Multifunctional tool for PDF files

PDF TO THE MAX

To manage the mountains of paper that cross our desks every day, we need to file, retrieve, copy, stamp, investigate, and classify documents. A special tool can help users keep on top of their electronic paperwork: pdftk – the PDF toolkit. **BY STEFAN LAGOTZKI**

Native Linux PDF utilities such as GhostScript are very useful if you're willing to click through the menus. But if you're looking for something faster, or if you would like to automate a recurring task, try pdftk (the PDF Toolkit). pdftk is a convenient command-line program for processing PDF files. According to creator Sid Steward, "If PDF is electronic paper, then pdftk is an electronic staple-remover, hole-punch, binder, secret-decoder-ring, and X-Ray-glasses."

Installation and Use

You can download the latest version of the PDF toolkit from one of the Sid Steward's [1] websites. The GPL program is available for Linux, Mac OS X (Panther), FreeBSD, Solaris, and Windows. The platform-specific install proved to be quite simple on the platforms we tested (including Debian and SuSE Linux).

After completing the install, you can run pdftk from a shell. The *pdftk --help* command gives you a list of commands and options with short help texts. Table 1 lists and explains the major operations. The generic syntax for processing PDF files with the program is:

```
pdftk inputfile(s) ⮐
operation [option] ⮐
output outputfile ⮐
[passwords] [userpermissions]
```

Input files have to be in PDF format. The tool additionally needs text files in a special format for some operations. pdftk outputs one or more PDF files and also the text files in special cases.

In the following sections, I have put together a few examples that demonstrate a few of pdftk's more interesting uses. These examples by no means explore the program's limits.

Adding Attachments to PDF Files

You can add an attachment to a PDF file just as you can add an attachment to an email message. Adobe Reader (version 6 or newer) stores the attachments at the recipient. pdftk allows users to attach files to PDF documents and also to save the attachments. Before the release of Adobe Reader 7, this was the only way Linux users could save attachments.

Attachments can be used to add source code or excerpts from literature databases to PDF files. The following example shows how to forward a PDF file and the accompanying source code. To add a source code file to a PDF document using pdftk, type the following commands:

```
pdftk form.pdf attach_files ⮐
form.tex output new.pdf
```

Alternatively, you could use pdfLaTeX and attachfile to add the source code to the finished PDF file. The recipient would then use pdftk to unpack the source code file and other attachments using in any directory:

```
pdftk example_attachment.pdf ➘
unpack_files output Source
```

In this example, pdftk saves the attachments in a directory named *Source*. Adding the directory name always makes sense if you are handling multiple attachments.

Watermarks and Background Colors

pdftk uses a similar approach to the LaTeX *eso-pic* package to add a watermark to a document. The *background* option handles this and also allows the user to assign a PDF background color.

The image you will be using as your watermark must be a PDF file. You could create the image with a vector graphics tool or write a PostScript program. If the watermark is not the same size as the document, pdftk will scale the watermark. Let's assume you want to stamp the word "DRAFT" on a document. The first step would be to create a PDF with the right page size, before going on to call pdftk as follows:

```
pdftk example.pdf background ➘
draft.pdf output draft1.pdf
```

The watermark looks like a stamp on any part of the document without content. You can create a small EPS file with a background color of your choice. The PostScript commands for an A4 size page look like this:

```
%!PS-Adobe-2.0
%%BoundingBox: 0 0 595 842
0.95 0.95 0.90 setrgbcolor
0 0 moveto 595 0 rlineto 0 842 ➘
rlineto -595 0 rlineto
closepath fill
showpage
```

It is easy to change the background color in the EPS code. You can then convert the EPS file to a PDF using *epstopdf* and then run pdftk to use the file as a background:

```
pdftk example.pdf background ➘
Bg.pdf output eg_color.pdf
```

Splitting and Assembling PDF Files

The *burst* operation allows you to split a PDF file into its component pages. To do so, you need to provide a generic name for the pages and specify the numbering format:

```
pdftk example.pdf burst ➘
output Page%03d.pdf
pdftk example.pdf burst ➘
output ./Pages/Page%03d.pdf
```

In both examples a three-digit page number will be added to the page names. In the second example, pdftk will store the PDF files in an existing subdirectory.

The *cat* operation tells pdftk to concatenate multiple PDF files to create a new document. You can use wildcards to specify the filenames of the individual source files.

```
pdftk example.pdf form.pdf ➘
attachment.pdf ➘
cat output example_concat.pdf
pdftk D=coversheet.pdf ➘
B=example.pdf ➘
cat D B1-4 output ➘
example_coversheet.pdf
```

As the second example demonstrates, you can use *cat* to rearrange documents by linking part of a PDF file with parts of other PDFs to create a new document.

Table 1: pdftk Operations

Operation	Explanation
attach_files	Adds files as attachments to a PDF document. This allows you to add an archive file to the PDF file.
background	Adds a watermark to each page of the PDF document. Also allows you to affix an electronic stamp to empty spaces.
burst	Splits a PDF document into individual pages.
cat	Concatenates a new PDF file from multiple files or pages from different PDF documents.
dump_data	Outputs information about a PDF file on standard output.
dump_data_fields	Outputs information about the form fields in a PDF file on standard output.
fill_form	Fills out PDF forms or links form data with the document.
unpack_files	Unpacks the enclose attachments of a PDF document in a directory.
update_info	Updates the meta-information (e.g. author, title, topic) in a PDF file.

