

PreviewPrint

Print and Print Preview for VB, v1.0

by Brodie Thiesfield (brodie@iname.com)

No warranty or guarantee of fitness for your use is provided with this software. Use it at your own risk. This software is freeware, however if you find it useful, it'd be great if you would send me an email and let me know.

Introduction

I was creating a database lookup program that required accurate printing and print previewing. After searching the internet for something which would provide me with this and failing to find anything satisfactory, I decided to write it myself and make it available for other peoples use. The result is PreviewPrint.

PreviewPrint is an MFC C++ OLE automation server, which accepts basic text and drawing commands to describe each page of a document. Once the document has been described fully, the print preview may be displayed and the document printed.

PreviewPrint provides basic print and print preview capability which can be used from a visual basic program. It is mainly oriented towards formatting and print previewing a text based report – the original use that I wrote it for. It does not make life easy for the programmer like the visual basic printer object, however it is quite powerful in its features for printing line drawings and text.

If more functionality is required of PreviewPrint, then I have provided the MFC C++ source code for free, and it may be modified to suit your purpose. If these modifications may be useful to other people, please email the source back to me with a short note outlining the changes and I will try to include them on this version.

Files in this package

| | |
|------------------|---|
| PreviewPrint.doc | This file, documentation for PreviewPrint |
| PreviewPrint.exe | Executable, ole automation server, distribute this with your program |
| Sample.exe | Executable, sample program using PreviewPrint. (open PreviewPrint.exe and then close it to register the ole automation interface, before running this program) |
| PreviewPrint.tlb | Type library for PreviewPrint interface (before opening the sample source code in VB, load this type library as a reference object interface) |
| Source*.* | C++ / MFC source files for PreviewPrint.exe |
| Sample*.* | VB5 source files for Sample.exe |

Using PreviewPrint

To use PreviewPrint:

1. copy the files previewprint.exe and previewprint.tlb into your program's directory
2. run the executable to register the ole automation server
3. go to Project->References and add a reference to PreviewPrint for the project. If PreviewPrint doesn't show up in the list, then click Browse and select the PreviewPrint.tlb file

A summary of the interfaces to PreviewPrint is included following. If you have access to the MFC help, then nearly all of the text and line output methods following are just a public duplication of the methods available on the MFC class CDC. Look up the online reference of this class for a more complete description of how they work.

Some points to note when using PreviewPrint:

- all coordinates used, supplied and returned in PreviewPrint are specified in tenths of millimeters, eg 254 = 25.4 mm = 2.54 cm = 1 inch.
- all coordinates originate from (0,0) in the top, left corner of the page. Positive coordinate directions are right and down.
- all text sizes are specified in tenths of points, not in tenths of millimeters.
- the tenths of millimeter size of a particular point size font may be found by setting the font using SetFont and calling GetTextExtent.

Properties

Property: Orientation

Type: short

Description: Get or set the orientation of the paper.

Valid values: PPO_PORTRAIT = 1, PPO_LANDSCAPE = 2

Property: PageWidth, PageHeight

Type: short

Description: Get the page width and height

Valid values: positive numeric

Property: hwndOwner

Type: long

Description: Set or get the hwnd of the owner of the print preview window. This should be set before displaying the print setup dialog, print dialog or print preview window.

Valid values: hwnd

Property: PreviewPrintState

Type: short

Description: Get the current state of the print preview window. IDLE = no window displayed. PREVIEW = print preview window is being displayed. PRINT = print window is being displayed.

Valid values: PPS_IDLE = 0, PPS_PREVIEW = 1, PPS_PRINT = 2

Property: StartPage

Type: short

Description: Set or get the page number of first page. This is the number which will be used in the print dialog box where the page range may be set, eg from page 4 to 7.

Valid values: positive numeric

Document Management Methods

Method: BeginDoc

Description: Call this method to begin a new document. This must be called before calling any drawing methods. Any existing document will be destroyed and a new document started when this method is called.

Parameters: none

Returns: true = success, false = failure

Method: NewPage

Description: Call this method to save the current page as final and start a new blank page in the document.

Parameters: none

Returns: true = success, false = failure

Method: EndDoc

Description: Call this method to complete a document. Only after this method has been called can the document be printed or print previewed.

Parameters: none

Returns: true = success, false = failure

Method: KillDoc

Description: Call this method to dispose of the saved document data. This will release any memory that is currently being used by this document. The document doesn't exist any longer after this method is called and so cannot be printed or print previewed.

Parameters: none

Returns: nothing

Method: SetTitle

Description: Set the title of the document and the name of the application which is printing / print previewing for display in the titlebar during print preview.

Parameters: lpszApp (string), lpszDoc (string)

Returns: nothing

Method: ShowPreview

Description: Show the print preview window for the current document. The document must have been created and drawn already and the EndDoc method must have been called.

