

Creación de archivos PDF con :



Manual de referencia

Introducción

WxPdfCOM es un componente generador de documentos PDF basado en wxWidgets. Se trata de un acrónimo. El prefijo wx significa que el componente está basado en la librería de código abierto wxWidgets. PDF son las conocidas siglas del formato de Adobe, Portable Document Format. Y por último COM, son las siglas de Microsoft que representan la tecnología del Modelo de Objetos de Componentes (Component Object Model).

Un objeto COM es realmente una librería de enlace dinámico, (DLL) con capacidad para exponer su interfaz de uso a otros programas. Una vez referenciado el componente en un proyecto de Visual Basic, el documento PDF se convierte en un objeto más del lenguaje, del mismo modo que los objetos Printer o App.

El objeto COM no es más que un envoltorio del verdadero motor generador de los documentos PDF, que no es otro que wxPdfDocument. wxPdfDocument es de código abierto, al igual que wxWidgets y a su vez es una traducción de la también libre FPDF.

Por si acaso tantos nombres introducen alguna confusión para el lector, en los siguientes apartados abordaremos detalladamente cada uno de ellos para que nos resulten más familiares.

FPDF

FPDF es una clase escrita en PHP que permite generar documentos PDF directamente, sin usar ninguna biblioteca externa. La F de FPDF significa Free (gratis y libre): se puede usar para cualquier propósito y modificarla al gusto para satisfacer las necesidades propias.

A continuación mostramos una lista de las principales características de FPDF:

- Elección de la unidad de medida, formato de página y márgenes
- Gestión de cabeceras y pies de página
- Salto de página automático
- Salto de línea y justificación del texto automáticos
- Admisión de imágenes (JPEG y PNG)
- Colores
- Enlaces
- Admisión de fuentes TrueType, Type1 y codificación
- Compresión de página

Puedes encontrar todo lo referente a FPDF en Internet (¿ como no ?) . La dirección es <http://fpdf.org>. Allí podrás encontrar el código fuente de la clase en cuestión, un manual con todos los comandos, ejemplos de uso y una sección de scripts creados por colaboradores que en algunos casos no sirve sino para extender aun más la capacidad inicial de la clase original.

Para usar FPDF solo necesitas un servidor de Internet con PHP instalado, tus páginas Web pueden contener código en PHP para generar los documentos PDF y grabarlos en un fichero o bien enviarlos a un navegador directamente. Como FPDF es libre, cualquiera puede tomar su código y modificarlo o copiarlo. Muchos ya lo han hecho. Gracias a esto existen versiones de FPDF en otros lenguajes, a saber: ASP, Lingo, Phyton, Ruby C++.

Por cierto, la versión en C++ se llama wxPdfDocument y pronto hablaremos de ella, pero antes hablaremos de wxWidgets.

wxWidgets

wxWidgets permite a los desarrolladores crear sus aplicaciones para OS X, Windows, Macintosh, GTK+ X11, Motif, WinCe y otros sistemas operativos, utilizando un mismo código base. Se puede utilizar desde lenguajes tales como C++, Python, Perl y C#.NET. Es extensible, gratis y libre, de código abierto y maduro. No se puede pedir más. Todo esto quiere decir que puedes tener versiones de tus programas para diferentes sistemas operativos con solo cambiar algunas opciones y volviendo a compilar. En el momento de crear este manual wxWidgets va por la versión 2.8.4 y existen multitud de programas comerciales que han sido creados sobre esta base.

La dirección en Internet es <http://www.wxwidgets.org/>. Allí, amigo lector, podrás profundizar en esta formidable herramienta que otros han decidido compartir. Como verás es todo un mundo, wxForo, wxWiki, wxComunidad, wxBlog, wxCode. Todo es wx y todo es gratis. Aunque siempre puedes hacer una donación o decirte a colaborar. A propósito de wxCode...

wxCode

Uno de los wxLugares que merece la pena conocer dentro del wxMundo de wxWidgets es este. <http://wxCode.com>. Allí encontrarás programas, utilidades y componentes para hacer tu vida de wxProgramador mucho más fácil y placentera. Eso sí, nada es compilado, todo va en código fuente. ¿ No es maravilloso ?. Por cierto que allí es donde puedes encontrar wxPdfDocument.

wxPdfDocument

Su autor se llama Ulrich Telle. wxPdfDocument es una librería basada en wxWidgets que permite generar documentos PDF. Se trata de una mas de las adaptaciones que se han realizado de FPDF, la clase original escrita en PHP. Algunos de los scripts en PHP que puedes encontrar en el sitio web de FPDF han sido incorporados a wxPdfDocument.

Además de las 14 fuentes standard de Adobe, es posible utilizar otras fuentes Type1 o TrueType que pueden se internas o externas al documento generado. En compilaciones Unicode es posible utilizar también fuentes CJK. Lo que significa que puedes generar documentos en idiomas con un juego de caracteres diferente al occidental. De hecho CJK

son las siglas de Chino, Japonés y Coreano (Korea en inglés).La propia documentación que acompaña a wxPdfDocument se señalan las siguientes características:

- Elección de la unidad de medida, formato de página y márgenes
- Definición de Cabecera y pie de página
- Salto automático de página
- Salto automático de línea y texto justificado
- Formatos de imagen soportados (GIF, JPEG, PNG y WMF)
- Colores (Escala de grises, RGB, CMYK, Spot colors)
- Enlaces (internos y externos)
- 14 fuentes Adobe Standard
- Fuentes TrueType y Type1 (incrustadas o sin incrustar), codificación
- Fuentes TrueType Unicode y Type0 (para Chino, Japonés y Coreano). Solo en compilaciones Unicode
- Compresión de página
- Primitivas gráficas para la creación de gráficos sencillos
- Máscaras
- Marcadores
- Rotación de texto e imagen
- Protección mediante claves y gestión de permisos
- Anotaciones
- Formularios (Acroforms). Campos tipo texto, combo box, check box, botones de radio, pulsadores)
- JavaScript
- Relleno con degradados
- Plantillas

COM

La tecnología COM de Microsoft(Component Object Model) permite la comunicación entre componentes de software. COM es utilizado por los desarrolladores para crear componentes de software reutilizable, enlazar componentes para construir aplicaciones y aprovechar los servicios de Windows. La familia de tecnologías COM incluye COM+, DCOM y ActiveX.

Ventajas e inconvenientes

wxPdfCOM es una librería que permite manejar wxPdfdocument desde lenguajes que soporten el modelo COM, como por ejemplo Visual Basic. Esto supone una serie de ventajas para el programador medio.

- No es necesario dominar C++ ni wxWidgets para sacar todo el provecho de wxPdfdocument.
- Tienes a tu disposición las mismas características y potencialidad de esa librería.

- Con respecto a PHP, no es necesario tener un servidor Web para poder ejecutar los scripts.
- Corre más rápido que PHP, y no presenta problemas de 'time out' en la creación de documentos extensos.
- Salvo leves diferencias, conserva la sintaxis original de FPDF, por lo que puedes seguir inspirándote o importar directamente los scripts de ejemplo que abundan en la Red.
- Al igual que los demás, es gratis.

En cuanto a las desventajas nos encontramos:

- Las funciones y procedimientos no admiten un número variable de argumentos, en cualquier llamada hay que pasarlos todos.
- En wxPdfCOM no existe el polimorfismo, los tipos de datos son rígidos. Esto impone una falta de expresividad que se hará patente para aquellos programadores acostumbrados PHP y C++.
- Aunque es una herramienta gratuita, no es de código abierto.

Configurando el entorno

Registrar el componente

Antes de empezar a utilizar el generador, es necesario registrar el componente en nuestra máquina y crear una referencia dentro del proyecto que lo va a invocar. Para ello debes abrir el menú proyecto y seleccionar la opción referencias. Aparecerá una ventana similar a la de la figura 1. En ella hay que marcar la referencia correspondiente a wxPdfCom tal como aparece en este ejemplo.

Es posible que tengas que utilizar el botón *Examinar* para buscar la librería DLL en el lugar donde haya sido desempaquetado el archivo de instalación del componente.

Una vez seleccionada la referencia pulsa *Aceptar* para regresar al entorno de Visual Basic. Todo estará listo para empezar a utilizar wxPdfCOM.

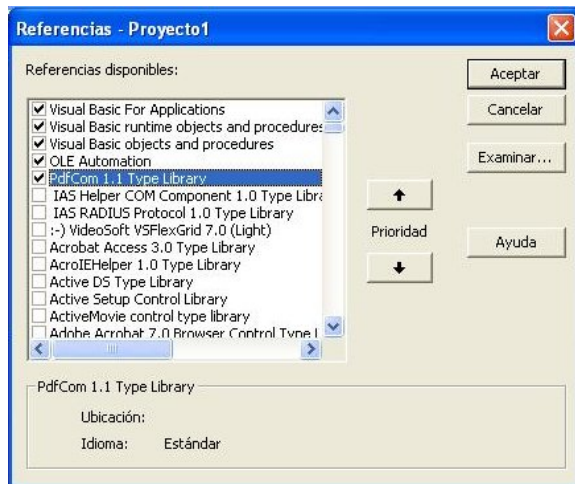
En algunas ocasiones es posible que al seleccionar la referencia y pulsar aceptar, se produzca un error.

Incluso puede producirse una salida inesperada del entorno de Vb6, con la consecuencia inmediata de que perdamos los últimos cambios que no hayan sido grabados.

En ese caso se puede intentar registrar manualmente el componente. Para ello hay que entrar en el menú *Inicio* de *Windows* y seleccionar la opción *Ejecutar*. En la ventana de ejecutar debemos invocar al programa regsvr32.exe y pasarlo como argumento el path completo de wxPdfCom. Por ejemplo suponiendo que hemos instalado la librería en c:/winnt/ pondremos:

regsvr32 c:/winnt/PDFCOM.DLL

¿ El sistema responde con un mensaje diciendo que el registro del archivo ha sido exitoso. ?
 ¡ Hurra ! En este momento, si todo ha ido bien, se supone que estas preparado para empezar a utilizar tu generador de archivos PDF favorito. Lo siguiente que podemos hacer es incluir el archivo de constantes predefinidas. Veamos qué es este archivo.



Archivo de constantes

El archivo de constantes es un modulo .bas, que en el proyecto de ejemplo lo encontramos con el nombre include.bas, una clara reminiscencia del lenguaje C. De hecho, los valores que en este archivo se definen son los mismos que utiliza wxPdfdocument para su funcionamiento interno. Dichas constantes nos servirán para dar mayor legibilidad a nuestro código fuente.

Por ejemplo, cuando queramos expresar que deseamos que un trozo de texto aparezca centrado, podemos poner el número 2, pero queda mucho mejor poner PDF_ALIGN_CENTER. Nadie te impide rebautizar estas constantes con otro nombre que nos guste más. Incluso en otros idiomas diferentes al Inglés.

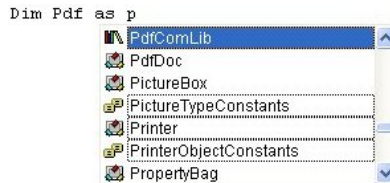
En los ejemplos de este manual axial como en los ejemplos incluidos en la librería y el sitio Web de wxPdfCOM, se conserva una nomenclatura similar a la original por varios motivos. En primer lugar por la comodidad de no tener que buscar otros nombres, en segundo lugar por deferencia a sus autores originales. Y por último por utilidad, son las constantes que utiliza la mayoría, (usuarios de FPDF, de wxPdfDocument y de wxPdfCOM).

Existen varias categorías de constantes que se agrupan del modo siguiente:

- Definición de marcos y bordes
- Alineaciones de texto (Izquierda, Derecha, Centro)
- Estilo de líneas
-

Declarando variables

Para poder crear nuestro primer documento PDF debemos declarar una variable del tipo adecuado. El entorno integrado nos ayudará mostrando una lista de los tipos disponibles nada más teclear dim, el nombre para nuestro objeto y un espacio.



Si hemos registrado correctamente el componente, en la lista desplegable que nos aparece, encontraremos la referencia que buscamos. Ver figura. Seleccionamos pdfcomlib.pdfdoc y nuestra línea de código quedará como la siguiente.

`Dim pdf as pdfcomlib.pdfcom`

Asignando memoria

Antes de utilizarla nos queda un último paso, asignar la memoria para el objeto con el operador new. Al igual que cuando declaramos la variable, nuestro ide nos mostrará una lista desplegable con las opciones disponibles para que no tengamos que teclear el nombre completo. La nueva línea de código quedará como la siguiente.

[Set pdf as new pdfdoc](#)

Comunicando con wxPdfCOM

Ya solo tenemos que empezar a utilizar el nuestra variable. Para ver una lista de los procedimientos y atributos del objeto, teclearemos el nombre de la variable y un punto. De nuevo obtendremos una lista desplegable con todos los atributos que componen la interfaz del objeto COM.



Manual del usuario

Consideraciones generales

Paso de argumentos

En wxPdfCOM todas las funciones tienen un número fijo de argumentos. Este inconveniente tiene algunas consecuencias que lo sitúan en desventaja con respecto a las versiones en otros lenguajes. Todas las llamadas a funciones deben pasar todos y cada uno de los argumentos que espera dicha función, aunque no se vayan a utilizar.

Afortunadamente, la mayoría de los procedimientos están preparados para asignar valores por defecto a los argumentos que se pasen en blanco o a cero. Pero incluso en ese caso, el argumento es obligatorio. En aquellos casos en que la función espera una cadena de caracteres y no necesites pasar una, deberás poner unas comillas dobles (""). En aquellos casos en que la función espera un valor numérico y no quieras suministrar uno, deberás escribir un cero (0).

Es de esperar que esta característica cambie en futuras versiones del componente con la introducción de argumentos opcionales.

Paso de arrays

El paso de arrays desde Visual Basic a wxPdfCOM requiere una atención especial. Una equivocación en la forma de declarar una array que ha de ser pasado como argumento a una función puede causar errores en el resultado, o incluso un error de automatización y un colapso del programa.

En Visual Basic existen varias posibilidades a la hora de declarar un array. Veamos diversas opciones.

