Adding Watermarks to PDF Documents with PeopleCode

PT 8.58
Randall Groncki

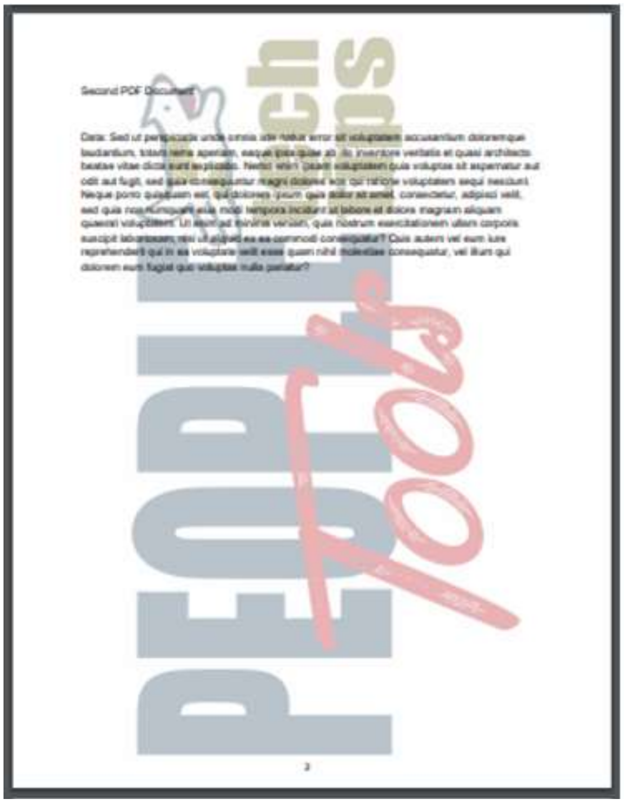
Introduction

A watermark is text or images on a document, usually feint. They convey ideas such as status, branding, quality, or ownership of the document rather than contribute to the document's specific content.

PeopleCode enables watermarking PDF documents through the delivered PSXP_ENGINE:PDFMerger class. One of that class's properties is the PSXP_ENGINE:Watermark class. By manipulating a few properties of this Watermark class, we can add a text or image watermark to a PDF document.

Why add a watermark on a PeopleSoft produced PDF?

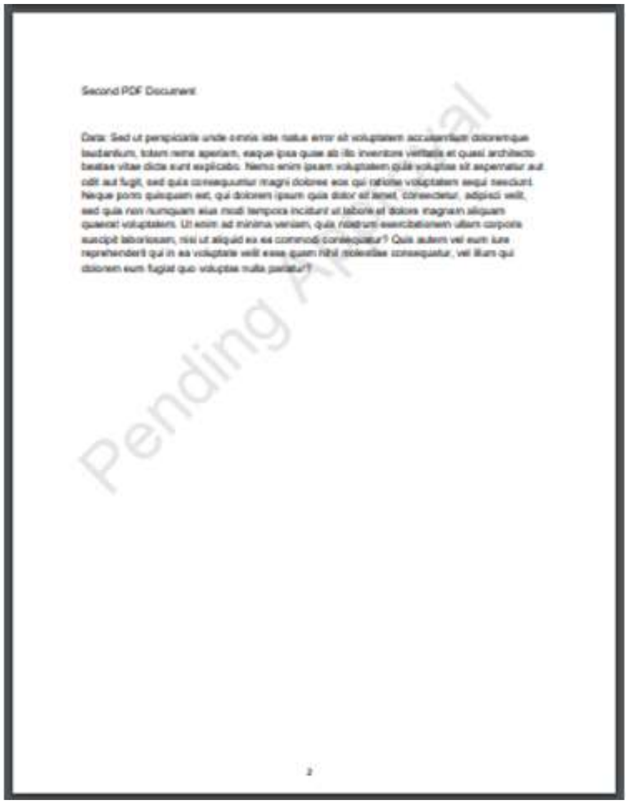
- The requirements specify a watermark
- Transaction status



Second PDF Document

Data: Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

2



Second PDF Document

Data: Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

2

Watermarking

PDFMerger Basics

PeopleCode uses the delivered PSXP_ENGINE:PDFMerger to manipulate PDF documents after they have been created.   This class can:

- Combine one or more PDF documents into one PDF document:
- Add Page Numbers
- Add Watermarks
- Change the properties of the new PDF document.

