

## L'impression des documents

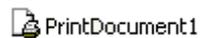
Les composants permettant le lancement d'une impression sont **PrintDocument** et **PrintPreviewDialog**. Dans la mesure où il est souhaitable de permettre à l'utilisateur de régler les marges de son document et de sélectionner une imprimante, les composants **PageSetupDialog** et **PrintDialog** sont également nécessaires. Le composant **PrintPreviewControl** permet seulement la visualisation du document désigné par **PrintDocument** et c'est au programmeur de prévoir la commande d'impression.

Les composants **PrintDocument** et **PrintPreviewDialog** provoquent l'événement **PrintPage** en réponse duquel il faut programmer le détail de l'impression. C'est notamment dans cette procédure événementielle qu'est réalisée la lecture du fichier contenant le document, à moins qu'il s'agisse d'impression directe.

Pour pouvoir utiliser les outils de lecture de fichier, l'Imports **System.IO** est indispensable et le fichier, qui doit être utilisé par plusieurs procédures, doit être déclaré au niveau de la classe. Dans cette situation, les premières lignes de codes de la classe sont :

```
Imports System.IO
Public Class FBase
    Dim MonDoc As StreamReader
```

### Le moteur d'impression : PrintDocument



Le composant **PrintDocument** est toujours invisible à l'exécution. L'impression directe d'un document, sans possibilité pour l'utilisateur de sélectionner l'imprimante, ni de modifier les marges, se commande par le code suivant :

```
PrintDocument1.Print()
```

L'exécution de la méthode **Print** provoque l'événement **PrintPage** et l'impression peut être réalisée comme ceci :

```
Private Sub PrintDocument1_PrintPage(ByVal sender As Object, ByVal e As
    System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
    Dim Ligne As String
    Dim Position As System.Drawing.PointF ' PointF et RectangleF sont les position et
    Dim Rectangle As RectangleF ' dimensions de la zone d'impression
    Dim Police As Font ' exprimées en nombres réels (Floating point)
    Dim HauteurLigne As Single

    Ligne = "Voici la ligne à imprimer" ' Une première ligne à imprimer

    Police = New Font("Courier New", 10, FontStyle.Bold)
    Position = New PointF(e.MarginBounds.Left, e.MarginBounds.Top)
    HauteurLigne = Police.GetHeight(e.Graphics)
    Rectangle.Location = Position
    Rectangle.Width = e.MarginBounds.Width ' Récupérer la largeur entre marges
    Rectangle.Height = HauteurLigne ' et utiliser la hauteur de ligne
    e.Graphics.DrawString(Ligne, Police, Brushes.Black, Rectangle)

    Ligne = "et en voici une autre" ' une autre ...

    Police = New Font("Times New Roman", 16, FontStyle.Bold)
    Position = New PointF(e.MarginBounds.Left, e.MarginBounds.Top + HauteurLigne)
    HauteurLigne = Police.GetHeight(e.Graphics)
    Rectangle.Location = Position
    Rectangle.Width = e.MarginBounds.Width
    Rectangle.Height = HauteurLigne
    e.Graphics.DrawString(Ligne, Police, Brushes.Black, Rectangle)

    ' ... tant qu'il y a des lignes ... ' encore une ...

    e.HasMorePages = False
End Sub
```

L'impression d'un fichier texte, sans possibilité pour l'utilisateur de sélectionner l'imprimante, ni de modifier les marges, se commande par le code suivant :

```
PrintDocument1.DocumentName = "X:\TXT\MonTexte.Txt"
MonDoc = New StreamReader(PrintDocument1.DocumentName)
PrintDocument1.Print()
MonDoc.Close()
```

