

Dokumente (stapelweise) einscannen

Wenn man keinen Dokumentenscanner mit Vorlageneinzug hat wird das Scannen mehrerer Seiten oder z.B. ganzer Zettelsammlungen schnell zu einer abendfüllenden Angelegenheit und man bekommt vom vielen Klicken in der GUI des Scannprogramms einen lahmen Zeigefinger. Ein kleines Skript schafft hier schnell abhilfe:

[batch_scan_to_pdf.sh](#)

```
#!/bin/bash
# Copyright (c) 2013 Christoph von Thuelen
#
# Author: Christoph von Thuelen <Christoph AT von-Thuelen DOT de>
#
# This program is free software; you can redistribute it and/or
# modify it under the terms of the GNU General Public License as
# published by the Free Software Foundation; either version 2 of the
# License, or (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# Following tools are required to do the job:
# * sane-utils
# * libtiff-tools
# * imagemagick
# * pdftk
# * usbreset (workaround for Epson Perfection V300 Photo ;-) )

# config parameters
SCANAREA="-l 0 -t 0 -x 210 -y 297" # app. DIN-A4 scan area
COUNTER=1 # page counter, starting at page
"1"
OUTPUTFILE="scan" # filename
MYSCANNER=epkowa # my scanner: Epson Perfection
V300 Photo aka "epkowa"
CANSHOWPROGRESS=1 # it is able to show the scan
progress during scan

# Check some tools we need ...
if [ ! -f /usr/bin/tiff2pdf ]
then
    echo "ERROR - tiff2pdf is not installed. Please install libtiff-
tools"
    exit 1
fi
if [ ! -f /usr/bin/scanimage ]
then
```

```
    echo "Error - scanimage is not installed.. Please install sane-utils"
    exit 1
fi
if [ ! -f /usr/bin/pdftk ]
then
    echo "Error - pdftk is not installed. Please install pdftk"
    exit 1
fi
if [ ! -f /usr/bin/convert ]
then
    echo "Error - convert is not installed. Please install imagemagick"
    exit 1
fi

function help {
    echo "Usage: `basename $0` -m {binary,gray,color} -r [resolution in
dpi] -o <outputfile>"
    echo "  -m: Scan mode can be: binary, gray or color"
    echo "  -r: Resolution in dpi"
    echo "  -o: Output file name w/o \".pdf\""
    exit 1
}

# Show help text
if [ $# -eq "0" ]; then
    help
elif [ $1 = "--help" -o $1 = "-help" -o $1 = "-?" -o $1 = "-h" ]; then
    help
fi

# read options
while getopts ":m:r:o:" Option
do
    case $Option in
        m) SCANMODE=$OPTARG ;;
        r) RESOLUTION=$OPTARG ;;
        o) OUTPUTFILE=$OPTARG ;;
    esac
done

# scan images
if [ -f $OUTPUTFILE.pdf ]; then
    echo "Error - Output file: $OUTPUTFILE already exists!"
    exit 1
else
    echo "Batch mode, Output file: $OUTPUTFILE"
    scanimage -d $MYSCANNER --format tiff --mode $SCANMODE $SCANAREA --
resolution $RESOLUTION $PROGRESS --batch --batch-prompt
    while [ -f out$COUNTER.tif ]; do
        # tiff2pdf -z out$COUNTER.tif -o out$COUNTER.pdf
        convert out$COUNTER.tif out$COUNTER.jpg
    done
fi
```

```

    convert out$COUNTER.jpg out$COUNTER.pdf
    COUNTER=$((COUNTER+1))
done

#concatenate scanned images to one single pdf
pdftk out*.pdf cat output $OUTPUTFILE.pdf

## clean up
rm out*.tif
rm out*.pdf

## workaround for reset Epson Perfection V300 Photo scanner after
batch scan
DEVICE=`lsusb | grep -i epson | cut -d' ' -f 4 | cut -d: -f1`
BUS=`lsusb | grep -i epson | cut -d' ' -f 2`
echo "BUS: $BUS, DEVICE: $DEVICE"
usbreset /dev/bus/usb/$BUS/$DEVICE
fi
exit 0

```

Nicht vergessen die Datei batch_scan_to_pdf.sh mit:

```
chmod +x batch_scan_to_pdf.sh
```

auch ausführbar zu machen



Da sich leider mein Scanner, ein Epson Perfection V300 Photo, nach jedem Batch Durchlauf aufgehängt hat verwende ich am Ende des o.a. Skriptes ein kleines Tool namens usbreset welches ich [hier](#) gefunden habe - funktioniert ausgezeichnet.

[usbreset.c](#)

```

/* usbreset -- send a USB port reset to a USB device */
/*
take from here: http://marc.info/?l=linux-usb&m=121459435621262&w=2

compile with: cc usbreset.c -o usbreset

usage:
$ lsusb
  Bus 002 Device 003: ID 0fe9:9010 XXYYZZ
$ chmod +x usbreset
$ sudo ./usbreset /dev/bus/usb/002/003
*/

#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>

```

```
#include <sys/ioctl.h>

#include <linux/usbdevice_fs.h>

int main(int argc, char **argv)
{
    const char *filename;
    int fd;
    int rc;

    if (argc != 2) {
        fprintf(stderr, "Usage: usbreset device-filename\n");
        return 1;
    }
    filename = argv[1];

    fd = open(filename, O_WRONLY);
    if (fd < 0) {
        perror("Error opening output file");
        return 1;
    }

    printf("Resetting USB device %s\n", filename);
    rc = ioctl(fd, USBDEVFS_RESET, 0);
    if (rc < 0) {
        perror("Error in ioctl");
        return 1;
    }
    printf("Reset successful\n");

    close(fd);
    return 0;
}
```

Übersetzten kann man das kleine Tool dann mit:

```
cc usbreset.c -o usbreset'
```

From:

<https://von-thuelen.de/> - Christophs DokuWiki

Permanent link:

<https://von-thuelen.de/doku.php/wiki/linux/batchscanning?rev=1448377916>

Last update: **2020/04/15 18:21**

