Remove metadata from a PDF file, using exiftool and qpdf. Note that embedded objects may still contain metadata.

README.md Raw

Anonymising PDFs

PDF metadata

Metadata in PDF files can be stored in at least two places:

- the Info Dictionary, a limited set of key/value pairs
- XMP packets, which contain RDF statements expressed as XML

PDF files

A PDF file contains a) objects and b) pointers to those objects.

When information is added to a PDF file, it is appended to the end of the file and a pointer is added.

When information is removed from a PDF file, the pointer is removed, but the actual data may not be removed.

To remove previously-deleted data, the PDF file must be rebuilt.

pdftk

pdftk can be used to update the Info Dictionary of a PDF file. See `pdftk-unset-info-dictionary-values.php` below for an example. As noted in the pdftk documentation, though, pdftk does not alter XMP metadata.

exiftool

exiftool can be used to read/write XMP metadata from/to PDF files.

- `exiftool -all:all` => read all the tags.
- `exiftool -all:all=` => remove all the tags.

`exiftool -all:all` also removes the pointer to the Info Dictionary, but does not completely remove it.

qpdf

qpdf can be used to linearize PDF files (`qpdf --linearize $FILE`), which optimises them for fast web loading and removes any orphan data.

Embedded objects.

After running qpdf, there may be new XMP metadata, as it extracts metadata from any embedded objects. To read the XMP tags of embedded objects, use `exiftool -extractEmbedded -all:all $FILE`.

pdftk-unset-info-dictionary-values.php

Raw

```php
<?php

$file = 'example.pdf';

// get the current metadata
$command = sprintf('pdftk %s dump_data', escapeshellarg($file));
$output = array(); $return = null; exec($command, $output, $return);

//print_r($output);

if ($return) {
    throw new Exception('There was an error reading metadata from the PDF file');
}

// set any metadata values to null
foreach ($output as $index => $line) {
    if (strpos($line, 'InfoValue:') === 0) {
        $output[$index] = 'InfoValue:';
    }
}

// write the updated metadata to a file
$metadataFile = tempnam(sys_get_temp_dir(), 'pdf-meta-');
file_put_contents($metadataFile, implode("\n", $output));

// create a new PDF using the updated metadata
$tmpFile = tempnam(sys_get_temp_dir(), 'pdf-tmp-');
$command = sprintf('pdftk %s update_info %s output %s',
    escapeshellarg($file), escapeshellarg($metadataFile), escapeshellarg($tmpFile));
$output = array(); $return = null; exec($command, $output, $return);

if ($return) {
    throw new Exception('There was an error writing metadata to the PDF file');
}

// clean up the temporary files
rename($tmpFile, $file);
unlink($metadataFile);
```

remove-pdf-metadata.sh

Raw

```sh
#!/bin/bash

FILE=example.pdf

# read tags from the original PDF
#exiftool -all:all $FILE

# remove tags (XMP + metadata) from the PDF
exiftool -all:all= $FILE

# linearize the file to remove orphan data
qpdf --linearize $FILE

# read XMP from the modified PDF
#exiftool -all:all $FILE

# read all strings from the modified PDF
#cat $FILE | strings > $FILE.txt

# read XMP from embedded objects in the modified PDF
#exiftool -extractEmbedded -all:all $FILE
```


Nichtraucher commented Dec 29, 2013

Could you possibly add functionality that makes it possible to a) remove metadata for files in a directory (and its subdirectories), and b) make it a Nautilus script (in order to edit metadata in selected files/directories)? That would make it a lot easier to use!
cheers!

m3nu commented Aug 28, 2015

This is short enough to make it a shell function.

```
clean_pdf() {
 pdftk $1 dump_data | \
  sed -e 's/\(InfoValue:\)\s.*/\1\ /g' | \
  pdftk $1 update_info - output clean-$1

 exiftool -all:all= clean-$1
 exiftool -all:all clean-$1
 exiftool -extractEmbedded -all:all clean-$1
 qpdf --linearize clean-$1 clean2-$1

 pdftk clean2-$1 dump_data
 exiftool clean2-$1
 pdfinfo -meta clean2-$1
}
```

via http://blog.snapdragon.cc/2015/08/28/shell-function-to-remove-all-metadata-from-pdf/

naught101 commented Aug 31, 2015

These methods don't seem to remove EXIF data from images embedded within a PDF. For example, the adobe photoshop editing history in a JPEG.

