# GeoELAN Manual, v1.1

## Table of Contents



**Please cite this document as**

> Larsson, J. 2021. GeoELAN Manual. Lund: Lund University Humanities Lab. https://gitlab.com/rwaai/geoelan


**Acknowledgments**

> GeoELAN was developed with support from the Bank of Sweden Tercentenary Foundation (Grant nos NHS14-1665:1 and IN17-0183:1).
>
> We would also like to acknowledge the The Language Archive, Max Planck Institute for Psycholinguistics in Nijmegen for their tireless efforts in developing ELAN.


Document date: 2021-03-07

# Changelog

- `v1.1`
  - General
    - Fixed various typos.
  - `eaf2geo`
    - Added: Geo-referencing annotations on dependent tiers is now supported, with the caveat that the tier can not be tokenized, nor can any of its parents.
    - Added number of annotations and whether tier is tokenized, when selecting tier to geo-reference.
    - Added: Polylines are now colour coded according to annotation value. Annotated events with the same annotation value will have the same color.
  - `check`
    - Changed: reporting of errors should be clearer (FIT parsing error and user input errors)
      - if recoverable error raised, the data summary will be printed together with the error
    - Changed: Specifying FIT Global ID via `-g`/`--global-id` no longer requires `--verbose` to print values.
    - Changed: Data summary table is now sorted on numerical FIT Global ID.
  - `match`
    - Fixed: `match` should now be considerably faster.
- `v1.0`
  - Initial release

# Introduction

GeoELAN is a command-line tool that geo-references time-aligned text-annotations of observed phenomena in audiovisual recordings, captured with a recent Garmin VIRB action camera (see Larsson et al 2021). By annotating a section representing an on-site utterance, a plant that is in view, or anything else that was captured, it can be automatically linked to the corresponding coordinates. The nature of the workflow also means consultants not physically present at the the time of recording may evaluate observed phenomena to be geo-referenced post-collection. As the name implies, the free ELAN annotation software plays a central role and is required to annotate events. The final output can be points or polylines in the form of an annotated KML-file. Henceforth, "VIRB" refers to the Garmin VIRB Ultra 30. Also see the note on GoPro.

**GeoELAN is multi-functional tool that can**

- geo-reference ELAN-annotations of VIRB footage and output these as annotated points or polylines.
- search your hard drive to locate and match all relevant VIRB-files, including FIT-files.
- automatically concatenate the clips for a specific recording, and generate an ELAN-file.
- inspect the content of your FIT-files, even outside of coordinates.

**Installation**

- See the `bin` directory for pre-compiled executables for Linux, macOS, and Windows.

**Requirements**

- Garmin VIRB action camera, such as the VIRB Ultra 30 (documentation)
- ELAN (documentation)
- FFmpeg (required for concatenating video)

**Quick help**

- Usage: `geoelan SUBCOMMAND OPTIONS`. E.g. to geo-reference an ELAN-file:
  - `geoelan eaf2geo --eaf MyElanFile.eaf --fit MyFitFile.fit`
- Running `geoelan` with no options will display an overview.
- Running `geoelan SUBCOMMAND --help` displays an overview for that sub-command, e.g.:
  - `geoelan eaf2geo --help`.
- Available sub-commands: `cam2eaf`, `eaf2geo`, `match`, `check`, `manual`
- The sub-command must be the first argument, but for those that follow, order is not important
- The `geoelan` executable contains the full manual for convenience, export with:
  - `geoelan manual --pdf`

## Compile and install from source

Most users will probably use the pre-compiled versions, but you can also compile GeoELAN yourself. Depending on your operating system, this may require installing additional software. The basic steps are:

1. Install Rust
2. Get the GeoELAN source from https://gitlab.com/rwaai/geoelan (via `git` or the zip)
3. `cd geoelan` (you should be in the folder containing `Cargo.toml`)
4. `cargo build --release`
5. `cargo install --path .` (makes `geoelan` a global command)

## Before you start

**Running GeoELAN**

- GeoELAN is a command-line tool and has no graphical user interface.
- GeoELAN is self-contained and can be run as is, but…
- …FFmpeg is required to concatenate clips. (only for sub-command `cam2eaf`)
- If you use macOS and GeoELAN does not run, see https://support.apple.com/en-us/HT202491.
- The terminal command is `geoelan` on linux/macOS, and `geoelan.exe` on Windows.

**Device compatibility**

- Other VIRB cameras may work, but only VIRB Ultra 30 has been tested extensively.

**GPS and the VIRB**

- Make sure the GPS is turned on and in reach of a satellite.
- Current GPS-modules may need a few minutes before starting to log coordinates.
- On the VIRB's screen the GPS-icon should be steady, not blinking.
- The VIRB may still log coordinates while the icon is blinking, but do not rely on this being the norm.

**Annotating in ELAN**

- Each kind of observed phenomena should be limited to a single ELAN-tier, so…
- …to keep e.g. mentioned locations and plant sightings within the same ELAN-file make a separate tier for each (see Example walkthrough below).

# Example walkthrough

Described below is an example of how GeoELAN can be used to geo-reference ELAN-annotations. Please refer to the detailed sections if you get stuck. Note that all input video clips must be the unprocessed, original VIRB files. The so-called FIT-files mentioned throughout this manual are where the VIRB logs GPS-data and other kinds of telemetry during a recording session. These need to be matched to the corresponding video recording (see *The FIT-format and the Garmin VIRB* for further information). GeoELAN will help with all of this, with the exception of annotating your data.

The basic steps are:

1. Record video with a recent Garmin VIRB action camera.
2. Use GeoELAN to concatenate the video clips and generate an ELAN-file.
3. Annotate spatially interesting sections in ELAN using the pre-generated ELAN-file.
4. Use GeoELAN to geo-reference the annotations, resulting in an annotated KML-file.

**Input files**

- `VIRB0001-1.MP4`, the first clip in one recording session (remaining clips located automatically)
- FIT-file with corresponding GPS-data (located automatically)

**Output files**

- `VIRB0001-1_point-single.kml`: the final output is a KML-file with ELAN annotation content synchronised and mapped to the corresponding points as descriptions. In this example, each annotation will generate a single point. See sub-command *eaf2geo* for other options.

**Example of VIRB SDCard file structure**

```
├── DCIM
│   └── 100_VIRB
│       ├── VIRB0001-1.GLV          Recording session split up
│       ├── VIRB0001-1.MP4          as 10 minutes clips. Low (GLV)
│       ├── VIRB0001-2.GLV          and high (MP4) resolution clip.
│       └── VIRB0001-2.MP4
└── GMetrix
    ├── 2017-01-01-12-00-00.fit  Telemetry files, a.k.a "FIT-files".
    └── 2017-01-02-12-00-00.fit  May contain data, such as GPS-logs,
                                 for multiple recording sessions.
```

## Step 1/3: Generate an ELAN-file with linked media files

**Command**

```
geoelan cam2eaf --video INDIR/VIRB0001-1.MP4 --indir INDIR/ --outdir OUTDIR/
```

**Output**

```
OUTDIR/VIRB0001-1/
├── VIRB0001-1.mp4     High-resolution video (concatenated)
├── VIRB0001-1_glv.mp4 Low-resolution video for ELAN (concatenated)
├── VIRB0001-1.wav     Extracted audio for ELAN (concatenated)
├── VIRB0001-1.eaf     ELAN-file with pre-linked media files
├── VIRB0001-1.kml     Overview KML-file with all points logged during the recording session
└── VIRB0001-1.txt     FFmpeg concatenation file, paths to input clips
```

**Explanation**

GeoELAN locates and concatenates all clips belonging to the recording session starting with `VIRB0001-1.MP4`, then generates an ELAN-file with the resulting audio and video files pre-linked.