An important to note, the merger class does not change the original documents, but produces a new combined document with these changes.   The merger class also works on a single document rather than requiring two or more documents to merge.

Here is the basic syntax of merging two PDF documents into new combined PDF document.

```
import PSXP_ENGINE:*;

&oMerger = create PSXP_ENGINE:PDFMerger();

&PDF_Merge_Error = "";
&PDF_Merge_Array = CreateArray(&FirstReportFilePath);
&PDF_Merge_Array.Push(&FirstReportFilePath;
&PDF_Merge_Array.Push(&SecondReportFilePath;

&PDF_Merge_Result = &oMerger.mergePDFs(&PDF_Merge_Array, &PDF_New_Merge_File_name,
&PDF_Merge_Error);
```

Watermark Basics

The PDFMerger class contains a watermark property, which is an object of type PSXP_ENGINE:Watermark. At design time, you must choose between a text or image watermark. The system will not generate both on a document. If both are defined, then the image watermark settings will override the text watermark settings.

Watermarks do work in conjunction with the PageNumber watermark object.   Any configuration of one does not affect the other.

Text Water Marks

Text watermarks take a text string and place it on the page as a layer underneath the content of the PDF document. Any control characters such as linefeeds, carriage returns and page feeds are stripped out of the string before adding it to the document.

* see Appendix A for a more detailed code listing

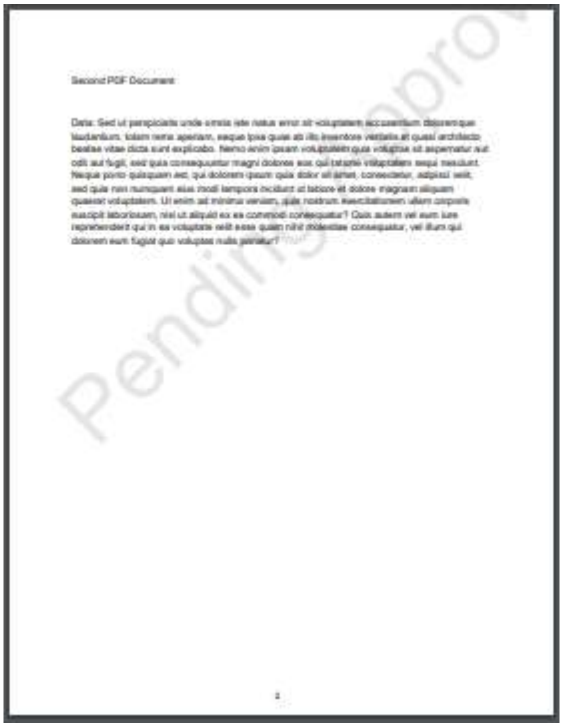
```
import PSXP_ENGINE:*;

Local PSXP_ENGINE:PDFMerger &oMerger = create PSXP_ENGINE:PDFMerger();
Local PSXP_ENGINE:Watermark &oWatermark = create PSXP_ENGINE:Watermark();

/* set the water mark, location and angle */
&oWatermark.Text = "PeopleTools Tech Tips";
&oWatermark.TextStartPosX = 150;
&oWatermark.TextStartPosY = 200;
&oWatermark.TextAngle = 45;

/* which page for the watermark?  0=all */
&oWatermark.PageIndex = 0;

&oMerger.Watermark = &oWatermark;
```

Properties

Text	The text to print on the document as the watermark. Control characters are stripped out by the PDFMerger engine. Take note of the first character in that string since this is what the X/Y Axis properties are placing on the page.
TextStartPosX	Default = 0 The X-Axis on the page from left to right for the first character of the printed string.
TextStartPosY	Default = 0 The Y-Axis on the page from bottom to top for the first character of the printed string.
TextAngle	Default = 0 The angle of the text appearing on the page. Angle = 0 is the normal, left to right text on a page. Angle = 90 is bottom to top printing on the page
TextFontName	Default = Helvetica A limited number of fonts available for the text watermark. The default is Helvetica. * As a note, there is a bug as of the 8.58.08 distribution where the chosen font only appears on the first page of the document. All other pages default to the Helvetica font.
TextFontSize	Font size of your text watermark.
PageIndex	Default = 0 (All Pages) This property which page to place the watermark on the merged document. The Default 0 places the watermark on all pages of the document.

