

Photos, Files & Raspberry Pi

- [Set the "date taken" on photos with ExifTool](#)
- [Convert images in bulk \(convert_images.sh\)](#)
- [Delete HEIC files that already have a JPG copy](#)
- [Rename all files in a folder to upper case](#)
- [Shrink Raspberry Pi SD-card images with PiShrink](#)

Set the "date taken" on photos with ExifTool

Scanned or undated photos often have no EXIF date, so photo libraries (Immich, PhotoPrism, Synology Photos) sort them by file date. ExifTool can set it.

Install

```
sudo dnf install perl-Image-ExifTool    # RHEL/Rocky (EPEL)
sudo apt install libimage-exiftool-perl # Debian/Ubuntu
```

Set one fixed date on every JPG in a folder (recursively)

```
exiftool -r -ext jpg -ext jpeg -AllDates="2005:11:06 03:20:00" /path/to/photos
```

`-AllDates` sets `DateTimeOriginal`, `CreateDate` and `ModifyDate` together.

Copy `DateTimeOriginal` into the other date tags

Quote the argument. Unquoted, bash reads `<` as "read input from a file".

```
exiftool -r -ext jpg '-AllDates<DateTimeOriginal' /path/to/photos
```

Backups

ExifTool keeps a `name.jpg_original` copy of every file it changes. Once you have checked the result:

```
exiftool -r -delete_original! /path/to/photos
```

Or use `-overwrite_original` up front if you have your own backup.

As a tiny script

```
#!/bin/bash
dir=${1:-.}; when=${2:?usage: setdate <dir> "YYYY:MM:DD HH:MM:SS"}
exiftool -r -ext jpg -ext jpeg -AllDates="$when" "$dir"
```

Convert images in bulk (convert_images.sh)

Location: `/opt/scripts/docker_scripts/convert_images.sh` (68 lines)

Purpose

Converts HEIC photos (iPhone) in a folder to JPG next to the originals, with `heif-convert`.

How it runs

By hand.

Usage

```
/opt/scripts/docker_scripts/convert_images.sh -n /path/to/photos # dry run  
/opt/scripts/docker_scripts/convert_images.sh /path/to/photos
```

Dependencies

`heif-convert` from `libheif-tools` (EPEL).

What it touches

Writes `<name>.jpg` beside each HEIC file. It does not recurse into subfolders and does not delete anything.

Delete HEIC files that already have a JPG copy

Location: `/opt/scripts/docker_scripts/delete_heic_jpg.sh` and `/opt/scripts/docker_scripts/delete_jpg_heic.sh` (56 lines each)

Purpose

Two mirror-image clean-up scripts for use after converting photos. Each works in the **current folder only**:

- `delete_heic_jpg.sh` deletes each `.heic` file that has a matching `.jpg` (keeps the JPG).
- `delete_jpg_heic.sh` deletes each `.jpg` file that has a matching `.heic` (keeps the HEIC).

Extensions are matched in any case.

How it runs

By hand, from inside the photo folder.

Usage

```
cd /path/to/photos
/opt/scripts/docker_scripts/delete_heic_jpg.sh dry-run # show what would go
/opt/scripts/docker_scripts/delete_heic_jpg.sh        # delete
```

Dependencies

bash only.

What it touches

Deletes files in the current folder. The log goes to stderr.

Rename all files in a folder to upper case

Location: `/opt/scripts/renameuppercase.sh` (May 2016, 1 line of logic)

Purpose

Renames every file in the **current folder** to upper case (for example `photo.jpg` becomes `PHOTO.JPG`).

How it runs

By hand, from inside the folder you want to change.

Usage

```
cd /path/to/folder && /opt/scripts/renameuppercase.sh
```

Dependencies

`tr`, `mv`.

What it touches

Every file name in the current folder. There is no undo.

Shrink Raspberry Pi SD-card images with PiShrink

Location: `/opt/scripts/pishrink.sh` (April 2018)

Purpose

A copy of the open-source **PiShrink** tool. It shrinks a Raspberry Pi SD-card image (`.img`) to the smallest size its root filesystem allows. Unless you pass `-s`, it also sets the Pi to grow the filesystem back to full size on first boot.

How it runs

By hand, as root.

Usage

```
sudo /opt/scripts/pishrink.sh pi-backup.img          # shrink in place
sudo /opt/scripts/pishrink.sh -s pi-backup.img small.img # copy first, no auto-expand
```

Dependencies

`parted`, `losetup`, `tune2fs`, `e2fsck`, `resize2fs`, `md5sum`, `which`.

What it touches

Only the image file. It attaches a loop device, mounts the image in a temporary folder, and may replace `/etc/rc.local` inside the image.