

# 1 Overview

This article provides some information about Cli alongside with examples on how to use it.

## 2 Info

Cli provide tools to some simple S2 file manipulation. In its current form it can generate basic information about S2 file, generate CSV file from data in S2 file, select desirable part of S2 file and merge two S2 files into one s2 file. We control program with flags and their arguments.

### 2.1 Flags

Whenever we run Cli there are two mandatory flags, task flag and input flag, followed by additional flags. Only one task flag can be used at the same time. Flags arguments must follow directly after flag and in correct order.

#### 2.1.1 statistics

**-s** provides us with some statistics. It is task flag. It has no arguments. By default it will write result to standard output. If we provide **-o** it will write result to this file instead. We will get version; total time of measurement; date, time and zone of the beginning; no. of special messages, comments, definitions, time stamps, unknown, errors, streams, packets per stream, samples per stream.

#### 2.1.2 CSV

**-r** partially convert S2 file into CSV format(it is irreversible). It is task flag. Has no arguments. By default it will

- write result to standard output,
- use all handles,
- start from time 0 and end on Long.MAXVALUE = 9,223,372,036,854,775,807.

If we provide **-o** it will write result to this file instead. If we provide **-h** it will only use those handles. If we provide **-t** it will only use data from that time interval.

We will get time stamps, data, and handles.

#### 2.1.3 Cut

**-c** cut original S2 file and save new one. It is task flag. Has no arguments. It is mandatory to accompany this flag with flag **-o**. By default it will

- use all handles,

- start from time 0 and end on Long.MAXVALUE = 9,223,372,036,854,775,807.
- keep comments, special messages,...

If we provide **-h** it will only use those handles. If we provide **-t** it will only use data from that time interval. If we provide **-d** it will only use those additional data types.

#### 2.1.4 Merge

**-m** merges two original S2 files and save them into new one. It is task flag. It has mandatory argument. If argument is true it check if files correspond and if they do it will merge them on same handles. If false it will give second file new handles. It is mandatory to accompany this flag with flags **-v**, **-o**.

#### 2.1.5 Input

**-i** is primary input flag. It has 2 mandatory arguments. First argument is directory and second one is name of input S2 file. **-v** is additional input flag. It has 2 mandatory arguments. First argument is directory and second one is name of input S2 file.

#### 2.1.6 Output

**-o** is additional flag. It has 2 mandatory arguments. First argument is directory and second one is name of output file. When used with

- **-s** file, extension must be **.txt**
- **-r** file, extension must be **.csv**
- **-c** file, extension must be **.s2**
- **-m** file, extension must be **.s2**

#### 2.1.7 Handles

**-h** is additional flag. It has mandatory argument. If argument in binary form have 1 on position  $i+1$  then it will keep handle  $i$ . If argument is negative it will instead keep handle  $i$  if argument has 0 on position  $i+1$ . Argument must be between  $-2^{32}$  and  $2^{32}$

#### 2.1.8 Time interval

**-t** is additional flag. It has 2 mandatory arguments. First argument must be smaller than second one. The arguments represent time interval.

### 2.1.9 Data types

**-d** is additional flag. It has mandatory argument. If we want comments, last digit must be 1. If we want special messages, second last digit must be 1. If we want meta data, third last digit must be 1.

## 3 Examples

In all examples to follow we will assume we have two **S2** files named **file1.s2** and **file2.s2** and both stored in **C:\files**. Examples are independent.

### 3.1 example

Lets say file1.s2 stores data about EKG measurement and we want to know how long did it last. For that we call Cli as follows.

```
Cli -s -i C:\files file1.s2
```

**-i** is always necessary and has 2 mandatory arguments file directory and name. After running the program we will get statistics of **file1.s2** on standard output.

Now we want to know the actual data for the first 30s. We want them saved in file **output1.csv** for later use:

```
Cli -r -i C:\files file1.s2 -o C:\files output1.csv -t 0 30
```

Flag **-r** tels the program we want actual data in CSV format, **-o** has 2 necessary arguments directory and name of file in which we will save our CSV data. There is also **-t** with 2 arguments which represent time interval.

### 3.2 example

Lets say the measurement on **file2.s2** is too long. We decide we only want part of data between 45s and 75s since the beginning. We call Cli as follows.

```
Cli -c -i C:\files file2.s2 -o C:\files newFile2.s2 -t 45 75
```

Now we want data from **file1.s2** and **newFile2.s2** to be in the same S2 file named **merged.s2**

```
Cli -m false -i C:\files file1.s2 -v C:\files newFile2.s2 -o C:\files merged.s2
```

### 3.3 example

Let say we have 2 mesurments saved on **file1.s2**. We are particularly interested in data between 25-30 seconds and 130-205 seconds in first mesurement. First we cut each part out and save it.

```
Cli -c -i C:\files file1.s2 -o C:\files cut1.s2 -t 25 30 -h 0
```

```
Cli -c -i C:\files file1.s2 -o C:\files cut2.s2 -t 130 205 -h 0
```

Now we merge them back into new file. Since they have data from same initial measurement we want them to look like it.

```
Cli -m true -i C:\files cut1.s2 -v C:\files cut2.s2 -o C:\files merged.s2
```