

Les options d'impression

Sommaire

1	Introduction.....	2
2	Le principe d'impression en Windows Form	2
2.1	Approche de l'objet PrintDocument.....	2
2.2	Principe de l'impression.....	3
3	Imprimer avec les boîtes de dialogues.....	4
3.1	La classe contenant tous les composants relatifs à l'impression : le <i>PrinterSettings</i>	4
3.2	Les composants <i>Dialog</i>	5
3.2.1	Le composant <i>PageSetupDialog</i>	5
3.2.2	Le composant <i>PrintPreviewDialog</i>	8
3.2.3	Le composant <i>PrintDialog</i>	11
4	Imprimer grâce à l'événement <i>PrintPage</i>	13
5	Les différents types d'impression.....	13
5.1	Imprimer un graphisme ou une image	13
5.2	Imprimer un texte	16
6	Point sur la sécurité d'impression	17
7	Aperçu avant impression avec le <i>PrintPreviewControl</i>	18
8	Conclusion	19

1 Introduction

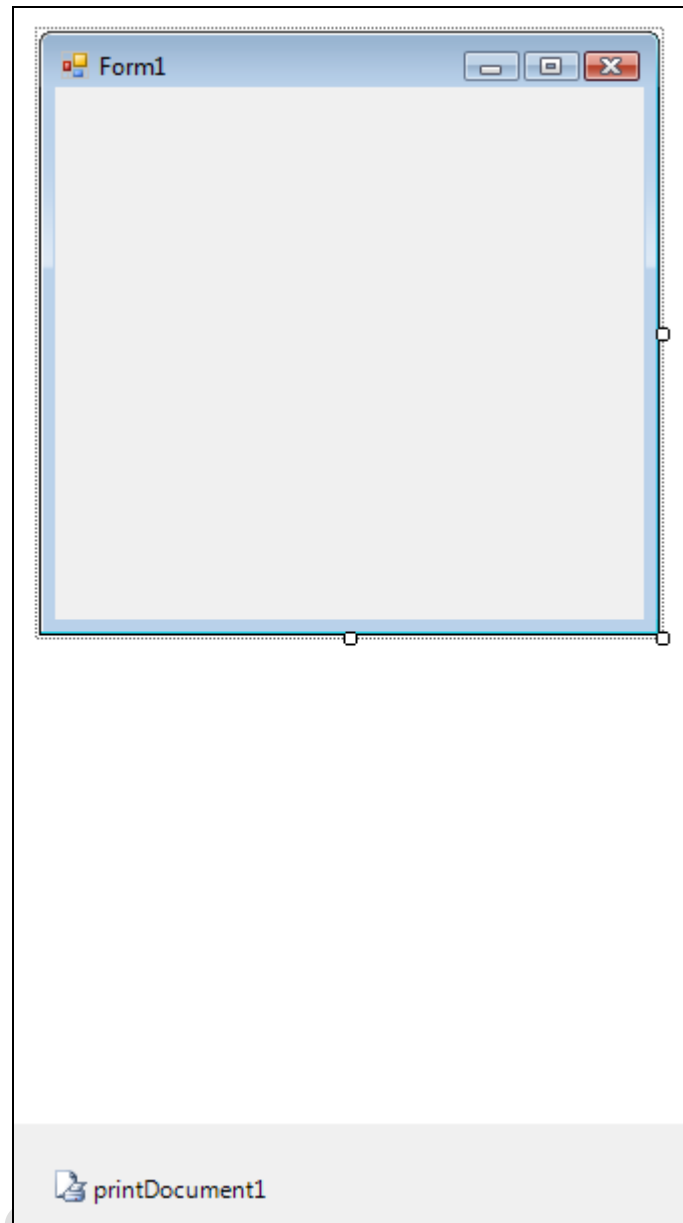
Dans ce chapitre sur les impressions en Windows Form, vous apprendrez à gérer les différents contrôles et objets permettant la gestion de l'impression d'un document. Pour cela il existe trois composants sous forme de boîtes de dialogues ainsi que plusieurs autres contrôles et objets. Ces éléments vous permettront de personnaliser l'impression autant à la conception qu'à l'exécution.