**Breakdown of the command**

The relevant sub-command is `cam2eaf`. The user specifies the *first clip in the session* (`--video`). The remaining clips, together with the corresponding FIT-file, will be automatically located as long as these exist somewhere in the specified input directory (`--indir`), including sub-directories. If low-resolution clips (`.GLV`) are located, a concatenated low-resolution video will be linked in the ELAN-file. If not, the concatenated high-resolution video will be linked instead. GeoELAN defaults to *not* insert a tier with geo-data in the ELAN-file due to the effect this may have on performance (to do so, see *Geo-data in ELAN* and the `--geotier` option for the sub-command *cam2eaf*). The result, including the FIT-file, is copied to a folder named `VIRB0001-1` under the specified output directory (`--outdir`).

For long recording sessions containing many clips, step 1 (i.e. running `cam2eaf`) is usually much faster if `--indir` and `--outdir` is not on the same physical hard drive. Those with an SSD (standard on most modern laptops) should be fine running step 1. on a single drive however.

## Step 2/3: Annotate events in ELAN

The user annotates events that are to be geo-referenced using the generated ELAN-file. Currently, the tool only supports extracting annotations from a single tier, selectable in step 3. So if the user wants to generate a KML-file with e.g. indigenous place names mentioned on-site during the recording, all information concerning the place names must be limited to a single tier. When the annotations are geo-referenced in step 3, their textual content will be used as descriptions for the corresponding points in the KML-file. Points corresponding to unannotated sections of the ELAN-file will either be

discarded or have no description, depending on the output options in step 3. The annotated event can relate to anything observed in the recording and can be represented as either points or polylines in the output KML-file. If you are unsure which best applies to what you had in mind for your data or how this may affect how you annotate, here are a few ideas for each kind.

**Point output**

*Points* could concern documenting the location of a plant or a geographical feature (on-site, or at the time either is visible in the video), or a place name or an animal cry (at the time either is uttered or heard on-site). For these specific cases, the exact time spans of the annotations are not that important. Making the annotation last for the duration of the place name being uttered, or for as long as the plant is visible should be enough. If unsure, add a another second. An average coordinate will be calculated for those that were logged within each annotation's time span, so as long as the camera wearer does not stray too far from the observation point, the result should be accurate enough. (This behaviour is specific to the `--geoshape point-single` option, used in step 3 below, refer to sub-command *eaf2geo* for other options)

**Line output**

*Lines* could concern documenting various types of movement through the landscape or a narrative reflecting on the current surroundings over time. For movement types, one could tag the movement of walking up-hill as an annotation (as it is observed visually in the recording, with the annotation's start time at the bottom and its end at the top), whereas geo-referencing the various parts of a narrative could represent comments on visible landscape features, or perhaps the re-construction of an historical event as it unfolded over space and time. In the KML-file, the line may be continuous (a mix of marked and unmarked events) or broken-up (marked events only).

Since the tool can be run as many times as needed, several kinds of observations can be stored within the same ELAN-file on separate tiers (plant observations and documenting narratives may both apply to a single recording session). Simply adjust the output options from points to lines, if necessary, and select a different tier in step 3 below. (see `--geoshape` for sub-command *eaf2geo* for a more detailed overview of the options).

## Step 3/3: Generate a KML-file from geo-referenced ELAN annotations

**Command**

```
geoelan eaf2geo --eaf VIRB0001-1.eaf --fit 2003-01-02-12-00-00.fit --geoshape point-single
```

**Output**

```
OUTDIR/VIRB0001-1/
├── ...                          Existing files
└── VIRB0001-1_point-single.kml  New KML-file, one point per ELAN-annotation in the selected tier
```

**Explanation**

GeoELAN geo-references all annotations in a single ELAN-tier (selectable from a list) for the specified ELAN-file and generates an annotated KML-file where each point represents a single annotation.

**Breakdown of the command**

The relevant sub-command is `eaf2geo`. By specifying an ELAN-file (`--eaf`) and the corresponding FIT-file (`--fit`) GeoELAN will synchronise the annotations with the coordinates contained within the FIT-file. This process is usually completely automatic, but in case it fails, the user will be presented with a list of recording sessions together with so-called UUIDs (Universally Unique Identifier) present in the FIT-file. UUIDs are embedded both within the original video clips and the FIT-files and are key to synchronise and extract relevant GPS-data (see *The FIT-format and the Garmin VIRB* below). As a help, the number of video clips and the *UUID for the first clip* in each session is listed. The detailed sections will mention other options for selecting recording session and the corresponding data, so it may be good to remember that GeoELAN always requires specifying the *first clip* or the *UUID for the first clip* to be able to filter and synchronise data.

*Selecting a recording session (UUIDs shortened to fit)*

```
 Session | Clips | First UUID in session
.....................................................................................
  1.     |  1    | VIRBactioncameraULTRA30_Tall_..._32eed236_1_17_2017-01-28-05-16-40.fit
  2.     |  1    | VIRBactioncameraULTRA30_Tall_..._32eed5ab_1_18_2017-01-28-05-16-40.fit
  3.     |  3    | VIRBactioncameraULTRA30_Tall_..._32eed7e9_1_19_2017-01-28-05-16-40.fit
  4.     |  1    | VIRBactioncameraULTRA30_Tall_..._32eedd83_1_20_2017-01-28-05-16-40.fit
.....................................................................................
Select session:
```

`--geoshape point-single` lets GeoELAN know that each, respective annotation should be distilled into a single point, meaning that the generated KML-file will contain as many points as there are annotations on the selected tier. Each point inherits the corresponding annotation text for the selected tier as its description. The KML-file is named according to the selected `--geoshape` option, in this case `VIRB0001-1_point-single.kml`.

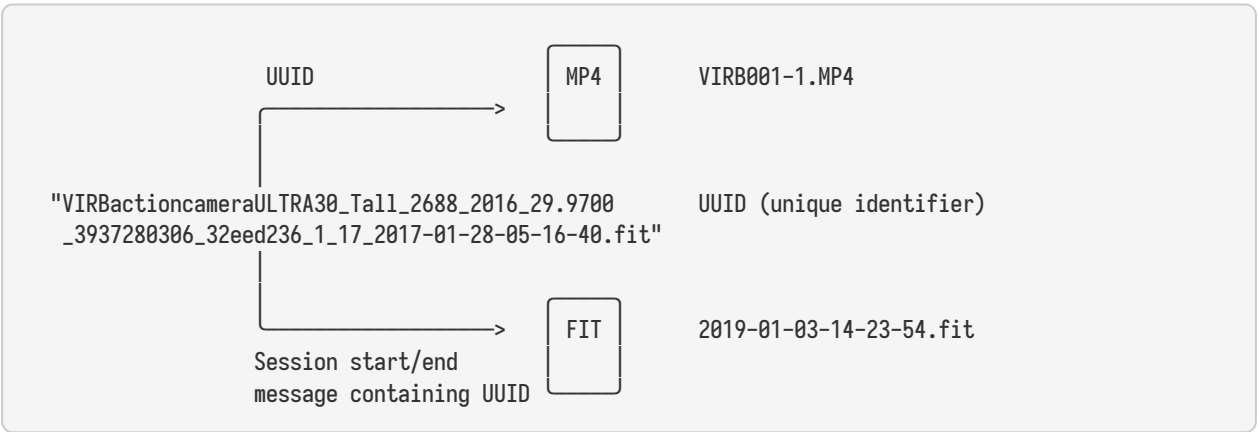
## The FIT-format and the Garmin VIRB

A single FIT-file may contain telemetry for multiple recording sessions. When the camera is turned on, it immediately starts logging data into a new FIT-file, regardless of a video being recorded or not. The camera will keep logging to this file until completely turned off. If turned on again, a new FIT-file will be created. All data points in a FIT-file are explicitly timestamped, which allows synchronisation against any data type in the file. Further, with the help of the built-in GPS, absolute timestamps can be generated. These can be used for documentation purposes or to synchronise against external data sources.