Usage

You will have to experiment with the properties to place the watermark in the location and orientation required.



Second PDF Document

Data: Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

The watermark is not constrained by the page. Should the text string be too large in any dimension for the page, only the portion of the text string on that page will display.

You will also have to adjust to the X and Y axis on the page depending on the Text Angle chosen. Should you choose a Text angle of 225, then your XY coordinates must be on the upper right of the page so that the text can land at a downward, left angle across the page.

The XY Coordinates are not the bottom left of where the displayed result string ultimately appears on the page. It is the bottom left of the first character of the string where ever that may start.

Image Watermarks

An image watermark places an image as a layer underneath the content of the document.

* see Appendix B for a more detailed code listing.

```
import PSXP_ENGINE:*;

Local PSXP_ENGINE:PDFMerger &oMerger = create PSXP_ENGINE:PDFMerger();
Local PSXP_ENGINE:Watermark &oWatermark = create PSXP_ENGINE:Watermark();

/* Watermark Image File URL (This example: 200x200 pixel image)*/
&oWatermark.ImageFile = &ServerImageFileURL;

/* lower left Coordinates  */
&oWatermark.ImageLowerLeftX = 50;
&oWatermark.ImageLowerLeftY = 20;

/* upper right Coordinates */
&oWatermark.ImageUpperRightX = 250;
&oWatermark.ImageUpperRightY = 220;

/* which page for the watermark?  0=all */
&oWatermark.PageIndex = 0;

&oMerger.Watermark = &oWatermark;
```

Properties

ImageFile	The URL (file server location) of the image file to use as the watermark
ImageLowerLeftX	Default = 0 The X-Axis on the page from left to right.
ImageLowerLeftY	Default = 0 The Y-Axis on the page from bottom to top.
ImageUpperRightX	Default = image will display in the true width of the image The X-Axis on the page from left to right.
ImageUpperRightY	Default = image will display in the true height of the image The Y-Axis on the page from bottom to top.
PageIndex	Default = 0 (All Pages) This property which page to place the watermark on the merged document. The Default 0 places the watermark on all pages of the document.

Usage

You will have to experiment with the properties to place the image watermark in the proper location. The Image will visually skew if you specify a top-right X/Y position that is not in proportion to the original image.

The watermark is not constrained by the page. Should the image be too large in any dimension for the page, only the portion of the image on that page will display.

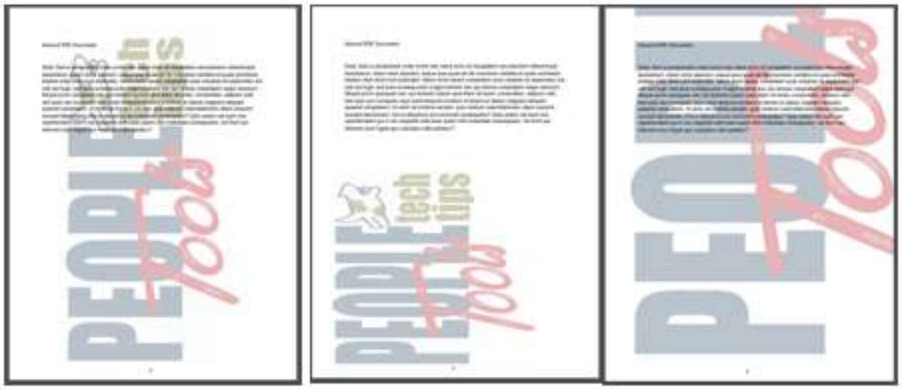


Image File Accessibility

The ImageFile property is a reference to the image file location on a file server, not an image file object itself. You must consider the architecture of your PeopleSoft implementation when designing your solution. This code can be called from any Application or Report server. Can all Application and Report servers reference the location of this image file?

If the App and Report servers do not use common file locations, you may want to place the image file on the executing machine at runtime to ensure the PDFMerger object can locate it.

Image File Opacity

Remember that this image is being placed as a layer underneath the contents of the PDF. How opaque the image is… matters. Placing a dark, busy image underneath the document text may obscure the document making it difficult or impossible to read.

