
senaite.serial.cli Documentation

Release 1.0.2

**Riding Bytes
Naralabs**

Nov 16, 2020

Contents

| | | |
|----------|-------------------------------------|-----------|
| 1 | Installation | 3 |
| 2 | Quickstart | 5 |
| 3 | Testing | 7 |
| 3.1 | Send messages from a file | 7 |
| 3.2 | Escape characters | 8 |
| 4 | Examples | 9 |
| 4.1 | Example 1 | 9 |
| 5 | License | 11 |

This package provides a command line for interfacing systems to SENAITE LIMS across the serial RS-232 communications port.

This interface is compliant with the industry supported standard CLSI (Clinical and Laboratory Standards Institute, formerly NCCLS) [LIS1-A](#) “*Specification for Low-Level Protocol to Transfer Messages Between Clinical Laboratory Instruments and Computer Systems*”, a revision of [ASTM E1381-02](#).

However, this tool is meant to be generic enough so it can be extended with other communication protocols and standards.

Note: In 2001, [ASTM Committee E31](#) decided to restructure its operations, with the intent of focusing on standards-development issues such as security, privacy, and the electronic health record. Part of the reorganization plan was to transfer responsibility for E31.13 standards to NCCLS. Following this transfer, standard ASTM E1381 was redesignated as NCCLS standards LIS1.

This command line tool works together with [senaite.lis2a](#), an add-on that enables the reception and interpretation of CLSI [LIS2-A2](#) messages for [SENAITE LIMS](#).

Table of Contents:

CHAPTER 1

Installation

You can install this tool by directly using *pip*:

```
$ pip install senaite.serial.cli
```

If you've downloaded the source code, you can install the package with:

```
$ pip install -e .
```


CHAPTER 2

Quickstart

Check the `--help` from `senaite.serial.cli` with:

```
$ senaite_serial -h
usage: senaite_serial [-h] [-v] [-b BAUDRATE] [-u URL] [-r RETRIES] [-d DELAY]
                    [-t]
                    port

SENAITE Serial client interface

positional arguments:
  port                  COM Port to connect. Serial client will listen to this
                        port for incoming data and use this same port to send
                        data back

optional arguments:
  -h, --help            show this help message and exit
  -v, --verbose          Verbose logging (default: False)
  -b BAUDRATE, --baudrate BAUDRATE
                        Baudrate (default: 9600)
  -u URL, --url URL     SENAITE full URL address, with username and password:
                        'http(s)://<user>:<password>@<senaite_url>'. (default:
                        None)
  -r RETRIES, --retries RETRIES
                        Number of attempts of reconnection when SENAITE
                        instance is not reachable. Only has effect when
                        argument --url is set (default: 3)
  -d DELAY, --delay DELAY
                        Time delay in seconds between retries when SENAITE
                        instance is not reachable. Only has effect when
                        argument --url is set (default: 5)
  -t, --dry-run          Dry run. Data won't be sent to SENAITE instance. This
                        argument only has effect when argument --url is set
                        (default: False)
```


CHAPTER 3

Testing

It is possible to use *socat* to create a pair of virtual serial ports:

```
$ socat -d -d pty,raw,echo=0 pty,raw,echo=0
2019/05/02 17:26:21 socat[24050] N PTY is /dev/ttys002
2019/05/02 17:26:21 socat[24050] N PTY is /dev/ttys007
```

In this example, two serial ports *ttys002* and *ttys007* were created.

Now we can start the command tool listening on one serial port:

```
$ serial -v /dev/ttys007
Listening on port /dev/ttys007, press Ctrl+c to exit.
```

And send messages to the other port:

```
$ echo "H|Hello|Serial|Port|\nL|1" > /dev/ttys002
```

3.1 Send messages from a file

If you have a file with the data to be transmitted, you can make use of *sed* to read and send the contents of the file line by line. Create a bash script file “send” first, like follows:

```
#!/bin/bash
echo -n -e `sed -n $3p $1` > $2
```

Give execution permissions:

```
$ chmod +x send
```

Having a file “input.txt”, you can send the contents of this file sequentially, by executing the following command:

```
$ send input.txt /dev/tty02 1
```

Where the last number represents the line number from the input file to send.

3.2 Escape characters

CHAPTER 4

Examples

4.1 Example 1

Input:

```
\x05
\x021H|\^&||ALCZC12033KD^Roche^AMPLILINK^3.3.7.1201^Roche ASTM+^CZC12033KD^169.254.
↪218.147|||||||1|20190822160121\x0D\x0324\x0D\x0A
\x022P|1\x0D\x033F\x0D\x0A
\x023O|1|BP19-24277|BP19-24277|^^^ALL||20190819215010||||A\x0D\x0370\x0D\x0A
\x024R|1|^^^HI2CAP96|61|cp/mL|20^10000000^
↪TiterRanges|N||V||BVURE|20190820092419|20190820122902|Cobas TaqMan\x0D\x0363\x0D\x0A
\x025C|1||Accepted|G\x0D\x0309\x0D\x0A
\x026C|2|I|TM40^ STEP_CORR-2|I\x0D\x03F0\x0D\x0A
\x027L|1|N\x0D\x030A\x0D\x0A
\x04
```

Output:

```
$ senaite_serial /dev/pts/7
Listening on port /dev/pts/7, press Ctrl+c to exit.

* Establishment Phase completed
* Transfer Phase started ...
Frame 1 received
Frame 2 received
Frame 3 received
Frame 4 received
Frame 5 received
Frame 6 received
Frame 7 received
* Transfer Phase completed
-----
H|\^&||ALCZC12033KD^Roche^AMPLILINK^3.3.7.1201^Roche ASTM+^CZC12033KD^169.254.218.
↪147|||||||1|20190822160121
```

(continues on next page)

(continued from previous page)

```
P|1
O|1|BP19-24277|BP19-24277|^^^ALL||20190819215010||||A
R|1|^^^HI2CAP96|61|cp/mL|20^10000000^
↪TiterRanges|N||V||BVURE|20190820092419|20190820122902|Cobas TaqMan
C|1||Accepted|G
C|2|I|TM40^ STEP_CORR-2|I
L|1|N
-----
* Entering Neutral state
```

CHAPTER 5

License

SENAITE.SERIAL.CLI Copyright (C) 2020 RIDING BYTES & NARALABS

SENAITE.SERIAL.CLI is available under the terms of the [GNU General Public License, version 2](#) as published by the [Free Software Foundation](#).

The source code of this software, together with a copy of the license can be found at this repository: <https://github.com/senaite/senaite.core>

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.