

# Contents

## Overview

SWBPrinter is an in-process server (ActiveX DLL) that allows Visual Basic programmers to print and preview reports easily without the use of a full-featured report writer. While it was not intended to fully replace the VB Printer object, it does offer most of the functionality required to produce simple reports and documents with print and preview capability. SWBPrinter has the following features:

- Output text to the page at any location specified.
- Address the page in any ScaleMode available to Visual Basic.
- User-defined line spacing for line-oriented reports.
- Automatic internal page counter.
- Optional automatic headings on each new page.
- Support for portrait and landscape page formats.
- Support for font control.
- Support for line and box drawing.
- Formats and prints memo text (long text strings) within a specified horizontal region.

## [Sample Application](#)

## Properties

[Orientation](#)

[X](#)

[Y](#)

[Page](#)

[Title](#)

[ShowPreviewSave](#)

[ScaleMode](#)

[TotalPages](#)

[Spacing](#)

[AutoHeadings](#)

[AutoHeadingsFontName](#)

[AutoHeadingsFontSize](#)

[AutoHeadingsFontAttr](#)

[AutoNewPage](#)

[PrintFile](#)

[ShowProgress](#)

## Methods

[SetFont](#)

[TextOut](#)

[NewPage](#)

[PreviewReport](#)

[PrintReport](#)

[BeginDoc](#)

[EndDoc](#)  
[DestroyReport](#)  
[LineOut](#)  
[BoxOut](#)  
[MemoOut](#)  
[ImageOut](#)  
[SaveReport](#)

**Events**

[OnNewPage](#)

# Sample Application

'Sample usage of SWBPrinter Print Preview Engine DLL

```
'Declare Constants for use with SWBPrinter
Private Const SWBPRINTER_FONT_BOLD = 1
Private Const SWBPRINTER_FONT_UNDERLINE = 2
Private Const SWBPRINTER_FONT_ITALIC = 4
```

```
'Declare SWBPrinter
Private WithEvents SWBPrintEngine As SWBPrinter
```

```
Private Sub Form_Load()
    'Create an instance of SWBPrinter
    Set SWBPrintEngine = New SWBPrinter
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
    'Allow user to save the report if desired.
    'The SaveReport method returns False if user clicked Cancel in Save dialog.
    If Not SWBPrintEngine.SaveReport("") Then MsgBox ("You chose not to save the current report.")

    'Clean up the temporary file used to create the report.
    SWBPrintEngine.DestroyReport
End Sub
```