L'exécution de la méthode `Print` provoque l'événement `PrintPage` et l'impression peut être réalisée, mais cette fois les lignes à imprimer sont fournies par la lecture du fichier. La procédure `PrintDocument1_PrintPage` devient par exemple :

```
Private Sub PrintDocument1_PrintPage(ByVal sender As Object, ByVal e As
    System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
    Dim Ligne As String
    Dim Position As System.Drawing.PointF
    Dim Rectangle As RectangleF
    Dim Police As Font
    Dim HauteurLigne As Single
    Dim NbrLignesParPage As Integer
    Dim NumLigne As Integer = 0
    ' Comptage des lignes imprimées
    ' Dans cet exemple, toutes les lignes
    ' utilisent la même police

    Police = New Font("Courier New", 10, FontStyle.Bold)
    HauteurLigne = Police.GetHeight(e.Graphics)
    ' Nombre de lignes par page est la
    ' hauteur de la zone imprimable
    ' divisée par la hauteur d'une ligne

    NbrLignesParPage = e.MarginBounds.Height / HauteurLigne

    Do Until MonDoc.Peek = -1
        ' Lire le contenu du fichier ligne par
        ' ligne et calculer la position de la
        ' ligne à imprimer
        Ligne = MonDoc.ReadLine()
        Position = New PointF(e.MarginBounds.Left, e.MarginBounds.Top +
            (NumLigne * HauteurLigne))
        ' Définir les coordonnées de la zone
        ' dans laquelle la ligne sera imprimée
        ' avec la position calculée et sa
        ' taille
        Rectangle.Location = Position
        Rectangle.Width = e.MarginBounds.Width
        ' Récupérer la largeur entre marges
        Rectangle.Height = HauteurLigne
        ' et utiliser la hauteur de ligne

        e.Graphics.DrawString(Ligne, Police, Brushes.Black, Rectangle)

        NumLigne += 1
        ' Incrémenter le compteur de lignes
        If NumLigne > NbrLignesParPage Then Exit Do
    Loop
    ' et vérifier si on est en fin de page

    If MonDoc.Peek <> -1 Then
        ' Relancer l'impression de la page
        e.HasMorePages = True
        ' suivante
    Else
        MonDoc.Close()
        MonDoc = New StreamReader(TDocument.Text)
        ' En prévisualisation, cette procédure
        ' est exécutée une première fois pour
        ' l'affichage et il faut rouvrir le
        ' fichier pour l'impression éventuelle
    End If
End Sub
```

Pour donner à l'utilisateur la possibilité de sélectionner l'imprimante et de modifier les marges, il faut associer les composants `PrintDialog` et `PageSetupDialog` à `PrintDocument` et permettre à l'utilisateur de les utiliser pour définir ses options. Il faut noter que `PageSetupDialog` permet aussi la désignation de l'imprimante. Ces associations se programment par ces lignes :

```
PageSetupDialog1.Document = PrintDocument1
PrintDialog1.Document = PrintDocument1
```

## La mise en page : PageSetupDialog



Une fois un **PrintDocument** et une **PageSetupDialog** associés, l'activation de la boîte de dialogue se programme par :

```
PageSetupDialog1.ShowDialog()
```

Les propriétés **AllowMargins**, **AllowOrientation**, **AllowPaper** et **AllowPrinter** dont la valeur par défaut est **True** peuvent recevoir la valeur **False** pour interdire l'accès aux fonctionnalités correspondantes.

## La sélection de l'imprimante : PrintDialog



Une fois un **PrintDocument** et une **PrintDialog** associés, l'activation de la boîte de dialogue se programme par :

```
PrintDialog1.ShowDialog()
```

Les propriétés **AllowSomePages**, **AllowSelection** et **AllowPrintToFile** peuvent aussi recevoir une des valeurs **True** ou **False** selon le degré de liberté accordé à l'utilisateur.

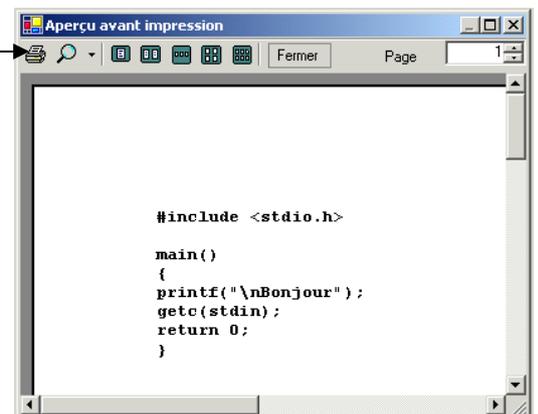
## La fenêtre de prévisualisation : PrintPreviewDialog



La boîte de dialogue de prévisualisation affiche le document tel qu'il sera imprimé et permet à l'utilisateur d'en commander l'impression.

Le composant **PrintPreviewDialog** doit être associé à un **PrintDocument** avant d'être activé. Il est bien entendu que le **PrintDocument** doit être réglé au préalable. Le code suivant réalise cette association et active la boîte de dialogue.

```
PrintPreviewDialog1.Document = PrintDocument1
MonDoc = New StreamReader(PrintDocument1.DocumentName)
PrintPreviewDialog1.ShowDialog()
MonDoc.Close()
```

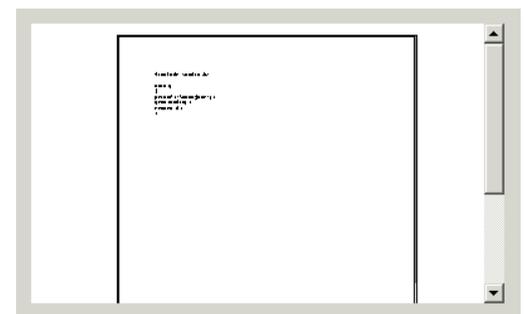


## Le composant de prévisualisation : PrintPreviewControl

Le composant de prévisualisation affiche le document tel qu'il sera imprimé, mais ne permet pas à l'utilisateur d'en commander l'impression.

