une déclaration pour chaque élément de menu. Comme le nom complet de l'élément est passé, j'ai simplement restreint à l'élément de menu cliqué.

Lancez maintenant votre form et utilisez le menu. Remarquez que Exit referme bien le Form.

*Note du traducteur :*

si vous utilisez VB en français (c'est très probablement votre cas), les noms des menus et sous-menus sont automatiquement francisés. Par conséquent, le sous-menu Exit devient Quitter. Vous l'intercepterez donc en VFP par :

IF cMenu = "QUITTER"



Encore merci à Mike S pour sa suggestion.

Première publication le 9 juillet 2007 par [bbout](#)