Parameters: none

Returns: true = success, false = failure

Method: ShowPrintSetup

Description: Show the print setup dialog to allow any printer settings to be changed, or a different printer to be selected. This dialog box must be used instead of any of the printer functions under VB.

Parameters: none

Returns: true = success, false = failure

Line Output Methods

Method: GetPos

Description: Return the current position of the pen on the current document page

Parameters: x (short), y (short)

Returns: true = success, false = failure

Method: MoveTo

Description: Move the current position of the pen to a new position on the current document page without drawing any lines or text

Parameters: x (short), y (short)

Returns: nothing

Method: SetLineStyle

Description: Set the line style to be used for all lines that are drawn subsequently. Setting the line style will not affect any lines that have already been drawn on the page.

Parameters: nStyle (short): PPL_SOLID = 0, PPL_DASH = 1 (- - - -), PPL_DOT = 2 (. . . .), PPL_DASHDOT = 3 (_ . _ . _), PPL_DASHDOTDOT = 4 (_ . . _ . . _)

nWidth (short): 0 = minimum width possible, 1 or more = width in tenths of millimeters
crColor (long): RGB color (use the RGB() function), or predefined color eg vbRed

Returns: true = success, false = failure

Method: LineTo

Description: Draw a line using the current line style, width and color from the current location (see GetPos) to the specified coordinates.

Parameters: x (short), y (short)

Returns: true = success, false = failure

Text Methods

Method: SetFont

Description: Set the text style for all text to be subsequently drawn. This will not affect the style of any text that has already been drawn.

Parameters: lpszFaceName (string): name of the font that you wish to use. This should be the exact name of the font face, for example, the name returned by a common font dialog.

nSize (short): size of the text in tenths of a point eg 80 = 8.0pt, 240 = 24.0pt. Note that these measurements are tenths of a point, not tenths of a millimeter.

nOptions (short): text options: PPF_BOLD = 1, PPF_ITALIC = 2, PPF_UNDERLINE = 4, PPF_STRIKEOUT = 8

nAngle (short): angle of the text compared to the horizontal baseline. This angle should be supplied in degrees.

Returns: true = success, false = failure

Method: TextOut

Description: Basic text output. Writes a character string at a reference location using the currently selected font. How the text is output in relation to the reference point (x, y) depends upon the setting of the text alignment using SetTextAlign.

Parameters: x (short), y (short): reference location for text string output

szText (string): text string to output

Returns: true = success, false = failure

Method: TextOutClip

Description: Advanced text output. Writes a character string within a rectangular region using the currently selected font. How the text is output in relation to the reference point (x, y) depends upon the setting of the text alignment using SetTextAlign. If the clip rectangle supplied (nClipLeft, nClipTop, nClipRight, nClipBottom) is not large enough to hold the text then any text outside that rectangle will be clipped.

Parameters: x (short), y (short): reference location for text string output

szText (string): text string to output

nClipLeft (short), nClipTop (short): absolute location of the clip rectangles top left point

nClipRight (short), nClipBottom (short): absolute location of the clip rectangles bottom right point

Returns: true = success, false = failure

Method: SetTextAlign

Description: Set the alignment of the text in relation to the reference point.

Note: It is recommended to use TA_NOUPDATECP all of the time so that the current position isn't changed by text output. It is unknown what affect using TextOut or TextOutClip has on the current position.

Parameters: nOptions (short): a combination of the following possibilities:

- one of: PPA_NOUPDATECP = 0, PPA_UPDATECP = 1
- plus, one of: PPA_LEFT = 0, PPA_CENTER = 6, PPA_RIGHT = 2
- plus, one of: PPA_TOP = 0, PPA_BASELINE = 24, PPA_BOTTOM = 8

These options should be added together, for example, to align the text with the baseline centered on the reference point, use SetTextAlign (PPA_NOUPDATECP + PPA_CENTER + PPA_BASELINE);

Returns: The previous text-alignment setting.

Method: GetTextAlign

Description: Returns the current text alignment settings.

Parameters: none

Returns: The current text-alignment setting.

Method: SetBkColor

Description: Sets the current background color.

Parameters: crColor (long): new background color, use a VB predefined color, eg vbRed or the RGB() function to set the color, eg RGB(0, 128, 64).

Returns: The previous background color.

Method: GetBkColor

Description: Gets the current background color.

Parameters: none

Returns: The current background color.

Method: SetTextColor

Description: Sets the text face color.

Parameters: crColor (long): new text face color, use a VB predefined color, eg vbRed or the RGB() function to set the color, eg RGB(0, 128, 64).

Returns: The previous text face color.

Method: GetTextColor

Description: Gets the current text face color.

Parameters: none

Returns: The current text face color.

Method: GetTextExtent

Description: Returns the dimensions of the text string given the current font.

Parameters: lpszText (string): string to test the dimensions of using the current font

nWidth (short), nHeight (short): dimensions of this text in tenths of millimeters

Returns: true = success, false = failure.