PDF Meld by FyTek, Inc.

PDF Meld

◆ FyTek Incorporated ◆

Web site: http://www.fytek.com

Introduction

PDF Meld is a program to manipulate pages of existing PDFs. There are three basic functions the program performs:

- Append two or more PDFs into a single PDF
- Extract a page or list of pages from a single PDF
- Overlay pages of one PDF onto the pages of a second PDF

The source PDFs in all cases above must be unencrypted. The page text must be either uncompressed or flate (zlib) compressed, common for most PDFs, when overlaying pages. PDFs with form fields and/or PDFs with revisions may not work in all situations.

Bookmarks are retained for each input PDF unless the option to remove them is used. The bookmark structure for each input PDF is placed under a new heading named after the input file itself.

The next sections describe the options for the executable version of the program and the methods of the DLL version.

Using the Executable

The program pdfmeld.exe is the Windows executable program. The general syntax is:
`pdfmeld.exe` *`filein1.pdf[,filein2.pdf...] fileout.pdf`* `[options]`

The above syntax without any options will merge two or more PDFs into a single PDF. fileout.pdf is the output PDF in the example above. There are several ways the input files may be specified:

- Use a comma separated list of file names (no space before/after the comma)
  For example, "pdfmeld.exe file1.pdf,file2.pdf,file3.pdf fileout.pdf"

- Specify a directory rather than a file to process all PDFs in the directory
  For example, "pdfmeld.exe c:\pdfiles\ fileout.pdf"

- Use an '@' in front of a file name containing a list of files to process.
  The file should contain a list of PDF files, one file per line in the file. For example, "pdfmeld.exe @mylist.dat fileout.pdf"

Use the -pages option to pull pages from an existing PDF. Only use one input file in this case (or two if using the -overlay option). Separate page numbers with a comma or use a - between numbers for a range. For example, "pdfmeld.exe filein.pdf fileout.pdf -pages 2,7,14-20,35" will extract pages 2, 7, 14 through 20 and page 35, a total of 10 pages, from filein.pdf and place them in fileout.pdf. The -pages option, when used along with the -overlay or -repeat options, applies only to the second input PDF.

Use the -overlay option to overlay pages from one PDF onto a second PDF. Only use two input files in this case. The first PDF specified is the background and the second will be placed on top of the first. The resulting PDF will have the same number of pages as the second PDF. The first PDF should have the same or fewer pages than the second PDF. Both PDFs should have the same page sizes. For example, "pdfmeld.exe filein1.pdf,filein2.pdf fileout.pdf -overlay" will take page 1 of filein1.pdf and place page 1 of filein2.pdf on top and this becomes page 1 of fileout.pdf. The same process is applied to the remaining pages of the two input PDFs. You can overlay more PDFs by re-running the overlay process using the output PDF as an input PDF along with a new input PDF if necessary.

The -repeat option can be used to overlay PDFs as well. This option will repeat through the first PDF, using it as a background, for as many pages as

there are in the second PDF. The resulting PDF will have the same number of pages as the second PDF. The first PDF should have the same or fewer pages than the second PDF. Both PDFs should have the same page sizes. The resulting PDF from using a background PDF with 1 page and a foreground PDF with 5 pages will be 5 pages and contain the single page background PDF on each page with the second PDF overtop.

-pages *list*	A range of pages to pull from the input PDF. Use a comma separated list with no spaces before or after the comma. A dash may be used to separate a range of numbers. For example, -pages 2,7,14-20,35.
-exclude	Specifies the values entered for -pages are the pages to exclude rather than include.
-overlay	Used to specify overlay mode rather than append mode. This option places the contents of each page from one PDF on top of the corresponding page from a second PDF. The first input PDF is the background and the second input PDF is placed over top.
-repeat	Used to specify overlay mode rather than append mode and repeat the first PDF. This option places the contents of each page from one PDF on top of the corresponding page from a second PDF. The first input PDF is the background and the second input PDF is placed over top. The background is repeated, if necessary, for as many pages as there are in the second PDF.
-nobm	Turns off pulling of bookmarks from source PDFs. This is off by default if -pages or -overlay is used.