'This subroutine defines the click event of a "Preview" button on a form.

```
Private Sub PrevBtn_Click()
    Dim I As Integer

    'Indicate the start of a new report job.
    SWBPrintEngine.BeginDoc

    'Set global settings ScaleMode, Orientation, and vertical line Spacing
    SWBPrintEngine.ScaleMode = vbInches
    SWBPrintEngine.Orientation = vbPRORPortrait
    SWBPrintEngine.Spacing = 1.2

    'Indicate that we want automatic headings on each page.
    SWBPrintEngine.AutoHeadings = True
    SWBPrintEngine.AutoNewPage = True
    SWBPrintEngine.ShowProgress = True
    SWBPrintEngine.AutoHeadingsFontName = "Arial"
    SWBPrintEngine.AutoHeadingsFontSize = 12
    SWBPrintEngine.AutoHeadingsFontAttr = SWBPRINTER_FONT_ITALIC Or SWBPRINTER_FONT_BOLD
    SWBPrintEngine.Title = "Sample SWBPrinter Report"

    'Set the initial font to Arial 10 point, with no special attributes.
    SWBPrintEngine.SetFont "Arial", 10, 0
```

```
'Loop and print some text down the page. NewLine advances according to Spacing property.
SWBPrintEngine.Y = 2
For I = 1 To 20
    SWBPrintEngine.TextOut 0.5, "The quick brown fox jumped over the lazy dog. Y position = " & SWBPrintEngine.Y &
" Line # " & I
    SWBPrintEngine.NewLine
Next I
```

```
'Set the font to a different font, and print some more text.
SWBPrintEngine.SetFont "Times New Roman", 10, SWBPRINTER_FONT_ITALIC
SWBPrintEngine.Y = SWBPrintEngine.Y + 0.25
For I = 1 To 20
    SWBPrintEngine.TextOut 0.5, "The quick brown fox jumped over the lazy dog. Y position = " & SWBPrintEngine.Y &
" Line # " & I
    SWBPrintEngine.NewLine
Next I
```

```
'Force a new page and display a box with wrapped text in it.
SWBPrintEngine.NewPage
```

```
SWBPrintEngine.BoxOut 2.5, 2.75, 7, 9
SWBPrintEngine.Y = 3
SWBPrintEngine.MemoOut 2.75, 6.75, Memo.Text

'Force a new page and display a box with a graphic image in it.
SWBPrintEngine.NewPage
SWBPrintEngine.BoxOut 2, 3, 7, 4.25
SWBPrintEngine.ImageOut App.Path & "\swbtitle.jpg", 2.1, 3.1, 4.8, 1
SWBPrintEngine.Y = 4.5
SWBPrintEngine.SetFont "Arial", 16, SWBPRINTER_FONT_BOLD
SWBPrintEngine.TextOut 3, "Sample Image Output"

'Indicate the end of the print job.
SWBPrintEngine.EndDoc

'Now preview the report immediately after generating it.
SWBPrintEngine.PreviewReport
End Sub

Public Sub SWBPrintEngine_OnNewPage(ByVal PageNum As Integer)
'Always start printing at vertical position of 2 inches on each new page.
SWBPrintEngine.Y = 2
End Sub
```

# Orientation Property

**Type:** Integer

**Access:** Read/Write

## **Description:**

Sets or returns the page orientation for the current report. This property should be set after calling the BeginDoc method. You cannot use more than one page orientation within a single document. The default value is portrait mode.

Valid values are:

|                        |   |                           |
|------------------------|---|---------------------------|
| <b>vbPRORPortrait</b>  | 1 | Prints in portrait mode.  |
| <b>vbPRORLandscape</b> | 2 | Prints in landscape mode. |

# X Property

**Type:** Double

**Access:** Read/Write

**Description:**

Sets or returns the current horizontal print position for subsequent text output. The value should be in terms of the current print job's ScaleMode.

# Y Property

**Type:** Double

**Access:** Read/Write

**Description:**

Sets or returns the current vertical print position for subsequent text output. The value should be in terms of the current print job's ScaleMode.

# Page Property

**Type:** Integer

**Access:** Read-Only

**Description:**

Returns the current page number. This value is automatically incremented when the NewPage method is used.

# Title Property

**Type:** String

**Access:** Read/Write

**Description:**

Sets or returns the title of the report used by the AutoHeadings feature. When the AutoHeadings property is True, this value is printed centered at the top of each new page.

# ShowPreviewSave Property

**Type:** Boolean

**Access:** Read/Write

**Description:**

Sets or returns the visible state of the Save button on the print preview screen. By setting this property to False prior to calling the PreviewReport method, the preview screen will not display the Save button its toolbar.

# ScaleMode Property

**Type:** Integer

**Access:** Read/Write

## **Description:**

Returns or sets a value indicating the unit of measurement for coordinates of text when positioning text on the page. This property should be set after calling the BeginDoc method. The default value is vbInches.

Valid values are:

**vbUser**

**vbTwips**

**vbPoints**

**vbPixels**

**vbCharacters**

**vbInches**

**vbMillimeters**

**vbCentimeters**

# TotalPages Property

**Type:** Integer

**Access:** Read-Only

**Description:**

Returns the total number of pages in a document after calling the EndDoc method. The value of this property is undefined prior to calling EndDoc.