Opción A- Declarar una variable sin tipo, (el compilador creará una variable del tipo variant), y poblarla posteriormente mediante la función Array(). Por ejemplo:

```
Dim ciudades  
ciudadesArray("Madrid", "Sevilla", "Valencia", "León")
```

Opción B - Declarar un array del tipo y tamaño específico y dar valor a cada uno de sus miembros individualmente:

```
Dim ciudades(4) as String
ciudades(0)"Madrid"
ciudades(1)"Sevilla"
ciudades(2)"Valencia"
ciudades(3)"León"
```

La representación en memoria de estos dos arrays es diferente en cada caso. Si quisiéramos crear un ComboBox con nombres de ciudades deberemos pasar a la función ComboBox el segundo tipo de array.

Todas las funciones que en su lista de argumentos contengan un array del tipo que sea, se pueden ver afectadas por esta peculiaridad. Se recomienda examinar detenidamente la documentación de esas funciones antes de utilizarlas o mirar en los ejemplos.

Si tienes problemas con alguna función de esta clase, puede que la solución esté en cambiar la declaración del array o la forma de poblarlo con datos. Estás avisado.

Ver : ComboBox(), ClippingPolygon().

Argumentos acumulables

Muchas de las funciones de wxPdfCOM tienen argumentos que denominaremos 'acumulables'. En esta sección explicaremos el porqué de estos argumentos y su mecanismo. En la referencia de los métodos que contengan uno o varios de sus argumentos acumulables se hará mención expresa de este hecho. Las opciones susceptibles de ser acumuladas irán acompañadas por el signo + para distinguirlas.

Los lenguajes C y C++ pueden trabajar a nivel de bits con facilidad. Con ellos se puede manipular una variable numérica y comprobar el estado de activación de cada uno sus bits. En numerosas ocasiones se utiliza una variable numérica para almacenar una serie de valores de estado, donde solo se almacena si el valor de un bit determinado es activado o desactivado, verdadero o falso. Estas variables de suelen denominar 'flags'.

El ahorro de memoria con el uso de flags es considerable pues en el caso de un entero que ocupa dos bytes, tenemos ocho variables en una. Veamos un caso real, el argumento 'border' del método Cell(). Este argumento nos permite especificar el tipo de marco que deseamos para la celda. Podemos seleccionar un borde para cada uno de los cuatro lados del rectángulo. En la siguiente tabla se muestran las constantes predefinidas para utilizar con el argumento 'border'.

PDF_BORDER_NONE 0
PDF_BORDER_LEFT 1

PDF_BORDER_RIGHT	2
PDF_BORDER_TOP	4
PDF_BORDER_BOTTOM	8
PDF_BORDER_FRAME	15

A continuación mostramos la misma tabla pero con los valores en binario:

PDF_BORDER_NONE	0000
PDF_BORDER_LEFT	0001
PDF_BORDER_RIGHT	0010
PDF_BORDER_TOP	0100
PDF_BORDER_BOTTOM	1000
PDF_BORDER_FRAME	1111

Como observarás, cada uno de los cuatro lados, tiene asignado un bit. El izquierdo tiene asignado el primer bit de la derecha, el lado derecho tiene asignado el segundo bit, el lado superior el tercero y el lado inferior tiene asignado el cuarto bit.

Para conseguir un marco completo utilizamos PDF_BORDER_FRAME, que tiene activados los cuatro bits (la representación binaria de 15 es 1111).

Si trabajamos en C, podemos utilizar el operador 'or' para combinar las opciones. Por ejemplo, la expresión:

```
(PDF_BORDER_LEFT | PDF_BORDER_RIGHT | PDF_BORDER_TOP | PDF_BORDER_BOTTOM)
```

es equivalente a

```
(PDF_BORDER_FRAME)
```

Para trabajar con wxPdfCOM en lugar de utilizar el operador 'or', tenemos que sumar los argumentos para obtener un efecto combinado de los mismos. Observa que : 1+2+4+8 suman 15. Ahora imagina que deseas imprimir con Cell() una celda con forma de U, es

decir, con todos los bordes activados menos el superior, el tercero de la derecha. Tendrás que acumular del siguiente modo :

PDF_BORDER_LEFT + PDF_BORDER_RIGHT + PDF_BORDER_BOTTOM

O lo que es igual a sumar los valores decimales, (1+2+8). El resultado es 11 que traducido a binario es 1011, donde el tercer bit está desactivado. Puedes utilizar la calculadora del sistema para hacer pruebas, en el modo 'calculadora científica' tienes botones de radio para efectuar conversiones entre diversas bases numéricas.

Definir el formato del documento

Definir un formato será lo primero que tengamos que hacer cada vez que tengamos que crear un nuevo documento PDF. No es lo mismo crear una carta, una novela o un calendario, cada tipo de documento se ajustará a unos estándares que en muchas ocasiones tendremos que seguir.

Al ser uno de los formatos más extendidos el A4, tamaño folio, es el formato por defecto en wxPdfCOM. Para crear un documento en formato A4, (210x297mm) no es necesario hacer nada especial, aparte de declarar una variable PDF y añadirle páginas. Solo en el caso de que necesitemos un formato diferente, necesitaremos utilizar la función Initialize(). En tal caso, ésta deberá ser la primera función a la que deberemos llamar. Para más información consulta la referencia de Initialize().

Propiedades del documento

Existe una serie de funciones que nos ayuda a documentar nuestro archivo pdf. Estas funciones son las que aparecen en Acrobat cuando consultamos las propiedades del documento. En la siguiente lista se enumeran las más importantes. En la referencia individual de cada uno de ellas encontrarás instrucciones más precisas para su utilización.

SetAuthor()	Define el autor del documento
SetCreator()	Establece el nombre de la aplicación que ha creado el documento
SetKeywords()	Define las palabras clave del documento
SetSubject()	Establece el asunto del documento
SetTitle()	Define el título del documento
SetDisplayMode()	Establece las opciones iniciales de visualización

Cabeceras y pies de página

Una de las pocas diferencias que existen entre wxPdfCOM y sus antecesores (FPDF y wxPdfDocument) estriba precisamente en el tratamiento de la cabecera y el pie de página.

Tanto el lenguaje C++ como PHP permiten la herencia de clases. En ambos casos, la clase original define la cabecera y pie de página como funciones virtuales. Las funciones virtuales no hacen absolutamente nada, tienen que ser definidas dentro de las instancias de la clase si se quiere utilizar su funcionalidad. Luego, durante la creación del documento PDF, las funciones definidas por el editor son llamadas automáticamente para cada nueva página que se inserte en el documento. Tanto la cabecera como el pie de página se definen una sola vez independientemente del número de páginas del documento.

wxPdfCOM no permite heredar ninguna función de la clase original aunque ello no es obstáculo para que podamos definir una cabecera o pie de página. Simplemente el procedimiento es diferente. La cabecera o pie de página se definen dentro de un bloque Head (o Foot). El bloque Head comienza con el comando Head(true) y termina con el comando Head(false). Todos los comandos que se invoquen dentro de este bloque, no se ejecutarán inmediatamente sino que se guardarán en memoria para ser utilizados después. No todos los comandos son aptos para ser incluidos dentro de una cabecera o pie de página, en la tabla siguiente se incluye una lista de los comandos aceptados. Los comandos no apropiados se ejecutarán directamente en lugar de ser almacenados.

Imprimir textos

WxPdfCOM cuenta con diversos métodos para imprimir texto, desde el sencillo Text() hasta el más complejo WriteXML. En este apartado vamos a enumerar y comparar los diferentes métodos. En la referencia individual de cada uno de ellos encontrarás instrucciones más precisas para su utilización. Los métodos más usuales para mostrar texto son :

Text()	Texto sencillo
RotatedText()	Texto con rotación
Write()	Texto sin justificar
Cell()	Texto en una sola línea
ClippedCell()	Texto recortado
MultiCell()	Texto multilinea
ImportText()	Tipo desde archivo
WriteCell()	Multilinea en celdas
TextBox()	Multilinea, alineación horizontal y vertical
WriteXML()	Texto con etiquetado XML

ClippingText()	Máscara de texto
ShapedText()	Texto con forma

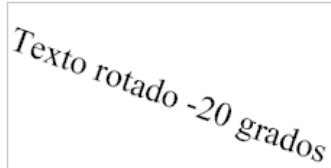
Text(x,y,txt)

Es tal vez el más sencillo de todos estos métodos. Toma unas coordenadas e imprime el texto que le pases como argumento. Si el texto es tan largo que no cabe en una sola línea se perderá por el lado derecho de la pantalla.

RotatedText(x,y,txt,angle)

Este comando es tan sencillo como Text(), pero tiene una característica especial, su capacidad de rotar el texto en un ángulo determinado que le suministramos como argumento adicional.

Los ángulos positivos provocan una rotación en el sentido contrario a las agujas del reloj y los ángulos negativos lo hacen en el mismo sentido que las agujas del reloj. El máximo valor del ángulo debería de ser 360 aunque no pasa nada si se especifica un número mayor.



Ejemplo de RotatedText()

Write(h,txt,link)

Otra de sus ventajas es que si el texto es demasiado largo, Write() salta al principio de la siguiente línea y continua escribiendo al alcanzar el margen derecho. Además, puedes convertir el texto en un hipervínculo que al ser pulsado llevará al usuario a la URL o enlace especificado en el argumento 'link'. Lo que Write no hace es justificar el texto, si la siguiente palabra no cabe en la línea, salta a la siguiente dejando un espacio entre la última palabra y el margen derecho.

Cell(w,h,txt,border,ln,align,fill,link)

Cell() es uno de los métodos más utilizados para mostrar texto. Cell() se basa en las coordenadas actuales para mostrar un texto. Opcionalmente se puede especificar el tipo de enmarcado, la alineación y el relleno. Puedes crear un hipervínculo y decidir en que posición quedará el puntero después de mostrar el texto. Si el texto es más largo que el espacio disponible, no se produce salto automático de línea y el texto puede desbordar el ancho de la celda. Cell() ser útil para construir tablas.

ClippedCell(w,h,txt,border,ln,align,fill,link)

Este comando tiene exactamente los mismos argumentos y funciona igual que Cell() salvo que no desborda los límites de la celda, el texto se recorta en caso de que su longitud supere el ancho disponible. ClippedCell tiene lo que podrá tratarse de un bug, el argumento w cuando es cero, hace que la celda ocupe el ancho de línea disponible, sin embargo el valor cero se aplica al recorte de modo que aparece una celda en blanco cuando w es igual a cero.

MultiCell(w,h,txt,border,align,fill,maxline)

El párrafo que estás leyendo en este momento está hecho con MultiCell(). Es otro de los métodos habituales, se asemeja mucho a Cell() pero tiene la ventaja de extenderse a varias líneas si es necesario al igual que hace Write(). Aunque a diferencia de Write(), MultiCell() si soporta la alineación.

ImportText (FileName)

ImportText es un caso particular de MultiCell(), parte de la posición actual, no utiliza borde ni relleno y muestra un texto justificado. Su particularidad es que el texto se lee desde un archivo de disco cuyo nombre es el único argumento de este método.

WriteCell(h,txt,border,fill,link)

En WriteCell, el ancho de celda se calcula automáticamente en función de la longitud del texto. Si supera el ancho disponible, se creará otra celda con su marco correspondiente en la línea siguiente y con la longitud precisa para acoger el texto restante.

TextBox(w,h,txt,halign,valign,border,fill)

Este es otro comando similar a MultiCell. Su característica más destacada es que admite alineación tanto vertical como horizontal.

WriteXML (txt)

Otro método de texto con un solo argumento, en este caso es una cadena de texto que puede contener etiquetas HTML que serán interpretadas acordemente. Este comando es multilinea.

 ... 	Negrita
<i> ... </i>	Itálica
<u> ... </u>	Subrayado
<o> ... </o>	Marcado
<s> ... </s>	Tachado

 ... 	Negrita
 ... 	Enfatizado, parecido a <i>
<small> ... </small>	Tamaño reducido
^{...}	Superíndice
_{...}	Subíndice
<h1> ... </h1>	Rótulos desde el uno al seis
<h6> ... </h6>	Texto con forma

Este es un ejemplo de WriteXML() para demostrar el efecto de las etiquetas **negrita**, *itálica*, subrayado, sobre marcado, tachado, small y subíndice

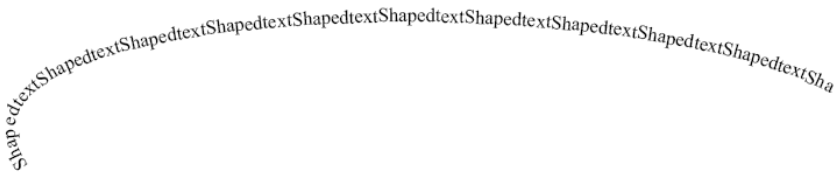
ClippingText(x,y,txt,outline)

Es una forma especial de texto que funciona como máscara. Viene acompañada del comando UnsetClipping(). Cualquier salida gráfica que se emita entre los dos comandos, solo será visible dentro de los límites del texto. El siguiente ejemplo ilustra este método.

ShapedText (shape,text,mode)



Este método permite realizar otro efecto especial. Consiste en definir una forma con comandos de forma (Shape) y hacer que el texto se dibuje siguiendo la trayectoria de la forma definida.



Gráficos

Uno de los puntos fuertes del formato PDF es su potencia gráfica. En WxPdfCOM permite explotar esta característica con numerosos métodos especializados en la salida gráfica.

Tanto usados aisladamente o en combinación entre si o con otros métodos de color o transformación los métodos gráficos te permitirán llevar a cabo gran cantidad de efectos. Lo siguiente es una lista de los métodos disponibles en wxPdfCOM. A continuación te mostramos su uso con algunos ejemplos.

Arrow	Dibuja una flecha
Clippingellipse	Máscara en forma de elipse
ClippingPath	Máscara de forma personalizada
ClippingPolygon	Máscara en forma de polígono
ClippingRect	Máscara rectangular
DrawCircle	Círculo
Ellipse	Elipse
Marker	Formas predefinidas
Polygon	Polígono
Rect	Rectángulo
RegularPolygon	Polígono regular
RoundedRect	Rectángulo de esquinas redondeadas
Sector	Sector
Shape	Forma personalizada
ShapeCurveTo	Definición de forma personalizada, curva de Bezier
ShapeMoveto	Definición de forma personalizada

Ejemplo de **Arrow()**



Código fuente :

```
x Pdf.GetX
y Pdf.GetY
Pdf.SetFillColor 0, 0, 0
Pdf.SetTextColor 0, 0, 0
Pdf.SetDrawColor 0, 0, 0
Pdf.Arrow x, y, db1 + 100, db2, 2, 5, 3
```

El siguiente ejemplo ilustra el uso de **Clippingellipse()**. La elipse se rellena con algo de texto desde un bucle. Finalmente se llama a **UnsetClipping()** para volver al modo normal.



