

How to export Flickr to a GPX file to Geotag photos

Using Flickr to add GPS metadata to my photos.

Recently I developed a need to fill my photo metadata in correctly, including GPS coordinates. Right now I have this information in

- some of the photos as EXIF/IPTC
- WordPress, as post meta
- Flickr
- and in my head

Neither WordPress, nor Flickr are ideal place to store this, and my head will eventually lose data as well, so I want the files themselves to be as self sustaining as possible. I'm planning to fix all potential problems as well with proper encodings (*yay for ASCII only metadata in a binary file!*) at the description, title, and keywords fields.

So I started looking for a software which would allow me to drag and drop images on a map and geotag them accordingly - this is where the problem started. I've tried out at least ten solutions for linux, and ended up using Flickr for this.

[digiKam](#)^[1] although it works, it's horrible, because the tiles don't reload if you move the map, only if you zoom. You also need to fiddle with Konqueror settings to make this work and it seems like digiKam starts messing with metadata I did not want it to mess with.

[geotag](#)^[2] and [picasa](#)^[3] needs Google Earth, which simple fails to work on current Linux Mint.

[gpsprune](#)^[4] is so arcane that I still haven't figured out how to use it correctly.

So I decided I'll go with the simplest, but most tedious solution: making a GPX file by hand and use [exiftool](#)^[5] to apply it when I realized, I could use Flickr for this, since all the photos I published on my site are cross-posted to Flickr.

Making the GPX file

Due to my day job, I'm exposed heavily to Perl and bash... so I made the data fetcher and GPX printer in bash, because - apart from wget - it doesn't have dependencies.

```
#!/bin/bash

FLICKR_API_KEY="YOUR_API_KEY"
GPX_FILE="/tmp/photo-geo.gpx"
FLICKR_USER="YOUR_FLICKR_USER"

function flickr_meta () {
    while read id; do
        &2 echo "getting flickr meta for $id";
        json=$(wget -q -O- "https://api.flickr.com/services/
rest/?method=flickr.photos.getInfo&api_key=${FLICKR_API_KEY}
&photo_id=${id}&format=json&nojsoncallback=1")

        if ! grep -q 'latitude:' <<< $json || ! grep -q
'longitude:' <<< $json; then
            &2 echo "no geo for $id";
            continue;
        fi

        if ! grep -q 'taken:' <<< $json; then
            &2 echo "no taken date for $id"
            continue
        fi

        lat=$(sed -r 's/. *latitude:"([0-9\.-]+)"/.*\/\1/' <<< "$
$json}")
        lon=$(sed -r 's/. *longitude:"([0-9\.-]+)"/.*\/\1/' <<<
"$json")
        d=$(sed -r 's/. *taken:"([0-9\ : -]+)"/.*\/\1/' <<< "$
$json" | sed -r 's/(.*)\s(.*)\/\1T\2Z/' )
        echo "<trkpt lat=\"$lat\" lon=\"$lon\"><time>$d</
time></trkpt>"
        done <<< "$(flickr_list)"
    }
}
```

```

function flickr_list () {
    >&2 echo "getting flickr id list";
    json=$(wget -q -O- "https://api.flickr.com/services/rest/?
method=flickr.people.getPhotos&api_key=${FLICKR_API_KEY}
&format=json&nojsoncallback=1&user_id=${FLICKR_USER}
&per_page=300");
    json=$(echo "${json}" | json_pp);

    while read line; do
        if ! grep -q "id" <<< "$line"; then
            continue;
        fi

        echo "$line" | sed -r 's/"id"\s+:\s+"([0-9]+)",?/\1/'
    done <<< "$json";
}

```

```

function generate_gpx () {
    :>$GPX_FILE
    echo '<?xml version="1.0" encoding="UTF-8"
standalone="no" ?>
<gpx xmlns="http://www.topografix.com/GPX/1/1"
xmlns:gpxx="http://www.garmin.com/xmlschemas/GpxExtensions/v3"
xmlns:gpxtpx="http://www.garmin.com/xmlschemas/
TrackPointExtension/v1" creator="Oregon 400t" version="1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.topografix.com/GPX/1/1 http://
www.topografix.com/GPX/1/1/gpx.xsd http://www.garmin.com/
xmlschemas/GpxExtensions/v3 http://www.garmin.com/xmlschemas/
GpxExtensionsv3.xsd http://www.garmin.com/xmlschemas/
TrackPointExtension/v1 http://www.garmin.com/xmlschemas/
TrackPointExtensionv1.xsd">
<trk>
    <trkseg>' >> $GPX_FILE
    echo $(flickr_meta) >> $GPX_FILE
    echo '</trkseg>
</trk>
</gpx>' >> $GPX_FILE
}

```

```
generate_gpx
```

This outputs a single file, by default to `/tmp/photo-geo.gpx`. I've limited the query to 300 images and I didn't go pagination, because I didn't need it. You may do, so please be aware of this.

Actual tagging

Just run

```
exiftool -geotag=/path/to/gpx /path/to/photos/dir
```

```
bash
```

and magic happens.

Links

1. <https://www.digikam.org/>
2. <https://sourceforge.net/projects/geotag/>
3. <http://picasa.google.com/>
4. <http://activityworkshop.net/software/gpsprune/>
5. <http://owl.phy.queensu.ca/~phil/exiftool/>

Created by Peter Molnar <mail@petermolnar.net>, published at 2016-10-13 19:00 UTC, last modified at 2021-05-11 11:49 UTC , to canonical URL <https://petermolnar.net/article/how-to-export-flickr-to-a-gpx-file-to-geotag-photos/> , licensed under CC-BY-4.0 .