

## 1<sup>ère</sup> Partie - Installation, Création de Contrôle & Événements

Microsoft propose maintenant la version 2.0 de la Boîte à Outils d'Interopérabilité des Formulaires ( Interop Forms Toolkit 2.0) sur <http://msdn2.microsoft.com/en-us/vbasic/aa701259.aspx> . Avec cette boîte à outils, les développeurs peuvent créer des contrôles ActiveX utilisables dans du VB, et des Forms qui peuvent être appelés depuis VB. Mais cette boîte à outils n'est pas réservée seulement à VB. On peut s'en servir aussi pour créer des Forms et des contrôles qu'on utilisera en VFP, comme je vais vous le montrer.

Commençons par le début. À partir du lien ci-dessus, téléchargez l'Installer. Vous n'avez besoin que du fichier InteropFormToolsInstaller.msi (5.2mb) ; installez-le sur votre machine.

Cette boîte à outils est conçue pour être un supplément à la version complète de VS.NET, ce dont nous n'avons pas vraiment besoin. Elle est livrée avec une série de fichiers modèles, qui serviront au plus gros travail en .Net. si vous avez la version complète, tout sera installé comme Add-in. Je l'utilise avec la version gratuite de VBExpress.Net, que vous pouvez télécharger chez Microsoft.



Figure 1

Il faut copier ces fichiers à un emplacement spécifique pour qu'ils apparaissent en tant que modèles dans VB. Ouvrez votre explorateur et allez à l'endroit spécifié ci-dessous. Vous avez besoin des fichiers ZIP. VB Express vous permet de définir l'emplacement de vos fichiers modèles personnels.

