

impresión de documentos

Los componentes para el lanzamiento de una impresión son **PrintDocument** y **PrintPreviewDialog**. En la medida en que es deseable para permitir al usuario ajustar los márgenes del documento y seleccionar una impresora, componentes **PageSetupDialog** y **PrintDialog** También son obligatorios. el componente

PrintPreviewControl sólo permite la visualización del documento designado por **PrintDocument** y este es el programador para proporcionar el comando de impresión. componentes **PrintDocument** y **PrintPreviewDialog** causa de eventos **PrintPage** en respuesta a lo que hay que programar la impresión por menor. En particular, en este procedimiento de evento que se realiza la lectura del archivo que contiene el documento, a menos que sea la impresión directa. Para utilizar las herramientas de reproducción de archivos, el **Imports System.IO** es esencial y el archivo para ser utilizado por varios procedimientos, deben ser declarados a nivel de clase. En esta situación, las líneas primer código de clase son:

Las importaciones System.IO

Clase Fbase Pública

Como StreamReader Dim miDoc

El motor de impresión: PrintDocument



el componente **PrintDocument** sigue siendo invisible en tiempo de ejecución. Impresión directa de un documento, sin posibilidad de que el usuario seleccione la impresora, o cambiar los márgenes, se controla con el código siguiente:

```
PrintDocument1.Print ()
```

La ejecución del método **impresión** provoca Evento **PrintPage** y la impresión se puede hacer así:

```
Private Sub PrintDocument1_PrintPage (remite de ByVal como objeto, ByVal e As
System.Drawing.Printing.PrintPageEventArgs) maneja PrintDocument1.PrintPage Línea As String Dim
```

Sun posición como System.Drawing.PointF ' PointF y RectangleF son la posición y

Sun rectángulo como RectangleF

'Las dimensiones de la zona de impresión

Como Sun Policia de fuente

'Se expresa como un número real (punto flotante)

Sun LineHeight As Single Line = "Esta es la línea que se

va a imprimir"

'La primera línea a imprimir

Policia = new Font ("Courier New", 10, FontStyle.BOLD) de posición = Nueva PointF (e.MarginBounds.Left, e.MarginBounds.Top) LineHeight = Police.GetHeight (e.Graphics) Rectangle.Location = rectángulo de posición. width = e.MarginBounds.Width ' Obtener la anchura entre las líneas

Rectangle.Height = LineHeight

'Y el uso de la altura de línea

e.Graphics.DrawString (Línea, Policia, Brushes.Black Rectángulo) Línea = "y aquí es otro"

' otra ...

Policia = Nueva Fuente ("Times New Roman", 16, FontStyle.BOLD)

Posición = Nueva PointF (e.MarginBounds.Left, e.MarginBounds.Top LineHeight +) = LineHeight Police.GetHeight (e.Graphics) Posición Rectangle.width Rectangle.Location = = e.MarginBounds.Width Rectangle.Height LineHeight

e.Graphics.DrawString (Línea, Policia, Brushes.Black Rectángulo)

' ... siempre y cuando hay líneas ...

'Sin embargo, ...

e.HasMorePages = False

End Sub

Impresión de un archivo de texto, sin posibilidad de que el usuario seleccione la impresora, o cambiar los márgenes, se controla con el código siguiente:

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

La ejecución del método **impresión** provoca Evento **PrintPage** y la impresión se puede llevar a cabo, pero esta vez las líneas que se imprimirán se proporcionan mediante la reproducción del archivo. El procedimiento **PrintDocument1_PrintPage** se convierte, por ejemplo:

```
Private Sub PrintDocument1_PrintPage (remitente de ByVal como objeto, ByVal e As
System.Drawing.Printing.PrintPageEventArgs) maneja PrintDocument1.PrintPage Línea As String Dim
```

```
Sun posición como System.Drawing.PointF sol rectángulo
como RectangleF Sun Sun Fuente Fuente Como LineHeight
As Single Dim NbrLignesParPage As Integer Dim ROWNUM
As Integer = 0
```

'Contando líneas impresas' En este ejemplo, todas las líneas utilizar el mismo tipo de letra

```
Policía = Nueva Fuente ( "Courier New", 10, FontStyle.BOLD) LineHeight = Police.GetHeight (e.Graphics) ' Número de líneas
```

por página es

'Altura del área imprimible' dividido por la altura de una fila

```
NbrLignesParPage = e.MarginBounds.Height / LineHeight Hacer hasta MonDoc.Peek = -1
```

```
MonDoc.ReadLine Línea = ()
```