Código fuente :

```
db1 Pdf.GetX + 50
db2 Pdf.GetY + 15
Pdf.Clippingellipse db1, db2, 10, 5, True
For x 0 To 200 Step 5
Pdf.RotatedText x, db2 + 10, "clipped text clipped text clipped text clipped text", 30
Next
Pdf.UnsetClipping
```

En el siguiente ejemplo llamamos a **Pdf.ClippingPath()** tras crear una forma con **pdf.ShapeMoveto()**, **ShapeLineTo()** y **ShapeCurveTo()**. La figura creada se rellena con texto como en el caso anterior. Finalmente volvemos al modo normal con **UnsetClipping()**



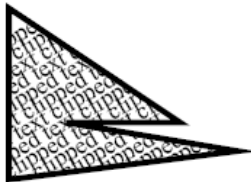
Código fuente :

```

db1 Pdf.GetX + 50
db2 Pdf.GetY + 15
Pdf.ShapeMoveTo 'cshape', db1, db2
Pdf.ShapeLineTo 'cshape', db1 + 20, db2 - 10
Pdf.ShapeLineTo 'cshape', db1 + 30, db2 + 10
Pdf.ShapeLineTo 'cshape', db1 + 40, db2 - 10
Pdf.ShapeLineTo 'cshape', db1 + 50, db2 + 15
Pdf.ShapeCurveTo 'cshape', db1, db2 + 20, db1, db2 + 20, db1, db2
Pdf.SetLineWidth 1
Pdf.SetFillColor 250, 0, 0
Pdf.ClippingPath 'cshape', PDF_STYLE_FILLDRAW
For x 0 To 200 Step 5
Pdf.RotatedText x, db2 + 20, 'clipped text clipped text clippe', 30
Next
Pdf.UnsetClipping

```

Ejemplo de **ClippingPolygon()**



Código fuente :

```

Dim y, z
y Array(70#, 100#, 80#, 110#, 70#)
z Array(175#, 195#, 195#, 200#, 205#)
Pdf.ClippingPolygon y, z, True
For x 0 To 200 Step 5
Pdf.RotatedText x, 205, 'clipped text clipped text clippe', 30
Next
Pdf.UnsetClipping

```

Ejemplo de **ClippingRect()**



Código fuente :

```
db1 Pdf.GetX
db2 Pdf.GetY + 20
Pdf.ClippingRect db1, db2, 60, 20, True
For x 0 To 500 Step 5
Pdf.RotatedText x, db2 + 20, "clipped text clipped text", 30
Next
Pdf.UnsetClipping
```

Ejemplos de **DrawCircle()**, (con y sin contorno) y **Ellipse()**



Código fuente :

```
db1 Pdf.GetX
db2 Pdf.GetY + 30
Pdf.DrawCircle db1, db2, 10, 0, 360, PDF_STYLE_FILLDRAW, 0
Pdf.DrawCircle db1 + 30, db2, 10, 0, 180, PDF_STYLE_FILL, 0
Pdf.Ellipse db1 + 60, db2, 10, 5, 20, 0, 360, PDF_STYLE_FILLDRAW, 0
```

Referencia de instrucciones

AcceptPageBreak

Cuando un salto de página es necesario, se produce una llamada a este método. El salto de página se produce siempre que el valor de retorno de AcceptPageBreak sea verdadero. Por defecto el valor devuelto depende del modo establecido por SetAutoPageBreak()

Esta llamada se produce de forma automática por lo cual AcceptPageBreak no tiene que ser llamado directamente por la aplicación. Sin embargo algunos scripts utilizan este método para cambiar los márgenes y conseguir textos en varias columnas.

* Ver SetAutoPageBreak()

AddFont

Importa una fuente TrueType, TrueTypeUnicode o Type1. El fichero de definición de la fuente debe de haber sido creado con carácter previo y estar disponible en un subdirectorio llamado 'fonts'. SetFontPath puede ser invocado para establecer una carpeta diferente. El archivo de la fuente debe de estar también esta carpeta siempre que se desee incrustar en el propio archivo PDF.

Ver :SetFont(), SetFontPath()

Argumentos

Family	Indica la familia a la que pertenece la fuente. Se puede adoptar cualquier nombre pero si coincide con un nombre estandar existente lo sustituirá
Style	Estilo de la fuente. Valores posibles en tabla siguiente (case insensitive)
File	Archivo de definición de la fuente. Se puede pasar una cadena vacía en cuyo caso el nombre se construye a partir de los valores family y style

Valores para style

(Vacío)	Normal
B	Negrita
I	Itálica
BI o IB	Negrita e Itálica

AddFontCJK

Importa una fuente CJK (Chino, Japonés o (K) coreana). El fichero de definición de la fuente debe de haber sido creado con carácter previo y estar disponible en el subdirectorio actual.

SetFontPath puede ser invocado para establecer una carpeta diferente. El archivo de la fuente debe de estar también esta carpeta siempre que se desee incrustar en el propio archivo PDF. Esas fuentes requieren la instalación del soporte CJK para Adobe.

Ver: SetFont()

Family	Indica la familia a la que pertenece la fuente. Se puede adoptar cualquier nombre pero si coincide con un nombre estándar existente lo sustituirá. No requiere especificar estilo ya que se instalan todos los disponibles El nombre del archivo de definición se construye automáticamente con el valor 'Family' en minúsculas y sin espacios
--------	--

Hay varios ficheros de definición de fuente predefinidos

BIG5	Chino tradicional
BIG5-HW	Chino (tradicional) Caracteres ASCII mitad de anchura
GB	Chino (simplificado)
GB-HW	Chino (simplificado) Caracteres ASCII mitad de anchura
SJIS	Japonés
SJIS-HW	Japonés, caracteres ASCII mitad de anchura
UHC	Coreano
UHC-HW	Coreano, caracteres ASCII mitad de anchura

AddLink

Crea un enlace interno. Un enlace interno es un área donde se puede hacer clic y saltar a otra parte del documento. Devuelve un identificador válido para pasar como argumento a Cell(), Write(), Image() o Link(). El punto de destino del salto se define con SetLink()

Ver Cell(), Write(), Image(), Link(), SetLink().

AddPage

Añade una página al documento. Si existe una página anterior, llama a Footer() antes de crear la nueva página y a Header() para la nueva. Al crear una nueva página, la posición se fija en la esquina superior izquierda de acuerdo con los márgenes establecidos. Los valores previamente establecidos, si es que hay alguno, para tipo de letra, colores y ancho de línea, son conservados en la nueva página. El origen del sistema de coordenadas está arriba y a la izquierda y estas se incrementan hacia abajo y hacia la derecha.

Ver header(), Footer(), SetMargins()

Orientation		Indica si la nueva página va a ser vertical o apaisada. Puede adoptar uno de los siguientes valores.
	1	Orientación vertical. También se puede utilizar la constante predefinida 'PORTRAIT'
	2	Página apaisada. La constante predefinida es 'LANDSCAPE'

AliasNbPages

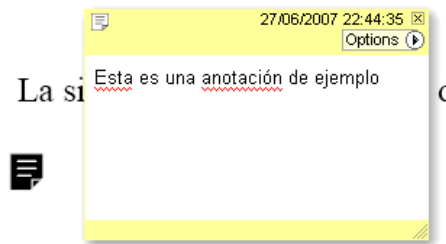
Define un alias para el número total de páginas. Este se sustituirá al cerrar el documento. En FPDF y wxPdfDocument el valor por defecto para esta función es 'nb' entre llaves {}, En wxPdfCPM se recomienda su uso por compatibilidad. Existe un alias interno para el número de página actual, es de uso restringido, su valor {pg} y no existe una función para cambiarlo.

Ver PageNo(), Footer()

alias	Cadena de caracteres que se sustituirá por el total de páginas al cierre del documento.
-------	---

Annotate

Inserta una anotación de texto en la posición indicada por las coordenadas x e y. La siguiente es una anotación de texto insertada con Annotate :



x	Abscisa de la anotación
---	-------------------------

y	Ordenada de la anotación
text	Texto de la anotación

AppendJavaScript

Añade una pieza de código Javascript a un objeto Javascript a nivel de documento.

javascript	Cadena de texto que contiene el código JavaScript.
------------	--

Arroz

Dibuja una flecha entre dos puntos.

Ver : SetLineWidth(), SetDrawColor(), SetFillColor()

x1	Abscisa del primer punto
y1	Ordenada del primer punto
x2	Abscisa del segundo punto
y2	Ordenada del segundo punto
linewidth	Anchura de la flecha
height	Altura de la punta de la flecha
height	Anchura de la punta de la flecha

AxialGradient

Define un sombreado con degradación axial. Devuelve un identificador entero.

col1	Primer color (RGB o CMYK)
col2	Segundo color (RGB o CMYK)
x1	Abscisa del punto de inicio del vector degradado. (rango 0..1)
y1	Ordenada del punto de inicio del vector degradado. (rango 0..1)

x2	Abscisa del segundo punto del vector degradado. (rango 0..1)
y2	Ordenada del segundo punto del vector degradado. (rango 0..1)
intexp	Exponente de interpolación

Bookmark

Añade una entrada nueva en el índice del documento.

txt	Título de la entrada
level	Nivel, 0 es el nivel más alto. 1 y siguientes representan los sub-epígrafes
y	Posición (ordenada) dentro de la página actual. -1 indica la posición actual

Cell

Imprime una celda (área rectangular), permite especificar el enmarcado, el color de fondo y un texto. El origen de la celda es la posición actual del puntero de escritura. El texto se puede alinear a uno de los lados o estar en el centro. Después de la llamada a Cell(), la posición se mueve justo a continuación o saltar a la línea siguiente. Cell permite que el texto funcione como un enlace. Si está habilitado el salto de página automático y Cell va más allá de límite se produce un salto de página previo.

Ver: SetFont(), SetDrawColor(), SetFillColor(), SetTextColor(), SetLineWidth(), AddLink(), Ln(),MultiCell(), Write(), SetAutoPageBreak().

w	Anchura de la celda. Cero significa hasta el final de línea
h	Altura de la celda. Si es cero toma el valor por defecto
border	Define el tipo de enmarcado de la celda según la tabla Border
ln	Sirve para indicar la posición del puntero de escritura después de Cell 0: A continuación, 1: Al inicio de la línea siguiente 2: Debajo
align	Establece el alineado para el texto según la tabla align
fill	Relleno

link	URL o identificador devuelto por Addlink()
------	--

Posibles valores para borde

BORDER_NONE	Sin borde
BORDER_LEFT	Borde a la izquierda
BORDER_RIGHT	Borde a la derecha
BORDER_TOP	Borde superior
BORDER_BOTTOM	Borde inferior
BORDER_FRAME	Borde en los cuatro lados

Al ser constantes numéricas se pueden sumar para combinar su efecto

Posibles valores para alineación

ALIGN_LEFT	Alineado a la izquierda
ALIGN_CENTER	Texto centrado
ALIGN_RIGHT	Alineado a la derecha
ALIGN_JUSTIFY	Texto justificado

CheckBox

Añade un campo del tipo 'check box' a la lista de campos del formulario.

name	Nombre asignado al campo
x	Abscisa de la posición
y	Ordenada de la posición
width	Anchura del campo
checked	Valor por defecto

ClippedCell

Esta función es la misma que Cell() salvo que la celda se recorta al superar los límites del rectángulo. Imprime una celda (área rectangular) , permite especificar el enmarcado, el color de fondo y un texto. El origen de la celda es la posición actual del puntero de

escritura. El texto se puede alinear a uno de los lados o estar en el centro. Después de la llamada a Cell(), la posición se mueve justo a continuación o saltar a la línea siguiente. Cell permite que el texto funcione como un enlace. Si está habilitado el salto de página automático y Cell va más allá de límite se produce un salto de página previo.

Ver: SetFont(), SetDrawColor(), SetFillColor(), SetTextColor(), SetLineWidth(), AddLink(), Ln(), MultiCell(), Write(), SetAutoPageBreak().

w	Anchura de la celda. Cero significa hasta el final de línea
h	Altura de la celda. Si es cero toma el valor por defecto
border	Define el tipo de enmarcado de la celda según la tabla Border In Sirve para indicar la posición del puntero de escritura después de Cell 0: A continuación, 1: Al inicio de la línea siguiente 2: Debajo
align	Establece el alineado para el texto según la tabla 'alineación'
fill	Relleno
link	URL o identificador devuelto por Addlink()

Posibles valores para borde

BORDER_NONE	Sin borde
BORDER_LEFT	Borde a la izquierda
BORDER_RIGHT	Borde a la derecha
BORDER_TOP	Borde superior
BORDER_BOTTOM	Borde inferior
BORDER_FRAME	Borde en los cuatro lados

Al ser constantes numéricas se pueden sumar para combinar su efecto

Posibles valores para alineación

ALIGN_LEFT	Alineado a la izquierda
ALIGN_CENTER	Texto centrado
ALIGN_RIGHT	Alineado a la derecha
ALIGN_JUSTIFY	Texto justificado

ClippingEllipse

Define un área de enmarcado con la forma de una elipse.

x	Abscisa de la posición
y	Ordenada de la posición
rx	Radio horizontal
ry	Radio vertical, si es cero el resultado será un círculo
outline	Determina si se va a dibujar el perfil

ClippingPath

Define una máscara con la forma de un objeto shape. Una máscara restringe la salida impidiendo que se muestre ningún elemento fuera del área definida por ella.

shape	Nombre del objeto Shape que define el área de máscara
style	Estilo de dibujo, puede ser uno de los siguientes:
PDF_STYLE_NOOP	Sin estilo, valor por defecto.
PDF_STYLE_DRAW	Dibuja contorno
PDF_STYLE_FILL	Relleno

PDF_STYLE_FILLDRAW	Dibuja contorno y relleno
--------------------	---------------------------

ClippingPolygon

Define un polígono como área de enmarcado. Un área de enmarcado restringe la salida e impide que los objetos se muestren por fuera de la misma.

x	Array de abscisas
y	Array de ordenadas
outline	Valor booleano de definición del contorno.