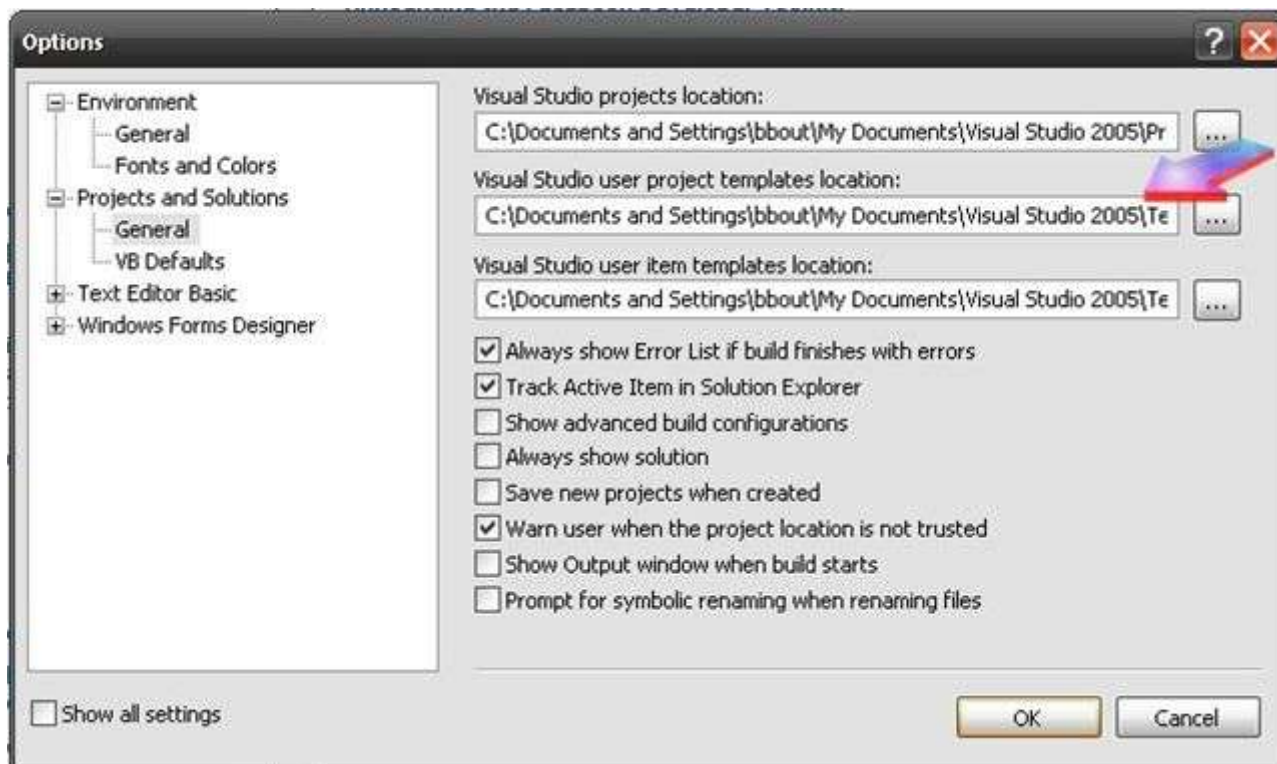


Figure 2

Si vous avez conserve les répertoires par défaut, l'emplacement des modèles de projet sera

C:\Documents and Settings\bout\Mes Documents\Visual Studio 2005\Templates\ProjectTemplates\Visual Basic\ pour les fichiers modèles suivants :

***VB6 InteropForm Library.zip***

***VB6 UserControl.zip***

Et les détails se trouveront à

C:\Documents and Settings\bout\My Documents\Visual Studio 2005\Templates\ItemTemplates\Visual Basic pour les fichiers suivants:

***VB6 InteropForm Library Info.zip***

***VB6 InteropForm.zip***

***VB6 UserControl Item.zip***

Une fois ceci effectué, ces modèles vous seront proposés chaque fois que vous choisirez un nouvel élément dans VB Express.

Revenons maintenant à VB Express, et cliquons sur Nouveau; nous voyons la boîte de dialogue pour un nouveau projet. Choisissez VB6 Interop UserControl et renommez votre projet comme indiqué.