'Leer el contenido del archivo línea por línea y el cálculo de la posición de la línea de impresión

```
Posición = Nueva PointF (e.MarginBounds.Left, e.MarginBounds.Top +
```

*(ROWNUM LineHeight *))*

'Definir las coordenadas de la zona' en el que se imprimirá 'la línea con la posición calculada y

```
Rectangle.Location = Posición
Rectangle.width = e.MarginBounds.Width
Rectangle.Height = LineHeight
```

*' tamaño
'Obtener la anchura entre las líneas
'Y el uso de la altura de la fila*

```
e.Graphics.DrawString (Línea, Policía, Brushes.Black, Rectángulo) ROWNUM + 1 =
```

'Incrementar el contador de línea

```
Si LineNum > NbrLignesParPage Entonces Salir Do ' y comprobar si hay al final de la página
bucle
```

```
Si MonDoc.Peek <> -1 Entonces
e.HasMorePages = True
otro
```

*'Repetir impresión Página
' siguiente, próximo*

```
MonDoc.Close ()
```

```
MyDoc = New StreamReader (TDocument.Text) ' En opinión, este procedimiento
```

```
End If
```

'Se ejecuta una vez para

```
End Sub
```

'Pantalla y es necesario abrir el' archivo para la posible impresión

Para dar al usuario la posibilidad de seleccionar la impresora y cambiar los márgenes, debe asociar los componentes **PrintDialog** y **PageSetupDialog** para **PrintDocument** y permitir al usuario utilizar para establecer sus opciones. Tenga en cuenta que **PageSetupDialog** también permite la designación de la impresora. Estas asociaciones se programan con estas líneas:

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

Diseño: PageSetupDialog



una vez que una **PrintDocument** y una **PageSetupDialog** asociado, la activación del programa de diálogo:

```
PageSetupDialog1.ShowDialog ()
```

Los propietarios **AllowMargins**, **AllowOrientation**, **AllowPaper** y **AllowPrinter** cuyo valor por defecto es **verdadero** puede recibir el valor **falso** para prohibir el acceso a las mismas características.

Selección de la impresora: PrintDialog



una vez que una **PrintDocument** y una **PrintDialog** asociado, la activación del programa de diálogo:

```
PrintDialog1.ShowDialog ()
```

Los propietarios **AllowSomePages**, **AllowSelection** y **AllowPrintToFile** también puede recibir valores **verdadero** o **falso** de acuerdo con el grado de libertad proporcionado al usuario.

La ventana de vista previa: PrintPreviewDialog



El diálogo de vista previa muestra el documento tal y como se imprimirá y permite al usuario controlar la impresión. el componente **PrintPreviewDialog** debe estar asociado con una

PrintDocument antes de ser activado. Se entiende que la **PrintDocument** debe ser resuelta de antemano. El siguiente código realiza esta asociación y el diálogo activo.

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

El componente previsualización: PrintPreviewControl

El componente de vista previa muestra el documento tal y como se imprimirá, pero no permite al usuario controlar la impresión. el componente **PrintPreviewControl** debe estar asociado con una

PrintDocument antes de ser activado. Se entiende que la **PrintDocument** debe ser resuelta de antemano. El código siguiente realiza la asociación y permite la visualización del documento en el componente.