Most graphics programs allow the designer to adjust the opacity and saturation of the image. The image in the examples above was modified to a 33% opacity and, most likely, still too dark. Adjust the image so the document can be comfortably consumed by the user.

Image Orientation

The Watermark class does not have a way to rotate or manipulate the image in any way other than specifying the dimensions for its placement on the page. Use graphics programs for any image manipulation before handing that image to the Watermark class.

Documentation

PeopleTools PeopleBooks Watermarks

As delivered PeopleCode objects, the Watermark class is documented in the PeopleTools PeopleBooks PeopleCode API Reference section. Look for the BI Publisher Classes, then scroll down to the Watermark Class and Properties.

*Note, the Watermark Class Properties section has an error as of the writing of this document.

Correct Property Names:

- ImageLowerLeftX
- ImageLowerLeftY
- ImageUpperRightX
- ImageUpperRightY

Wrong Property Names in PeopleBooks:

- Image**File**LowerLeftX
- Image**File**LowerLeftY
- Image**File**UpperRightX
- Image**File**UpperRightY

What is the difference? Notice the lack of "**File**" in the correct property names.

Appendix A

Text Watermark Code Example

```
/*  Merge Documents */
/* create the objects */
Local PSXP_ENGINE:PDFMerger &oMerger = create PSXP_ENGINE:PDFMerger();
Local PSXP_ENGINE:PageNumber &oPageNumber = create PSXP_ENGINE:PageNumber();
Local PSXP_ENGINE:Watermark &oWatermark = create PSXP_ENGINE:Watermark();

/* load the pdfs to be merged into an array */
&PDF_Merge_Array.Push(&BI_Pub_Generator.First_PDF_URL);
&PDF_Merge_Array.Push(&BI_Pub_Generator.Second_PDF_URL);
&PDF_Merge_Array.Push(&BI_Pub_Generator.First_PDF_URL);
&PDF_Merge_Array.Push(&BI_Pub_Generator.Second_PDF_URL);

/* define result file location and name */
&PDF_Merge_File = GetFile("PDF_MERGE.pdf", "W");
&PDF_Merge_File_name = &PDF_Merge_File.Name;
&PDF_Merge_File.Close();

/* set the page number */
&oPageNumber.FontName = "Helvetica";
&oPageNumber.FontSize = 10;
&oPageNumber.PositionX = 300;
&oPageNumber.PositionY = 20;
&oPageNumber.StartNum = 1;
&oMerger.PageNumber = &oPageNumber;

   /* which page for the watermark?  0=all */
&oWatermark.PageIndex = 0;

/* set the Text water mark, location and angle */
&oWatermark.Text = &Rec_X_PDF_WMRK_WRK.DESCR.Value;

/* Font - these are the same as from the Page Number class in PeopleBooks */
rem &oWatermark.TextFontName = "Courier";
rem &oWatermark.TextFontName = "Courier-Bold";
rem  &oWatermark.TextFontName = "Courier-BoldOblique";
rem &oWatermark.TextFontName = "Helvetica";
&oWatermark.TextFontName = "Helvetica-Bold";
rem &oWatermark.TextFontName = "Helvetica-BoldOblique";
rem &oWatermark.TextFontName = "Helvetica-Oblique";
rem &oWatermark.TextFontName = "Symbol";
rem &oWatermark.TextFontName = "Times-Bold";
rem &oWatermark.TextFontName = "Time-BoldItalic";
rem &oWatermark.TextFontName = "Time-Italic";
rem &oWatermark.TextFontName = "Time-Roman";
rem &oWatermark.TextFontName = "ZapfDingbats";

&Returncode = ViewAttachment("record://X_FILEATTACH", "PDF_MERGE.pdf", "PDF_MERGE.pdf");
```