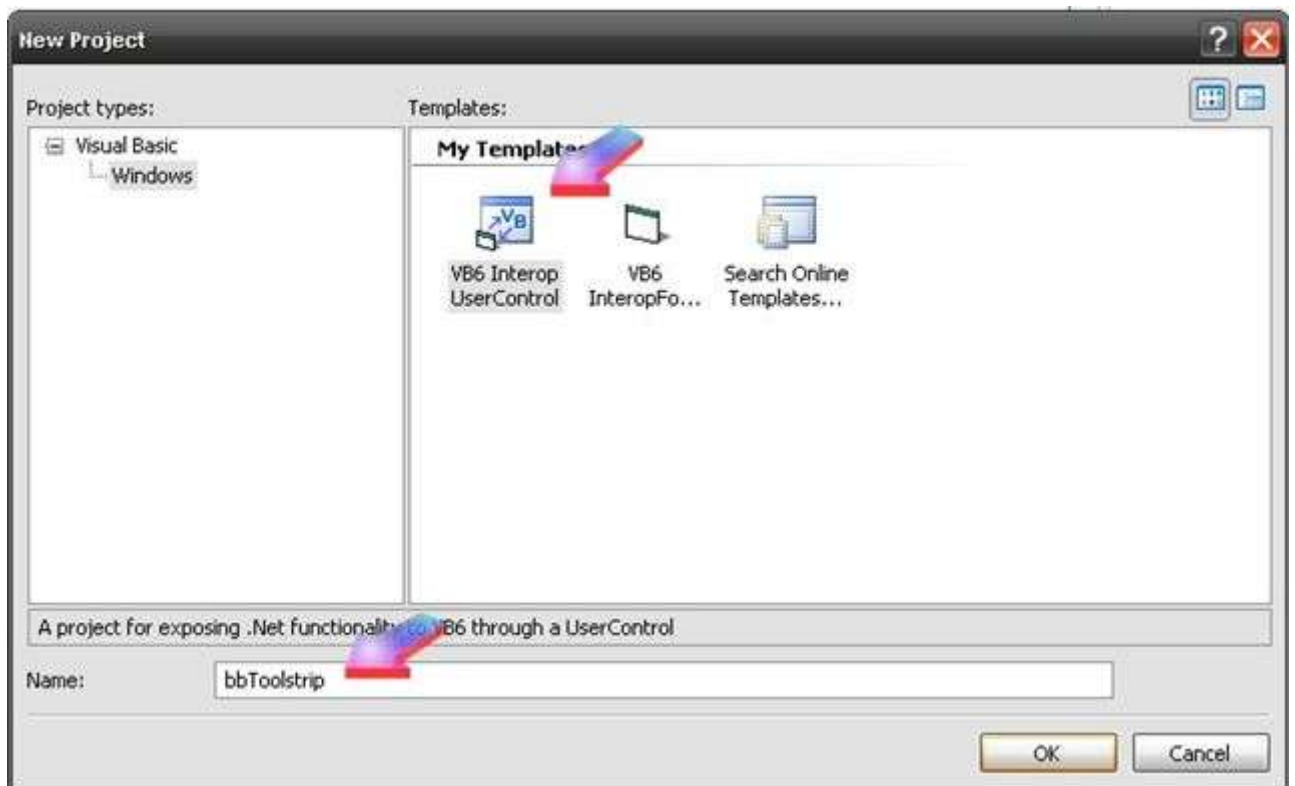


Figure 3

VB Express va créer les fichiers nécessaires à votre nouveau contrôle utilisateur. Il faut maintenant enregistrer le projet afin de créer les répertoires ; ce que nous faisons ici.

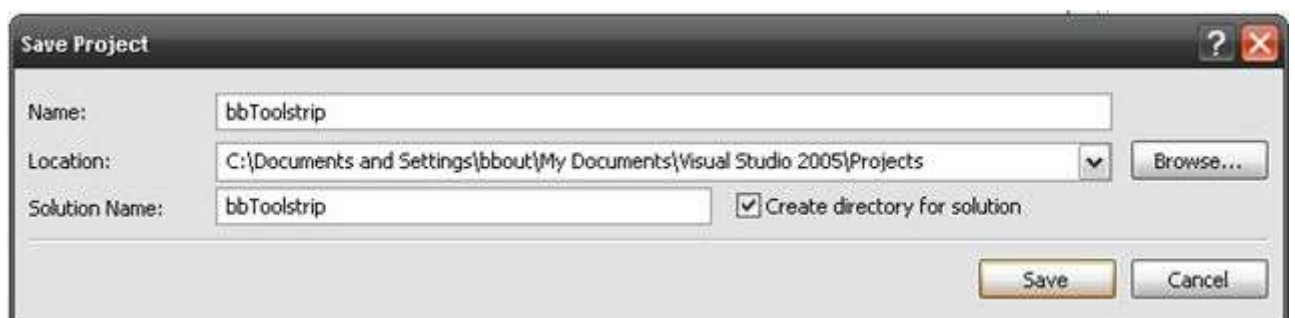


Figure 4

Un des fichiers créés sera **InteropUserControl.vb**, que nous allons renommer en **MyToolstrip.vb**.



Figure 5

Double-cliquez dessus, ou bien cliquez droit et choisissez « Concepteur de vues ». Vous ouvrez ainsi ce fichier pour le modifier. Dans la Boite à Outils à gauche de l'écran, sélectionnez le contrôle « ToolStrip » puis cliquez sur le contrôle utilisateur qui a été généré pour lui ajouter.

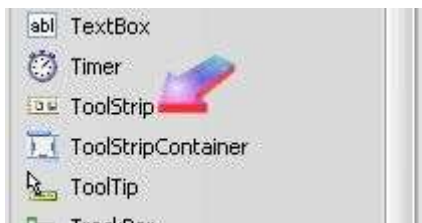


Figure 6

Comme ce contrôle utilisateur n'a aucun bouton, cliquez droit dessus et choisissez « Insérer des éléments standard ».