2 Le principe d'impression en Windows Form

2.1 Approche de l'objet PrintDocument

Le PrintDocument définit un objet représentant un document qui le renvoie à la sortie de l'imprimante. Cette classe vous permet d'imprimer des documents contenant des textes ou encore des images. Ainsi lorsque vous imprimez un document celui-ci est symbolisé par la classe PrintDocument. En effet il n'existe pas de représentation visuelle mais il apparaît dans les délais de conception. Elle possède toutes les informations nécessaires pour établir l'impression du document défini.

Remarque : PrintDocument se situe dans l'espace de nom System.Drawing.Printing



Comme l'objet PrintDocument ne possède pas de représentation visuelle celui-ci est affiché en bas lorsque vous l'insérez dans votre Formulaire de puis la ToolBox

2.2 Principe de l'impression

Vous devez donc créer une instance de la classe PrintDocument afin d'imprimer votre document tout en définissant les propriétés nécessaires à votre impression. Puis vous appelez la méthode Print dans le but de lancer l'impression. En effet cette méthode fait elle-même appel aux méthodes OnStartPrint, OnEndPrint, OnStartPage, OnEndPage qui indiquent le mode d'impression du document. Cela engendre ainsi un ou plusieurs événements PrintPage. Par ailleurs un événement PrintPage sera créé pour chaque page si l'impression comporte plusieurs pages.

3 Imprimer avec les boîtes de dialogues

3.1 La classe contenant tous les composants relatifs à l'impression : le *PrinterSettings*

Lorsqu'un utilisateur voudra imprimer un document, il voudra surement pouvoir personnaliser son impression. Ces options sont configurables et données par le concepteur. Tout ce qui concerne l'impression se trouve dans la classe *PrinterSetting*. Il permet à l'aide des composants *Dialog* (décrits dans la prochaine partie) de personnaliser nos impressions : nombres de pages, imprimante à utiliser, ...

Voici les principales propriétés du *PrinterSettings* :

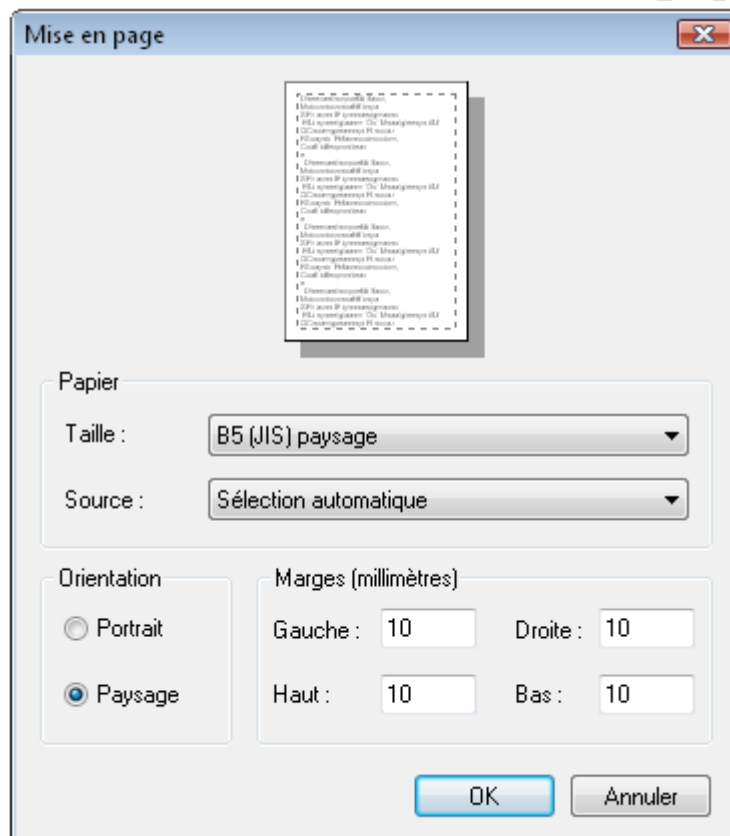
Propriété	Description
Copies	Elle permet de connaître et de définir le nombre de fois que le document devra être imprimé.
DefaultPagesettings	Elle permet de connaître les paramètres par default d'une page pour une imprimante.
InstalledPrinters	Elle permet de connaître le nom de toutes les imprimantes installées sur l'ordinateur. Cette propriété est <i>Static</i> .
PaperSizes	Elle permet de connaître les formats d'impression pris en charge par une imprimante.
PaperSources	Elle permet de connaître les bacs à papier disponibles sur une imprimante.
PrinterName	Elle permet de connaître et définir le nom de l'imprimante à utiliser.
PrinterResolutions	Elle permet de connaître toutes les résolutions prises en charge par une imprimante.
SupportsColor	Elle permet de savoir si une imprimante permet l'impression en couleur.