ClippingRect

Define un rectángulo como área de enmarcado. Un área de enmarcado restringe la salida e impide que los objetos se muestren por fuera de la misma.

x	Abscisa de la esquina superior izquierda
y	Ordenada de la esquina superior izquierda
w	Anchura del rectángulo
h	Altura del rectángulo
outline	Valor booleano que define si se dibuja el contorno

ClippingText

Define un texto como área de enmarcado. Un área de enmarcado restringe la salida e impide que los objetos se muestren por fuera de la misma.

x	Abscisa de la esquina superior izquierda
y	Ordenada de la esquina superior izquierda
txt	Texto
outline	Valor booleano que define si se dibuja el contorno

CloseFile

Finaliza la edición del documento PDF. No es necesario llamar a este método explícitamente ya que SaveFile() lo hace de forma automática. Si no existe ninguna página en el documento, Close() llama a AddPage() para evitar la creación de un documento inválido.

Ver :Open(), SaveAsFile()

ClosePath

Cierra el path de enmarcado. Un área de enmarcado restringe la salida impidiendo que algún elemento se dibuje fuera del área definida.

style	Estilo
-------	--------

Están disponibles las siguientes opciones de estilo

PDF_STYLE_NOOP	ninguno
PDF_STYLE_DRAW	Dibujo del contorno
PDF_STYLE_FILL	Relleno
PDF_STYLE_FILLDRAW	Relleno y contorno

ComboBox

Añade un campo de tipo combo en las coordenadas especificadas o en la posición actual. La lista de valores será un array de strings donde cada elemento se poblará independientemente. Utilizar la función array() de Visual Basic para asignar elementos a un variant provocaría un error si intentamos pasarlo a ComboBox.

name	Nombre del campo
x	Abscisa, si se deja a cero toma el valor de GetX()
y	Ordenada, si se deja a cero toma el valor de GetY()
width	Ancho
height	Alto
values	Array de cadenas para el contenido

Convert2Roman

Toma como argumento un número entero y devuelve una cadena de caracteres con su representación en números romanos. Esta función puede ser útil para numerar páginas o capítulos de un documento.

value	Valor entero que se quiere convertir
-------	--------------------------------------

Curve

Dibuja una curva de Bezier. Una curva de Bezier es tangente a la línea que une los puntos de control a cada lado de la curva.

x0	Abscisa del punto inicial
y0	Ordenada del punto inicial
x1	Abscisa del punto de control primero
y1	Ordenada del punto de control primero
x2	Abscisa del punto de control segundo
y2	Ordenada del punto de control segundo
x3	Abscisa del punto final
y3	Ordenada del punto final
style	Estilo de la curva. Contorno/Relleno
PDF_STYLE_DRAW	Contorno
PDF_STYLE_FILL	Relleno

CurveTo

Añade una curva de Bezier cúbica al path o subpath actual. La curva se extiende desde el punto actual hasta el punto (x3,y3), usando los puntos de control 1 y 2. Establece (x3,y3) como nuevo punto actual.

Ver Curve()

x1	Abscisa del punto de control primero
y1	Ordenada del punto de control primero
x2	Abscisa del punto de control segundo
y2	Ordenada del punto de control segundo
x3	Abscisa del punto final
y3	Ordenada del punto final

DrawCircle

Dibuja un círculo dadas las coordenadas de su centro, el radio y los ángulos de inicio y fin. Los valores de ángulo deben estar comprendidos entre 0 y 360. Un ángulo final 0 se interpreta como 360.

x0	Abscisa de la posición
y0	Ordenada de la posición
r	Radio del círculo
astart	Angulo inicial
afinish	Angulo final
style	Estilo dibujo y/o relleno
nseg	El círculo se compone de hasta nSeg curvas de Bezier

Ellipse

Dibuja una elipse

x0	Abscisa del centro
y0	Ordenada del centro
rx	Radio horizontal

ry	Radio vertical. Si ry es cero, el resultado será un círculo
angle	Angulo de rotación. Contrario a las agujas del reloj
astart	Angulo de salida
afinish	Angulo final
style	Estilo (Contorno/Relleno)
nSeg	La elipse se compone de nSeg curvas de Bezier

EndTemplate

Termina una plantilla. Restaura las variables inicializadas con BeginTemplate(). Si existe una plantilla en construcción, devuelve su identificador. En caso contrario devuelve cero.

Ver : BeginTemplate(), UseTemplate().

Footer

Inicia o termina la definición del pie de página. Toma como argumento un valor lógico (true o false) que define el comienzo y el fin del bloque de definición del pie de página. No todos los comandos son aptos para ser incluidos en el bloque. Es necesario cerrar un bloque que ha sido abierto.

define	Valor de comienzo/fin de la definición del bloque pie de página
True :	Comienza la grabación del bloque
False :	Termina la grabación del bloque

GetBreakMargin

Devuelve el valor del margen para el salto de página.

GetCellMargin

Devuelve el margen de Cell().

GetFontFamily

Devuelve el nombre de familia de la fuente actual.

Ver : SetFont()

GetFontPath

Devuelve el path de la carpeta donde se buscarán los ficheros de definición de las fuentes. Por defecto su nombre es /fonts pero se puede cambiar con SetFontPath().

GetFontSize

Devuelve el tamaño en puntos de la fuente actual

Ver : SetFont()

GetFontStyle

Devuelve el estilo de la fuente actual.

Ver : SetFont()

GetFontSubsetting

Devuelve verdadero o falso, dependiendo de si está activada la inclusión parcial de fuentes Unicode. Si utilizamos alguna fuente diferente de las 14 fuentes básicas de Adobe, deberemos incluir su definición en el archivo PDF. Incrustar fuentes dentro de los documentos hace que estos aumenten su tamaño, especialmente al utilizar fuentes Unicode, que contienen miles de caracteres. Para reducir el tamaño de estos archivos se pueden incluir sub-conjuntos (subsets) de los caracteres que se vayan a utilizar. wxPdfDocument y en consecuencia, wxPdfCOM, admiten la inclusión parcial de fuentes para fuentes Unicode.

GetImageScale

Devuelve la escala de la imagen.

GetLeftMargin

Devuelve el margen izquierdo.

Ver : SetLeftMargin().

GetLineHeight

Devuelve la altura de una línea de texto.

GetLineWidth

Devuelve el ancho de línea actual.

Ver Line(), Rect(), Cell(), MultiCell()

GetPageHeight

Devuelve la altura de página actual en unidades.

GetPageWidth

Devuelve el ancho actual de página.

GetScaleFactor

Devuelve el factor de escalado (el número de puntos por unidad de usuario).

GetStringWidth

Devuelve la longitud de una cadena de caracteres en la unidad del usuario. Tiene en cuenta el tipo de letra actual, su estilo y su tamaño. Debe haber una fuente seleccionada.

s	Cadena de caracteres cuya longitud devolverá la función
---	---

GetTemplateBBox

Obtiene las dimensiones de un modelo externo Las dimensiones de las páginas importadas desde documentos PDF externos suelen ser de interés. Esta función toma parte de sus argumentos por referencia y deposita en ellos el resultado. Los valores devueltos se corresponden con las coordenadas de punto inferior izquierdo, el ancho y el alto del modelo o plantilla.

Ver: SetTemplateBBox(), BeginTemplate(), ImportPage().

templateId	Es un identificador de plantilla válido
x	Abscisa de la esquina inferior izquierda
y	Ordenada de la esquina inferior izquierda
width	Anchura del modelo
height	Altura del modelo

GetTemplateSize

Obtiene el tamaño calculado de un modelo o plantilla. El ancho y/o el alto deben tener un valor menor o igual que cero para ser calculados. Si uno de estos valores se pasa con valor cero ,será devuelto con el valor calculado en proporción al otro elemento.

Ver :BeginTemplate(), EndTemplate(), UseTemplate(), ImportPage()

templateId	Es un identificador de plantilla válido
width	Anchura del modelo
height	Altura del modelo

GetTopMargin

Devuelve el margen superior.

GetX

Devuelve la abscisa de la posición actual.

Ver SetY(), GetY(), SetX()

GetY

Devuelve la ordenada de la posición actual.

Ver SetY(), GetX(), SetX()

HEADER

Inicia o termina la definición de la cabecera de página. Toma como argumento un valor lógico (true o false) que define el comienzo y el fin del bloque de definición de la cabecera. No todos los comandos son aptos para ser incluidos en el bloque. Es necesario cerrar un bloque que ha sido abierto.

define	Valor de comienzo/fin de la definición del bloque de cabecera de página
True :	Comienza la grabación del bloque
False :	Termina la grabación del bloque

Image

Imprime una imagen en la posición y con las dimensiones especificadas. Opcionalmente puede incluir un enlace o se puede mostrar delimitada mediante una máscara. Las coordenadas de la esquina superior izquierda deben ser especificadas.

En cuanto a las dimensiones pueden darse expresamente o hacer que wxPdfCOM las calcule. Si se pasa solo una, la otra es calculada proporcionalmente. Si se pasan ancho y alto igual a cero, la imagen se mostrará a 72 puntos por pulgada. Los formatos reconocidos son JPEG, PNG, GIF y WMF. Para JPEG se admiten los siguientes tipos: Escala de grises, color verdadero (24 bits), y CMYK (32 bits). Para PNG se admiten los formatos siguientes : Escala de grises de hasta 8 bits (256 niveles), color indexado, color verdadero (24 bits). Sin embargo no soporta el entrelazado ni el canal alfa. Si se define un color transparente será tenido en cuenta pero solo será interpretado por Acrobat 4 y superiores.

Los formatos GIF soportados dependen del decodificador de wxWidgets. wxPdfCOM está compilado con la última versión disponible a la hora de confeccionar este manual (la 2.8.3). Los archivos WMF contienen imágenes vectoriales descritas mediante comandos del GDI de Windows (Graphics Device Interface). Existen unos 80 comandos diferentes en el WMF estándar. Este método solamente soporta algunos de ellos, aunque son suficientes para la mayoría de las imágenes WMF existentes. Con la finalidad de optimizar el tamaño de los documentos creados, las imágenes incluidas varias veces solo se almacenan una vez. Esto supone un gran ahorro cuando la misma imagen se utiliza en repetidas ocasiones.

file	Nombre del fichero que contiene la imagen
x	Abscisa de la posición superior izquierda donde se mostrará la imagen
y	Ordenada de la esquina superior izquierda de la imagen
w	Anchura de la imagen mostrada o cero para cálculo automático
h	altura de la imagen mostrada o cero para cálculo automático
link	Dirección del enlace
maskimage	Identificador de máscara creado previamente con ImageMask()

ImageMask

Prepara una imagen para su uso como máscara. Devuelve un identificador de máscara válido o cero si se produce algún error.

file	Nombre del archivo que contiene la imagen
------	---

ImportPage

Importa una página del documento PDF externo. El contorno utilizado será el ArtBox de la página importada, si es que se ha definido. En su defecto se utilizara el CropBox y si no el MediaBox. El valor de retorno es un índice de la página importada que podrá utilizarse con UseTemplate(). Si la página especificada está fuera de rango o no se ha seleccionado la fuente ImportPage() devolverá cero.

Ver : SetSourceFile(), UseTemplate().

pageno	Número de página para importar
--------	--------------------------------

ImportText

Lee un fichero de texto y lo imprime en el documento actual. Respeta los saltos de línea manuales e introduce automáticamente los necesarios si una línea es demasiado larga. El texto aparece justificado.

filename	Nombre del archivo de texto para importar
----------	---

Initialize

Define valores iniciales del documento. No es imprescindible llamar a este método, pero si se hace deberá ser antes que a ningún otro. Sirve para determinar la orientación por defecto que tendrán las páginas, la unidad de medida y el tipo de papel. En realidad esta función no debería tener el primero de sus argumentos puesto que la orientación de deberá indicar para cada página que se añada. Esta redundancia debida a que wxPdfCOM no admite argumentos vacíos es posible que desaparezca en futuras versiones donde previsiblemente se introduzcan los argumentos por defecto.

Los tipos de papel soportados son los mismos que wxWidgets. Actualmente hay 116 tipos diferentes entre los que elegir el que más se acerque a nuestras necesidades sin que sea posible crear tamaños diferentes a estos. Si se omite la llamada a Initialize, el documento tendrá un tamaño A4.

orientation	Tipo de orientación por defecto
unit	Unidad de medida
paper	Tipo de papel

Lin

Dibuja una línea entre dos puntos. Lin tiene en cuenta los valores establecidos para anchura de línea y color. No confundir con Ln() que sirva para saltar líneas.

Ver: SetLineWidth(), SetDrawColor()

x1	Abscisa del punto de origen
y1	Ordenada del punto de origen
x2	Abscisa del punto de destino
y2	Ordenada del punto de destino

LineCount

Calcula la cantidad de líneas que ocuparía un texto teniendo en cuenta el ancho disponible y el tamaño de la fuente actual.

w	Ancho disponible
text	Texto para el cual se desea hacer el cálculo

Ln

Provoca saltos de línea de altura variable.

La nueva posición se fija en el margen izquierdo. La altura del salto depende del argumento, por defecto se toma la altura de la última celda impresa.