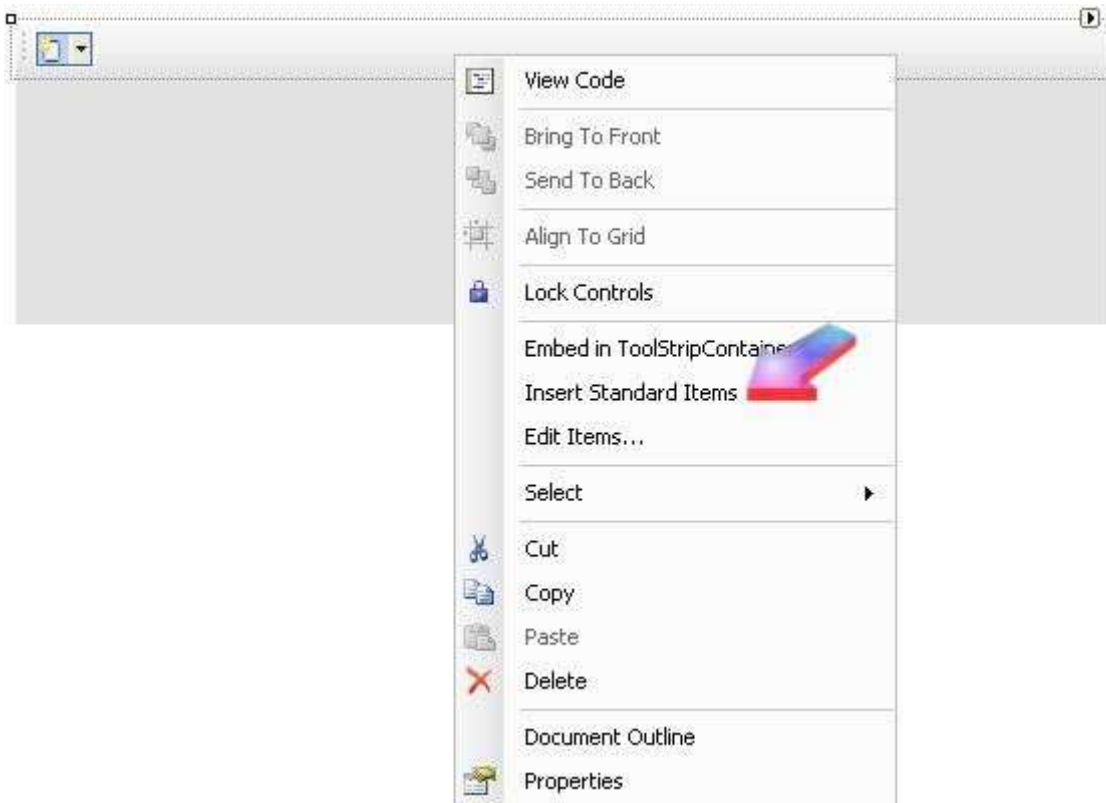


Figure 7

Et vous aurez automatiquement les boutons standard

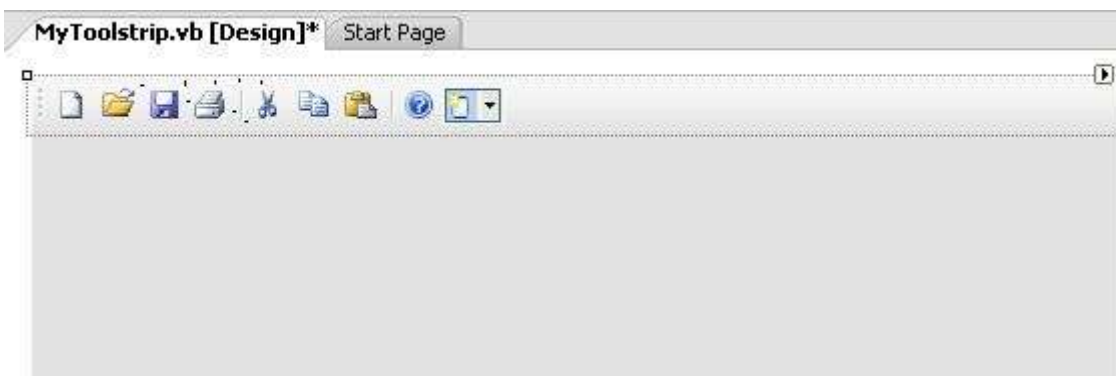


Figure 8

Ouvrez maintenant la fenêtre de Code (click droit, puis « Afficher le code ») ; vous verrez alors cette zone dans laquelle vous allez ajouter votre propre code. Il nous faut trouver une solution pour exposer les évènements des boutons : commençons par ces quelques lignes de code qui vont créer les évènements pour chaque bouton.

```

VB6 Interop Code

'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 Button1Click()

```

}

we add these events

Figure 9

Double-cliquons maintenant sur chacun des boutons dans la fenetre de Design, pour leur ajouter un gestionnaire d'événement. Déléguons le click aux événements que nous venons de créer, en nous aidant de l'Intellisense.

```

'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 Button1Click()

```

---

```

Private Sub NewToolStripButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Raiseevent
End Sub
End Class

```

- Button1Click
- Click
- Copy
- Cut
- DbClick
- Help
- NewDocument
- Open
- Paste
- Print

Public Event NewDocument()

Figure 10

```

' 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 Button1Click()

Private Sub NewToolStripButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    RaiseEvent NewDocument()
End Sub

Private Sub OpenToolStripButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    RaiseEvent Open()
End Sub

Private Sub SaveToolStripButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    RaiseEvent Save()
End Sub

Private Sub PrintToolStripButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    RaiseEvent Print()
End Sub

Private Sub CutToolStripButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    RaiseEvent Cut()
End Sub

Private Sub CopyToolStripButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    RaiseEvent Copy()
End Sub

Private Sub PasteToolStripButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    RaiseEvent Paste()
End Sub

```

Figure 11

Tout est prêt! Faites exécuter le contrôle pour voir à quoi il ressemble dans VB Express. La dernière étape consiste à générer le projet : le code du modèle va enregistrer ce contrôle pour nous.

