

File/Data Formats in WirelessEye

Philipp H. Kindt, philipp.kindt@informatik.tu-chemnitz.de

January 21, 2024

1 Overview

WirelessEye provides the following 3 different file formats for exporting CSI data:

- **Simple CSV Format:** A comma-separated value (CSV) format optimized for simplicity. It is the only format used for live-export (e.g., streaming data to an external program such as a classifier in real-time). It can also be used for recording data to files.
- **Compact CSV Format:** A CSV format which demands only approximately 30% of the space of the *simple CSV* format. It can be used for recording data into files.
- **WirelessEye Binary:** A non-standard, proprietary file format with minimalistic space requirements. It requires less than about 20% of the space of the *Simple CSV* format and only 60% of the space of the *Compact CSV* format. It can be used for recording data into files.

In addition, there is a dedicated exchange format to read classification results from a classifier into WirelessEye for the purpose of real-time annotations. Figure ?? gives an overview on the different formats used in different data paths. This document describes all of these formats.

2 Simple CSV Format

The *simple CSV* format uses one line per subcarrier. Data of multiple subcarriers that belong to the same frame always appear subsequently (i.e., without any data belonging to a different frame in between), in ascending order by their subcarrier index. The disadvantage of this format is that data remaining the same for each subcarrier (e.g., the timestamp and the MAC address) is repeated for every single subcarrier, thereby occupying a considerable amount of additional space.

This format is used both for live-export to an external program, as well as for data-recording. The format is identical for both purposes, barring the only

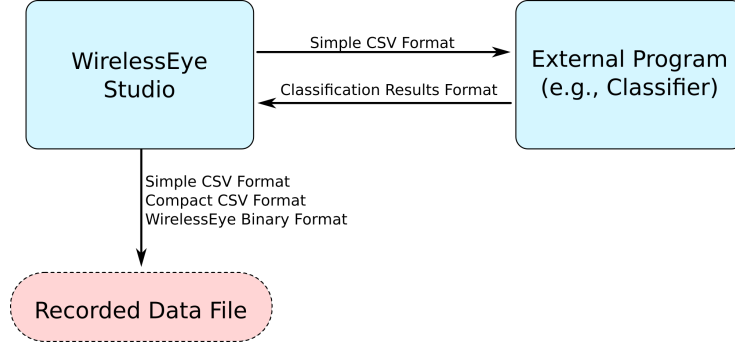


Figure 1: Overview of the formats used.

exception that for live-export, no CSV header is generated. The CSV-header (which is used for recording data into files) is as follows.

```
timestamp;MAC;subcarrier;amplitude;phase;RSSI;frame_control
```

Each line of CSV data, which belongs to a single subcarrier, consists of the following data (in this order), each separated by a semicolon (“;”).

1. The timestamp as a string “yyyy-mm-dd hh:ii:ss:uuuuuu”. *y* represents the year, *m* the month, *d* the day, *h* the hour, *i* the minute, *s* the second and *uu* the microseconds.
2. The MAC address of the sender as a string with hexadecimal numbers. Each two characters are separated by a colon. E.g., “ca:ff:ee:ca:ff:ee”.
3. The subcarrier index as a counting number.
4. The CSI amplitude as a floating point number. The decimal separator is always a dot (“.”), there are 10 digits behind the dot. Make sure that the *en_US* locale is installed before recording to ensure the right decimal separator is used.
5. The CSI phase as a floating point number. The decimal separator is a dot (“.”), there are 10 digits behind the dot.
6. The RSSI of the frame as a floating point number. The decimal separator is always a dot (“.”).
7. The *frame control* field as an integer between 0 and 255.

Example file (truncated after 10 rows of CSI data):

```
timestamp;MAC;subcarrier;amplitude;phase;RSSI;frame_control
2021-10-29 18:07:33:885002;34:31:c4:68:33:ba;0;0.0000000000;0.0000000000;-16.3306273036;128
2021-10-29 18:07:33:885002;34:31:c4:68:33:ba;1;0.0000000000;0.0000000000;-16.3306273036;128
2021-10-29 18:07:33:885002;34:31:c4:68:33:ba;2;0.0000000000;0.0000000000;-16.3306273036;128
2021-10-29 18:07:33:885002;34:31:c4:68:33:ba;3;0.0000000000;0.0000000000;-16.3306273036;128
```

```

2021-10-29 18:07:33:885002;34:31:c4:68:33:ba;4;0.0821877068;0,0000000000;-16.3306273036;128
2021-10-29 18:07:33:885002;34:31:c4:68:33:ba;5;0.0713229456;-1,7858112085;-16.3306273036;128
2021-10-29 18:07:33:885002;34:31:c4:68:33:ba;6;0.0376520662;-4.1719695729;-16.3306273036;128
2021-10-29 18:07:33:885002;34:31:c4:68:33:ba;7;0.0408393998;-4.1932428661;-16.3306273036;128
2021-10-29 18:07:33:885002;34:31:c4:68:33:ba;8;0.0091319674;-8.5526415515;-16.3306273036;128
2021-10-29 18:07:33:885002;34:31:c4:68:33:ba;9;0.0547918045;-9.1277138412;-16.3306273036;128