Ver : Cell();

h	Altura del salto, si es cero toma el alto de la última celda
---	--

Marker

Dibuja una marca.

x	Abscisa del centro de la marca
y	Ordenada del centro de la marca
markerType	Uno de los tipos de la tabla siguiente
size	Tamaño de la marca

Lista de tipos para Marker()

PDF_MARKER_CIRCLE

PDF_MARKER_SQUARE
PDF_MARKER_TRIANGLE_UP
PDF_MARKER_TRIANGLE_DOWN
PDF_MARKER_TRIANGLE_LEFT
PDF_MARKER_TRIANGLE_RIGHT
PDF_MARKER_MARKER_DIAMOND
PDF_MARKER_PENTAGON_UP
PDF_MARKER_PENTAGON_DOWN
PDF_MARKER_PENTAGON_LEFT
PDF_MARKER_PENTAGON_RIGHT
PDF_MARKER_STAR
PDF_MARKER_STAR4
PDF_MARKER_PLUS
PDF_MARKER_CROSS
PDF_MARKER_SUN
PDF_MARKER_BOWTIE_HORIZONTAL
PDF_MARKER_BOWTIE_VERTICAL
PDF_MARKER_MARKER_ASTERISK

MirrorH

Realiza una transformación de reflejo horizontal. Esta función equivale a escalar en -100 en la dirección de x. Para regresar a un escalado normal hay que pasarle el valor 100.

x	Abscisa del eje de reflexión
---	------------------------------

MirrorV

Realiza una transformación de reflejo vertical. Esta función equivale a escalar a -100 la dirección y. Para regresar a un escalado normal hay que pasarle el valor 100.

y	Abscisa del eje de reflexión
---	------------------------------

MoveTo

Hace saltar el puntero de dibujo a una nueva posición. Esta debe ser la primera operación después de ClippingPath().

x	Abscisa de la nueva posición
y	Ordenada de la nueva posición

MultiCell

Este método permite imprimir texto con saltos de línea. Estos se insertarán automáticamente tan pronto el texto alcance el borde derecho de la celda, o manualmente. Se dibujarán las celdas que sean necesarias, una debajo de otra. El texto puede aparecer alineado justificado o centrado. Se puede especificar borde y un relleno.

w	Anchura de las celdas. Cero significa hasta el final de línea
txt	Texto para imprimir
border	Define el tipo de enmarcado de la celda según la tabla Border
ln	Sirve para indicar la posición del puntero de escritura después de Cell
0:	A continuación
1:	Al inicio de la línea siguiente
2:	Debajo
align	Establece el alineado para el texto según la tabla align
fill	Relleno
maxline	Máximo número de líneas. Si es cero no hay limitación

Posibles valores para borde

BORDER_NONE	Sin borde
BORDER_LEFT	Borde a la izquierda
BORDER_RIGHT	Borde a la derecha
BORDER_TOP	Borde superior
BORDER_BOTTOM	Borde inferior
BORDER_FRAME	Borde en los cuatro lados

Al ser constantes numéricas se pueden sumar para combinar su efecto

Posibles valores para alineación

ALIGN_LEFT	Alineado a la izquierda
ALIGN_CENTER	Texto centrado
ALIGN_RIGHT	Alineado a la derecha
ALIGN_JUSTIFY	Texto justificado

OpenPdf

Este método inicia la generación de un documento PDF. OpenPdf() no crea ninguna página. AddPage() invoca al método interno Open() de forma automática si es necesario.

Ver : AddPage(), Close().

PageNo

Devuelve el número de página actual.

Ver: AliasNbPages()

Polygon

Dibuja un polígono con línea de contorno y relleno opcionales.

x	Array de abscisas
y	Array de ordenadas
style	Estilo (contorno y/o relleno)
PDF_STYLE_DRAW :	Dibuja contorno
PDF_STYLE_FILL :	Dibuja relleno
PDF_STYLE_FILLDRAW :	Dibuja contorno y relleno

PushButton

Añade un botón a la página en la posición actual. El botón se incluye en la lista de campos del formulario (acroform), se puede incluir un fragmento de código JavaScript como acción del botón.

name	Nombre del campo
width	Anchura del botón
height	Altura del botón
caption	Texto que aparece en la cara del botón
action	Código JavaScript asociado

RadioButton

Añade un botón de radio en la posición actual.

group	Nombre del grupo al que pertenece
name	Nombre de campo
width	Achura del botón de radio

Rect

Dibuja un rectángulo. El contorno y el relleno son opcionales. Tiene en cuenta los valores de color y ancho de línea establecidos.

Ver `SetLineWidth()`, `SetDrawColor()`, `SetFillColor()`.

x	Abscisa de la esquina superior izquierda
y	Ordenada de la esquina superior izquierda
w	Anchura
h	Altura
style	Estilo (contorno y/o relleno)
PDF_STYLE_DRAW :	Dibuja contorno
PDF_STYLE_FILL :	Dibuja relleno
PDF_STYLE_FILLDRAW :	Dibuja contorno y relleno

RegularPolygon

Dibuja un polígono regular, opcionalmente se puede inscribir un círculo en su interior.

x0	Abscisa del centro
y0	Ordenada del centro
r	Radio del círculo interior
ns	Número de lados
angle	Angulo de orientación
circle	Valor lógico. Indica si se dibuja el círculo interno
style	Estilo del polígono (draw/fill)
circlestyle	Estilo del círculo inscrito (draw/fill)

Rotate

Realiza una rotación con respecto a un centro. La rotación afecta a todos los elementos impresos a continuación, con la excepción de las zonas 'clickables'. Los métodos GetX y GetY no se ven afectados. La rotación tampoco afecta al mecanismo de salto automático de página. La rotación no se mantiene de una página a otra. Cada nueva página comienza con rotación cero.

angle	Angulo de rotación en grados
x	Abscisa del centro de rotación
y	Ordenada del centro de rotación

RotatedImage

Muestra una imagen con rotación Hay que especificar las coordenadas de origen de la imagen mostrada. Las dimensiones pueden ser explícitas o calculadas al igual que ocurre con Image(). Admite los formatos habituales (JPG,PNG,GIF,BMP Y WMF). El tipo de imagen es inferido automáticamente de la extensión de los archivos. La imagen puede funcionar como enlace. Al igual que ocurre con Image(), las imágenes utilizadas varias veces en un mismo archivo PDF, solo se almacenan una vez para ahorrar espacio.

Ver :Image()

x	Abscisa de la esquina superior izquierda
y	Ordenada de la esquina superior izquierda
w	Anchura de la imagen. Si es cero se calcula
h	Altura de la imagen. Si es cero se calcula
angle	Angulo de rotación
link	Enlace
maskImage	Identificador de máscara creado con ImageMask()

RotatedText

Imprime un texto con rotación.

x	Abscisa del centro de rotación
y	Ordenada del centro de rotación
txt	Texto para mostrar con rotación

RoundedRect

Dibuja un rectángulo con las esquinas redondeadas. En la selección de las esquinas que se van a dibujar redondeadas, se pueden sumar los valores para obtener resultados intermedios. De esta forma no es necesario curvar solo una esquina (o todas), sino que puedes dibujar rectángulos con dos o tres esquinas curvadas.

x	Abscisa de la esquina superior izquierda
y	Ordenada de la esquina superior izquierda
w	Anchura del rectángulo
h	Altura del rectángulo
r	Radio de la curva de las esquinas

roundedCorner	Selección de esquinas curvadas, no han de ser las cuatro esquinas
PDF_CORNER_NONE :	Ninguna esquina curvada. Para esto te recomiendo Rect()
PDF_CORNER_TOP_LEFT :	Selecciona la esquina superior izquierda
PDF_CORNER_TOP_RIGHT :	Selecciona la esquina superior derecha
PDF_CORNER_BOTTOM_LEFT :	Selecciona la esquina bajo izquierda
PDF_CORNER_BOTTOM_RIGHT :	Selecciona la esquina bajo derecha
PDF_CORNER_ALL :	Todas las esquinas redondeadas.
style	Estilo (contorno y/o relleno)
PDF_STYLE_DRAW :	Dibuja contorno
PDF_STYLE_FILL :	Dibuja relleno
PDF_STYLE_FILLDRAW :	Dibuja contorno y relleno

SaveAsFile

Guarda el documento creado en un archivo de disco. Close() es llamado en caso necesario. Si se le pasa un nombre vacío se grabará un documento llamado 'doc.pdf' Si existe el archivo se sobrescribirá. Si existe y se encuentra abierto, es posible que se produzca un error o que simplemente no se grabe. Por lo tanto tienes que asegurarte de que el archivo, si existe, está cerrado.

Ver Close()

name	Nombre del archivo. Una cadena vacía equivale a 'doc.pdf'
------	---

Skale

Modifica la escala en las direcciones de X e Y. La transformación se aplica de forma independiente para cada dimensión.

sx	Factor de escalado en anchura. No se admite cero.
sy	Factor de escalado en altura. No se admite cero.

x	Abscisa del centro de escalado
y	Ordenada del centro de escalado

ScaleX

Modifica la escala solo en la dirección de x.

ScaleXY

Efectua un escalado a lo ancho y a lo alto.

s	Factor de escalado en anchura. No se admite cero.
x	Abscisa del centro de escalado
y	Ordenada del centro de escalado

ScaleY

Modifica la escala solo en la dirección de y.

sy	Factor de escalado en altura. No se admite cero.
x	Abscisa del centro de escalado
y	Ordenada del centro de escalado

Sector

Dibuja un sector

x0	Abscisa del punto central.
y0	Ordenada del punto central
r	Radio
astart	Angulo inicial.

afinish	Angulo final
style	Estilo
PDF_STYLE_DRAW :	Dibuja contorno
PDF_STYLE_FILL :	Dibuja relleno
PDF_STYLE_FILLDRAW :	Dibuja contorno y relleno
clockwise	Sentido de las agujas del reloj (si verdadero), al contrario (si falso)
origin	Origen de los ángulos.
0 :	A las tres
90 :	A las doce
180:	A las 9
270:	A las 6

SetAlpha

Establece los valores para Alpha y BlendMode. Devuelve un identificador para su posterior uso.

lineAlpha	Valor de Alpha para trazos. Es un valor entre cero (transparente) y 1 (opaco)
fillAlpha	Valor de Alpha para relleno. Es un valor entre cero (transparente) y 1 (opaco)
blendmode	Valor para BlendMode. Una de las constantes siguientes :
PDF_BLENDMODE_NORMAL	
PDF_BLENDMODE_MULTIPLY	
PDF_BLENDMODE_SCREEN	
PDF_BLENDMODE_OVERLAY	
PDF_BLENDMODE_DARKEN	
PDF_BLENDMODE_LIGHTEN	

PDF_BLENDMODE_COLORDODGE	
PDF_BLENDMODE_COLORBURN	
PDF_BLENDMODE_HARDLIGHT	
PDF_BLENDMODE_SOFTLIGHT	
PDF_BLENDMODE_DIFFERENCE	
PDF_BLENDMODE_EXCLUSION	
PDF_BLENDMODE_HUE	
PDF_BLENDMODE_SATURATION	
PDF_BLENDMODE_COLOR	
PDF_BLENDMODE_LUMINOSITY	

SetAlphaState

Activa un estado de Alpha previamente definido.

id	Identificador devuelto por SetAlpha()
----	---------------------------------------

SetAuthor

Define el autor del documento.

Ver : SetCreator(), SetKeywords(), SetSubject(), SetTitle()

author	Nombre del autor
--------	------------------

SetAutoPageBreak

Activa o desactiva el salto automático de página. En caso de activado, el segundo parámetro define el margen inferior. El estado por defecto es Activado con un margen de dos centímetros.

Ver : Cell(), MultiCell(), AcceptPageBreak()

autopagebreak	Valor de estado activado/desactivado
margin	Distancia desde el final de la página

SetCellMargin

Define el margen para las celdas. Se puede llamar antes de crear la primera página.

margin	Valor del margen
--------	------------------

SetCompression

Activa o desactiva la compresión de página. Si se activa, la representación interna de cada página se comprime y el documento resultante se reduce a la mitad aproximadamente. La compresión está activada por defecto.

compress	Valor lógico de compresión (Verdadero o Falso)
----------	--

SetCreator

Define el creador del documento. Lo normal es poner el nombre de la aplicación que creó el documento. En las versiones freeware de wxPdfCOM, el valor por defecto es 'wxPdfCOM' y no se puede cambiar con SetCreator().

Ver: SetAuthor(), SetKeywords(), SetSubject(), SetTitle()

creator	El nombre del programa que ha creado el documento.
---------	--

SetDisplayMode

Define la forma en que el documento deberá mostrarse en el visor. De puede establecer el nivel de zoom inicial a uno de los niveles preestablecidos, a un factor preciso o dejar que el propio visor fije el nivel por defecto. (En Acrobat, el zoom por defecto se puede escoger en el menú 'Preferencias'). La configuración de página se puede especificar también. El modo por defecto es pantalla completa y continua.

zoom	Tipo de zoom. Uno de los siguientes
PDF_ZOOM_FULLPAGE:	Página completa
PDF_ZOOM_FULLWIDTH :	Ancho de pantalla completo

PDF_ZOOM_REAL:	Tamaño real
PDF_ZOOM_DEFAULT :	Zoom por defecto del visor
PDF_ZOOM_FACTOR :	Valor del tercer argumento zoomFactor
layout	Configuración de página: Uno de los siguientes
PDF_LAYOUT_SINGLE :	Muestra una página cada vez
PDF_LAYOUT_CONTINUOUS	
PDF_LAYOUT_TWO	
PDF_LAYOUT_DEFAULT	
zoomFactor	Si zoom es PDF_ZOOM_FACTOR el porcentaje de zoom será el de este argumento

SetDrawColor

Define el color usado para las operaciones de dibujo. Las operaciones afectadas son las líneas, rectángulos y bordes de celdas. El color se expresa por la combinación de los colores rojo, verde y azul (RGB). El método se puede invocar antes de crear la primera página y su valor se conserva para todas las páginas o hasta que se cambie.

Ver SetFillColor(), SetTextColor(), Line(), Rect(), Cell(), MultiCell()

red	Nivel de rojo. Es un valor entre 0 y 255
green	Nivel de verde. Es un valor entre 0 y 255
blue	Nivel de azul. Es un valor entre 0 y 255

SetFillColor

Define el color para las operaciones de relleno. Las operaciones donde se utiliza el relleno son los rectángulos con la opción de relleno activada, los fondos de las celdas etc. Se puede establecer antes de crear la primera página y el valor se conserva para todas las páginas o hasta que se vuelva a establecer con SetFillColor().