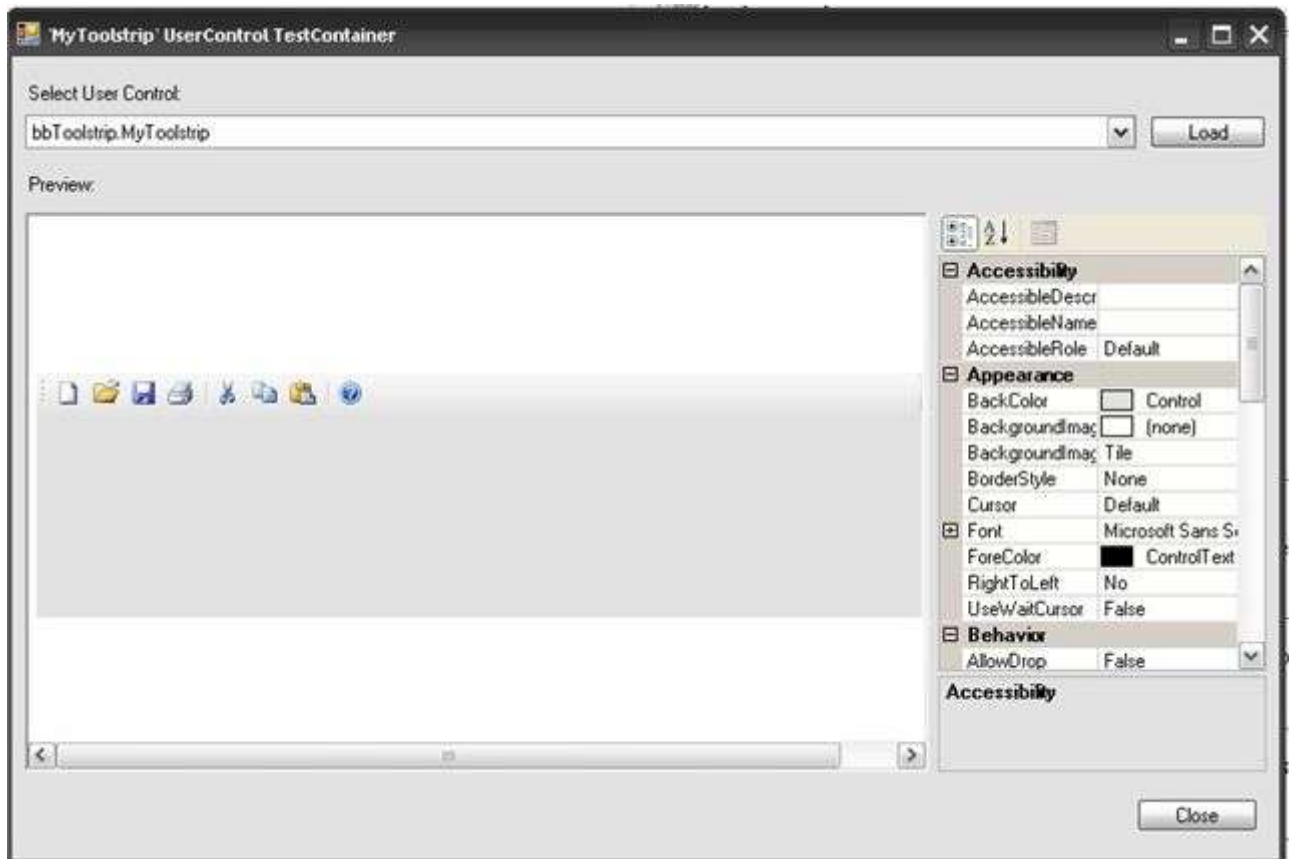


Figure 12

**Passons à VFP.** Créons un nouveau Form et ajoutons-y un OleControl. Dans la boîte de dialogue, recherchez et sélectionnez le contrôle que vous venez de créer, il sera ajouté à votre Form.



Figure 13





Figure 14

Cliquez sur le contrôle, redimensionnez-le et réglez la propriété **Anchor** sur **12**

En lançant de formulaire vous verrez le controle, mais les clicks sur les boutons seront sans effet. Il nous faut les raccorder aux événements que nous avons créés précédemment.

Double-cliquez sur le contrôle pour ouvrir la fenêtre de code de VFP : les événements que nous avons créés apparaissent dans la liste déroulante. Commençons par l'événement **NewDocument**.