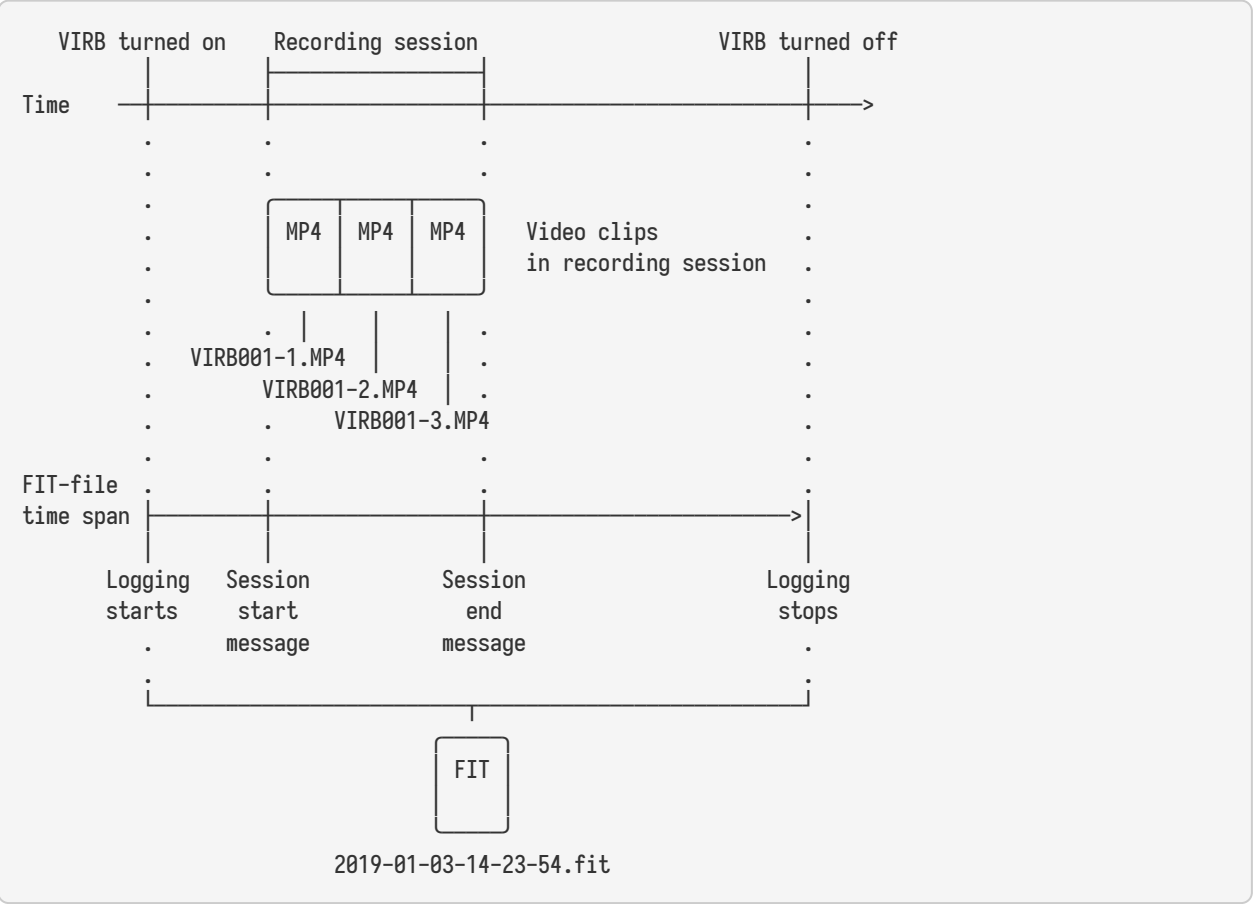
For geo-referenced annotations, `geoelan` always embeds absolute timestamps in the resulting KML-file.

The VIRB cameras split up recording sessions into video clips, each approximately 10 minutes in length, with no option to turn this off. To link VIRB video to its corresponding telemetry (e.g. coordinates logged by the GPS during the recording session), both the clips and the FIT-file contain unique identifiers (UUID). When the user starts recording, a "video recording session start" message is logged to the current FIT-file together with the UUID embedded in the first clip, denoting the start of a recording session. Similarly, when recording ends, a "video recording session end" message is logged together with the UUID embedded in the last clip in the session. Since all logged FIT-data is timestamped, this creates a timeline for the session that can be related to any logged data in the FIT-file.

*Matching MP4 and FIT-files via embedded UUIDs*



*Logging telemetry and boundaries for a recording session in a FIT-file*




The VIRB logs location, barometric pressure, and rotation among many other data types. Since the FIT-format is not a text based data format, and thus cannot be inspected using a text editor, the `check` sub-command allows for some exploration of a FIT-file (see sub-command *check*). GeoELAN will also help out with matching recording sessions to the corresponding FIT-files (see sub-commands *cam2eaf*, and *match*).

## Preserving UUIDs

Concatenating or converting the video clips will usually remove the UUIDs, so the user is advised to save the original video clips. To mitigate the risk of losing UUIDs, GeoELAN embeds these as metadata within the concatenated video files, and also within the ELAN and KML-files.

Most of the sub-commands allow for selecting UUID from those present in the relevant FIT-file when matching files or geo-referencing annotations. The `match` sub-command can also be used to locate all files for a specific session, or to generate a CSV-file listing all matched clips and FIT-files in the specified directory, together with the path and UUID for each encountered file.

## Video file management and options

On the VIRB MicroSD card, the low-resolution clips have a `.GLV` extension. These are generated by the VIRB for quick viewing on the internal camera display. If available, GeoELAN will prefer to link these in the ELAN-file over the high-resolution video due to their smaller size (both will be concatenated by default). GeoELAN will not be able to identify the low-resolution `.GLV` as such if renamed to `.MP4` and they may even be mistaken for the high-resolution versions. If you only require the low-resolution videos to be concatenated, use the `--low-res-only` flag when running `cam2eaf`. This will ignore the high-resolution `.MP4`-files as a concatenation target, with an option to copy these as-is (`--copy`) to the output directory (see the *cam2eaf* section for further information).

---

# FFmpeg and video concatenation

The `cam2eaf` sub-command requires FFmpeg for concatenating the high and low-resolution MP4-clips and to extract the audio track as a WAV-file. There are two main options for installing FFmpeg:

1. Download the *static build* of FFmpeg, and specify its path using the `--ffmpeg` option
2. Install via a *package manager*. FFmpeg will be automatically available to `cam2eaf` in this case.

**Static build**

The *static build* option means that the relevant media codecs are included in a single, executable file that can be used as is. The FFmpeg download page provides links to static builds for macOS, Windows and Linux. Put the downloaded `ffmpeg`-file in a convenient location and use the `--ffmpeg` option when running `cam2eaf`. Optionally moving or symlinking this file to a directory in PATH will yield the same result as using a package manager below.

