

# Python3 GPS Client Tools

clazzes.org's python3 gpsclient is a python library to connect to GPSD and is packaged with a versatile command line client and GPSD simulator daemon.

## gpsclient Command Line Tool

Use the debian package `python3-clazzes-gpsclient` in order to run the following examples.

### Connect to GPSD

The following command show basic usage of `gpsclient` to connect to a running `gpsd/gpstrack-runner` instance.

```
# connect to tcp://localhost:2947 and log the received records
gpsclient

# connect to tcp://localhost:2948 and log the received records
gpsclient -u tcp://localhost:2948

# connect to tcp://localhost:2947 and view the records in a curses
interface
gpsclient -c

# connect to tcp://localhost:2948 and view the records in a curses
interface
gpsclient -c -u tcp://localhost:2948
```

### Connect to GPSD and write a gpstrack file

The following examples use `gpsclient` to log JSON records to a gzipped `gpstrack` file.

```
# connect to tcp://localhost:2947 and log the received records to tmp.
gpstrack.gz
gpsclient -q -l tmp.gpstrack.gz

# connect to tcp://localhost:2948 and log the received records to tmp.
gpstrack.gz
gpsclient -q -u tcp://localhost:2948 -l tmp.gpstrack.gz

# connect to tcp://localhost:2947, show records in a curses GUI and log
the received records to tmp.gpstrack.gz
gpsclient -c -l tmp.gpstrack.gz

# connect to tcp://localhost:2948, show records in a curses GUI and log
the received records to tmp.gpstrack.gz
gpsclient -c -u tcp://localhost:2948 -l tmp.gpstrack.gz
```

### Replay a gpstrack file

The following examples replay a previously recorded `gpstrack` instead of connecting to `GPSD/gpstrack-runner`. There are two options to modify timestamps:

1. `fake_timestamp=true` in order to emit the records with the current timestamp

2. `start_timestamp=2014-04-19T22:34:57Z` in order to map the first TPV timestamp to the given timestamp and output all subsequent records offset by the so calculated time difference between the recording timestamp and the given first timestamp.

The option `interval=1` is used to emit records in a given time intervals given in seconds or fraction of a second like `0.001`.

```
# Replay file tmp.gpstrack.gz, emitting one TPV record a second.
gpsclient -u 'tmp.gpstrack.gz?interval=1'

# Replay file tmp.gpstrack.gz, emitting one TPV record a second and
faking timestamps.
gpsclient -u 'tmp.gpstrack.gz?interval=1&fake_timestamps=true'

# Replay file tmp.gpstrack.gz, emitting one TPV record a second mapping
timestamps.
gpsclient -u 'tmp.gpstrack.gz?interval=1&start_timestamp=2014-05-02T10:
00:00Z'
```

## Getting information on a track

A summary on the contents of a gpstrack file may be retrieved by calling

```
gpsclient -q -i -u /var/spool/gpstrack-runner/20170221_BerlinUmgeb.
gpstrack.gz
```

## Converting NMEA logfiles

NMEA logfiles created by `gpsmon -l` may be converted by the schell script `gpslog2gpstrack`

```
gpslog2gpstrack tmp.log tmp.gpstrack.gz
```

## gpstrack-runner Daemon

The gpstrack-runner Daemon is contained in the debian package `gpstrack-runner` and may be started by

```
/etc/init.d/gpstrack-runner start
```

By default, the daemon listens on port 2948 (intentionally one port beside the original `gpsd` Server).

The daemon replays all gpstrack files contained in the directory `/var/spool/gpstrack-runner/` by faking timestamps and emitting one record per second.

For each newly connected client, the daemon replay all the logfiles from the beginning, the set of gpstrack files is scanned anew for each new client connection.

The behavior might be customized by editing the config file `/etc/gpstrack-runner/runner.cnf`.