3.2 Les composants *Dialog*

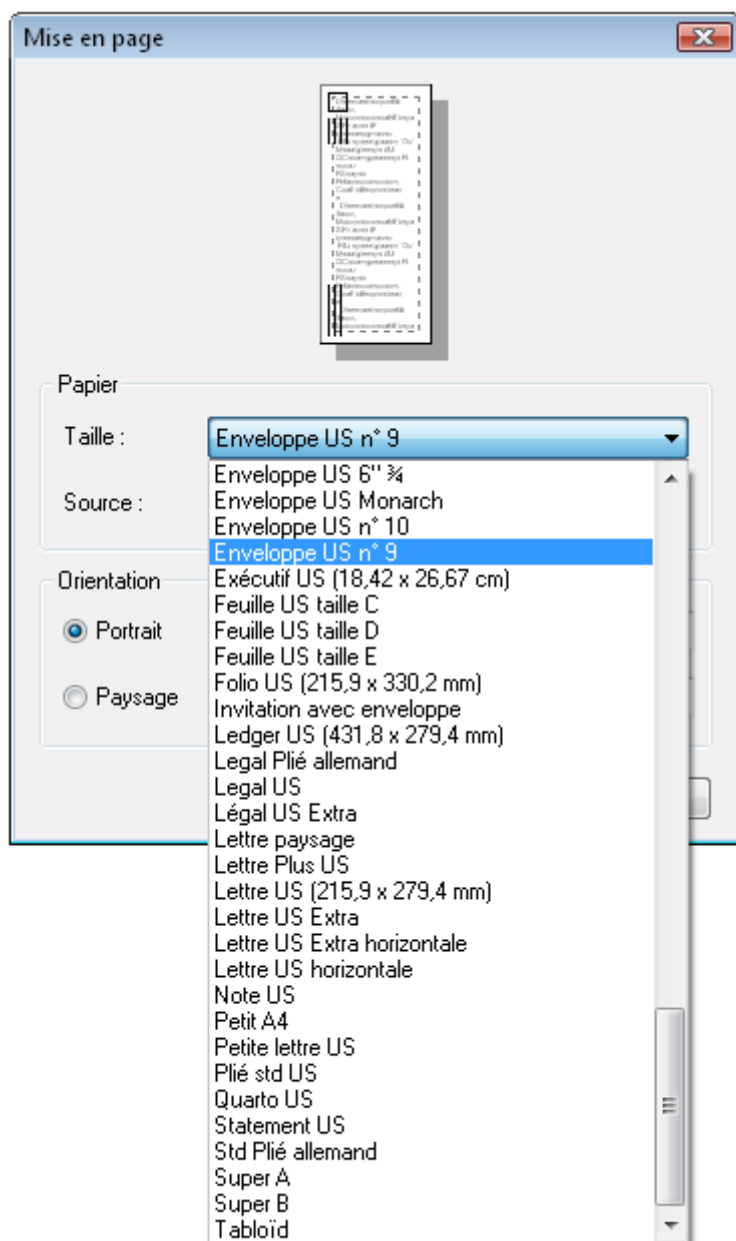
Les composants *Dialog* sont tous les composants permettant d'afficher une boîte de dialogue permettant certaines actions. Il existe trois boîtes de dialogues utiles pour gérer les options d'impression en Windows Form : le *PageSetupDialog*, le *PrintPreviewDialog* et le *PrintDialog*.