**Package manager**

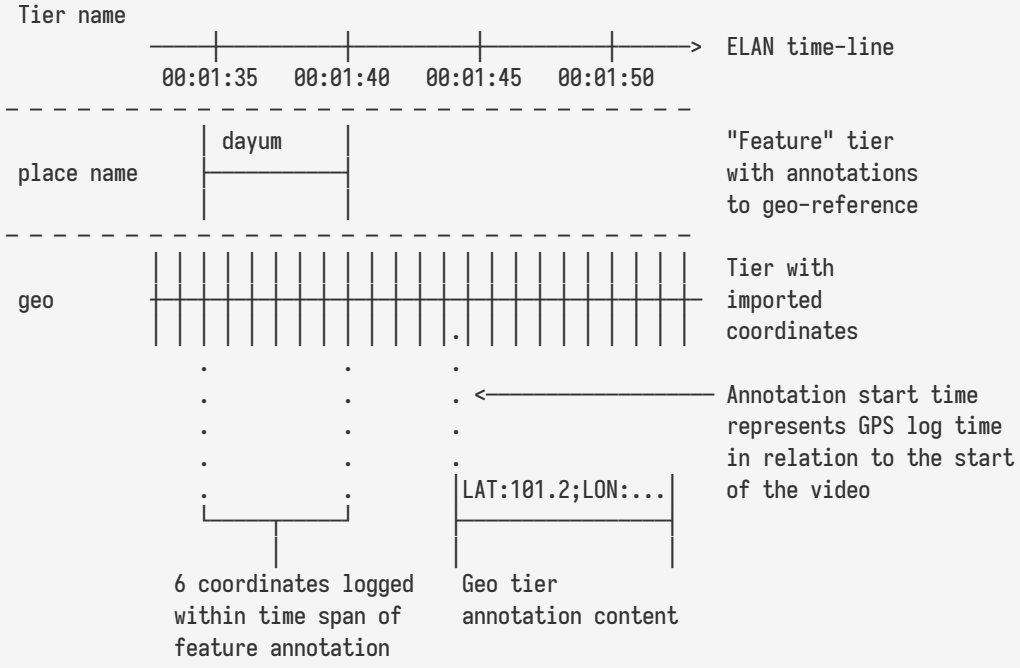
Installing via a *package manager* means the `ffmpeg` command can be executed from anywhere in a terminal. Linux distributions usually come with one pre-installed. For macOS Homebrew is a popular choice, whereas Windows has Chocolatey (or WSL). This option means you do not have to specify the location of `ffmpeg` each time `cam2eaf` is run. If a package manager is not for you, go with the *static build* for your platform.

The video and audio streams are by default only concatenated, not converted, to avoid data loss and to save time. There is also an option to convert the concatenated low-resolution clips (`.GLV`) to mpeg2-video to ensure compatibility with some older software. See the *cam2eaf* section for more information.

## Geo-data in ELAN

It is possible to import GPS-data from the relevant FIT-file into its own annotation tier (see the `--geotier` option for *cam2eaf*). Note that the ELAN-file will become quite large if full resolution 10Hz GPS-data is imported, so an option to reduce the full GPS-log into a more manageable size exists for most of the sub-commands (see *The sub-commands* and the `--downsample` option). ELAN seems to handle fairly large "geo-tiers" well as long as the "Text" tab is not selected for that tier ("Grid" works fine so far). When geo-referencing annotations, synchronising directly against the original FIT-file is currently the only option (i.e. using the "geo-tier" for this is currently not implemented). `cam2eaf` always copies the relevant FIT-file to the output directory together with the rest of the files for convenience.

*Example ELAN tier-structure with optional geo-tier*



## The sub-commands

Five sub-commands are available:

| Sub-command | Description |
|---|---|
| `cam2eaf` | Generate an ELAN-file, with corresponding concatenated media files |
| `eaf2geo` | Geo-reference ELAN-annotations and generate an annotated KML-file |
| `match` | Locate and match VIRB video clips with FIT-files |
| `check` | Inspect the contents of a FIT-file |
| `manual` | View or save this manual to disk |

The most relevant sub-commands are probably `cam2eaf` and `eaf2geo`. `match` is there to help with locating and matching which clips and FIT-files belong together, but this functionality partly exists in `cam2eaf` as well. `check` will print the data contents of a FIT-file, but will do so in an unprocessed form (e.g. longitude and latitude will be printed as semicircles, rather than degrees). It is intended more as a technical aid for troubleshooting or to verify the contents of a FIT-file.

**Time adjustment**

If the VIRB has not adjusted for the current time zone in the FIT-file, several of the sub-commands have a `--time-offset` option. It takes a +/- value in hours that will be applied to all timestamps in the output, e.g. `--time-offset 7` will add seven hours to all timestamps.

**Reducing the number of coordinates with `--downsample`**

For the sub-commands `cam2eaf`, `eaf2geo`, and `check`, the output may contain coordinates in some form (a KML-file or a geo-tier in an ELAN-file). Since the VIRB logs 10 points per second, the resulting files may become too heavy to work with in some cases, such as loading a KML-file in Google Earth. `--downsample` can be used for both ELAN-files and KML-files to reduce the number of coordinates that are imported/exported. It is especially advised for KML-files that are to be loaded into Google Earth, since a 2 hour recording may contain up towards 72 000 logged points. It takes a positive numerical value that is effectively a divisor: `--downsample 10` means an average coordinate will be calculated for every cluster of 10 points. For 72 000 logged points, a value of 100 means the output will contain 720 points and so on. If `--downsample` exceeds the total number of points logged by the GPS, it will be set to the largest applicable value (resulting in a single point for the entire recording as opposed to none at all). Extreme values may also affect the result in unexpected ways, depending on gaps in the GPS-data.

**If `cam2eaf` or `eaf2geo` return errors**

Both `cam2eaf` and `eaf2geo` have the flag `--force`. While it is very rare for a FIT-file to be corrupt, if it does occur it may still be that data can be partially extracted or even in full. If an error was returned,

try re-running, but also add `--force`. While `--force` may overcome some errors it could also have unpredictable results. Only use in the event the normal procedure fails. Before using `--force`, try the `check` sub-command on corrupt FIT-files first, since you may get a brief description of the error.

GeoELAN will never overwrite existing files without permission. Should you accidentally delete the generated ELAN-file, just re-run the `cam2eaf` sub-command. It will automatically skip concatenating videos, but still generate a new ELAN-file.

In the sub-command sections, arguments listed under 'Flags' do not take a value, whereas those listed under 'Options' do. If a `default` value is listed, it will be automatically set, unless the user specifies otherwise.

## cam2eaf

- **Basic usage:** `geoelan cam2eaf --indir INDIR/ --video VIRB0001-1.MP4 --outdir OUTDIR/`
- **Help:** `geoelan cam2eaf --help`

`cam2eaf` locates and concatenates all clips for the specified recording session found in the input directory (`--indir`) and exports a concatenated WAV-file. An ELAN-file is then generated with the media files pre-linked. The result is copied together with the corresponding FIT-file to the specified output directory (`--outdir`). By specifying the first clip in the relevant session (via `--video`, `--uuid` or `--fit`), the remaining files will be automatically located. Optionally, the resulting low-resolution video can be converted to mpeg2 (`--mpeg2`), and the high resolution clips can be copied as-is (`--copy`).