Options

-noimg	Specifies that images should be removed from the PDF(s). Line drawn images are not removed - only images that came from an external source such as a jpeg, tiff, bmp, etc.
-open	Automatically opens Acrobat and loads the newly created PDF.
-print	Automatically prints the newly created PDF to the default printer. Must have Acrobat or Acrobat Reader installed.
-mail	Opens the user's e-mail program to a composition window with the newly created PDF attached. May not work with all e-mail programs.
-force	Turns off the prompt to overwrite the output file if it already exists.
-s	Include subdirectories when the input is a directory rather than individual files.
-pagenum	Adds page numbers to each page. Default position is bottom left hand corner.
-pagelmargin *number*	The amount of space from the left edge of the page for the page number. Each unit is 1/72 of an inch so a number of 144 means 2 inches from the left edge of the page.
-pagebmargin *number*	The amount of space from the bottom edge of the page for the page number. Each unit is 1/72 of an inch so a number of 144 means 2 inches from the bottom edge of the page. This value is used to mean from the top of the page if -pagetop is specified.

Options

-pagetop	Place the page numbers at the top of the page rather than the bottom. Note that -pagebmargin then specifies space from the top rather than bottom edge of the page.
-fontsize *number*	Point size of the font for the page number. Default is 10.
-pagefmt *text*	Format string for the page number. Use %1 for the current page and %2 for the total number of pages. For example, you could use "Page %1 of %2" to have "Page 1 of 3" print on the first page, "Page 2 of 3" on the second and so on. The default string is "%1/%2".
-title *title*	Sets the document title.
-subject *subject*	Sets the document subject.
-keywords *keywords*	Sets the document keywords.
-author *author*	Sets the document author.
-creator *creator*	Sets the document creator.
-producer *producer*	Sets the document producer.
-creationdate *YYYYMMDDHHmmSS* or *today*	Sets the document creation date. Provide a date in the format shown or use the word today to use the current system date/time. For example, to set to January 15th, 2002 at 4:15:30 PM, enter 20020115161530.
-moddate *YYYYMMDDHHmmSS* or *today*	Sets the document modification date. Provide a date in the format shown or use the word today to use the current system date/time.
-e *path-file*	Sets the error/result file. Use this option to check if any errors were encountered. This file will contain the word OK if the new PDF was created.

Using the DLL (Dynamic Link Library)

The file pdfmeld.dll is the dynamic link library. This file should reside in your Windows or Winnt directory under the system32 sub-directory. You first must register the DLL on your system (note this step happens automatically when you run the setup program). Do this by running

regsvr32 pdfmeld.dll

You should see a message box that reads:

DllRegisterServer in pdfmeld.dll succeeded.

Click OK to continue. You are now ready to use the DLL.

The program will append multiple PDFs together into a single PDF by default. The setPages, setOverlay and setRepeat modify this behavior.

Use the setPages method to pull pages from an existing PDF. Only use one input file in this case (or two if using the setOverlay method). Separate page numbers with a comma or use a - between numbers for a range. For example, setPages "2,7,14-20,35" will extract pages 2, 7, 14 through 20 and page 35, a total of 10 pages, from the input PDF. The setPages method, when used along with the setOverlay or setRepeat methods, applies only to the second input PDF.

Use the setOverlay method to overlay pages from one PDF onto a second PDF. Only use two input files in this case. The first PDF specified is the background and the second will be placed on top of the first. The resulting PDF will have the same number of pages as the second PDF. The first PDF should have the same or fewer pages than the second PDF. Both PDFs should have the same page sizes. You can overlay more PDFs by re-running the overlay process using the output PDF as an input PDF along with a new input PDF if necessary.