Listing 1: Typical PDF Meta-Data

```
InfoKey: Title
InfoValue: Working with pdftk
InfoKey: Subject
InfoValue: Practical pdftk
   examples
InfoKey: Keywords
InfoValue: pdftk, iText,
   OpenSource applications
InfoKey: Author
InfoValue: Stefan Lagotzki
InfoKey: City
InfoValue: Dresden
```

Querying and Updating Meta-Information

Most PDF files co ntain meta-information with details of the author, the topic, or the software used to create the document. Pdftk allows you to send this data to standard output or to redirect it into a file:

```
pdftk example.pdf ➘
dump_data output info.txt
```

This command saves the meta-information from the PDF document to a file titled *info.txt*. The information comprises a key field and the matching value (see Listing 1). Before forwarding or archiving PDF documents, it often makes sense to update the meta-data. Pdftk allows you to do so without having to recreate or translate the PDF file.

To update the meta-information, first create a text file with the meta-data; the file should look something like this (shortened for the sake of brevity):

```
InfoKey: Creator
InfoValue: TeX
InfoKey: Corporation
InfoValue: Sample and Sons
```

This file does not need to contain all the information that a PDF file can store. Fields that already contain values are not touched by the update if the text file does not specify them. You can even add new key fields (Corporation in our example) and assign values to them. The following call updates the meta-information:

```
pdftk example.pdf ➘
update_info info.txt ➘
output eg_meta.pdf
```

The input and output files are not permitted to have the same name. In other words, you either need to manually rename the output file, or use a shell script to do so.

Filling Out PDF Forms

PDF files can contain forms with known form fields. Adobe developed the proprietary but open FDF format for PDF form data. Listing 2 shows an example of a short FDF file.

In Listing 2, *T* is the title, and *V* is the form field value. You can now merge the PDF file with the FDF file and decide whether the form data should remain editable or be indelibly merged with the document:

```
pdftk form.pdf fill_form ➘
eg.fdf output edit.pdf
pdftk form.pdf fill_form ➘
eg.fdf output end.pdf flatten
```

The first example gives you editable results; whereas the *flatten* option in the second file indicates that the form fields should be indelibly merged with the PDF file.

The form feature allows you to use pdftk to create completed PDF forms on an Internet or Intranet server. The user fills out the form fields in his or her browser. Then a PHP or Perl script running in the background creates the FDF file; finally, pdftk combines the two parts. The completed PDF file can then be mailed.

Listing 2: Example FDF File

```
%FDF-1.2
1 0 obj <<
/FDF << /Fields [
<< /V (Dresden)/T (city) >>
<< /V (Stefan Lagotzki)/T
   (author)>>
]/F (form.pdf) >>
>>
endobj
trailer
<<
/Root 1 0 R
>>
%%EOF
```

Passwords and User Permissions in PDFs

PDF files can be protected by user and owner passwords. pdftk allows you to set both the passwords and the permissions for a PDF file. The following example sets both passwords:

```
pdftk file.pdf output ➘
file_new.pdf owner_pw ➘
Lie5quai user_pw phupaefu
```

The passwords in this example were generated using the *pwgen* tool. You must choose different strings for the user and owner passwords.

The owner of a PDF file can assign specific permissions. Table 2 has a list of permissions that you can set with pdftk. The following example first creates a PDF file that can only be printed. The second line creates a PDF file that can be printed and also copied.

Table 2: PDF Permissions

Option	Explanation
Printing	The document can be printed in best quality.
DegradedPrinting	The document print out will be of limited quality.
ModifyContents	The document content can be changed.
Assembly	The PDF document can be concatenated with other PDF documents.
CopyContents	Text and images can be copied from the file.
ModifyAnnotations	Comments and annotations can be changed.
FillIn	Forms in the PDF file can be filled out.
AllFeatures	The user has all specified privileges.

```
pdftk example.pdf output ➘
file_new.pdf owner_pw ➘
Lie5quai user_pw phupaefu ➘
allow printing
pdftk example.pdf output ➘
file_new.pdf owner_pw ➘
Lie5quai user_pw phupaefu ➘
allow printing CopyContents
```

PDF files can be encrypted with different levels of encryption. To encrypt a file with pdftk, add one of the following as the final option: *encrypt_40bit* or *encrypt_128bit*. You also need to supply a password for a password-protected PDF file. If you are processing multiple files, you can bind variables to the filenames and then assign a password to each file. In the following example, only file A is password protected:

```
pdftk A=file_new.pdf ➘
B=eg_color.pdf input_pw ➘
A=Lie5quai cat output ➘
egl_pw.pdf user_pw Abraxas
```

As the previous example does not allow you to concatenate the PDF file for *file_new.pdf*, you need to supply the owner password.

Conclusions

If you are looking for a quick, simple, and efficient tool for editing PDF files from the command line, try the PDF Toolkit. pdftk is a versatile, multifunctional PDF manipulation tool without the burden of a GUI. If you want to dig deeper into the subject of manipulating PDF files, see Sid Steward's book on *PDF Hacks* [2].

pdftk is written in C++ and based on the iText library [3], which in turn was written in Java. The whole program was complied and linked with tools from the free GNU Compiler Collection [4], This makes pdftk easily portable and extensible. The pdftk website has links to ports.

Development work on the pdftk program still continues. The program's author, Sid Steward, will answer queries on pdftk and PDF programming posted in the *comp.text.pdf* newsgroup and in his own PDF forum [1]. ■

INFO

[1] Sid Steward: pdftk; Version 1.12 (Nov. 2004): *http://www.accesspdf.com/pdftk/*

[2] Sid Steward, *PDF Hacks*; O'Reilly, 2004.

[3] Bruno Lowagie, Paulo Soares: iText-Library; Version 1.1 (Nov. 2004): *http://itext.sourceforge.net*

[4] GNU Compiler Collection, Version 3.4.3 (Nov. 2004): *http://gcc.gnu.org*