3.2.1 Le composant *PageSetupDialog*

Le *PageSetupDialog* est le composant permettant de travailler sur la configuration des pages d'un document à imprimer. Il permet d'avoir une boîte de dialogue donnant les options suivantes : taille, source, orientation et marges :



Il possède de base un grand nombre de tailles de papiers disponibles comme vous pourrez le constater :



Pour utiliser le *PageSetupDialog*, il faut instancier un *PrintDocument* puis le relier à un *PageSetupDialog* comme dans l'exemple suivant. Vous pouvez créer un *PageSetupDialog* à la conception en faisant simplement un Drag and Drop sur votre formulaire. Dans cet exemple nous appellerons notre *PrintDocument* « monDocument » et notre *PageSetupDialog* « pageSetupDialog1 » :

```
'VB  
  
Dim monDocument As New PrintDocument()  
  
Public Class Form1  
    InitializeComponent() ' Le PageSetupDialog sera lancé au démarrage  
  
    PageSetupDialog1.Document = monDocument  
    PageSetupDialog1.ShowDialog() ' Pour afficher la boîte de dialogue  
End Class
```

```
//c#  
  
PrintDocument monDocument = new PrintDocument();  
  
public Form1()  
{  
    InitializeComponent(); // Le PageSetupDialog sera lancé au démarrage  
  
    pageSetupDialog1.Document = monDocument;  
    pageSetupDialog1.ShowDialog(); // Pour afficher la boîte de dialogue  
}
```

Il existe des propriétés afin de personnaliser votre *PageSetupDialog* :

Propriété	Description
<i>AllowMargins</i>	Elle permet de connaître si la section des marges du <i>PageSetupDialog</i> est activée.
<i>AllowOrientation</i>	Elle permet de connaître si la section d'orientation du <i>PageSetupDialog</i> est activée.
<i>AllowPaper</i>	Elle permet de connaître si la section papier du <i>PageSetupDialog</i> est activée.
<i>AllowPrinter</i>	Elle permet de connaître si le bouton Imprimer est activé.
<i>Document</i>	Elle permet de connaître quelle page est associée à un <i>PrintDocument</i> .
<i>EnableMetric</i>	Elle permet de connaître les paramètres des marges. Si elles sont affichées en millimètre, elles seront converties automatiquement en centième de pouces.
<i>MinMargins</i>	Elle permet de connaître les marges minimales sélectionnables par un utilisateur.
<i>PageSettings</i>	Elle permet de connaître les paramètres d'une page à modifier.
<i>PrinterSettings</i>	Elle permet de connaître les paramètres d'impression modifiable par l'utilisateur lors d'une impression.
<i>ShowHelp</i>	Elle permet de définir si le bouton « Help » est visible.
<i>ShowNetwork</i>	Elle permet de définir si le bouton « Réseau » est visible.

3.2.2 Le composant *PrintPreviewDialog*

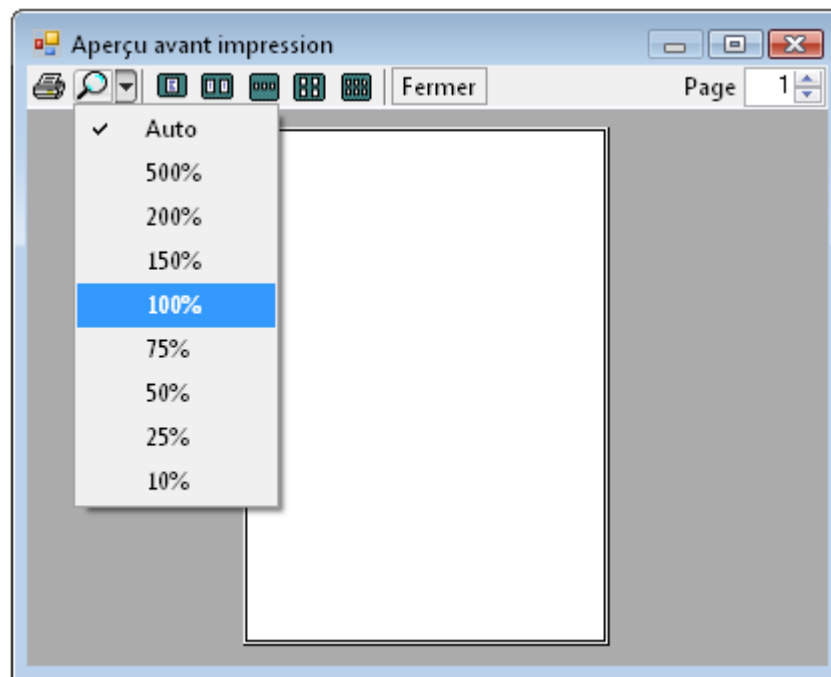
Le *PrintPreviewDialog* est un composant permettant d'avoir un « aperçu avant impression ». Il possède plusieurs outils utiles : un zoom, plusieurs types d'affichage des pages, le moyen de tourner page par page, un bouton fermer ainsi qu'un bouton imprimer. Le bouton imprimer permet d'appeler la méthode *Print* du *PrintDocument* qui dirige tout vers la sortie de l'imprimante.