Ver SetDrawColor(), SetTextColor(), Rect(), Cell(), MultiCell()

red	Nivel de rojo. Es un número entre cero y 255
-----	--

green	Nivel de verde. Es un número entre cero y 255
blue	Nivel de azul. Es un número entre cero y 255

SetFillGradient

Dibuja una área rectangular con degradado.

x	Abscisa de la esquina superior izquierda del rectángulo
h	Altura
gradient	Identificador de degradado

SetFont

Establece el tipo de letra utilizado para mostrar cadenas de caracteres. Es necesario establecer un tipo de letra antes de mostrar ningún texto. De lo contrario el resultado no será el esperado. En realidad la fuente por defecto de un documento recién creado es 'ZapfDingbats'. Si no seleccionas ningún tipo de letra e imprimes algo, es posible que lo único que veas sean corazones, círculos y otros símbolos. El tipo de letra puede ser uno de los estándar o bien uno de los añadidos por medio del método AddFont(). Las fuentes estándar utilizan el código cp1252 (Western Europe) de windows. El método se puede invocar antes de crear la primera página y así su efecto afectará a todo el documento o hasta que se establezca otra fuente.

El argumento 'family' puede ser una cadena vacía. En tal caso se conserva la actual familia y solo se tienen en cuenta los otros argumentos. Podemos usar SetFont() para cambiar el tamaño de la fuente aunque es más directo hacerlo con SetFontSize() si es el tamaño lo único que queremos cambiar. Las opciones de estilo se pueden combinar para obtener efectos combinados. En los tipos de letra simbólicos no tienen efecto los estilos 'Negrita' e 'Itálica'. En tamaño se puede poner cero en cuyo caso se conservará el tamaño previamente establecido o 12, (el valor por defecto), si no se ha especificado en ningún momento.

Ver AddFont(), SetFontSize(), Cell(), MultiCell(), Write()

family	Familia de la fuente. estándar de las que se muestran o una definida con AddFont()
Courier	(ancho fijo)
Helvetica or Arial	(sinónimos; sans serif)
Times	(serif)

Symbol	(símbolos)
ZapfDingbats	(símbolos)
style	Estilo de la fuente
Cadena vacía :	Normal
B :	Negrita)
I :	Itálica
BI o IB :	Negrita/Itálica
U :	Subrayado
size	Tamaño de la fuente en puntos.

SetFontPath

Declara la localización de los archivos de definición de fuentes. wxPdfCOM utiliza archivos XML de definición de fuentes para incrustar fuentes en los documentos PDF. El fichero de definición y el fichero de la fuente, en caso de inclusión, deben estar presentes en el path declarado por SetFontPath(). Si se le pasa una cadena vacía tomará el path de la variable de entorno WXPDF_FONTPATH. Si no existe la variable de entorno WXPDF_FONTPATH, se buscarán los archivos en la carpeta 'fonts' del directorio actual.

fontPath	Path por defecto para buscar los ficheros de fuentes
----------	--

SetFontSize

Define el tamaño de la letra.

Ver SetFont()

size	Tamaño de la letra definido en puntos
------	---------------------------------------

SetFontSubsetting

Establece el modo de inclusión de fuentes. Si un documento PDF utiliza fuentes adicionales a las 14 fuentes básicas estándar, hay que incrustar en el documento la información de las mismas. El resultado es que los documentos aumentan considerablemente de tamaño, máxime si se utilizan fuentes Unicode, que suelen incluir miles de símbolos.

Para reducir el tamaño de estos ficheros, se pueden incluir solo los caracteres que se utilicen en el documento. `SetFontSubsetting()` sirve precisamente para activar o desactivar esta opción, aunque solo funciona para fuentes TrueType Unicode.

<code>fontSubsetting</code>	Activa/desactiva la inclusión parcial de fuentes Truetype Unicode
-----------------------------	---

SetFormBorderStyle

Establece el estilo de borde de los campos de formulario. Se puede modificar el grosor y el estilo de todos los campos que se incluyan hasta una nueva llamada a `SetFormBorderStyle()` con diferentes valores.

<code>borderStyle</code>	Estilo del borde de los campos de formulario
<code>PDF_BORDER_SOLID</code>	
<code>PDF_BORDER_DASHED</code>	
<code>PDF_BORDER_BEVELED</code>	
<code>PDF_BORDER_INSET</code>	
<code>PDF_BORDER_UNDERLINE</code>	
<code>borderWidth</code>	Ancho del borde

SetFormColors

Establece los colores de los campos de formulario. Define un color para el borde, otro para el fondo y un color para el texto. Los tres colores se expresan en RGB.

<code>BRed</code>	Componente rojo para el borde. Es un valor entre 0 y 255
<code>BGreen</code>	Componente verde para el borde. Es un valor entre 0 y 255
<code>BBlue</code>	Componente azul para el borde. Es un valor entre 0 y 255
<code>BKRed</code>	Componente rojo para el fondo. Es un valor entre 0 y 255
<code>BKGreen</code>	Componente verde para el fondo. Es un valor entre 0 y 255
<code>BKBlue</code>	Componente azul para el fondo. Es un valor entre 0 y 255
<code>TRed</code>	Componente rojo para el texto. Es un valor entre 0 y 255

TGreen	Componente verde para el texto. Es un valor entre 0 y 255
TBlue	Componente azul para el texto. Es un valor entre 0 y 255

SetImageScale

Establece la escala de imagen

scale	Escala de imagen
-------	------------------

SetKeywords

Asocia al documento una serie de palabras clave. Las palabras clave son una pista sobre el contenido del documento, al igual que ocurre con las palabras clave de los archivos .html. Los buscadores de Internet suelen leer estas palabras para clasificar las páginas según su contenido. El argumento es una cadena de caracteres que contiene varias palabras separadas mediante espacios.

Ver : SetAuthor(), SetCreator(), SetSubject(), SetTitle()

keywords	Lista de palabras clave separadas por espacios
----------	--

SetLeftMargin

Define el margen izquierdo. Este método se puede llamar antes de crear la primera página. Si el valor actual de X queda fuera del margen, su valor se modifica automáticamente.

Ver SetTopMargin(), SetRightMargin(), SetAutoPageBreak(), SetMargins()

margin	Valor para el margen izquierdo
--------	--------------------------------

SetLineHeight

Establece la altura de una línea de texto.

height	Altura de línea
--------	-----------------

SetLineWidth

Define el ancho de línea. Por defecto el ancho de línea es de 0.2 mm. El ancho de línea se puede establecer antes de crear la primera página y se conserva para las siguientes.

Ver Line(), Rect(), Cell(), MultiCell()

width	Valor del ancho de línea
-------	--------------------------

SetLink

Define la página y la posición a la que apunta un enlace.

Ver: AddLink()

link	Identificador del enlace creado con AddLink()
y	Ordenada de la posición. -1 indica la posición actual. 0 indica el principio de página
page	Número de la página de destino. 0 y -1 indican la página actual

SetMargins

Define los márgenes superior, izquierdo y derecho. El valor por defecto para los márgenes es de 1 cm. Para modificar el margen inferior utiliza el método SetAutoPageBreak().

Ver : SetLeftMargin(), SetTopMargin(), SetRightMargin(), SetAutoPageBreak()

left	Margen izquierdo
top	Margen superior
right	Margen derecho

SetProtection

Define los permisos que tendrá el documento. Establece las claves de usuario y propietario. Si no se utiliza esta función, el documento tendrá todos los permisos salvo el de modificación. En caso de utilizarla, solo tendrá los permisos que se activen expresamente.

No es necesario establecer las claves de usuario y propietario. Si especificas una clave de usuario el visualizador, (Acrobat), la pedirá para poder abrir el documento.

Si especificas una clave de propietario, esta servirá para abrir el documento sin restricciones. A la hora de escoger una longitud para la clave ten en cuenta que el valor por defecto (si pones cero) adoptarás el método estándar, revisión 2 con una longitud de 40 bits. Si pones un valor mayor que cero adoptarás el método estándar, revisión 3 con la longitud de clave dada, pero ten en cuenta que esta deberá estar entre 40 y 128 y además ser múltiplo de 8. En caso contrario su longitud se adaptará a estas condiciones. A la hora de establecer claves de encriptación, en cuenta que Acrobat solo soporta claves de 40 y 128 bits.

permissions	Código para activar los permisos. (+) Argumento acumulable
-------------	--

PDF_PERMISSION_NONE	
+ PDF_PERMISSION_PRINT	
+ PDF_PERMISSION_MODIFY	
+ PDF_PERMISSION_COPY	
+ PDF_PERMISSION_ANNOT	
PDF_PERMISSION_ALL	
userPassword	Clave de usuario
ownerPassword	Clave de propietario
encryptionMethod	Método de encriptación
PDF_ENCRYPTION_RC4V1	
PDF_ENCRYPTION_RC4V2	
PDF_ENCRYPTION_AESV2	
keyLength	Longitud de la clave

SetRightMargin

Define el margen derecho. Este método se puede llamar antes de crear la primera página.

Ver : SetLeftMargin(), SetTopMargin(), SetAutoPageBreak(), SetMargins().

margin	Margen derecho
--------	----------------

SetSourceFile

Establece un documento externo para utilizarlo como plantilla. Con wxPdfCOM puedes importar páginas de un documento externo. WxPdfCOM/wxPdfDocument por lo general producen documentos PDF versión 1.3 aunque puede importar documentos hasta la 1.6. En estos casos se cambiará la versión del documento producido a la versión del documento importado.

Es necesario que los permisos para imprimir, copiar y extraer texto y gráficos estén activados en el documento original. De lo contrario no será posible utilizarlo de plantilla externa. SetSourceFile() devuelve el número de páginas del documento analizado o cero si ha habido un error.

Ver : ImportPage(), UseTemplate().

filename	Nombre del documento externo
password	Clave de usuario, solo en caso necesario

SetSubject

Establece el asunto del documento.

subject	Asunto del documento
---------	----------------------

SetTemplateBBox

Define el sector visible de un modelo. Será un área rectangular. El modelo se cargará completo, pero podemos definir que parte es la que será visible en nuestro documento.

Ver : GetTemplateBBox(), BeginTemplate(), ImportPage().

templateId	Identificador del modelo
x	Abscisa de la esquina superior izquierda
y	Ordenada de la esquina superior izquierda
width	Anchura del área visible del modelo
height	altura del área visible del modelo

SetTextColor

Define el color utilizado para mostrar texto. Se expresará por componentes RGB. Podrá establecerse antes de la creación de la primera página y su valor se recordará para el resto del documento.

Ver SetDrawColor(), SetFillColor(), Text(), Cell(), MultiCell().

red	Indica el nivel de rojo. Es un valor entre 0 y 255
green	Indica el nivel de verde. Es un valor entre 0 y 255
blue	Indica el nivel de azul. Es un valor entre 0 y 255

SetTitle

Define el título del documento.

Ver : SetAuthor(), SetCreator(), SetKeywords(), SetSubject().

title	Título del documento
-------	----------------------

SetTopMargin

Define el margen superior. Este método se puede llamar antes de crear la primera página y conserva su valor hasta nuevos cambios.

Ver `SetLeftMargin()`, `SetRightMargin()`, `SetAutoPageBreak()`, `SetMargins()`.

margin	Margen superior
--------	-----------------

SetX

Define la abscisa de la posición actual. Si el valor del argumento es negativo, se entiende referido al extremo derecho de la página.

Ver : `GetX()`, `GetY()`, `SetY()`, `SetXY()`.

x	Valor de la abscisa de la posición actual
---	---

SetXY

Define la abscisa y la ordenada de la posición actual. Si el valor de x es negativo, se entiende relativo al extremo derecho de la página. Si el valor de y es negativo, se entenderá relativo al extremo inferior de la página.

Ver : `SetX()`, `SetY()`.

x	Valor de la abscisa
y	Valor de la ordenada

SetY

Mueve la abscisa al margen izquierdo y ajusta la ordenada al valor del argumento. Si el valor de y es negativo, se entiende relativo a la parte inferior de la página.

Ver : `GetX()`, `GetY()`, `SetY()`, `SetXY()`.

y	Valor de la ordenada de la posición actual
---	--

Shape

Dibuja una forma que haya sido definida. El identificador de una forma es el nombre asignado durante su definición.

shape	Nombre asignado a la forma
-------	----------------------------

style	Estilo de dibujo. Uno de los siguientes o una combinación válida.
+ PDF_STYLE_DRAW	
+ PDF_STYLE_FILL	
+ PDF_STYLE_FILLDRAW	
+ PDF_STYLE_DRAWCLOSE	Cierra el path y dibuja, combinable con PDF_STYLE_FILL

ShapeCurveTo

Añade una curva de Bezier a una forma. La curva se extiende desde la posición actual hasta el punto (x3,y3) utilizando (x1,y1) y (x2,y2) como puntos de control. La posición del puntero de la forma queda en el punto (x3,y3).

name	Nombre asignado a la forma
x1	Abscisa del punto de control 1
y1	Ordenada del punto de control 1
x2	Abscisa del punto de control 2
y2	Ordenada del punto de control 2
x3	Abscisa del punto de destino
y3	Ordenada del punto de destino

ShapedText

Imprime un texto a lo largo de un recorrido definido por una forma. Dependiendo de la longitud del texto y de la forma, es posible definir varios modos de visualización: Texto se muestra una sola vez, texto estirado a la longitud de la forma o texto repetido a lo largo de la forma.

shape	Nombre de la forma
text	Texto para imprimir
mode	Modo: Ver opciones siguientes

PDF_SHAPEDTEXTMODE_ONETIME :	Una vez
PDF_SHAPEDTEXTMODE_STRETCHTOFIT :	Estirado a lo largo de la forma
PDF_SHAPEDTEXTMODE_REPEAT :	Repetido

ShapeLineto

Añade una línea a una forma. La línea parte desde la posición actual de la forma hasta el punto suministrado. La posición actual de la forma se mueve al nuevo punto.

shape	Nombre de un objeto Shape (Nuevo o existente)
x	Abscisa del punto
y	Ordenada del punto

ShapeMoveto

Define un nuevo punto para una forma. Si la forma no ha sido definida con anterioridad, se creará con el nombre dado en esta función y se fijará este punto como arranque de la nueva forma.

shape	Nombre de un objeto Shape (Nuevo o existente)
x	Abscisa del punto
y	Ordenada del punto

Skale

Modifica la escala en las direcciones de X e Y. La transformación se aplica de forma independiente para cada dimensión.

sx	Factor de escalado en anchura. No se admite cero.
sy	Factor de escalado en altura. No se admite cero.
x	Abscisa del centro de escalado
y	Ordenada del centro de escalado

Skew

Produce una inclinación en las direcciones de x e y.

xAngle	Angulo expresado en grados entre -90 (inclinación izquierda) y + 90 (derecha)
yAngle	Angulo expresado en grados entre -90 (inclinación abajo) y + 90 (arriba)
x	Abscisa del eje de inclinación
y	Ordenada del eje de inclinación

SkewX

Produce una inclinación en el sentido de x.

xAngle	Angulo expresado en grados entre -90 (inclinación izquierda) y + 90 (derecha)
x	Abscisa del eje de inclinación
y	Ordenada del eje de inclinación

SkewY

Produce una inclinación en el sentido de y.

yAngle	Angulo expresado en grados entre -90 (inclinación abajo) y + 90 (arriba)
x	Abscisa del eje de inclinación
y	Ordenada del eje de inclinación

StartTransform

Entra en un entorno de transformación. Este método debería ser invocado antes de aplicar transformaciones. Todos los métodos de transformación invocan a este método de forma implícita si es necesario. El estado inicial se restaura automáticamente con el final de página. No necesita argumentos.