# Spacing Property

**Type:** Double

**Access:** Read/Write

**Description:**

Sets or returns the vertical line spacing for the current report. This property should be set after calling the BeginDoc method. The default value is 1.3. Smaller values will decrease the line spacing, causing lines to appear closer together. Larger values will increase the line spacing, causing lines to appear farther apart.

# AutoHeadings Property

**Type:** Boolean

**Access:** Read/Write

## **Description:**

Sets or returns the use of automatic page headings. Automatic page headings can be produced whenever the NewPage method is called. Automatic page headings print the date left-justified, the report title (see Title property) centered on the page, and the page number right-justified, all on the top line of the page. The font name, size, and attributes are based on the AutoHeadingsFontName, AutoHeadingsFontSize, and AutoHeadingsFontAttr properties, respectively.

# AutoHeadingsFontName Property

**Type:** String

**Access:** Read/Write

**Description:**

Sets or returns the name of the font to be used when printing automatic headings.

# AutoHeadingsFontSize Property

**Type:** Integer

**Access:** Read/Write

**Description:**

Sets or returns the size of the font (in points) to be used when printing automatic headings.

# AutoHeadingsFontAttr Property

**Type:** Long

**Access:** Read/Write

**Description:**

Sets or returns the attributes of the font to be used when printing automatic headings. Use font attribute values specified for the SetFont method.

# AutoNewPage Property

**Type:** Boolean

**Access:** Read/Write

## **Description:**

Sets or returns the use of the automatic new page feature. When AutoNewPage is true and printing has reached the bottom margin of the page, the next TextOut call will issue the NewPage method first, then output its text. If AutoHeadings is also true, page headings will be generated for the new page.

# PrintFile Property

**Type:** String

**Access:** Read/Write

**Description:**

Returns the name of the temporary file used to store the page output during this session. This file will only be deleted when the DestroyReport method is issued.

# ShowProgress Property

**Type:** Boolean

**Access:** Read/Write

**Description:**

Sets or returns the display of a progress bar window while printing.

# SetFont Method

SetFont (ByVal Name As String, ByVal Size As Integer, ByVal Attributes As Long)

## Description:

This method sets the text font for subsequent text output using the TextOut method.

**Name**                    The name of the font.  
**Size**                    The size of the font in points.  
**Attributes**            The attributes to apply to the font.

The following values are valid and can be OR'ed together in combination:

**SWBPRINTER\_FONT\_BOLD**    1    Causes font to appear bold  
**SWBPRINTER\_FONT\_UNDERLINE** 2    Causes font to appear underlined.  
**SWBPRINTER\_FONT\_ITALIC**    4    Causes font to appear italicized.

## TextOut Method

`TextOut (ByVal XPos As Double, ByVal Text As Variant)`

### Description:

This method places text on the page at the current vertical position.

**XPos** The horizontal position where the text will begin. Value is in terms of ScaleMode  
**Text** The value to be printed.

# NewPage Method

NewPage

## **Description:**

Ejects the current page, increments the page counter by one, and prepares a new page for output. Resets the X and Y properties to zero.

# PreviewReport Method

PreviewReport

## **Description:**

Displays the print preview screen and allows user to preview the completed report. User can change zoom level, page through report, and print the report being viewed. This method should only be called after calling the EndDoc method.

# PrintReport Method

PrintReport

## **Description:**

Displays a print option window to allow user to select printer device, pages to print, and number of copies. This method should only be called after calling the EndDoc method.

# BeginDoc Method

BeginDoc

## **Description:**

Begins a new document and prepares it for output. This method should be called before setting any properties or calling any other methods.

# EndDoc Method

EndDoc

## **Description:**

Ends the current document. This method should be called before calling the PrintReport or PreviewReport methods.