The setRepeat method can be used to overlay PDFs as well. This method will repeat through the first PDF, using it as a background, for as many pages as there are in the second PDF. The resulting PDF will have the same number of pages as the second PDF. The first PDF should have the same or fewer pages than the second PDF. Both PDFs should have the same page sizes. The resulting PDF from using a background PDF with 1 page and a foreground PDF with 5 pages will be 5 pages and contain the single page background PDF on each page with the second PDF overtop.

DLL Methods

setInFile(path-file)	There are several ways the input files may be specified: • Use a comma separated list of file names (no space before/after the comma) For example, setInFile ("file1.pdf,file2.pdf,file3.pdf") • Specify a directory rather than a file to process all PDFs in the directory For example, setInFile ("c:\pdfiles\") • Use an '@' in front of a file name containing a list of files to process. The file should contain a list of PDF files, one file per line in the file. For example, setInFile ("@mylist.dat")
setOutFile(path-file)	Full path and name of the output PDF file.
pdfMeld	Call this method to build the PDF. The return values are: 0 = Output PDF was built successfully -1 = Can't open input file -2 = Can't open output file
setPages(pagelist)	A range of pages to pull from the input PDF. Use a comma separated list with no spaces before or after the comma. A dash may be used to separate a range of numbers. For example, setPages ("2,7,14-20,35").
setPagesExclude	Specifies the values entered for setPages are the pages to exclude rather than include.

setOverlay	Used to specify overlay mode rather than append mode. This method places the contents of each page from one PDF on top of the corresponding page from a second PDF. The first input PDF is the background and the second input PDF is placed over top.
setRepeat	Used to specify overlay mode rather than append mode and repeat the first PDF. This method places the contents of each page from one PDF on top of the corresponding page from a second PDF. The first input PDF is the background and the second input PDF is placed over top. The background is repeated, if necessary, for as many pages as there are in the second PDF.
setNoBookmarks	Turns off pulling of bookmarks from source PDFs. This is off by default if setPages or setOverlay is used.
setNoImages	Specifies that images should be removed from the PDF(s). Line drawn images are not removed - only images that came from an external source such as a jpeg, tiff, bmp, etc.
setOpen	Automatically opens Acrobat and loads the newly created PDF.
setPrint	Automatically prints the newly created PDF to the default printer. Must have Acrobat or Acrobat Reader installed.
setMail	Opens the user's e-mail program to a composition window with the newly created PDF attached. May not work with all e-mail programs.

setSubDir	Include subdirectories when the input is a directory rather than individual files.
setPageNum	Adds page numbers to each page. Default position is bottom left hand corner.
setPageLMargin(number)	The amount of space from the left edge of the page for the page number. Each unit is 1/72 of an inch so a number of 144 means 2 inches from the left edge of the page.
setPageBMargin(number)	The amount of space from the bottom edge of the page for the page number. Each unit is 1/72 of an inch so a number of 144 means 2 inches from the bottom edge of the page. This value is used to mean from the top of the page if setPageTop is specified.
setPageTop	Place the page numbers at the top of the page rather than the bottom. Note that setPageLMargin then specifies space from the top rather than bottom edge of the page.
setFontSize(number)	Point size of the font for the page number. Default is 10.
setPageFmt(string)	Format string for the page number. Use %1 for the current page and %2 for the total number of pages. For example, you could use "Page %1 of %2" to have "Page 1 of 3" print on the first page, "Page 2 of 3" on the second and so on. The default string is "%1/%2".
setTitle(title)	Sets the document title.
setSubject(subject)	Sets the document subject.
setAuthor(author)	Sets the document author.
setCreator(creator)	Sets the document creator.
setProducer(producer)	Sets the document producer.

setCreationDate Year, Month, Day, Hour, Minute, Second	Sets the document creation date. Provide the values or just pass in 0 to set to current system date/time. The only required value is Year. So, for instance, you may specify Year, Month, Day and leave the rest off.
setModDate Year, Month, Day, Hour, Minute, Second	Sets the document modification date. Provide the values or just pass in 0 to set to current system date/time. The only required value is Year. So, for instance, you may specify Year, Month, Day and leave the rest off.
setErrFile(path-file)	Sets the error/result file. Use this method to check if any errors were encountered. This file will contain the word OK if the new PDF was created.