It is possible to insert the corresponding coordinates as a tier in the ELAN (`--geotier`). This can be practical for confirming that the GPS had actually started logging at the relevant time, as a means of quick confirmation of location, or even for using the ELAN-file as an all-in-one data format. If a "geo-annotation" is ever longer than 100 ms (equal to the VIRB's 10Hz logging rate), it means the GPS only logged one point within that time span. Latitude, longitude, altitude, heading and an absolute timestamp is included for each inserted point. Use `--downsample` to reduce the number of coordinates that are imported.

To preserve information such as UUIDs and the relevant FIT-file, custom metadata fields are added to

the concatenated MP4-video. If this is unwanted, use the `--no-meta` flag to opt out. Tools such as MediaInfo can show these fields. Alternatively, run `ffmpeg -i VIDEO.MP4`. The following fields are embedded:

| Field name | Description | Example value |
|---|---|---|
| `fit_file` | FIT-file | `2017-05-29-13-05-42.fit` |
| `fit_sha256` | FIT-file SHA256 checksum | `64b5039f5bfa3dbdd477b64870297dcc9680ad06ba3828fc28b4fc62349ef0cd` |
| `fit_uuid` | All UUIDs in the session | `VIRBactioncameraULTRA30_..._1_44_2017-05-29-13-05-42.fit;...` |
| `fit_start` | Session start time | `2017-05-29T11:08:34.768` |
| `fit_end` | Session end time | `2017-05-29T11:08:51.068` |

### Flags

| Short | Long | Description |
|---|---|---|
| | `--copy` | Copy, do not concatenate, high resolution clips |
| | `--force` | Try forcing a partial FIT data extraction if the process fails |
| | `--geotier` | Insert tier with synchronised coordinates in ELAN-file |
| `-l` | `--low-res-only` | Only concatenate low resolution clips (.GLV) |
| `-n` | `--no-meta` | Do not embed FIT metadata in output MP4 |
| | `--mpeg2` | Use mpeg2 compression for low-resolution video output (.GLV) |
| | `--quiet` | Do not print file-by-file search progress |

### Options

| Short | Long | Description | Default | Required |
|---|---|---|---|---|
| | `--ffmpeg` | Custom path to FFmpeg | | |
| `-f` | `--fit` | VIRB FIT-file | | unless `-u` or `-v` |
| `-i` | `--indir` | Input path for locating files | | yes |
| `-o` | `--outdir` | Output path for resulting files | `OUTPUT` | |
| `-d` | `--downsample` | Downsample factor for coordinates | `1` | |
| `-t` | `--time-offset` | Time offset in +/- hours | `0` | |
| `-u` | `--uuid` | UUID for first VIRB clip in a session | | unless `-f` or `-v` |
| `-v` | `--video` | First VIRB clip in a session | | unless `-f` or `-u` |

Recording session can be specified using one of `--fit`, `--uuid`, `--video`. These options are mutually exclusive. `--fit` returns a list of sessions present in the FIT-file, from which the user can select the relevant one. `--uuid` and `--video` require no further user input, but either must be the first in the recording session they are part of.

**Example 1**

| `geoelan` | `cam2eaf` | `-v VIRB0001-1.MP4` | `-i INDIR/` | `-o OUTDIR/` | `--geotier` |
|---|---|---|---|---|---|
| | sub-command | first clip in session | input directory | output directory | insert coordinate tier |

**Result**: Locates all clips for the recording session starting with the clip `VIRB0001-1.MP4` (`-v`) in the input directory `INDIR/` (`-i`). These will be concatenated, and the audio track exported as a WAV for use in ELAN. The resulting files are then copied together with the corresponding FIT-file to the output directory `OUTDIR/` (`-o`). The generated ELAN-file will also have synchronised coordinates inserted as a tier (`--geotier`).

**Example 2**

| `geoelan` | `cam2eaf` | `-f 2017-01-28-05-16-40.FIT` | `-i INDIR/` | `-o OUTDIR/` | `--mpeg2` | `--low-res-only` |
|---|---|---|---|---|---|---|
| - | sub-command | FIT-file | input directory | output directory | convert low-res MP4 to mpeg2 | do not concatenate hi-res MP4 |

**Result**: The relevant recording session is specified via the FIT-file `2017-01-28-05-16-40.fit` (`-f`). This presents the user with a list of sessions to select from, which allows GeoELAN to locate the clips in the input directory `INDIR/` (`-i`). Only the low-resolution clips (`--low-res-only`) will be concatenated, and also converted to mpeg2 (`--mpeg2`). All resulting files are then copied together with the corresponding FIT-file to the output directory `OUTDIR/` (`-o`).

If you are unsure of the whereabouts of the FIT-file, make the search wider. Specifying the root of an external hard drive as input directory (`--indir`) will make the search process slightly longer, but should work well. Otherwise, just specify the FIT-file separately (`--fit`), which can be useful if it is located outside of the input directory.

If you intend to use the *static build* for FFmpeg as described in *FFmpeg and video concatenation*, point to it using `--ffmpeg PATH/TO/FFMPEG/ffmpeg` (`ffmpeg.exe` on Windows). If the `--ffmpeg` option is not used, `geoelan` will assume `ffmpeg` is available as a global command and complain accordingly if it is not.

## eaf2geo

- **Basic usage:** `geoelan eaf2geo --eaf VIRB0001-1.eaf --fit 2017-01-28-05-16-40.fit`
- **Help:** `geoelan eaf2geo --help`

`eaf2geo` generates a KML-file by geo-referencing the annotations in one tier for the specified ELAN-file. The user is presented with a list of tier names to select from. Several options exist for the KML-file, depending on the `--geoshape` option. The result can be either points or polylines (see below). The resulting KML-file contains absolute timestamps and will embed annotation values as a description for any point that was logged within an annotation's timespan. If the relevant FIT-file can not be automatically located, it must be specified separately.

**Flags**

| Short | Long | Description |
|---|---|---|
| | `--cdata` | KML-option, added visuals in Google Earth |
| | `--force` | Try forcing a partial FIT data extraction if the process fails |

**Options**

| Short | Long | Description | Default | Possible | Required |
|---|---|---|---|---|---|
| `-e` | `--eaf` | ELAN-file | | | yes |
| `-f` | `--fit` | VIRB FIT-file | | | yes |

| Short | Long | Description | Default | Possible | Required |
|---|---|---|---|---|---|
| | `--geoshape` | Output options for KML-file | `point-all` | `point-all`, `point-multi`, `point-single`, `line-all`, `line-multi` | |
| `-d` | `--downsample` | Downsample factor for coordinates | `1` | | |
| `-t` | `--time-offset` | Time offset, +/- hours | `0` | | |
| `-u` | `--uuid` | UUID for first VIRB clip in a session | | | unless `-e` or `-v` |
| `-v` | `--video` | First VIRB clip in a session | | | unless `-e` or `-u` |

*Selecting an ELAN-tier to geo-reference*

```
Found the following tiers:
    1: geo                                1647 annotations. Tokenized: false
    2: default                               7 annotations. Tokenized: false
Select feature tier:
```

It is possible to geo-reference annotations on a dependent (or "referring") tier, with the caveat that it cannot be a tokenized tier, nor can any of its parents. Tokenization is there to help with glossing, but the individual tokenized annotations have no temporal connection to the media on their own. When selecting a tier this is noted, together with the number of annotations in the available tiers.

**Example**

| `geoelan` | `eaf2geo` | `-f 2017-01-28-05-16-40.fit` | `-e VIRB0001-1.eaf` | `--geoshape point-single` |
|---|---|---|---|---|
| - | sub-command | FIT-file | ELAN-file | output option |

**Result**: Geo-references annotations and generates a KML-file with a single point per annotation (`--geoshape point-single`) in the ELAN-file `VIRB0001-1.eaf` (`-e`). Since no video is specified (i.e. the first clip in the recording session), the user will be presented with a list of UUIDs in the specified FIT-file `2017-01-28-05-16-40.fit` (`-f`) to choose from, each representing the first clip in a recording session.