Ver : StopTransform().

StopTransform

Abandona el contexto de transformación. Restaura los valores gráficos salvados con la última llamada a StartTransform(). Este método debe ser invocado para cancelar un entorno de transformación iniciado con StartTransform. Todos los entornos de transformación se cierran implícitamente con el final de página. No necesita argumentos.

Text

Imprime una cadena de caracteres en un lugar exacto. El punto suministrado determinará la base izquierda del primer carácter de la cadena. Las otras funciones para imprimir texto Cell(), MultiCell() o Write() son más potentes y efectivas, por lo que se recomienda utilizar Text() solo cuando sea necesario controlar con precisión el punto de inserción del mismo.

Ver : SetFont(), SetTextColor(), Cell(), MultiCell(), Write().

x	Abscisa del origen
y	Ordenada del origen
txt	Texto para imprimir

TextBox

Este método cuenta el número de líneas que ocupa el texto para un ancho máximo.

w	Ancho de celda. Si es cero, ocupa toda el ancho disponible
h	Altura de celda
txt	Texto para imprimir
halign	Permite centrar o alinear el texto. Admite los valores siguientes : PDF_ALIGN_LEFT PDF_ALIGN_CENTER PDF_ALIGN_RIGHT PDF_ALIGN_JUSTIFY
valign	Controla la alineación vertical. Admite los valores siguientes PDF_ALIGN_TOP ALIGN_MIDDLE PDF_ALIGN_BOTTOM
border	Indica qué bordes se van a dibujar. Admite los valores siguientes.(+)Acumulables) PDF_BORDER_NONE + PDF_BORDER_LEFT + PDF_BORDER_RIGHT

	+ PDF_BORDER_TOP + PDF_BORDER_BOTTOM PDF_BORDER_FRAME
fill	Valor lógico de relleno. 1 Verdadero, 0 Falso

TextField

Añade un campo de texto en la posición actual. El campo formará parte del Acroform.

name	Nombre del campo
width	Ancho del campo de texto
height	Altura del campo
value	Valor por defecto para el campo
multiline	Determina si el campo tendrá múltiples líneas

Translate

Mueve el origen relativamente a la posición actual.

tx	Movimiento a la derecha
ty	Movimiento hacia abajo

TranslateX

Mueve la abscisa relativamente a la posición actual

tx	Movimiento hacia la derecha
----	-----------------------------

TranslateY

Mueve la ordenada con respecto a la posición actual.

ty	Movimiento hacia abajo
----	------------------------

UnsetClipping

Elimina el área de máscara. Después de trabajar con máscaras, hay que eliminarlas con una llamada a UnsetClipping(). No tiene argumentos.

UseTemplate

Utiliza un modelo en la página actual o en otro modelo. Se utiliza como si fuera una imagen. Si se especifican a cero, el ancho y el alto se averiguan con una llamada interna a `GetTemplateSize()` o por los valores pasados a `BeginTemplate()`. El modelo puede quedar distorsionado si las dimensiones suministradas no se corresponden con las reales del modelo.

Ver : `BeginTemplate()`, `EndTemplate()`, `ImportPage()`

templateId	Identificador de modelo válido
x	Abscisa de la posición
y	Ordenada de la posición
width	Ancho
height	Alto

Write

Imprime texto en la posición actual. Cuando se alcanza el margen derecho o se encuentra con un carácter de salto de línea (`\n`), `Write` salta al principio de la línea siguiente. Se puede crear un enlace con `Write`. La posición queda fijada al final del texto mostrado.

Ver : `SetFont()`, `SetTextColor()`, `AddLink()`, `MultiCell()`, `SetAutoPageBreak()`

h	Altura de la línea
txt	Texto para mostrar
link	Enlace

WriteCell

Este método imprime texto con atributos de celda en la posición actual. Provoca un salto de línea cuando se alcanza el margen derecho o se encuentra el carácter `\n`. La impresión continua en el margen derecho de la siguiente línea. después de mostrar el texto la posición queda establecida al final del mismo. Se puede crear un enlace.

Ver: `SetFont()`, `SetTextColor()`, `AddLink()`, `MultiCell()`, `SetAutoPageBreak()`

h	Altura de la línea
txt	Texto para imprimir
border	Borde. Hay los siguientes (Acumulables) + PDF_BORDER_NONE + PDF_BORDER_LEFT + PDF_BORDER_RIGHT + PDF_BORDER_TOP + PDF_BORDER_BOTTOM PDF_BORDER_FRAME
fill	Relleno
link	Enlace

WriteXML

Imprime un texto con etiquetas HTML.
La impresión comienza en la posición actual.

str	Cadena de texto con etiquetas HTML
-----	------------------------------------

Anexos

Listado alfabético de instrucciones

AcceptPageBreak	Controla el salto de página
AddFont	Importa fuentes TrueType, TrueTypeUnicode o Type1
AddFontCJK	Importa una fuente CJK (Chino, Japonés,(K)Coreano)
AddLink	Crea un enlace interno
AddPage	Añade una página nueva al documento
AppendJavaScript	Añade código JavaScript
Arrow	Dibuja una flecha
AxialGradient	Define un sombreado con degradación axial
BeginTemplate	Marca el inicio de un modelo o plantilla (template)
BookMark	Añade una entrada de índice
Cell	Dibuja un texto enmarcado con varias opciones
CheckBox	Añade un campo tipo Check al formulario
ClippedCell	Dibuja una celda con recorte
ClippingEllipse	Define una mascara en forma de elipse
ClippingPath	Define una máscara con la forma de un objeto Shape
ClippingPolygon	Define un polígono como área de enmarcado
ClippingRect	Define un área de enmarcado rectangular
ClippingText	Define un texto como mascara
CloseFile	Cierra el documento
ClosePath	Sirve para dibujar un texto

ComboBox	Sirve para dibujar un texto
Convert2Roman	Convierte un entero a números romanos
Curve	Dibuja una curva de Bezier
CurveTo	Dibuja una curva de Bezier utilizando el actual como punto de salida
DrawCircle	Dibuja un círculo
Ellipse	Dibuja una elipse
EndTemplate	Finaliza la construcción de una plantilla. Devuelve su identificador
Footer	Controla la grabación del pie de página
GetBreakMargin	Devuelve el margen para el salto de página
GetCellMargin	Devuelve el margen de celda
GetFontFamily	Devuelve el nombre de la familia de la fuente actual
GetFontPath	Devuelve el path de la carpeta de fuentes
GetFontSize	Devuelve el tamaño en puntos de la fuente actual
GetFontStyle	Devuelve el estilo de la fuente
GetFontSubsetting	Devuelve el estado de activación de la inclusión parcial de fuentes Unicode
GetImageScale	Devuelve la escala de la imagen
GetLeftMargin	Devuelve el margen izquierdo
GetLineHeight	Devuelve la altura de la línea de texto
GetLineWidth	Devuelve el ancho de línea actual
GetPageHeight	Devuelve la altura de la página
GetPageWidth	Devuelve el ancho actual de página
GetRightMargin	Devuelve el margen derecho
GetScaleFactor	Devuelve el factor de escalado

GetStringWidth	Devuelve la longitud de una cadena.
GetTemplateBBox	Obtiene las dimensiones de una plantilla externa
GetTemplateSize	Obtiene el ancho y el alto de una plantilla
GetTopMargin	Devuelve el margen superior
GetX	Devuelva la abscisa de la posición actual
GetY	Devuelve la ordenada de la posición actual
HEADER	Controla la grabación de la cabecera de página
Image	Muestra una imagen
ImageMask	Prepara una imagen para su uso como máscara.
ImportPage	Importa una página de un documento PDF externo
ImportText	Crea un párrafo justificado a partir del contenido de un archivo de texto
Initialize	Inicializa el documento
Lin	Dibuja una línea entre dos puntos
LineCount	Cuenta el número de líneas que ocuparía un texto
Ln	Provoca saltos de línea de altura variable
Marker	Dibuja una marca
MirrorH	Crea un reflejo horizontal
MirrorV	Crea un reflejo vertical
MoveTo	Reposiciona el puntero de dibujo
MultiCell	Imprime un texto en varias líneas
OpenPdf	Inicia la generación de un documento Pdf
PageNo	Devuelve el número de la página actual
Polygon	Dibuja un polígono

PushButton	Crea un botón de formulario
RadioButton	Crea un botón de radio en la posición actual
Rect	Dibuja un rectángulo
RegularPolygon	Dibuja un polígono regular. Círculo inscrito opcional
Rotate	Activa la rotación de la salida gráfica.
RotatedImage	Muestra una imagen con rotación
RotatedText	Muestra un texto con rotación
RoundedRect	Muestra un rectángulo con alguna o todas sus esquinas redondeadas
SaveAsFile	Guarda el documento en un archivo de disco
Scale/Skale	Muestra una imagen con rotación
ScaleX	Muestra un texto con rotación
ScaleXY	Muestra un rectángulo con alguna o todas sus esquinas redondeadas
ScaleY	Guarda el documento en un archivo de disco
Sector	Dibuja un sector
SetAlpha	Establece valores para Alpha y para BlendMode
SetAlphaState	Establece un estado de Alpha definido previamente con SetAlpha()
SetAuthor	Define el autor del documento
SetAutoPageBreak	Establece el estado de salto de página automático así como el margen inferior
SetCellMargin	Establece el margen para las celdas
SetCompression	Activa/Desactiva la compresión de páginas
SetDrawColor	Establece el color para pincel de dibujo
SetCreator	Establece el nombre de la aplicación que ha creado el documento
SetDisplaymode	Establece la opciones de visualización iniciales del documento

SetFillColor	Establece el color para operaciones de relleno
SetFillGradient	Dibuja un área rectangular con degradado de color
SetFont	Establece indistintamente el tipo de letra, el estilo y el tamaño
SetFontPath	Establece el path para buscar los archivos de las fuentes
FontSize	Establece tamaño de letra
SetFontSubsetting	Activa o desactiva la inclusión parcial de fuentes TrueType Unicode
SetFormBorderStyle	Establece el estilo del borde de los campos de formulario
SetImageScale	Establece la escala de imagen
SetKeywords	Define las palabras clave del documento
SetLeftMargin	Establece el margen izquierdo
SetLineHeight	Establece indistintamente el tipo de letra, el estilo y el tamaño
SetLineWidth	Establece el ancho de línea
SetLink	Define la página y la posición a la que apunta un enlace.
SetMargins	Establece los márgenes superior, izquierdo y derecho
SetProtection	Activa los permisos y claves del documento
SetRightMargin	Establece el margen derecho
SetSourceFile	Establece un documento externo para utilizarlo como plantilla
SetSubject	Establece el asunto del documento
SetTemplateBBox	Define el sector visible de un modelo. Será un área rectangular
SetTextColor	Define el color utilizado para mostrar texto
SetTitle	Define el título del documento.
SetTopMargin	Define el margen superior
SetX	Define la abscisa de la posición actual

SetXY	Define la abscisa y la ordenada de la posición actual
SetY	Mueve la abscisa al margen izquierdo y ajusta la ordenada al valor del argumento
Shape	Dibuja una forma que haya sido definida.
ShapeCurveTo	Añade una curva de Bezier a una forma.
ShapedText	Imprime un texto a lo largo de un recorrido definido por una forma
ShapeLineto	Añade una línea recta a una forma
ShapeMoveto	Define un nuevo punto para una forma
Skale	Modifica la escala en las direcciones de X e Y.
Skew	Produce una inclinación en las direcciones de x e y
SkewX	Produce una inclinación en el sentido de x.
SkewY	Produce una inclinación en el sentido de y.
StartTransform	Entra en un entorno de transformación.
StopTransform	Abandona el contexto de transformación.
Text	Imprime una cadena de caracteres en un lugar exacto.
TextBox	Este método cuenta el número de líneas que ocupa el texto para un ancho máximo.
TextField	Añade un campo de texto en la posición actual.
Translate	Mueve el origen relativamente a la posición actual.
TranslateX	Mueve la abscisa relativamente a la posición actual
TranslateY	Mueve la ordenada con respecto a la posición actual.
UnsetClipping	Elimina el área de máscara.
UseTemplate	Utiliza un modelo en la página actual o en otro modelo.
Write	Imprime texto en la posición actual.
WriteCell	Este método imprime texto con atributos de celda en la posición actual.

