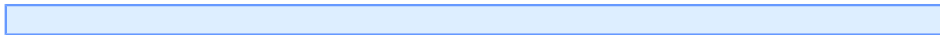


# Tutoriel 29 : Les applications Multi Documents Interface (MDI)

par Gilbert Miralles ([gilmir.developpez.com](http://gilmir.developpez.com))

Date de publication : Mardi 14 octobre 2003

Dernière mise à jour : Lundi 4 février 2008



Visual Basic comporte trois styles d'interfaces :

- Interface monodocument(SDI)

- Interface multidocument(MDI)

- Interface Explorateur

- Terminologie MDI

- L'application BlocNote

Prochain tutoriel

Visual Basic comporte trois styles d'interfaces :

## Interface monodocument(SDI)

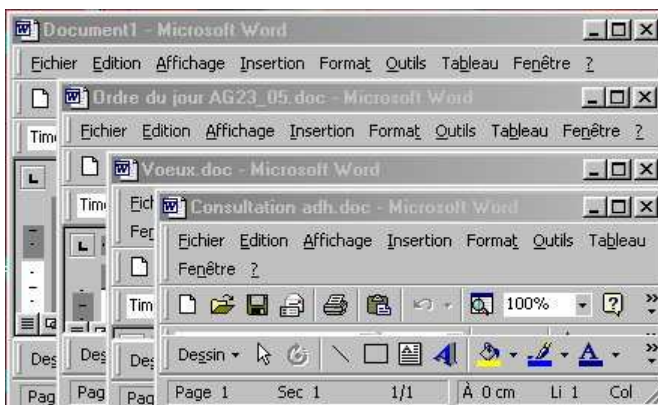
Une application SDI contient une seule fenêtre de données. Lorsque l'utilisateur ouvre un nouveau fichier de données, le contenu vient prendre la place de la fenêtre précédemment ouverte.(Style Bloc Notes que tout le monde connaît)

## Interface multidocument(MDI)

Une application MDI contient plusieurs fenêtres de données.

Microsoft Word, par exemple, permet d'ouvrir autant de documents que l'on veut.

Le code MDI de Word garantit que chaque document apparaît dans sa propre fenêtre (voir figure ci-dessous) Dans cet exemple nous avons ouvert 4 feuilles sous Word.



On passe d'une feuille à l'autre en cliquant sur la fenêtre appropriée ou en la sélectionnant dans la barre de menu. Lorsque l'utilisateur bascule d'une feuille à l'autre, la fenêtre sélectionnée reçoit le focus et devient la feuille active.

Le focus dans le cas d'une feuille a sa barre de titre marquée par une couleur plus visible de nuance bleue.

Le focus peut être passé à un objet par programme, grâce à

la méthode suivante : [Objet.SetFocus](#)

les propriétés Enabled et Visible de l'objet doivent être à "True".

## Interface Explorateur

On retrouve l'interface explorateur dans le cas des feuilles d'aide de Visual Basic et dans l'explorateur de Windows. La feuille active est représentée par deux fenêtres, l'une à droite et l'autre à gauche. La fenêtre de gauche affiche une vue détaillée des données dans la fenêtre de droite.

## Terminologie MDI

Pour pouvoir exploiter correctement le mode MDI, vous devez en connaître la terminologie.

La feuille principale qui sert de conteneur aux autres feuilles est souvent appelée feuille Parent ou fenêtre Parent.

dans notre projet nous l'appellerons [frmParent](#).

Cette feuille contient toutes les autres feuilles qui seront créées et dont l'appellation sera feuille Fille ou fenêtre fille, dans notre projet nous l'appellerons [frmChild](#).

Chaque document nouvellement créé s'ouvre dans une fenêtre fille à l'intérieur d'une feuille parent.

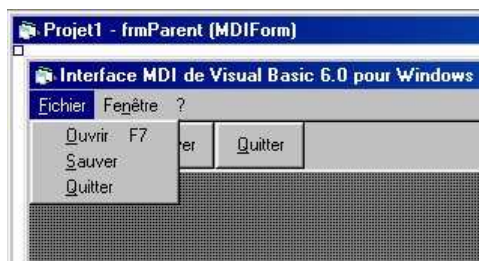
Les fenêtres filles ne peuvent pas sortir du cadre de la feuille parent.

Techniquement parlant, la fenêtre fille se distingue d'une fenêtre classique par sa propriété MDIChild qui aura comme caractéristique, la valeur True.