## The *geoshape* option

For the output KML-file, five possible `--geoshape` values are accepted:

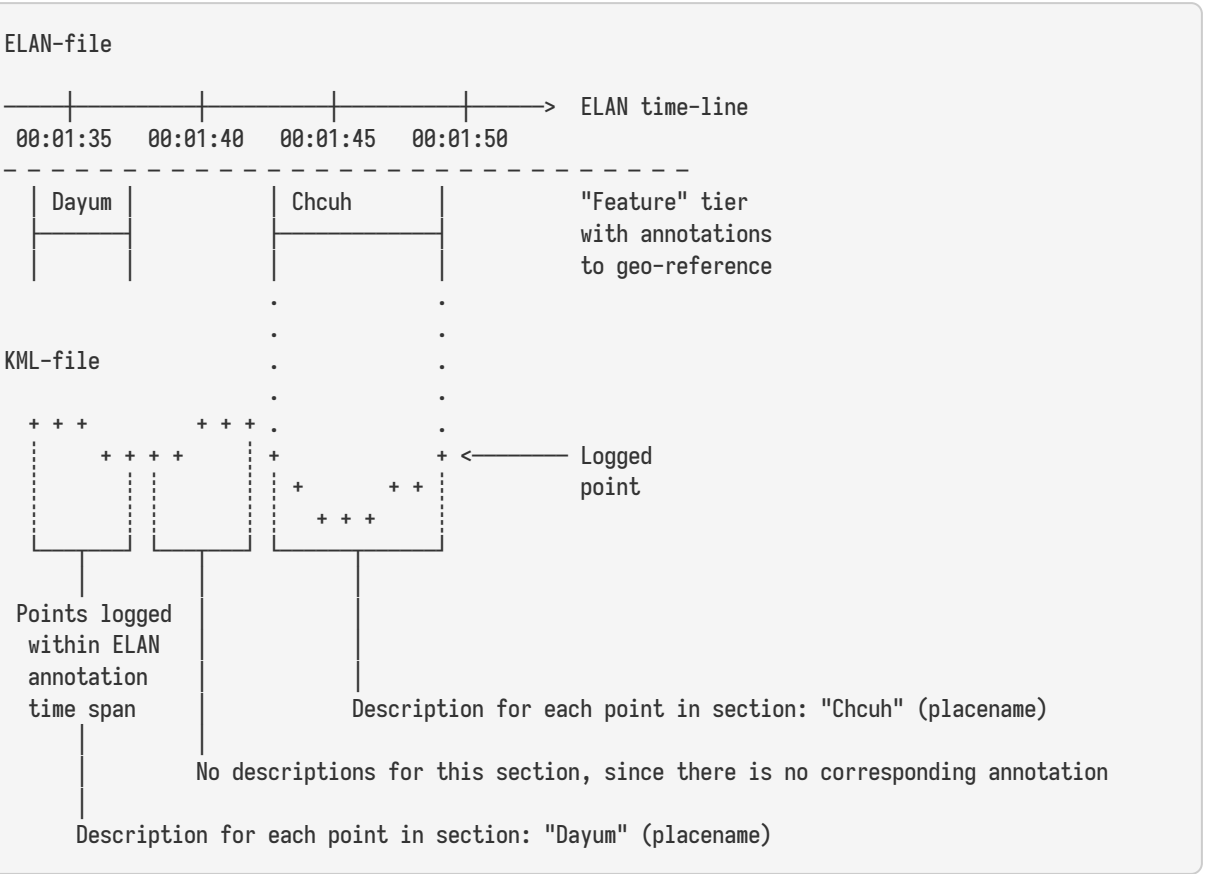
| | |
|---|---|
| `point-all` | All logged points exported (default if no option passed) |
| `point-multi` | Exported points correspond to marked/annotated events only |
| `point-single` | A single, averaged point for each annotation |
| `line-all` | Polyline from all logged points |
| `line-multi` | Polyline corresponds to marked/annotated events only |

`--downsample` can be used with all these options, but will be ignored for `point-single`.

**`point-all`**

All points logged during the recording session will be exported. Any point that intersects with the time span of an annotation will inherit the annotation text as the coordinate description. Points that do not, will have no description.

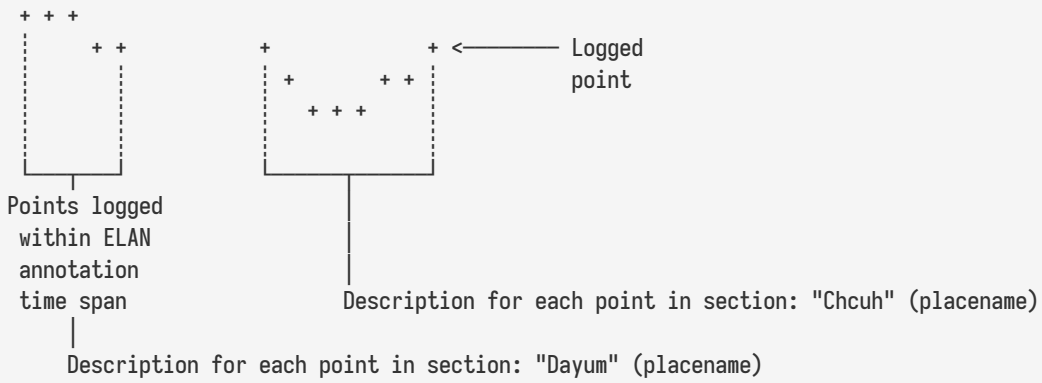
*Example `--geoshape point-all` option, KML/map output*




**`point-multi`**

Only points that intersect with the time span of an annotation will be exported and will inherit the annotation text as the coordinate description. Points that have no corresponding annotation will be discarded. *Useful for including points corresponding to marked events only*.

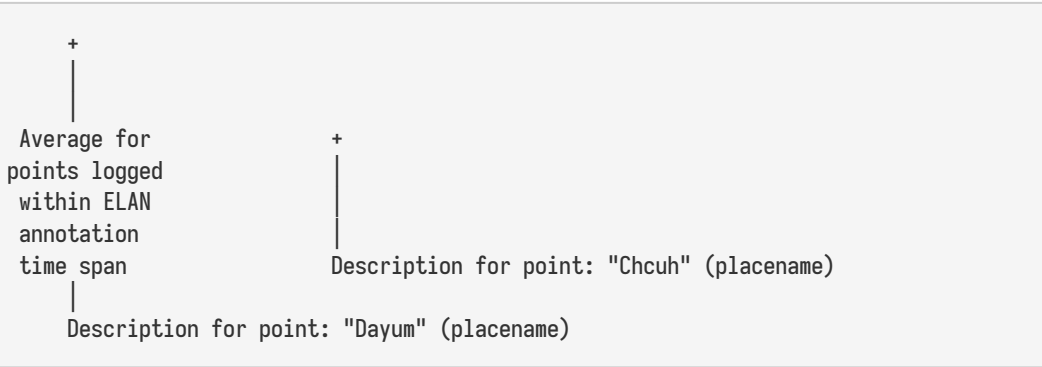
*Example* `--geoshape point-multi` *option, KML/map output*




`point-single`

Only points that intersect with the time span of an annotation will be considered for export. The difference to `point-multi` is that each annotation will only generate a single point: an average of those logged within the annotation's time span. Note that a custom `--downsample` value will be ignored for `point-single` since it may affect the result negatively (it also has little use, since the number of points in the output will not change and will be quite low compared to the other options). *Useful for distilling marked events, such as place names, to a single point for each event*.