Pour utiliser le *PrintPreviewDialog*, il faut instancier un *PrintDocument* puis le relier à un *PrintPreviewDialog* comme dans l'exemple suivant. Vous pouvez créer un *PrintPreviewDialog* à la conception en faisant simplement un Drag and Drop sur votre formulaire. Dans cet exemple nous appellerons notre *PrintDocument* « monDocument » et notre *PrintPreviewDialog* « printPreviewDialog1 » :

```
'VB  
  
Dim monDocument As New PrintDocument()  
  
Public Class Form1  
    InitializeComponent() ' Le PrintPreviewDialog sera lancé au démarrage  
  
    PrintPreviewDialog1.Document = monDocument  
    PrintPreviewDialog1.ShowDialog() ' Pour afficher la boîte de dialogue  
End Class
```

```
//C#  
  
PrintDocument monDocument = new PrintDocument();  
  
public Form1()  
{  
    InitializeComponent(); // Le PrintPreviewDialog sera lancé au  
    démarrage  
  
    printPreviewDialog1.Document = monDocument;  
    printPreviewDialog1.ShowDialog(); // Pour afficher la boîte de  
    dialogue  
}
```

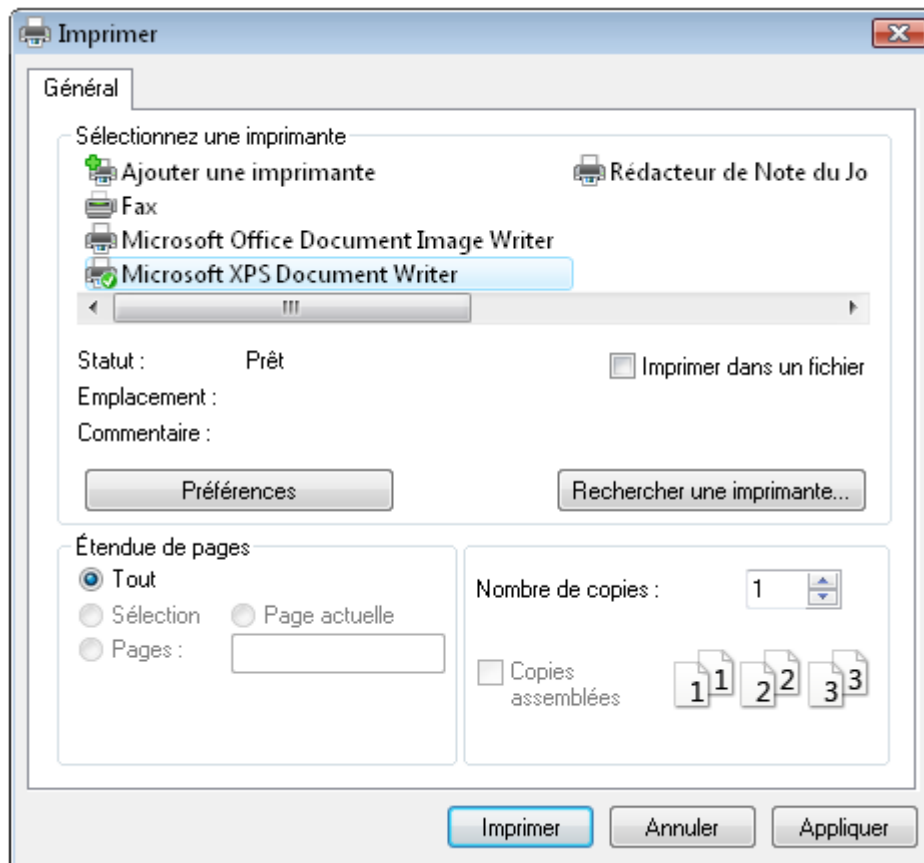
Vous devriez par la suite apercevoir une boîte de dialogue comme celle-ci :



Vous pouvez gérer l'aperçu de votre boîte de dialogue grâce à beaucoup de propriétés. Vous pouvez en avoir la liste [en cliquant ici](#).