DLL Examples

Here is an example of calling the DLL using Visual Basic.

```
Set PDF = CreateObject("pdf.Meld")
PDF.setInFile "c:\temp\filein1.pdf,d:\mypdfs\filein2.pdf"
PDF.setOutFile "c:\temp\fileout.pdf"
PDF.setPageNum
PDF.setPageTop
PDF.setPageLMargin (144)
PDF.setPageBMargin (72)
PDF.setPageFmt ("Page %1 of %2")
PDF.setCreationDate 2002, 2, 15
PDF.setModDate 0
rslt = PDF.pdfMeld
If rslt <> 0 Then
  MsgBox ("Error " & rslt)
End If
Set PDF = Nothing
```

Here is an example of calling the DLL using PowerBuilder.

```
OLEObject PDF
PDF = CREATE OLEObject
li_rc = PDF.ConnectToNewObject("pdf.Meld")
PDF.setInFile "c:\temp\filein1.pdf,d:\mypdfs\filein2.pdf"
PDF.setOutFile "c:\temp\fileout.pdf"
PDF.setPageNum
PDF.setPageTop
PDF.setPageLMargin 144
PDF.setPageBMargin 72
PDF.setPageFmt "Page %1 of %2"
PDF.setOverlay
PDF.pdfMeld
```

Here is an ASP example creating the PDF and redirecting the browser.

```
<%
Dim PDF
Set PDF = Server.CreateObject("pdf.Meld")
PDF.setInFile ("c:\temp\filein1.pdf,d:\mypdfs\filein2.pdf")
PDF.setOutFile ("c:\inetpub\webpub\output\fileout.pdf")
PDF.setPageNum
PDF.setPageTop
PDF.setPageLMargin (144)
PDF.setPageBMargin (72)
PDF.setPageFmt ("Page %1 of %2")
PDF.pdfMeld
Response.redirect("output/fileout.pdf")
set PDF = nothing
%>
```

DLL Examples

Here is an example using C.

```
#include <iostream.h>

// The import directive reads the typelib information from the DLL
// and creates pdfmeld.tlh and pdfmeld.tli, which are included.
// These define wrappers for each of the pdfmeld object methods.

#import <pdfmeld.dll>

// Using VC++ 5.0 Smart Pointers makes this much easier.
// The parameter string for a method is converted to Unicode, allocated
// and passed as a variant. The wrappers call IDispatch::Invoke
// This is all compatible with MFC (use AfxOleInit insead of CoInitialize, etc.).
int main(int argc, char* argv[])
{
  HRESULT      hr;

using namespace pdfMeld_TypeLib;

  hr = CoInitialize (NULL);    // Initialize COM
  if (SUCCEEDED(hr))
  {
    try    // Each of the following lines can throw exceptions
    {
      // Create the instance and get a pointer to the interface
      IpdfMeldPtr pPDF(__uuidof(pdfMeld));
      pPDF->setInFile (_bstr_t(L"c:\\temp\\filein1.pdf,d:\\mypdfs\\filein2.pdf"));
      pPDF->setPageNum ();
      pPDF->setPageTop ();
      pPDF->setPageLMargin (_bstr_t(L"144"));
      pPDF->setPageBMargin (_bstr_t(L"72"));
      pPDF->setPageFmt (_bstr_t(L"Page %1 of %2"));
      pPDF->setOutFile (_bstr_t(L"c:\\temp\\fileout.pdf"));

      _variant_t outval = pPDF->pdfMeld (); // Build the PDF file
    }
    catch (_com_error e)
    {
      cout << e.ErrorMessage() << endl;
    }
  }
  else
    cout << "CoInitialize Failed" << endl;

  CoUninitialize(); // Uninitialize COM

  return 0;
}
```