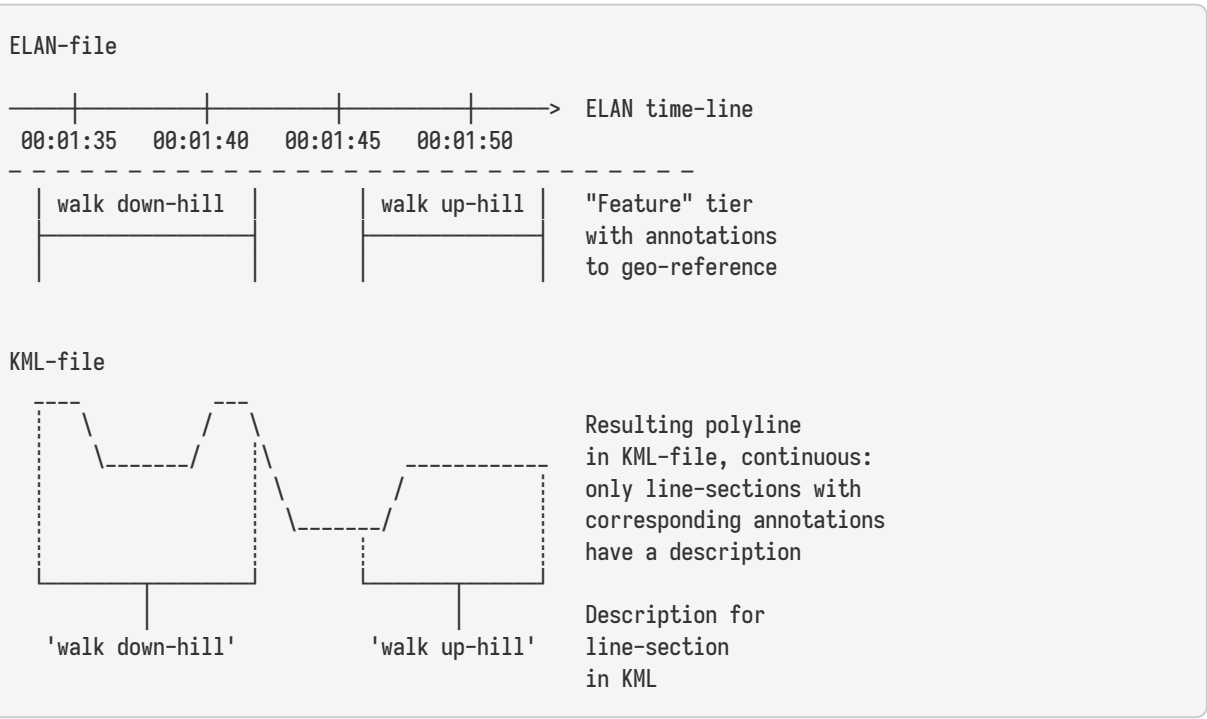
*Example* `--geoshape point-single` *option, KML/map output*




`line-all`

All points logged during the recording session will be exported, resulting in a continuous polyline. Sub-sections that intersect with an annotation inherit the annotation text as a description, whereas those that do not will have no description.

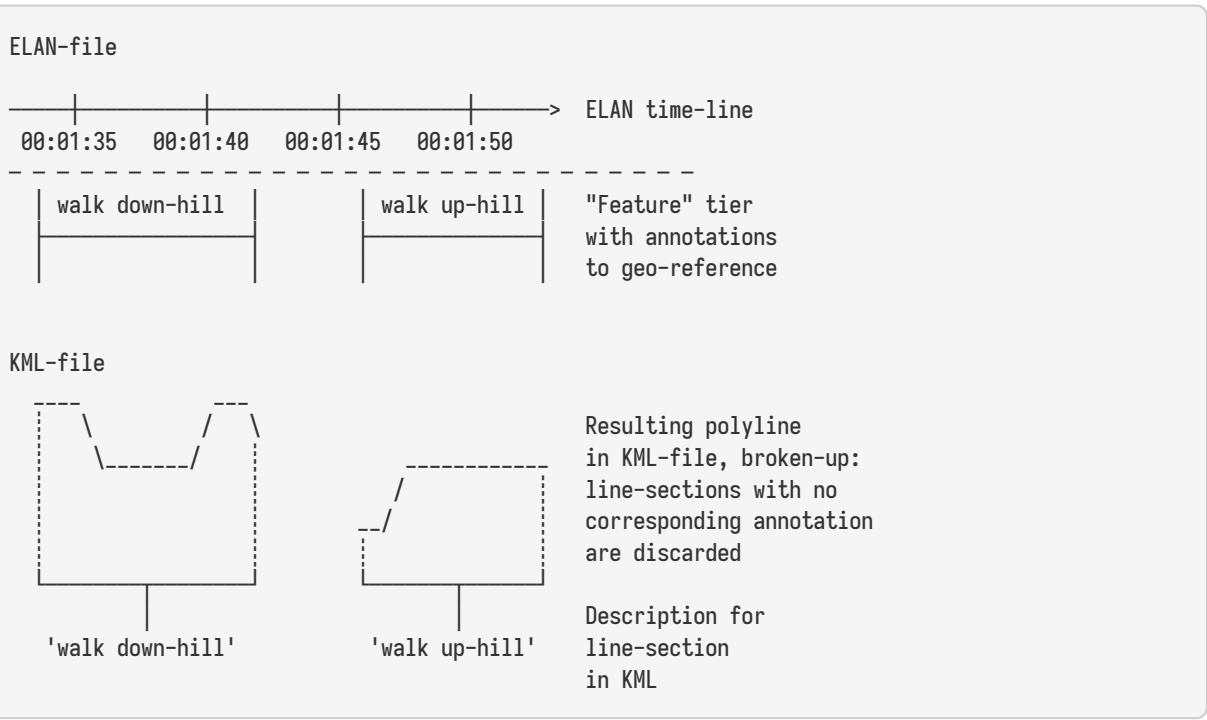
*Example* `--geoshape line-all` *option, KML/map output*




**`line-multi`**

Only points that intersect with the time span of an annotation will be exported, resulting in a broken-up line. Each sub-section inherits the text value of the annotation it intersects with. *Useful for representing paths corresponding to marked events only*.

*Example* `--geoshape line-multi` *option, KML/map output*



### The 'cdata' option

Using `--cdata` will insert extra information into the KML-file in the form of HTML inside the `<description>` element for each point (see the CDATA section in Google's KML documentation). This will cause an information bubble to pop-up in Google Earth when a point is clicked on, as a visual flair for e.g. presentations.

---

## match

- **Basic usage:** `geoelan match --indir INDIR/`
- **Help:** `geoelan match --help`

`match` will locate original VIRB clips and match these with any corresponding FIT-file/s found in the input path. By optionally specifying the first UUID (`--uuid`, `--fit`) or the first clip (`--video`) for a specific session, only paths for the files in that recording session will be returned. A CSV-file of the result can also be saved for future reference. If you are unsure of the location of all VIRB-files, use an input path closer to the root, such as the root of an external hard drive. If duplicate files are found, only the first one encountered will be reported. To include these, use the `--duplicates` flag.