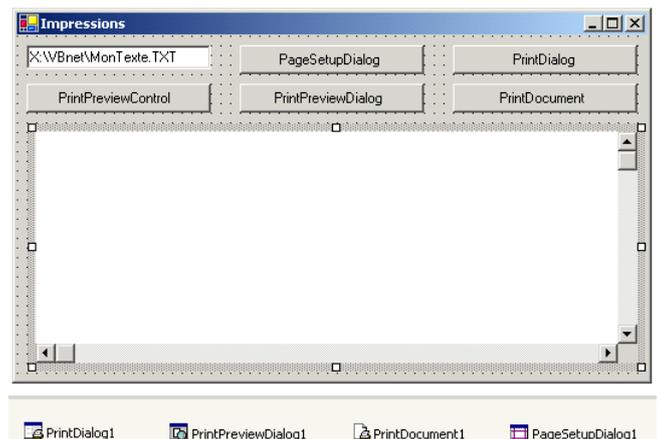
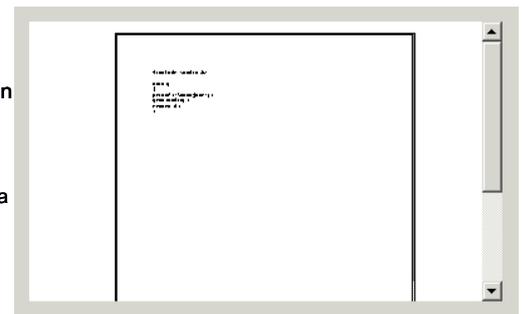
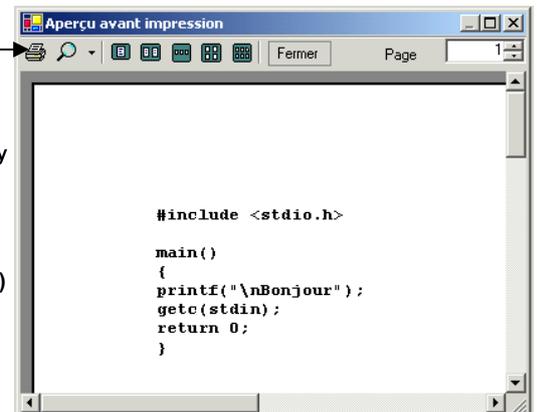
```
MyDoc = New StreamReader (PrintDocument1.DocumentName)
PrintPreviewControl1.Document PrintDocument1
PrintPreviewControl1.InvalidatePreview = ()
```

ejemplo completo

Todos estos componentes antes estudiados se combinan en una sola aplicación que permite realizar pruebas de cada tratamiento en una situación de un archivo de texto que permite la selección de la impresora y la modificación de los márgenes por el usuario.

ajuste de los componentes de códigos comunes se han combinado en un procedimiento **InitialiseComposants** la cual es invocado cada procedimiento de evento en respuesta al clic de un botón.

El cuadro de texto **TDocument** contiene la ruta y nombre de archivo de proceso.



Las importaciones System.IO

Clase Fbase Pública

Como StreamReader Dim miDoc

InitialiseComposants Sub privada (Prueba de ByVal como Byte)

Seleccionar caso de prueba

caso 1

PageSetupDialog1.Document = PrintDocument1

Asociación para PageSetupDialog

caso 2

PrintDialog1.Document = PrintDocument1

Asociación para PrintDialog

caso 3

PrintPreviewDialog1.Document = PrintDocument1 ' *Asociación PrintPreviewDialog*

caso 4

PrintPreviewControl1.Document = PrintDocument1 ' *Asociación PrintPreviewControl*

caso 0

'No hay nada que hacer aquí. Se espera que el parámetro 0 para entrar en el procedimiento, a pesar de que no tiene ninguna asociación para ser establecido (como en BPrintDocument_Click). Las 2 líneas escritas después End Select siempre deben llevarse a cabo.

End Select

PrintDocument1.DocumentName TDocument.Text miDoc = = New StreamReader

(PrintDocument1.DocumentName) End Sub

Private Sub (remite de ByVal como objeto, ByVal e As System.EventArgs) BPageSetup_Click

maneja BPageSetup.Click

InitialiseComposants (1)

PageSetupDialog1.ShowDialog () End Sub

Private Sub BPrintDial_Click (remite de ByVal como System.Object, ByVal e As

System.EventArgs) Maneja BPrintDial.Click

InitialiseComposants (2)

PrintDialog1.ShowDialog () End Sub

Private Sub BPrintPreviewDial_Click (remite de ByVal como System.Object, ByVal e As

System.EventArgs) Maneja BPrintPreviewDial.Click

InitialiseComposants (3)

PrintPreviewDialog1.ShowDialog () MonDoc.Close ()

End Sub

Private Sub BPrintPreviewControl_Click (remite de ByVal como System.Object, ByVal e As

System.EventArgs) Maneja BPrintPreviewControl.Click

InitialiseComposants (4)

PrintPreviewControl1.InvalidatePreview ()

'Forzar el control de la actualización

End Sub

Private Sub BPrintDocument_Click (remite de ByVal como System.Object, ByVal e As

System.EventArgs) Maneja BPrintDocument.Click

InitialiseComposants (0)

PrintDocument1.Print () MonDoc.Close ()

End Sub

Private Sub PrintDocument1_PrintPage (remite de ByVal como objeto, ByVal e As

System.Drawing.Printing.PrintPageEventArgs) Maneja PrintDocument1.PrintPage

'Sustituir por el procedimiento PrintDocument1_PrintPage ya escrito para imprimir un archivo de ' texto

End Sub End

Class