WriteXML	Imprime un texto con etiquetas HTML.
----------	--------------------------------------

Tablas

Valores de orientación del papel

PDF_ORIENTATION_PORTRAIT	1
PDF_ORIENTATION_LANDSCAPE	2

Constantes para la definición del borde de Cell

PDF_BORDER_NONE	0
PDF_BORDER_LEFT	1
PDF_BORDER_RIGHT	2
PDF_BORDER_TOP	4
PDF_BORDER_BOTTOM	8
PDF_BORDER_FRAME	15

Opciones de alineado

PDF_ALIGN_LEFT	0
PDF_ALIGN_CENTER	1
PDF_ALIGN_RIGHT	2
PDF_ALIGN_JUSTIFY	3
PDF_ALIGN_TOP	0
PDF_ALIGN_MIDDLE	1
PDF_ALIGN_BOTTOM	2

Opciones para el argumento Ln de Cell

PDF_LN_NO	0
PDF_LN_LN	1
PDF_LN_LN2	2

Opciones de estilo

PDF_STYLE_NOOP	0
PDF_STYLE_DRAW	1
PDF_STYLE_FILL	2
PDF_STYLE_FILLDRAW	3
PDF_STYLE_DRAWCLOSE	4
PDF_STYLE_MASK	7

Formas admitidas por el método Marker

PDF_MARKER_CIRCLE	0
PDF_MARKER_SQUARE	1
PDF_MARKER_TRIANGLE_UP	2
PDF_MARKER_TRIANGLE_DOWN	3
PDF_MARKER_TRIANGLE_LEFT	4
PDF_MARKER_TRIANGLE_RIGHT	5
PDF_MARKER_MARKER_DIAMOND	6

PDF_MARKER_PENTAGON_UP	7
PDF_MARKER_PENTAGON_DOWN	8
PDF_MARKER_PENTAGON_LEFT	9
PDF_MARKER_PENTAGON_RIGHT	10
PDF_MARKER_STAR	11
PDF_MARKER_STAR4	12
PDF_MARKER_PLUS	13
PDF_MARKER_CROSS	14
PDF_MARKER_SUN	15
PDF_MARKER_BOWTIE_HORIZONTAL	16
PDF_MARKER_BOWTIE_VERTICAL	17
PDF_MARKER_MARKER_ASTERISK	18
PDF_MARKER_MARKER_LAST	19

Valores para BlendMode

PDF_BLENDMODE_NORMAL	0
PDF_BLENDMODE_MULTIPLY	1
PDF_BLENDMODE_SCREEN	2
PDF_BLENDMODE_OVERLAY	3
PDF_BLENDMODE_DARKEN	4
PDF_BLENDMODE_LIGHTEN	5
PDF_BLENDMODE_COLORDODGE	6
PDF_BLENDMODE_COLORBURN	7
PDF_BLENDMODE_HARDLIGHT	8
PDF_BLENDMODE_SOFTLIGHT	9
PDF_BLENDMODE_DIFFERENCE	10
PDF_BLENDMODE_EXCLUSION	11
PDF_BLENDMODE_HUE	12
PDF_BLENDMODE_SATURATION	13
PDF_BLENDMODE_COLOR	14
PDF_BLENDMODE_LUMINOSITY	15

Valores de Zoom

PDF_ZOOM_FULLPAGE	0
PDF_ZOOM_FULLWIDTH	1
PDF_ZOOM_REAL	2
PDF_ZOOM_DEFAULT	3
PDF_ZOOM_FACTOR	4

Opciones de presentación de página

PDF_LAYOUT_CONTINUOUS	0
PDF_LAYOUT_SINGLE	1
PDF_LAYOUT_TWO	2
PDF_LAYOUT_DEFAULT	3

Tipos de permiso para el documento

PDF_PERMISSION_NONE	0
PDF_PERMISSION_PRINT	4
PDF_PERMISSION_MODIFY	8
PDF_PERMISSION_COPY	16
PDF_PERMISSION_ANNOT	32
PDF_PERMISSION_ALL	60

Métodos de encriptado

PDF_ENCRYPTION_RC4V1	0
PDF_ENCRYPTION_RC4V2	1
PDF_ENCRYPTION_AESV2	2

Tipos de papel

PAPER_NONE	0 Sin definir
PAPER_LETTER	1 Carta, 8 1/2 x 11 pulgadas
PAPER_LEGAL	2 Legal, 8 1/2 x 14 pulgadas
PAPER_A4	3 Folio A4, 210 x 297 milímetros
PAPER_CSHEET	4 C Folio, 17 x 22 pulgadas
PAPER_DSHEET	5 D Folio, 22 x 34 pulgadas
PAPER_ESHEET	6 E Folio, 34 x 44 pulgadas
PAPER_LETTERSMALL	7 Carta pequeña, 8 1/2 x 11 pulgadas
PAPER_TABLOID	8 Tabloid, 11 x 17 pulgadas
PAPER_LEDGER	9 Ledger, 17 x 11 pulgadas
PAPER_STATEMENT	10 Statement, 5 1/2 x 8 1/2 pulgadas
PAPER_EXECUTIVE	11 Executive, 7 1/4 x 10 1/2 pulgadas
PAPER_A3	12 A3 Folio, 297 x 420 milímetros
PAPER_A4SMALL	13 A4 small Folio, 210 x 297 milímetros
PAPER_A5	14 A5 Folio, 148 x 210 milímetros
PAPER_B4	15 B4 Folio, 250 x 354 milímetros
PAPER_B5	16 B5 Folio, 182 x 257 milímetros
PAPER_FOLIO	17 Folio, 8 1/2 x 13 pulgadas
PAPER_QUARTO	18 Quarto, 215x275 milímetros
PAPER_10X14	19 Folio 10 x 14 pulgadas
PAPER_11X17	20 Folio 11 x 17 pulgadas
PAPER_NOTE	21 Nota, 8 1/2 x 11 pulgadas
PAPER_ENV_9	22 Sobre #9, 3 7/8 x 8 7/8 pulgadas
PAPER_ENV_10	23 Sobre #10, 4 1/8 x 9 1/2 pulgadas
PAPER_ENV_11	24 Sobre #11, 4 1/2 x 10 3/8 pulgadas
PAPER_ENV_12	25 Sobre #12, 4 3/4 x 11 pulgadas
PAPER_ENV_14	26 Sobre #14, 5 x 11 1/2 pulgadas
PAPER_ENV_DL	27 Sobre DL, 110 x 220 milímetros
PAPER_ENV_C5	28 Sobre C5, 162 x 229 milímetros
PAPER_ENV_C3	29 Sobre C3, 324 x 458 milímetros
PAPER_ENV_C4	30 Sobre C4, 229 x 324 milímetros

PAPER_ENV_C6	31 Sobre C6, 114 x 162 milímetros
PAPER_ENV_C65	32 Sobre C65, 114 x 229 milímetros
PAPER_ENV_B4	33 Sobre B4, 250 x 353 milímetros
PAPER_ENV_B5	34 Sobre B5, 176 x 250 milímetros
PAPER_ENV_B6	35 Sobre B6, 176 x 125 milímetros
PAPER_ENV_ITALY	36 Sobre Italy, 110 x 230 milímetros
PAPER_ENV_MONARCH	37 'Sobre Monarch , 3 7/8 x 7 1/2 pulgadas
PAPER_ENV_PERSONAL	38 Sobre 6 3/4, 3 5/8 x 6 1/2 pulgadas
PAPER_FANFOLD_US	39 USA Std Fanfold, 14 7/8 x 11 pulgadas
PAPER_FANFOLD_STD_GERMAN	40 German Std Fanfold, 8 1/2 x 12 pulgadas
PAPER_FANFOLD_LGL_GERMAN	41 German Legal Fanfold, 8 1/2 x 13 pulgadas
PAPER_ISO_B4	42 B4 (ISO) 250 x 353 mm
PAPER_JAPANESE_POSTCARD	43 Japanese Postcard 100 x 148 mm
PAPER_9X11	44 9 x 11 pulgadas
PAPER_10X11	45 10 x 11 pulgadas
PAPER_15X11	46 15 x 11 pulgadas
PAPER_ENV_INVITE	47 Sobre Invite 220 x 220 mm
PAPER_LETTER_EXTRA	48 Letter Extra 9 1/2 x 12 pulgadas
PAPER_LEGAL_EXTRA	49 Legal Extra 9 1/2 x 15 pulgadas
PAPER_TABLOID_EXTRA	50 Tabloid Extra 11.69 x 18 pulgadas
PAPER_A4_EXTRA	51 A4 Extra 9.27 x 12.69 pulgadas
PAPER_LETTER_TRANSVERSE	52 Letter Transverse 8 1/2 x 11 pulgadas
PAPER_A4_TRANSVERSE	53 A4 Transverse 210 x 297 mm
PAPER_LETTER_EXTRA_TRANSVERSE	54 'Letter Extra Transverse 9 1/2 x 12 pulgadas
PAPER_A_PLUS	55 SuperA/SuperA/A4 227 x 356 mm
PAPER_B_PLUS	56 SuperB/SuperB/A3 305 x 487 mm
PAPER_LETTER_PLUS	57 Letter Plus 8.5 x 12.69 pulgadas
PAPER_A4_PLUS	58 A4 Plus 210 x 330 mm
PAPER_A5_TRANSVERSE	59 A5 Transverse 148 x 210 mm
PAPER_B5_TRANSVERSE	60 B5 (JIS) Transverse 182 x 257 mm
PAPER_A3_EXTRA	61 A3 Extra 322 x 445 mm
PAPER_A5_EXTRA	62 A5 Extra 174 x 235 mm
PAPER_B5_EXTRA	63 B5 (ISO) Extra 201 x 276 mm
PAPER_A2	64 A2 420 x 594 mm
PAPER_A3_TRANSVERSE	65 A3 Transverse 297 x 420 mm
PAPER_A3_EXTRA_TRANSVERSE	66 A3 Extra Transverse 322 x 445 mm
PAPER_DBL_JAPANESE_POSTCARD	67 Japanese Double Postcard 200 x 148 mm
PAPER_A6	68 A6 105 x 148 mm
PAPER_JENV_KAKU2	69 Japanese Envelope Kaku #2
PAPER_JENV_KAKU3	70 Japanese Envelope Kaku #3
PAPER_JENV_CHOU3	71 Japanese Envelope Chou #3
PAPER_JENV_CHOU4	72 Japanese Envelope Chou #4
PAPER_LETTER_ROTATED	73 Letter Rotated 11 x 8 1/2 in
PAPER_A3_ROTATED	74 A3 Rotated 420 x 297 mm

PAPER_A4_ROTATED	75 A4 Rotated 297 x 210 mm
PAPER_A5_ROTATED	76 A5 Rotated 210 x 148 mm
PAPER_B4_JIS_ROTATED	77 B4 (JIS) Rotated 364 x 257 mm
PAPER_B5_JIS_ROTATED	78 B5 (JIS) Rotated 257 x 182 mm
PAPER_JAPANESE_POSTCARD_ROTATED	79 Japanese Postcard Rotated 148 x 100 mm
PAPER_DBL_JAPANESE_POSTCARD_ROTATED	80 Double Japanese Postcard Rotated 148 x 200 mm
PAPER_A6_ROTATED	81 A6 Rotated 148 x 105 mm
PAPER_JENV_KAKU2_ROTATED	82 Japanese Envelope Kaku #2 Rotated
PAPER_JENV_KAKU3_ROTATED	83 Japanese Envelope Kaku #3 Rotated
PAPER_JENV_CHOU3_ROTATED	84 Japanese Envelope Chou #3 Rotated
PAPER_JENV_CHOU4_ROTATED	85 Japanese Envelope Chou #4 Rotated
PAPER_B6_JIS	86 B6 (JIS) 128 x 182 mm
PAPER_B6_JIS_ROTATED	87 B6 (JIS) Rotated 182 x 128 mm
PAPER_12X11	88 12 x 11 pulgadas
PAPER_JENV_YOU4	89 Japanese Envelope You #4
PAPER_JENV_YOU4_ROTATED	90 Japanese Envelope You #4 Rotated
PAPER_P16K	91 PRC 16K 146 x 215 mm
PAPER_P32K	92 PRC 32K 97 x 151 mm
PAPER_P32KBIG	93 PRC 32K(Big) 97 x 151 mm
PAPER_PENV_1	94 PRC Envelope #1 102 x 165 mm
PAPER_PENV_2	95 PRC Envelope #2 102 x 176 mm
PAPER_PENV_3	96 PRC Envelope #3 125 x 176 mm
PAPER_PENV_4	97 PRC Envelope #4 110 x 208 mm
PAPER_PENV_5	98 PRC Envelope #5 110 x 220 mm
PAPER_PENV_6	99 PRC Envelope #6 120 x 230 mm
PAPER_PENV_7	100 PRC Envelope #7 160 x 230 mm
PAPER_PENV_8	101 PRC Envelope #8 120 x 309 mm
PAPER_PENV_9	102 PRC Envelope #9 229 x 324 mm
PAPER_PENV_10	103 PRC Envelope #10 324 x 458 mm
PAPER_P16K_ROTATED	104 PRC 16K Rotated
PAPER_P32K_ROTATED	105 PRC 32K Rotated
PAPER_P32KBIG_ROTATED	106 PRC 32K(Big) Rotated
PAPER_PENV_1_ROTATED	107 PRC Envelope #1 Rotated 165 x 102 mm
PAPER_PENV_2_ROTATED	108 PRC Envelope #2 Rotated 176 x 102 mm
PAPER_PENV_3_ROTATED	109 PRC Envelope #3 Rotated 176 x 125 mm
PAPER_PENV_4_ROTATED	110 PRC Envelope #4 Rotated 208 x 110 mm
PAPER_PENV_5_ROTATED	112 PRC Envelope #5 Rotated 220 x 110 mm
PAPER_PENV_6_ROTATED	113 PRC Envelope #6 Rotated 230 x 120 mm
PAPER_PENV_7_ROTATED	114 PRC Envelope #7 Rotated 230 x 160 mm
PAPER_PENV_8_ROTATED	115 PRC Envelope #8 Rotated 309 x 120 mm
PAPER_PENV_9_ROTATED	116 PRC Envelope #9 Rotated 324 x 229 mm
PAPER_PENV_10_ROTATED	117 PRC Envelope #10 Rotated 458 x 324 m

Introducción.....	2
FPDF.....	2
COM.....	4
Ventajas e inconvenientes.....	4
Configurando el entorno.....	6
Registrar el componente.....	6
Archivo de constantes.....	7
Declarando variables.....	7
Asignando memoria.....	8
Comunicando con wxPdfCOM.....	8
Manual del usuario.....	9
Consideraciones generales.....	9
Paso de argumentos.....	9
Paso de arrays.....	9
Argumentos acumulables.....	10
Definir el formato del documento.....	12
Propiedades del documento.....	12
Cabeceras y pies de página.....	12
Imprimir textos.....	13
Gráficos.....	17
Referencia de instrucciones.....	21
Anexos.....	67
Listado alfabético de instrucciones.....	67
Tablas.....	74
Índice.....	79