3.2.3 Le composant *PrintDialog*

Le *PrintDialog* est un composant permettant d'avoir une boîte de dialogue qui permettra à un utilisateur lors de son exécution de paramétrer l'impression. Elle permet par exemple de sélectionner une imprimante ou de choisir les parties à imprimer d'un document.



Pour utiliser le *PrintDialog*, il faut instancier un *PrintDocument* puis le relier à un *PrintDialog* comme dans l'exemple suivant. Vous pouvez créer un *PrintDialog* à la conception en faisant simplement un Drag and Drop sur votre formulaire. Dans cet exemple nous appellerons notre *PrintDocument* « monDocument » et notre *PrintDialog* « printDialog1 » :

```
'VB
Dim monDocument As New PrintDocument()

Public Class Form1
    InitializeComponent() ' Le PrintDialog sera lancé au démarrage

    PrintDialog1.Document = monDocument
    PrintDialog1.ShowDialog() ' Pour afficher la boîte de dialogue
End Class
```

```
//c#

PrintDocument monDocument = new PrintDocument();

public Form1()
{
    InitializeComponent(); // Le PrintPreviewDialog sera lancé au
    démarrage

    printDialog1.Document = monDocument;
    printDialog1.ShowDialog(); // Pour afficher la boîte de dialogue
}
```

Voici les propriétés principales associées au *PrintDialog* :

Propriété	Description
AllowCurrentPage	Elle permet d'indiquer si le bouton « Current Page » est affiché.
AllowPrintToFile	Elle permet d'indiquer si la check box « Print To File » est affichée.
AllowSelection	Elle permet d'indiquer si le bouton « Selection » est affiché.
AllowSomePages	Elle permet d'indiquer si le bouton « Pages » est affiché.
Document	Elle permet de connaître quelle page est associée à un <i>PrintDocument</i> .
PrinterSettings	Elle permet d'indiquer les paramètres d'impression modifiable par la boîte de dialogue.
PrintToFile	Elle permet d'indiquer si la check box « Print To File » est sélectionnée.
ShowHelp	Elle permet de définir si le bouton « Help » est visible.
ShowNetwork	Elle permet de définir si le bouton « Réseau » est visible.

4 Imprimer grâce à l'évènement PrintPage

L'évènement PrintPage est le principal évènement impliqué dans une impression de document. Cet évènement renvoie tous les objets et toutes les informations nécessaires à une impression. Il est compris dans le gestionnaire d'état d'évènement PrintPageEventArgs (contenant les informations relatives à un évènement levé qui doit être spécifique à celui-ci). Le contenu est envoyé par l'objet Graphics de PrintPageEventArgs.

Remarque : La page imprimée se comporte comme un formulaire, un contrôle ou tous autres dessins représentés par l'objet Graphics.

Vous trouverez ci-dessous les différentes propriétés que possède l'objet PrintPageEventArgs :

Propriétés	Description
Cancel	Définit selon une valeur attribuée à la propriété si l'impression est annulée.
Graphics	Renvoie l'objet Graphics pour l'impression d'images.
HasMorePages	Indique si une autre page doit être imprimée.
MarginBounds	Obtient une zone rectangulaire qui est la partie de la page à l'intérieur des marges.
PageBounds	Obtient aussi une zone rectangulaire mais pour la totalité de la page
PageSettings	Renvoie les paramètres de la page actuelle.

5 Les différents types d'impression

5.1 Imprimer un graphisme ou une image

Pour utiliser une impression graphique, vous devez générer un gestionnaire d'état d'évènement PrintPageEventArgs et incorporer un graphique. Si vous voulez imprimer une image vous utilisez aussi le même principe que pour une impression graphique, cependant vous devrez utiliser la méthode Graphics.DrawImage.