```

3 Compact CSV Format

A more compact data format, which can be used for recording data into files, is obtained by writing one line per frame (and not per subcarrier, as in the *simple CSI* format). The CSV header depends on the number of subcarriers (and hence on the channel bandwidth). For the most simple case of a 20 MHz channel that has 64 subcarriers, it looks as follows.

```

timestamp;MAC;RSSI;frame_control;a0;p0;a1;p1;a2;p2;a3;p3;a4;p4;a5;p5;a6;p6;a7;p7;a8;p8;a9;p9;
a10;p10;a11;p11;a12;p12;a13;p13;a14;p14;a15;p15;a16;p16;a17;p17;a18;p18;a19;p19;a20;p20;a21;
p21;a22;p22;a23;p23;a24;p24;a25;p25;a26;p26;a27;p27;a28;p28;a29;p29;a30;p30;a31;p31;a32;p32;
a33;p33;a34;p34;a35;p35;a36;p36;a37;p37;a38;p38;a39;p39;a40;p40;a41;p41;a42;p42;a43;p43;a44;
p44;a45;p45;a46;p46;a47;p47;a48;p48;a49;p49;a50;p50;a51;p51;a52;p52;a53;p53;a54;p54;a55;p55;
a56;p56;a57;p57;a58;p58;a59;p59;a60;p60;a61;p61;a62;p62;a63;p63

```

(note that there are no linebreaks in the CSV header).

Here, *axy* represents the amplitude of subcarrier *xy*. Similarly, *pxy* corresponds to the phase of subcarrier *xy*. The highest value of *xy* and hence the header length depends on the number of subcarriers. Every line contains the following data (in this order, separated by semicolons):

1. The timestamp as a string “yyyy-mm-dd hh:ii:ss:uuuuuu”. *y* represents the year, *m* the month, *d* the day, *h* the hour, *i* the minute, *s* the second and *uu* the microseconds.
2. The MAC address of the sender in hexadecimal format. Each two chars are separated by a colon. E.g., “ca:ff:ee:ca:ff:ee”.
3. The RSSI of the frame as a floating point number. The decimal separator is a dot (“.”).
4. The *frame control* field as an integer between 0 and 255.
5. As many pairs of values of CSI amplitudes and phases as there are subcarriers. Each amplitude or phase is represented by a floating point number. The decimal separator is a dot (“.”), there are 10 digits behind the dot.

An example is omitted, since already one line of data would require excessive space in this document. You obtain an example by capturing a few frames using WirelessEye. The format is actually fully explained by the header.

4 WirelessEye Binary (.wbin) Format

The *WirelessEye binary* format can be used for recording CSI data. Different from the other formats, the file ending is *.wbin* instead of *.csv*. It is structured as follows.

1. (12 Bytes) File header, consisting of the string “WirelessEyeBinary”.
2. (4 Bytes) The number of subcarriers as a 32-bit unsigned integer.
3. For every received frame:
 - (a) (16 bytes) A timestamp in format *struct timespec_16bytes* (see below). It is identical to *struct timespec*, but always has 16 bytes of size, while the size of *struct timespec* might be platform dependant. (16 bytes)
 - (b) (6 bytes) The MAC address.
 - (c) (8 bytes) The RSSI as a double precision floating point value.
 - (d) (1 byte) The frame-control field, which is usually interpreted as an unsigned integer.
 - (e) For each subcarrier of the frame:
 - i. (8 bytes) The CSI amplitude of that subcarrier as a double precision floating point value.
 - ii. (8 bytes) The CSI phase of that subcarrier as a double precision floating point value.

Below is a definition of *struct timespec_16bytes*:

```
struct timespec_16bytes{
uint64_t tv_sec;
uint64_t tv_nsec;
};
```

Figure ?? provides an example of the WirelessEye binary format. A C-program to read this data can be found under the name *readBinaryFormat.c* in the *tools* folder of the WirelessEye repository.

5 Data Format for Classification Results

The format for signaling classification results consists of pairs of the class number to which the most recent data has been assigned to, and a confidence value. Each such pair belongs to a certain classifier - the number of classifiers can be arbitrarily high. Each value is separated by a semicolon (“:”) as follows:

```
class0:confidence0:class1:confidence1:class2:confidence2,...
```

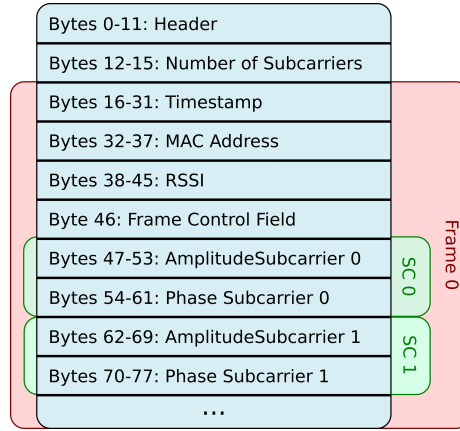


Figure 2: Example of the WirelessEye Binary (*.wbin*) format.

Here, the numbers (0,1,2,...) identify the classifier. The class number is an integer and the highest possible class number needs to be defined in the settings of WirelessEye studio. The *confidence* is a floating point value between 0 (the classifier is not at all confident in the results) and 1 (the classifier is 100 % confident in the results). Each such classifier output is terminated by a new-line. WirelessEye assumes that a classifier output belongs to all data since the previous output was received. Example:

5:0.01:9:0.99

Note that each line must contain an output of *all* classifiers. If some classifiers have not yet computed a new result, the previous result needs to be repeated in such cases.