**Flags**

| Short | Long | Description |
|---|---|---|
| | `--duplicates` | Include duplicate files in match results |
| | `--quiet` | Do not print file-by-file search progress |
| | `--csv` | Write result to CSV plain-text file |

**Options**

| Short | Long | Description | Required |
|---|---|---|---|
| `-f` | `--fit` | VIRB FIT-file for selecting session | |
| `-i` | `--indir` | Input path for locating files | yes |
| `-u` | `--uuid` | UUID for first VIRB clip in a session | |
| `-v` | `--video` | First VIRB clip in a session | |

**Example 1**

| geoelan | match | -i INDIR/ | --csv |
|---|---|---|---|
| | sub-command | input directory | save result as CSV |

**Result**: Returns paths to all VIRB clips found in `INDIR/` (`-i`) together with their corresponding FIT-file. The result is saved as a CSV-file to the current directory (`--csv`).

*Figure 10: CSV-layout, FITDATETIME lists the date and time for when the FIT-file on the same row was created.*

| MP4 | GLV | FIT | UUID | FITDATETIME |
|---|---|---|---|---|
| VIRB0001-1.MP4 | VIRB0001-1.GLV | 2018-10-27-22-44-41.fit | VIRBaction...16-44-22.fit | 2018-10-27T22:44:41.389 |
| VIRB0001-2.MP4 | VIRB0001-2.GLV | 2018-10-27-22-44-41.fit | VIRBaction...16-44-22.fit | 2018-10-27T22:44:41.389 |
| ... | ... | ... | ... | ... |

**Example 2**

| geoelan | match | -i INDIR/ | -v VIRB0001-1.MP4 |
|---|---|---|---|
| | sub-command | search directory | first clip in session |

**Result**: Returns paths to all clips found in `INDIR/` (`-i`) for the session starting with `VIRB0001-1.MP4` (`-v`) together with the corresponding FIT-file.

## check

- **Basic usage:** `geoelan check --fit 2017-01-28-05-16-40.fit`
- **Help:** `geoelan check --help`

`check` prints an overview or the detailed contents of a FIT-file. Options include filtering to print only a sub-set of the data, such as GPS-data only, data corresponding to a specific recording session, or both. As previously mentioned, it is more of a technical aid or for example to verify that the GPS really did

log coordinates. Optionally, a KML-file can also be generated.

**Flags**

| Short | Long | Description |
|---|---|---|
| | `--debug` | Print FIT definitions and data while parsing |
| | `--debug-unchecked` | Same as `--debug`, but strings are also unchecked UTF-8 |
| | `--kml` | Generate a KML-file |
| `-s` | `--select` | Select UUID from a list of all UUIDs present in the FIT-file |

**Options**

| Short | Long | Description | Default | Required |
|---|---|---|---|---|
| `-f` | `--fit` | FIT-file | | yes |
| `-g` | `--global-id` | FIT data type (see FIT SDK) | | |
| `-d` | `--downsample` | Downsample factor for coordinates | `1` | |
| `-u` | `--uuid` | UUID, first in session | | |
| `-v` | `--video` | VIRB video clip, first in session | | |

## Inspecting FIT data

Inside a FIT-file, data is identified by a numerical id. For example, data logged by the GPS is identified by the number `160`, also referred to as `gps_metadata` in the FIT Software Development Kit. `check` lists both identifiers in the summary table. To print a specific type of data message, find the data message of interest in the summary table, then re-run specifying `--global-id`, for example `--global-id 160` to print GPS-data to screen. Use `--select` or one of the other options (`--video`, `--uuid`) to print data limited to a specific recording session. The limitation will also apply to the optionally generated KML-file. Specifying a UUID via one of the options will also return the absolute time stamps for the start/end of the session. Many non-VIRB FIT-files, from e.g. watches, bike computers, will work with `check`. However some features, such as compressed timestamp headers are not yet implemented. In such cases, the tool will report the error and will either fail or return partial data.

The FIT Software Development Kit contains a spreadsheet, `Profile.xlsx`, which lists the kinds of data a FIT-file may contain. Not all of those apply to VIRB FIT-files, however, and a manufacturer may include undocumented data types.

If a FIT-file can not be properly parsed, GeoELAN will often return an error message that may hint at the issue. If possible, any data that could be extracted up until the error occurred will also be returned.

## Required FIT-data

For the full workflow to work, the FIT-file must contain the message types in the table below. While it is very unusual for any of these to be missing (perhaps the user turned off the GPS), if GeoELAN complains on missing FIT-data, `check` can be used to verify whether it was logged or not. Note that only VIRB FIT-files can be assumed to contain all necessary data for this workflow.

| Global ID | Message type | Description | Frequency | Required amount |
|---|---|---|---|---|
| 160 | `gps_metadata` | GPS-data (latitude, longitude, altitude…) | Logged roughly 10 times/second | One or more points within the time span of a specified recording session |
| 161 | `camera_event` | Start/end time, UUID for each clip in a recording session. Required for synchronising video with GPS data | Logged at the start/end of a session and each time a new clip is created. | Start/end messages for a recording session |
| 162 | `timestamp_correlation` | Time offset required for generating absolute time stamps | Logged only once, at the time the GPS module syncs with a satellite | Once anywhere in the FIT-file |

**Example 1**

| `geoelan` | `check` | `-f 2017-01-28-05-16-40.fit` |
|---|---|---|
| | sub-command | FIT-file |

**Result**: Prints a summary of all the data contained in the FIT-file `2017-01-28-05-16-40.fit` (`-f`), together with all logged UUIDs (each corresponding to a video clip) and checks whether the required data `160`, `161`, `162` exists or not. Use this to get the numerical global ID for a specific data type, such as GPS-data (listed as `gps_metadata`).

**Example 2**

| `geoelan` | `check` | `-f 2017-01-28-05-16-40.fit` | `-g 160` | `-s` | `--verbose` |
|---|---|---|---|---|---|
| | sub-command | FIT-file | global id | select session | print data |

**Result:** Prints data messages (`--verbose`) with the global id `160` (`-g`). The extracted data will be limited to the specified recording session, selected from a list (`-s`).

---

## manual

- **Basic usage:** `geoelan manual --pdf`
- **Help:** `geoelan manual --help`

`manual` exports or prints the contents of this file to screen. The full manual is embedded within the compiled executable for convenience. Running `geoelan manual` with no flag prints the full manual to screen.

**Flags**

| Short | Long | Description |
|---|---|---|
| | | Print full plain text version to screen |
| | `--pdf` | Save the full manual as a PDF to current directory |
| | `--pdf-a4` | Save the A4-guide as a PDF to current directory |

---

## A note on GoPro

GoPro action cameras are currently not supported. At the time this method was piloted, no public documentation for GoPro's GPMF data format existed (now available). Both FIT and GPMF are binary formats, meaning the content can't be viewed in a text editor or parsed without documentation. Since the FIT-format was already well established in other products and developer tools and documentation were freely available, a decision was made to use the Garmin VIRB Ultra 30. GPMF also lacks some of the features in FIT, such as explicit timestamps for all individual data points and data logging outside of recording video. There is no immediate plan to support GoPro, but if necessary a limited implementation may be possible.

---

## References


Larsson, Jens, Niclas Burenhult, Nicole Kruspe, Ross. S Purves, Mikael Rothstein and Peter Sercombe. 2020. Integrating behavioral and geospatial data on the timeline: towards new dimensions of analysis. *International Journal of Social Research Methodology*. doi: 10.1080/13645579.2020.1763705

ELAN (Version 5.9) [Computer software]. 2020. Nijmegen: Max Planck Institute for Psycholinguistics. Retrieved from https://archive.mpi.nl/tla/elan