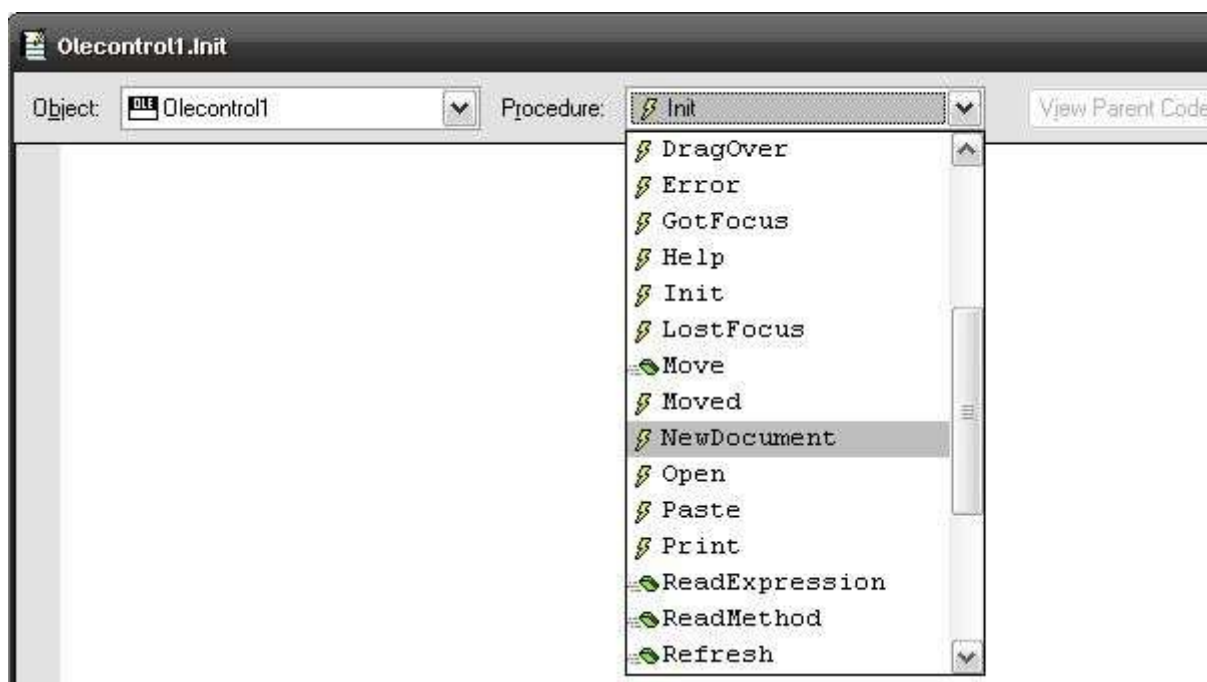


Figure 15

Et ajoutons le code qui sera exécuté quand cet événement surviendra. Pour le moment, saisissons simplement le code indiqué en exemple.

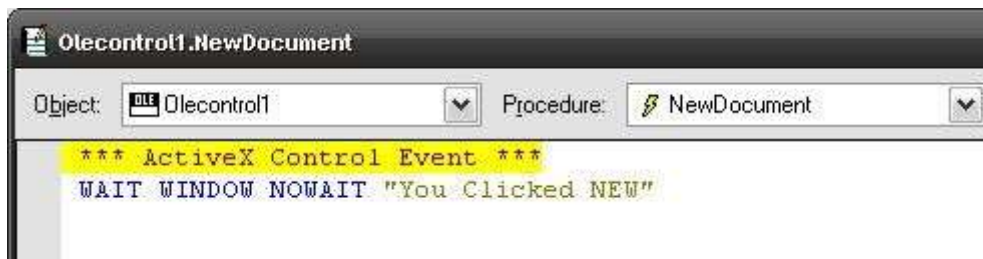


Figure 16

Répetons cela pour chaque événement créé.