Pour créer une application MDI, choisissez dans le menu Projet, ajoutez une feuilleMDI, après avoir créé la première feuille (MDIForm) vous la renommez en frmParent, vous créez une seconde feuille à l'aide de l'étiquette "Projet" ajoutez une feuille, (frm2) que vous renommez sous le nom de frmChild, puis définissez la valeur True à la propriété MDIChild de la nouvelle feuille.

Vous pouvez télécharger l'exemple type d'une application MDI prête à fonctionner.

## Téléchargement



Vous avez appris dans mes précédents Tutoriels à réaliser des menus à l'aide du créateur de menu.

Vous créez les menus correspondants à la figure ci-contre, à savoir :

**Menu Fichier** - Le menu Fichier contient trois sous menu intitulés Ouvrir, Sauver et Quitter

**Menu Fenêtre**, contient les sous menu Cascade et Mosaïque et "?" contient l'étiquette A propos de...

Un projet MDI peut également contenir des feuilles standard servant en particulier de boîte de dialogue modales, car les feuilles secondaires ne peuvent pas être affichées de manière modale. Pour les distinguer, Visual Basic affecte à chacun de ces trois types de feuilles une icône spécifique dans leur barre de titre et dans l'explorateur de projets.

Les applications MDI sont généralement équipées d'une barre d'outils et d'une barre d'état. Pour insérer des images ou des boutons dans une feuille MDI il faut obligatoirement insérer un objet image de type PictureBox et affecter la propriété Align à la valeur 1-Align Top pour insérer la barre sur le haut de la feuille(barre d'outils) et à la propriété 2 - Align Bottom pour insérer la barre dans le bas de la feuille(barre d'état).

Vous comprenez que pour installer deux barres dans la feuille MDI il faudra insérer deux objets PictureBox.

La barre d'outils vient automatiquement s'insérer au dessous de la barre de titre qui est partie intégrante de la feuille Parent.

A partir de ces deux barres précédemment citées, nous pouvons construire l'arborescence des instructions de commande nécessaire à l'organisation de notre application.

Pour vous aider à comprendre les applications MDI, nous allons écrire un projet de bloc notes qui ne demande pas énormément d'écriture de code et qui va nous permettre de vous familiariser avec la technique relative aux applications Multi Document Interface (MDI).

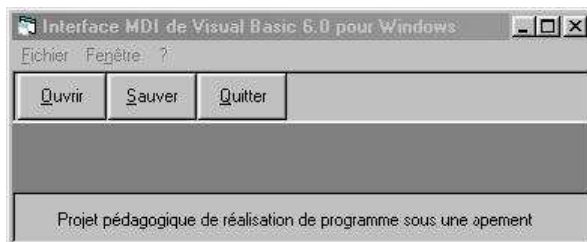
## L'application BlocNote

Après avoir dézippé le fichier téléchargé, nous insérons dans la feuille frmParent deux objets PictureBox comme précédemment expliqué en modifiant leurs propriétés pour que l'affichage de ces nouvelles barres soient conforme à la description indiquée.

Dans la barre supérieure nous insérons trois boutons, le bouton "Ouvrir", le bouton "Sauver" et le bouton "Quitter".

Nous insérons également un contrôle de boîtes de dialogue intitulé CommonDialog que nous positionnons n'importe où puisque à l'utilisation il devient invisible. Néanmoins pour nous habituer à cacher ce qui ne doit pas être vu par l'utilisateur et uniquement par soucis pédagogique, je vous demanderai de le positionner à l'intérieur de la barre d'outils nouvellement crée afin que ce contrôle ne soit plus visible.

Si ce contrôle n'est pas chargé dans votre boîte à outils, il faut aller le chercher depuis le générateur de programme(VB) et charger comme déjà expliqué dans des leçons précédentes le fichier [Comdlg32.ocx](#) qui se trouve dans le répertoire de Windows\system de votre système d'exploitation.



Voici un aperçu de l'application qui a été réduite à sa plus simple expression pour des raisons d'espaces mémoire.

Nous visualisons dans la barre de menu les étiquettes Fichier, Fenêtre et le point d'interrogation(commande About).

Dans la barre d'outils les boutons Ouvrir, Sauver et Quitter

Dans la barre de tache nous avons inséré un Label et inscrit une annotation qui peut éventuellement contenir votre Copyright ainsi que votre nom.