Text Watermark Code Example Continued…

```
   /* starting position on the page for the watermark string */
   &oWatermark.TextStartPosX = &Rec_X_PDF_WMRK_WRK.X_BOTTOM_X.Value;
   &oWatermark.TextStartPosY = &Rec_X_PDF_WMRK_WRK.X_BOTTOM_Y.Value;

   /* Agle:  0 = Standard left to right horizontal printing */
   &oWatermark.TextAngle = &Rec_X_PDF_WMRK_WRK.PTPG_3DROT_ANGLE.Value;

   If All(&Rec_X_PDF_WMRK_WRK.FONTSIZE.Value) Then
      &oWatermark.TextFontSize = &Rec_X_PDF_WMRK_WRK.FONTSIZE.Value;
   End-If;

   &PDF_Merge_Error = "";
   &PDF_Merge_Result = &oMerger.mergePDFs(&PDF_Merge_Array, &PDF_Merge_File_name,
 &PDF_Merge_Error);

   /* use the attachment functions to pop the new merged document to a new window */
   &Returncode = PutAttachment("record://X_FILEATTACH", "PDF_MERGE.pdf", &PDF_Merge_File_name);
   &Returncode = ViewAttachment("record://X_FILEATTACH", "PDF_MERGE.pdf", "PDF_MERGE.pdf");
```

Appendix B

Image Watermark Code Example

```
/*  Merge Documents */
 /* create the objects */
 Local PSXP_ENGINE:PDFMerger &oMerger = create PSXP_ENGINE:PDFMerger();
 Local PSXP_ENGINE:PageNumber &oPageNumber = create PSXP_ENGINE:PageNumber();
 Local PSXP_ENGINE:Watermark &oWatermark = create PSXP_ENGINE:Watermark();

 /* load the pdfs to be merged into an array */
 &PDF_Merge_Array.Push(&BI_Pub_Generator.First_PDF_URL);
 &PDF_Merge_Array.Push(&BI_Pub_Generator.Second_PDF_URL);
 &PDF_Merge_Array.Push(&BI_Pub_Generator.First_PDF_URL);
 &PDF_Merge_Array.Push(&BI_Pub_Generator.Second_PDF_URL);

 /* define result file location and name */
 &PDF_Merge_File = GetFile("PDF_MERGE.pdf", "W");
 &PDF_Merge_File_name = &PDF_Merge_File.Name;
 &PDF_Merge_File.Close();

 /* set the page number */
 &oPageNumber.FontName = "Helvetica";
 &oPageNumber.FontSize = 10;
 &oPageNumber.PositionX = 300;
 &oPageNumber.PositionY = 20;
 &oPageNumber.StartNum = 1;
 &oMerger.PageNumber = &oPageNumber;

  /* Get the image name from a Rowset on the page */
   &Str_Image_Name = &Rs_X_PT3_WMRK_IMG(&i).X_PT3_WMRK_IMG.PHOTO_IMGNAME.Value;

   If All(&Str_Image_Name)  Then

     /* Place the image file on the server, reachable by this code event */
      &ServerImage = create X_PT3_PDF_WATERMARK:Set_Server_Image(&Str_Image_Name);

      /* which page for the watermark?  0=all */
      &oWatermark.PageIndex = 0;

      /* Watermark Image File URL */
      &oWatermark.ImageFile = &ServerImage.GetImageURL();

      /* lower left Coordinates  */
      &oWatermark.ImageLowerLeftX = &Rec_X_PDF_WMRK_WRK.X_BOTTOM_X.Value;
      &oWatermark.ImageLowerLeftY = &Rec_X_PDF_WMRK_WRK.X_BOTTOM_Y.Value;

      /* upper right Coordinates */
      &oWatermark.ImageUpperRightX = &Rec_X_PDF_WMRK_WRK.X_TOP_X.Value;
      &oWatermark.ImageUpperRightY = &Rec_X_PDF_WMRK_WRK.X_TOP_Y.Value;

      &oMerger.Watermark = &oWatermark;

   End-If;
```

Image Watermark Code Example Continued…

```
   &PDF_Merge_Error = "";

   &PDF_Merge_Result = &oMerger.mergePDFs(&PDF_Merge_Array, &PDF_Merge_File_name,
&PDF_Merge_Error);

   /* use the attachment functions to pop the new merged document to a new window */
   &Returncode = PutAttachment("record://X_FILEATTACH", "PDF_MERGE.pdf", &PDF_Merge_File_name);
   &Returncode = ViewAttachment("record://X_FILEATTACH", "PDF_MERGE.pdf", "PDF_MERGE.pdf");
```