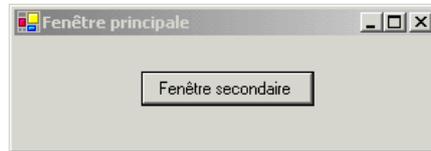


Programmation multi formulaires

L'ouverture de formulaires secondaires

Une application Windows dispose d'un formulaire de démarrage, qui est de ce fait son formulaire principal, et toute fenêtre doit être instanciée et affichée à la demande.

Soit une application qui contient deux formulaires, **FBase** et **FTest**. Le formulaire de démarrage est **FBase**. Il s'affiche automatiquement dès l'application chargée. Il est muni d'un bouton *Fenêtre secondaire* destiné à ouvrir **FTest**. Ce dernier est muni d'un bouton *Fin* pour le retour à **FBase**.



Voici le code d'ouverture de **FTest** dans **FBase** :

```
Private Sub BOuvreFTest_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles BOuvreFTest.Click
    Dim FSecondaire As New FTest
    FSecondaire.Show()
End Sub
```

Cette écriture de **BOuvreFTest** permet l'instanciation simultanée de plusieurs fenêtres **FTest**, chacune ayant ses éventuelles données réinitialisées. Les 2 lignes de codes de cette procédure peuvent être remplacées par **FTest.Show()**. Dans ce cas, **FTest** est unique et conserve ses éventuelles données d'un appel à l'autre.

Le code de fermeture de **FTest** est le suivant :

```
Private Sub BFin_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles BFin.Click
    Me.Close()
End Sub
```

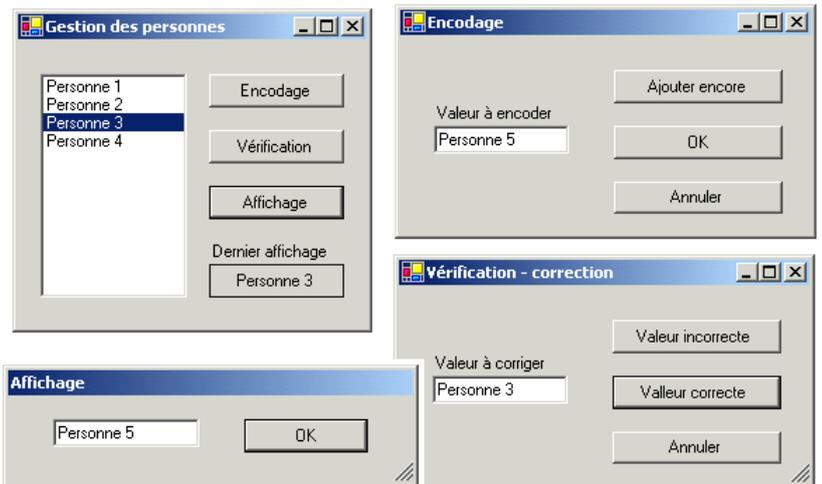
Dans cette situation multi formulaires, l'usage de la méthode **Me.Close** n'équivaut pas à utiliser l'instruction **End**. Cette dernière provoque l'arrêt de toute l'application et ne doit être utilisée qu'à bon escient. Dans cet exemple, il n'est pas question de tout arrêter, mais seulement de fermer **FTest** et de continuer les traitements avec **FBase**.

Le passage de données entre formulaires

La communication d'informations entre deux fenêtres est chose courante. Les techniques qui permettent cette communication sont multiples.

Afin d'en illustrer quelques unes, voici une hypothétique application de gestion d'un signalétique complexe de personnes comportant notamment les quatre formulaires suivants :

FBase, dont le titre est *Gestion des personnes*, est une fenêtre de sélection d'une personne. C'est par elle aussi que se commande l'encodage d'une nouvelle personne.



FAffiche, dont le titre est *Affichage*, est la fenêtre de consultation de l'ensemble des informations concernant la personne sélectionnée dans **FBase**. Dans cet exemple, la seule information affichable est le nom de la personne.

FVerifie, dont le titre est *Vérification - correction*, est la fenêtre de modification des informations concernant la personne sélectionnée dans **FBase**. Dans cet exemple, la seule information modifiable est le nom de la personne.

FEncode, dont le titre est *Encodage*, est la fenêtre d'encodage des informations concernant une nouvelle personne. Dans cet exemple, la seule information à encoder est le nom de la personne.