Ainsi dans un premier temps avant de créer votre code, vous devez insérer l'espace de nom System.Drawing.Printing. Ensuite incorporez un objet PrintDocument dans votre formulaire dans la partie design. Puis dans cette même partie, vous double-cliquez sur PrintDocument afin de créer la méthode d'impression. Ensuite vous pouvez insérer votre image grâce à la méthode Graphics.DrawImage. Vous trouverez un exemple de code d'impression d'une image ci-dessous :

```
'VB

Public Sub New()
    InitializeComponent()
    Dim monPrintDocument As New PrintDocument()
    AddHandler monPrintDocument.PrintPage, New
PrintPageEventHandler(AddressOf Me.ImageAImprimer)
    monPrintDocument.Print()

End Sub

Private Sub ImageAImprimer(ByVal sender As System.Object, ByVal ev As
System.Drawing.Printing.PrintPageEventArgs) Handles
PrintDocument1.PrintPage

ev.Graphics.DrawImage(Image.FromFile("C:\Users\Ketin\Desktop\DOTNET\Logo
DotNet\Logo.png"), ev.Graphics.RenderingOrigin)
End Sub
```

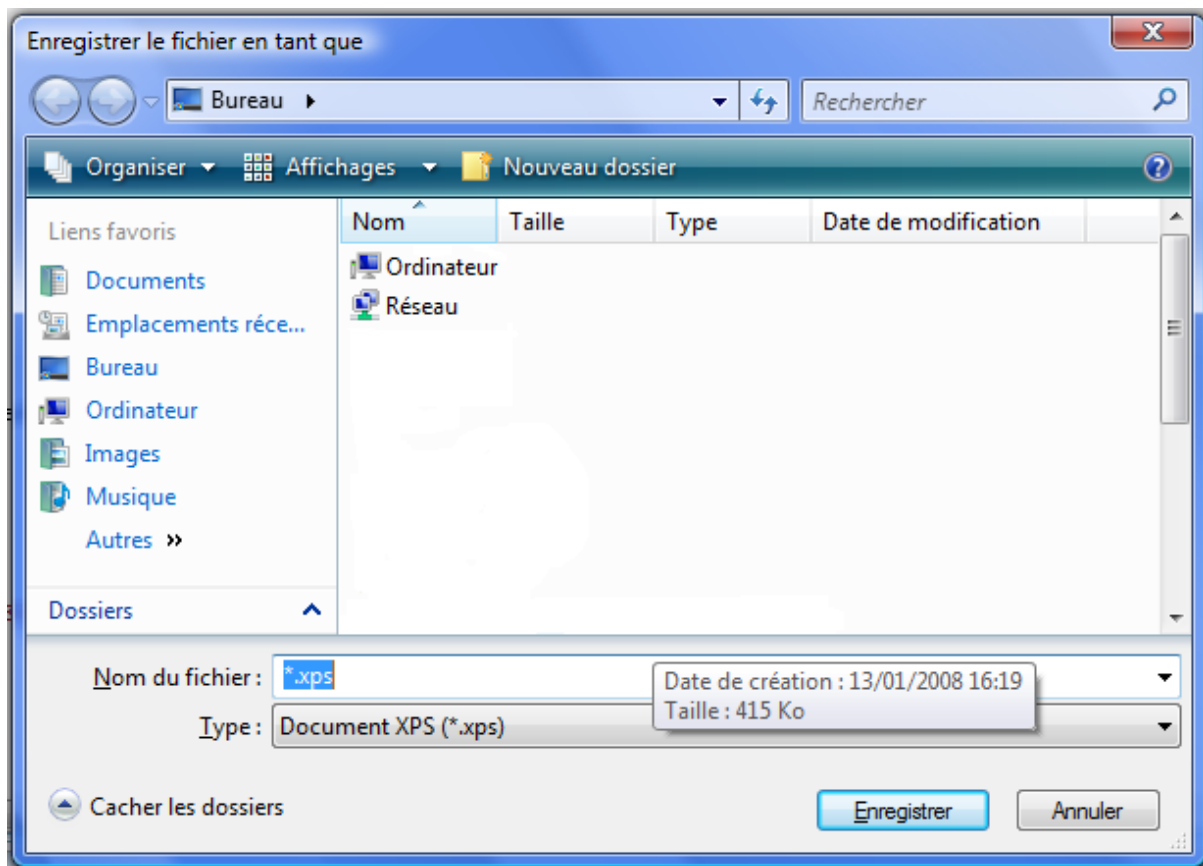
```
//C#

using System.Drawing.Printing;

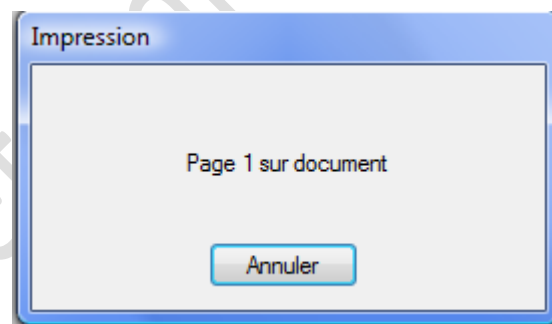
public Form1()
{
    InitializeComponent();
    PrintDocument monPrintDocument = new PrintDocument();
    monPrintDocument.PrintPage += new
PrintPageEventHandler((this.ImageAImprimer));
    monPrintDocument.Print();
}

private void ImageAImprimer(object sender, PrintPageEventArgs ev)
{
    ev.Graphics.DrawImage(Image.FromFile(@"C:\Users\Ketin\Desktop\DOTNET\Logo
DotNet\Logo.png"), ev.Graphics.RenderingOrigin);
}
```