# DestroyReport Method

DestroyReport

## **Description:**

Deletes the temporary print file created during the printing of the report. This method should only be called after EndDoc, and only after the user is done previewing or printing the report (using PreviewReport or PrintReport methods). Once a report is destroyed, it must be regenerated by the main application before it can be printed or previewed again.

# LineOut Method

LineOut (ByVal FromX as Double, ByVal FromY as Double, ByVal ToX as Double, ByVal ToY as Double)

## Description:

Draws a line from the coordinates FromX, FromY to the coordinates ToX, ToY. All values are in terms of ScaleMode

|              |  |
|--------------|--|
| <b>FromX</b> | The horizontal position where the line will begin. |
| <b>FromY</b> | The vertical position where the line will begin.   |
| <b>ToX</b>   | The horizontal position where the line will end.   |
| <b>ToY</b>   | The vertical position where the line will end.     |

## BoxOut Method

BoxOut (ByVal FromX as Double, ByVal FromY as Double, ByVal ToX as Double, ByVal ToY as Double)

### Description:

Draws a box from the upper-left corner coordinates FromX, FromY to the lower-right corner coordinates ToX, ToY. All values are in terms of ScaleMode.

|              |  |
|--------------|--|
| <b>FromX</b> | The horizontal position of the upper-left corner.  |
| <b>FromY</b> | The vertical position of the upper-left corner.    |
| <b>ToX</b>   | The horizontal position of the lower-right corner. |
| <b>ToY</b>   | The vertical position of the lower-right corner.   |

# MemoOut Method

MemoOut (ByVal FromX as Double, ByVal ToX as Double, Text as String)

## Description:

Displays memo-style text in a word-wrapped format starting at the current vertical position (see Y property). The text begins horizontally at the FromX position and wraps when it exceeds the ToX position. It continues to print vertically down the page between these two horizontal "margins" until all of Text is printed. All values are in terms of ScaleMode.

|              |   |
|--------------|---|
| <b>FromX</b> | The horizontal position of the left margin for the text.  |
| <b>ToX</b>   | The horizontal position of the right margin for the text. |
| <b>Text</b>  | The text to be printed.                                   |

# ImageOut Method

ImageOut (ByVal FileName as String, ByVal XPos as Double, ByVal YPos as Double, ByVal Width as Double, ByVal Height as Double)

## Description:

Displays the image contained in the file specified by FileName at the position specified by XPos and YPos (upper-left corner coordinates). The image is resized to the Height and Width specified. The image file must be at the location specified at the time the report is printed/previewed. Note that if this report is saved, the image file must be present when the report is subsequently loaded. The saved report does not contain the image data. All values are in terms of ScaleMode.

|                 |   |
|-----------------|---|
| <b>FileName</b> | The name of the image file to be printed.                 |
| <b>XPos</b>     | The horizontal position of the image's upper-left corner. |
| <b>YPos</b>     | The vertical position of the image's upper-left corner.   |
| <b>Width</b>    | The Width of the image in terms of ScaleMode.             |
| <b>Height</b>   | The Height of the image in terms of ScaleMode.            |

# SaveReport Method

`SaveReport (ByVal DestFileName As String) as Boolean`

## **Description:**

Saves the current report as a text file using the filename specified. You must use `SaveReport` after calling `EndDoc`, but before calling `DestroyReport` if you wish to save the contents of the. If the caller does not specify a file name, `SWBPrinter` prompts for one using the Save common dialog. The function returns *True* if the user selected a file name and a save was attempted. It returns *False* if the user clicked Cancel in the Save dialog.

**Note:** In some reports, `SWBPrinter` will not be able to emulate the layout of the report in the text file due to the translation from proportional-spaced to fixed-width fonts.

# OnNewPage Event

OnNewPage (ByVal PageNum as Integer)

## **Description:**

Fires once per page, specifically when the NewPage method is issued. This event is fired while the report is being generated, not during the preview or print operations. The new page number is passed to the calling application as PageNum.