Le composant **PrintPreviewControl** doit être associé à un **PrintDocument** avant d'être activé. Il est bien entendu que le **PrintDocument** doit être réglé au préalable. Le code suivant réalise cette association et active l'affichage du document dans le composant.

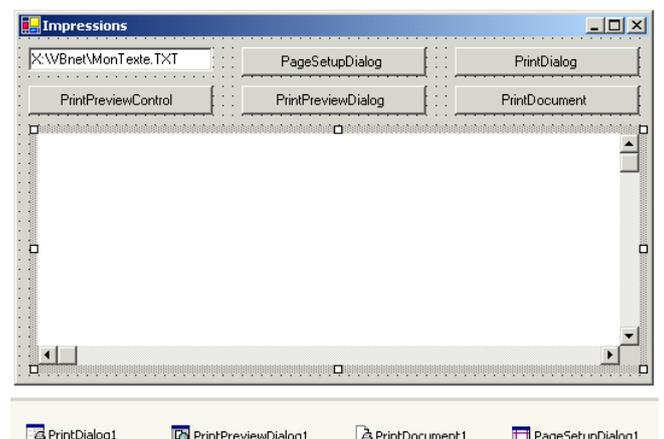
```
MonDoc = New StreamReader(PrintDocument1.DocumentName)
PrintPreviewControl1.Document = PrintDocument1
PrintPreviewControl1.InvalidaterPreview()
```



## Exemple récapitulatif

Tous les composants étudiés ci avant sont regroupés dans une même application permettant l'expérimentation de chacun dans une situation de traitement d'un fichier texte en autorisant la sélection de l'imprimante et la modification des marges par l'utilisateur.

Les codes communs de réglage des composants ont été réunis dans une procédure **InitialiseComposants** qui est appelée par chaque procédure événementielle de réponse au clic d'un bouton.



La boîte de texte **TDocument** contient le chemin et le nom du fichier à traiter.

```

Imports System.IO
Public Class FBase
    Dim MonDoc As StreamReader

Private Sub InitialiseComposants(ByVal Test As Byte)
    Select Case Test
        Case 1
            PageSetupDialog1.Document = PrintDocument1      ' Association pour PageSetupDialog
        Case 2
            PrintDialog1.Document = PrintDocument1          ' Association pour PrintDialog
        Case 3
            PrintPreviewDialog1.Document = PrintDocument1   ' Association pour PrintPreviewDialog
        Case 4
            PrintPreviewControll.Document = PrintDocument1  ' Association pour PrintPreviewControl
        Case 0
            ' Rien à faire ici. Le paramètre 0 est prévu pour entrer dans la procédure, même s'il n'y
            ' a aucune association à régler (comme dans BPrintDocument_Click). Les 2 lignes écrites
            ' après End Select doivent toujours être exécutées.
    End Select
    PrintDocument1.DocumentName = TDocument.Text
    MonDoc = New StreamReader(PrintDocument1.DocumentName)
End Sub

Private Sub BPageSetup_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    InitialiseComposants(1)
    PageSetupDialog1.ShowDialog()
    Handles BPageSetup.Click
End Sub

Private Sub BPrintDial_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles BPrintDial.Click
    InitialiseComposants(2)
    PrintDialog1.ShowDialog()
End Sub

Private Sub BPrintPreviewDial_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles BPrintPreviewDial.Click
    InitialiseComposants(3)
    PrintPreviewDialog1.ShowDialog()
    MonDoc.Close()
End Sub

Private Sub BPrintPreviewControl_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles BPrintPreviewControl.Click
    InitialiseComposants(4)
    PrintPreviewControll.InvalidatePreview()      ' Force la mise à jour du contrôle
End Sub

Private Sub BPrintDocument_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles BPrintDocument.Click
    InitialiseComposants(0)
    PrintDocument1.Print()
    MonDoc.Close()
End Sub

Private Sub PrintDocument1_PrintPage(ByVal sender As Object, ByVal e As
    System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
    ' Remplacer ici la procédure PrintDocument1_PrintPage déjà écrite pour l'impression d'un fichier
    ' texte
End Sub

End Class

```