Telekor commented Sep 29, 2015

I have scripyt bymanuelRiel: now the word "clean" is added at the end of the file name (without extension). The *tricky* line is this:

```
FILE="${FILE%%.*}"
```

And this is the fool script:

```
clean_pdf() {
    FILE=$1
    FILE="${FILE%%.*}"
    echo "##############"
    echo $1
    echo "##############"
    if [ -e $1 ]
        then
        pdftk $1 dump_data | \
        sed -e 's/\(InfoValue:\)\s.*/\1\ /g' | \
        pdftk $1 update_info - output ${FILE}.clean.pdf
        exiftool -all:all= ${FILE}.clean.pdf
        exiftool -all:all ${FILE}.clean.pdf
        exiftool -extractEmbedded -all:all ${FILE}.clean.pdf
        qpdf --linearize ${FILE}.clean.pdf ${FILE}.clean2.pdf
        pdftk ${FILE}.clean2.pdf1 dump_data
        exiftool ${FILE}.clean2.pdf
        echo "##############"
        echo "Metadata of file "${FILE}.clean2.pdf
        pdfinfo -meta ${FILE}.clean2.pdf
        echo "##############"
    else
        echo "File not found!"

        fi
}
```

rbreton74 commented Sep 30, 2015

Sorry, there is a small mistake. This one works fully:

```
clean_pdf() {
    FILE=$1
    FILE="${FILE%%.*}"
    echo "#############"
    echo $1
    echo "#############"
    if [ -e $1 ]
        then
        pdftk $1 dump_data | \
        sed -e 's/\(InfoValue:\)\s.*/\1\ /g' | \
        pdftk $1 update_info - output ${FILE}.clean.pdf
        exiftool -all:all= ${FILE}.clean.pdf
        exiftool -all:all ${FILE}.clean.pdf
        exiftool -extractEmbedded -all:all ${FILE}.clean.pdf
        qpdf --linearize ${FILE}.clean.pdf ${FILE}.clean2.pdf
        pdftk ${FILE}.clean2.pdf dump_data
        exiftool ${FILE}.clean2.pdf
        echo "#############"
        echo "Metadatos de fichero "${FILE}.clean2.pdf
        pdfinfo -meta ${FILE}.clean2.pdf
        echo "#############"
    else
        echo "File not found!"

        fi
}
```

ande2101 commented Nov 22, 2015

The previous script doesn't work with files with spaces in the filename.

Changaco commented Dec 20, 2015

Other option: install pdf-redact-tools and run `pdf-redact-tools -s $FILE`

danielneis commented Mar 7, 2016

This pdf-redact-tools uses exiftool to remove some tags as you can see in https://github.com/firstlookmedia/pdf-redact-tools/blob/master/pdf-redact-tools#L115

bluesceada commented Sep 24, 2016 • edited

Hi, I wonder if exiftool is still a valid (or ever was) approach.
If I run exiftool on my file it warns me that tags are not really removed:

```
$ exiftool -all:all= myfile.pdf
Warning: [minor] ExifTool PDF edits are reversible. Deleted tags may be recovered! - myfile.pdf
    1 image files updated
```

My files also grow from 542.9 kB to 543.2 kB by exiftool and then from 543.2 kB to 544.6 kB by qpdf. So it seems there is actually more information added?

Let's see if these pdf-redact-tools do anything more. However I do for sure not want to follow one of their approaches, stacking PNG files and call it a PDF (that won't be searchable, has no vector graphic figures, and is probably larger in file size...)

//edit: OK that is actually the only approach they support, that's not applicable for me (and shouldn't be for most people that don't want to give away very larger or bad quality PDFs)

bertalanimre commented Feb 2, 2017

I'm wondering if there is any enhancement for this script to remove the embeded meetadata as well. For example, I have metadata about embeded Word documents. Unfortunatelly after the cleaning some sensitive data remains like filename, DocumentID and Instance ID.
How can I delete these embeded metadata in the fist place?

9991212 commented Feb 18, 2017

blueseceada:
Unfortunately, exiftool was never a really sanitizing approach due to its limitation: http://www.sno.phy.queensu.ca/%7Ephil/exiftool/ - “Writer Limitations: PDF - The original metadata is never actually removed.”
But
qpdf --pages myfile.pdf 1-z -- --empty clean-myfile.pdf
/* creates a new (empty) PDF document from scratch and add (all: 1-z) the pages from the original PDF file into it */
does the trick as the top-level (=file itself) metadata are concerned. It does not clean metadata of embedded objects.
(Remark 1.: It is possible to use
pdftk myfile.pdf cat 1-end output clean-myfile.pdf
instead abovementioned as well.
Remark 2.: On MS Windows, you can use BeCyPDFMetaEdit to obtain the same result, too; but for PDF version >1.6 the result is not guaranteed.)

bertalanimre:
It may perhaps be done by filtering the PDF file through an editor (sed, tr?) capable of deleting characters between (and including) "<x:xmpmeta" and "</x:xmpmeta>" strings. But I have never needed it so never tried it.

RootLUG commented Mar 23, 2017

Did you guys find saome way how to remove metadata from file like this: https://publications.usa.gov/USAFileDnld.php?PubType=P&PubID=6099&httpGetPubID=0 ?

I also tried the approach suggested by @9991212 but there is still a lot of metadata left like Creator, For, Create Date etc... which should be on the top level PDF