Pour votre information, les propriétés de la "Form" frmParent que j'ai réalisé ont les valeurs suivantes :

Height : 7215

Width : 10350

Vous constatez que cela ne correspond pas à l'image qui est affichée ci-dessus, à vous de présenter votre interface en fonction de vos aptitudes créatrices, n'oublions pas que Visual Basic est aussi un environnement graphique et que la présentation des programmes est fonction du bon vouloir du programmeur que vous êtes.

Voici la saisie du code à insérer dans l'application

### [Voici le code à insérer dans la fenêtre frm Parent](#)

*A l'emplacement "Général" "Déclarations"*

```

Dim FenetreMDI As Form
'
Private Sub butOuvrir_Click( )
'Définitions des propriétés de al boîte de dialogue
CMDialog1.DialogTitle = "Choisissez un fichier"
CMDialog1.CancelError = True
CMDialog1.Filter = True
CMDialog1.Filter = "Fichiers Textes(*.txt)"
CMDialog1.FilterIndex = 1
CMDialog1.InitDir = "C:\"
'
'Définition de la routine de renvoi d'erreur
On Error GoTo ExitbutOuvrir
'Affichage de la boîte de dialogue
CMDialog1.ShowOpen
'Ouverture du fichier sélectionné
Set FenetreMDI = New frmChild
FenetreMDI.Show
FenetreMDI.Width = 5000
FenetreMDI.Height = 5800
crlf = Chr$(13) + Chr$(10)
Open CMDialog1.FileName For Input As #1
Line Input #1, texte
lire = texte
While Not EOF(1)
Line Input #1, texte
lire = lire + crlf + texte
Wend
FenetreMDI.Text1.Text = lire
FenetreMDI.Caption = CMDialog1.FileName
Close #1
Exit Sub
ExitbutOuvrir:
'Sortie sans provoquer d'erreur
End Sub
'
Private Sub butQuitter_Click( )
'Fermer toutes les feuilles et quitter le programme
T$ = "Attention"
M$ = "Voulez vous vraiment quitter le programme ?"
Reponse% = MsgBox(M$, 4 + 32, T$)
If Reponse% = 6 Then
'Sortir du programme
End
End If
End Sub
'
Private Sub butSauver_Click( )
'Définition des propriétés de la boîte dedialogue
CMDialog1.DialogTitle = "Enregistrer le fichier sous..."
CMDialog1.CancelError = True
CMDialog1.Filter = "txt (*.TXT | (*.TXT)"
CMDialog1.FilterIndex = 1
CMDialog1.InitDir = "C:\"
CMDialog1.FileName = "Saver.txt"
'
'Définition de la routine de traitement des erreurs
On Error GoTo ExitSaveFile
'Affichage de la boîte de dialogue
CMDialog1.ShowSave
'
'Sauvegarde du fichier
Open CMDialog1.FileName For Output As #1
Print #1, FenetreMDI.Text1.Text
Close #1
Exit Sub
ExitSaveFile:
'Sortie sans provoquer d'erreurs

```

```
End Sub
'
Private Sub MDIForm_Load( )
'Centrer la feuille à l'écran
CenterForm Me
End Sub
'
Private Sub mnu_About_Click( )
frmAbout.Show 1
End Sub
'
Private Sub mnu_Cascade_Click( )
frmParent.Arrange vbCascade
End Sub
'
Private Sub mnu_Mosaique_Click( )
frmParent.Arrange vbTileVertical
End Sub
'
Private Sub mnu_Ouvrir_Click( )
butOuvrir_Click
End Sub
'
Private Sub mnu_Quitter_Click( )
butQuitter_Click
End Sub
'
Private Sub mnu_Sauver_Click( )
butSauver_Click
End
```

Nous créons un module et saisissons la formule de centrage de feuilles précédemment décrites dans d'autres Tutoriels

```
Sub CenterForm(frm As Form)
'Centre la feuille à l'écran
frm.Left = (Screen.Width - frm.Width) / 2
frm.Top = (Screen.Height - frm.Height) / 2
End Sub
```

La fenêtre fille ne contient rien de particulier si ce n'est qu'un objet Texte qui aura comme propriété Name Text1.

**Télécharger.**

Vous pouvez bien sur améliorer ce programme qui est libre de droits et qui vous est offert par votre serviteur en espérant que vous appréciez les concepts que je développe pour vous.

## Prochain tutoriel

 ***Les attributs de fichiers, La modification des attributs système .***

