# QuickAnnotator Dataset Requirements

The QuickAnnotator (QA) tool assists in the annotation of datasets that may contain various data streams like video, audio, movement, and ECG data. The tool requires the data to be organized in a specified format and location. This document describes the requirements regarding this format and location.

## Location

The QA tool can load data from any location with a directory structure as shown Figure 1. The base directory, which can have any arbitrary name (like processeddata in this example), must contain a directory that reflects a participant id in its name e.g.:

participant<id>

with

|  |  |
| --- | --- |
| <id> | Participant id (number). |

|  |
| --- |
|  |
| Figure 1 Required directory structure from which he QuickAnnotator can retrieve data streams. |

## Video data

Each participant directory must contain a **camera directory**, as the main source of the QA tool is video. The camera directory itself must contain directories which are named a date corresponding the creation date of the contained data (see Figure 1). The video and audio files are stored separately and must have a filename conforming to the following format:

<id>.<timestamp>.<extension>

with

|  |  |
| --- | --- |
| <id> | Participant id (number). |
| <timestamp> | Timestamp represented as Unix (POSIX) time. |
| <extension> | One of   * + avi   + vid.h264   + vid.h264.mkv |

## Bender data

Optionally, a **bender directory** may exist containing bender sample data which is organized into directories which are named a date corresponding the creation date of the contained data (see Figure 1). The filenames of the bender files must conform to the following format:

Samplefile<timestamp>\_<id>.txt.gz

with

|  |  |
| --- | --- |
| <id> | Participant id (number). |
| <timestamp> | Timestamp represented as Unix (POSIX) time. |

## IntelliView data

Optionally, an **intelliview directory** may exist containing sample data which is organized into directories which are named a date corresponding the creation date of the contained data. The filenames of the contained bender files must have the following format:

IntelliView\_<timestamp>\_<id>.mat

with

|  |  |
| --- | --- |
| <id> | Participant id (number). |
| <timestamp> | Timestamp represented as Unix (POSIX) time. |

Note that the timestamp must be equal to the timestamp set in the video file. The timestamp value will be used to associate the IntelliView data file (MATLAB MAT-file) with a video file.

## Time and Date Conventions

The QA tool uses the [Unix (POSIX) timestamp](https://en.wikipedia.org/wiki/Unix_time) in the filename and the date from the selected directories to sync the video stream to external data like the bender data or ECG data. The Unix (POSIX) timestamp is the number of seconds elapsed since 1-jan-1970 00:00:00 (UTC) whilst not taking into account any leap seconds.

Offsets due to timezones, daylight saving, etc. are to be set in the configuration of the tool and may not be applied to the naming of the directories nor applied to the Unix (POSIX) timestamp in the filenames.

*NOTE: The bender data files are an exception on this, as these files may not contain the correct timestamp due to errors during the acquisition. As a workaround, the QA takes the file’s last modification time logged by the file system as being the timestamp of the last sample in that particular file.*

## Step-by-Step Procedure

The following steps describe a procedure on how to setup a dataset for use with the QA application.

1. Create a directory structure (hierarchy) as stated in the section `Location'.
2. Split the videos in chunks of ~10 minutes. See [here](http://unix.stackexchange.com/questions/1670/how-can-i-use-ffmpeg-to-split-mpeg-video-into-10-minute-chunks) for some advice on how this can be done using ffmpeg. Make sure that the filenames conforms to to the format stated in the section `video data' and that the <timestamp> reflects the start time of the video correctly. Store the filenames in the correct folders.

*Note: The date of the start time determines the folder in which the video should be stored. A 10 minute video of participant 4 that started on 2016-10-16 23:55 (UTC) (and thus ended the following day at 2016-10-17 00:05 (UTC)) should therefore be stored in folder*

.\processeddata\participant4\camera\2016-10-16\

1. Split the IntelliView data accordingly. Again make sure that:

* Each file contains the signals ECG and EDR and that
* That the timestamp of the first sample of each of these signals is equal to the start time of the associated video. Again, this timestamp is stated in the filename as described in section ‘IntelliView data’.

As an example, the 10 minute video file of participant 4 which started on 2016-10-16 23:55 (UTC) is stored as

.\processeddata\participant4\camera\2016-10-16\4.1476662100.avi

The associated MATLAB-file with IntelliView signal data is stored as

.\processeddata\participant4\intelliview\2016-10-16\ IntelliView\_1476662100\_4.mat

Observe that 2016-10-16 23:55 (UTC) is equivalent to 1476662100 in Unix (POSIX) time, see also [here](https://www.epochconverter.com/) and/or [here](http://www.unixtimestamp.com/).

## Bender data format

To be specified

## IntelliView

The MATLAB MAT-file with IntelliView data should contain the following signals as variables:

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Size | Class | Description |
| ECG | 1 × 1 | struct | ECG signal |
| EDR | 1 × 1 | struct | EDR signal |

Each MATLAB structure having the following fields represents a signal:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Size | Class | Description |
| name | 1 × N | char | Signal name (here: ECG or EDR). |
| unit | 1 × N | char | Signal unit (e.g. uV). |
| sample\_time | 1 × 1 | double | Sample time of the signal in [s]. |
| values | N × 1 | double | Signal values |

Figure 2 shows an example of a signal that conforms to this specification.

|  |
| --- |
|  |
| Figure 2 Example of a MATLAB structure that can be read by the QuickAnnotator Application |