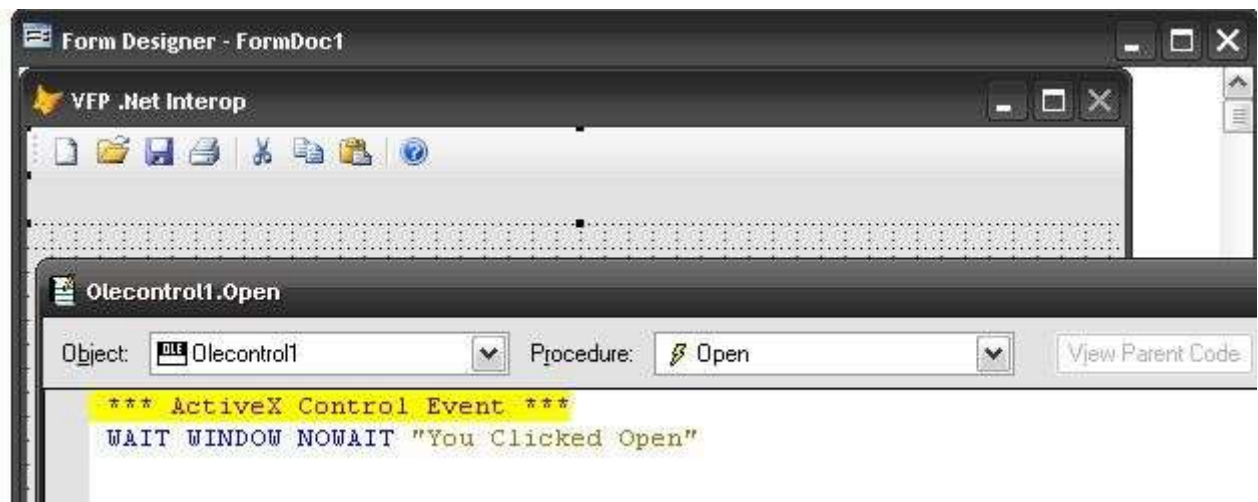


Figure 17

Relançons notre Form et cliquons sur les boutons. Notre code est bien exécuté. Nous venons de créer notre premier contrôle utilisateur Interop.

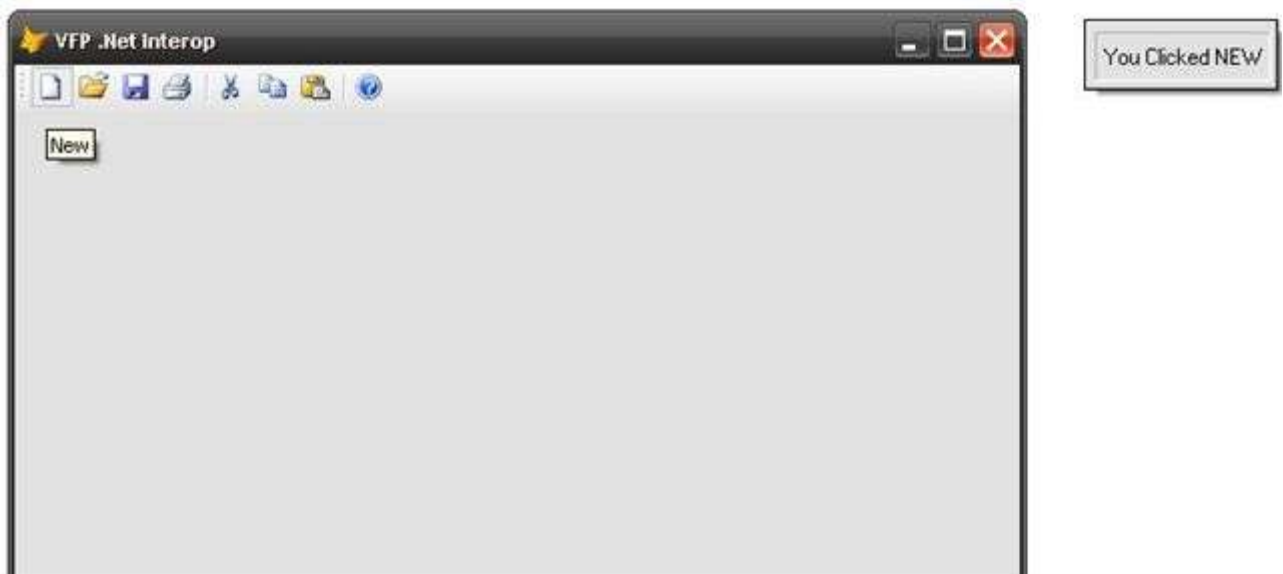


Figure 18

Amusez-vous ! créez vos contrôles .Net !

Dans un prochain article, nous ajouterons d'autres contrôles, et nous exposerons des propriétés.

Première publication le 20 juin 2007 par [bbout](#)