L'affichage du signalétique complet pour consultation se commande par le bouton *Affichage* de **FBase**. L'affichage du signalétique complet pour édition se commande par le bouton *Vérification*. Dans un cas comme dans l'autre, il faut qu'une personne soit sélectionnée dans la **ListBox MesPersonnes**.



Voici le code de **FBase** qui provoque l'ouverture des fenêtres concernées en leur passant l'information nécessaire, à savoir le nom de la personne dans cet exemple.

```
Private Selection As Integer
```

```
' Seule variable de la classe FBase, pour  
mémoriser l'indice d'une sélection
```

Une procédure privée a été créée pour l'affichage d'un message en cas de tentative d'affichage ou d'édition alors qu'il n'y a pas de sélection. Cette procédure peut donc être appelée par **BAffiche_Click** et par **BVerification_Click**. Le libellé du bouton concerné lui est passé en paramètre de sorte à rendre le message plus précis.

```
Private Sub MessageErreur(ByVal TexteDuBouton As String)
```

```
    MessageBox.Show("Une personne doit être sélectionnée pour utiliser " & TexteDuBouton)  
End Sub
```

Pour l'affichage, la méthode choisie pour la communication de l'information à **FAffiche** consiste à profiter de la portée par défaut de tout objet déposé sur un formulaire, qui est **Friend**, pour affecter directement la **TextBox** d'affichage **TUnePers**.

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

```
    If MesPersonnes.SelectedItems.Count = 1 Then ' Seulement si une sélection existe  
        Dim Fs As New FAffiche  
        Fs.TUnePers.Text = MesPersonnes.SelectedItem ' Affecter la sélection à la TextBox  
        Fs.Owner = Me ' Passer le nom du Form appelant  
        Fs.ShowDialog()  
    Else  
        MessageErreur(BAffiche.Text) ' Passer le libellé du bouton au message  
    End If  
End Sub
```

Pour l'édition, la méthode choisie pour la communication de l'information à **FVerifie** consiste à utiliser une propriété **Entree** créée à cette fin dans cette classe. Cette façon de faire est conforme au contrôle d'accès préconisé en POO : on accède aux propriétés d'une classe par les méthodes qu'elle expose pour cela.

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

```
    If MesPersonnes.SelectedItems.Count = 1 Then  
        Dim Fs As New FVerifie  
        Selection = MesPersonnes.SelectedIndex ' Mémoriser l'indice tant qu'on a le focus  
        Fs.Entree = MesPersonnes.SelectedItem ' Affecter à une propriété du form  
        Fs.Owner = Me ' Passer le nom du Form appelant  
        Fs.ShowDialog()  
    Else  
        MessageErreur(BVerification.Text) ' Passer le libellé du bouton au message  
    End If  
End Sub
```

Dans cette application, le nom du formulaire appelant doit être passé aux formulaires instanciés. C'est la propriété **Owner**, au nom *symbolique*, qui est utilisée à cet effet mais la propriété **Tag** aurait tout aussi bien fait l'affaire.

Quand un formulaire est la propriété d'un autre, il est réduit et fermé en même temps que son formulaire propriétaire. Par exemple, si *Form2* appartient à *Form1*, la fermeture ou la réduction de *Form1* entraîne celle de *Form2*. En outre, un formulaire détenu par un autre formulaire n'est jamais affiché derrière son propriétaire.

Si ces caractéristiques de la relation de propriété ne conviennent pas, il est toujours possible de créer une propriété dans le formulaire secondaire pour lui transmettre le nom de l'appelant. Il suffit d'affecter ensuite cette propriété de la même façon que n'importe quelle autre valeur.

Voici le code complet de **FAffiche** :

```
Public Class FAffiche

Dim Fp As FBase ' Référence à un Form de type FBase

Private Sub FAffiche_Activated(ByVal sender As Object, ByVal e As System.EventArgs)
    Handles MyBase.Activated
    Fp = Me.Owner ' Désignation du formulaire appelant à Fp
    BOK.Select()
End Sub

Private Sub BOK_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    Handles BOK.Click
    Fp.LDernierAffiche.Text = Me.TUnePers.Text ' Accès direct à un objet de FBase
    Me.Close()
End Sub

End Class
```

L'application conserve dans l'étiquette **LDernierAffiche** le dernier affichage réalisé par **FAffiche**. Pour l'exemple, c'est de la classe **FAffiche** qu'est affecté ce **Label**. La méthode choisie pour la communication de l'information à **FBase** est la même que celle utilisée par **FBase** pour transmettre le texte à **FAffiche**, c'est-à-dire l'affectation directe d'un objet du formulaire destinataire.