Voici les étapes que vous rencontrerez lorsque vous aurez débuggé votre application :



Vous enregistrez votre image afin de l'imprimer



Une boîte de dialogue s'ouvrira ensuite pour confirmer l'impression



Un icône d'imprimante s'affichera dans la barre des tâches de votre ordinateur, il correspond à la file d'attente.



Voici un aperçu de l'impression de l'image que nous avons utilisé. Grâce à la propriété `ev.Graphics.RenderingOrigin`, l'image est imprimée selon sa taille d'origine.

5.2 Imprimer un texte

Tout simplement, la méthode `Graphics.DrawString` vous permettra d'imprimer des textes. Vous devrez spécifier la police, la taille, la couleur et les coordonnées de l'emplacement du texte.

6 Point sur la sécurité d'impression

La permission d'imprimer peut être gérée grâce à la classe `PrintingPermission`. Dans celle-ci se trouve l'énumération `PrintingPermissionLevel` qui assure les autorisations d'impression selon des niveaux de sécurité. Il existe quatre niveaux qui sont les suivants (du plus restreint au plus accessible):

Les Différents Niveaux	Description
NoPrinting	Ce niveau n'autorise aucun accès à l'imprimante, et donc aucune impression n'est possible. NoPrinting est un sous-ensemble de SafePrinting.
SafePrinting	Offre une impression selon un accès restreint via une boîte de dialogue. SafePrinting est un sous-ensemble de DefaultPrinting.
DefaultPrinting	Selon un accès semi-restreint via une boîte de dialogue, DefaultPrinting offre une impression sécurisée. C'est le niveau par défaut. DefaultPrinting est un sous-ensemble d'AllPrinting.
AllPrinting	Fournit un accès total, sans restriction, à toutes les imprimantes.

7 Aperçu avant impression avec le PrintPreviewControl

Le contrôle *PrintPreviewControl* vous permettra d'avoir un aperçu brut de votre impression. Cependant il ne possède aucun bouton ou boîte de dialogue.

Voici quelques propriétés du contrôle *PrintPreviewControl* :

Propriétés	Description
AutoZoom	Définit selon une valeur qui ajuste la propriété Zoom suivant un redimensionnement du contrôle ou à une modification du nombre de pages affichées.
Zoom	Définit le niveau du Zoom du document.
Columns	Indique le nombre de pages qui sont affichées horizontalement à l'écran.
Rows	Indique le nombre de pages qui sont affichées verticalement à l'écran.
Document	Définit le document associé au contrôle.
StartPage	Indique le numéro de la page qui commence la prévisualisation.
UseAntiAlias	Indique si l'anti-alias est utilisé. Celui-ci est employé pour un aperçu avant impression ayant un affichage plus précis mais cela engendre une baisse de la performance d'affichage.

Lorsque vous instanciez la classe *PrintPreviewControl*, certaines propriétés posséderont des valeurs initiales :

Propriétés	Valeur par défaut
AutoZoom	True
Document	Null
Column	1
Rows	0
StartPage	0

8 Conclusion

Vous avez ainsi terminé ce chapitre sur les Impressions en Windows Form et vous possédez tous les atouts nécessaires pour élaborer une impression grâce aux différents objets PrintPreviewDialog, PrintDocument, PrintPreviewControl, etc.

L'équipe *Windows Form*

Dotnet-France Association