
NetSSH2 Documentation

Release 0.1.7

Jakub Krysl

Jul 29, 2019

Contents:

1	About NetSSH2	3
2	Installation	5
3	Installation from source	7
4	API Documentation	9
4.1	netssh2.session	9
4.2	netssh2.exceptions	11
4.3	netssh2.globals	11
5	Changelog	13
5.1	0.1.7 (2019-04-26)	13
5.2	0.1.6 (2019-04-12)	13
5.3	0.1.5 (2019-04-02)	14
5.4	0.1.4 (2019-03-27)	14
5.5	0.1.3 (2019-03-26)	14
5.6	0.1.2 (2019-03-26)	14
5.7	0.1.1 (2019-03-26)	14
5.8	0.1.0 (2019-03-26)	15
5.9	0.0.1 (2019-03-21)	15
6	Indices and tables	17
	Python Module Index	19
	Index	21

Library for communicating with network devices using [ssh2-python](#).

CHAPTER 1

About NetSSH2

The goals of this project are to create alternative to [netmiko](#) by replacing *Paramiko* with *ssh2-python*.

As *ssh-python* is designed to be faster than *Paramiko*, so should this library.

Contributions are most welcome!

CHAPTER 2

Installation

The recommended installation method is using *pip*.

```
pip install netssh2
```


CHAPTER 3

Installation from source

```
git clone https://gitlab.com/jkrysl/netssh2
cd netssh2
python setup.py install
```


4.1 netssh2.session

Module to handle basic session

```
class netssh2.session.Session(host="", user="", passwd="", key_file_private=None,  
                             key_file_public=None, key_passwd="", port=22, timeout=5000,  
                             buffer_size=1024, target_buffer_size=1024, auth_retries=5,  
                             auth_delay=2000, socket_retries=100, socket_delay=100,  
                             prompt=None, command_prompt=None, header_newline=True,  
                             newline='n', invoke_shell=False, verbose=True, log-  
                             ging_level='INFO')
```

Defines default session to be used across multiple vendors.

authenticate_session()

Tries to authenticate existing session

Raises netssh2.exceptions.NetSsh2AuthenticationError when authentication fails

Raises netssh2.exceptions.NetSsh2TooManyRetriesException if number of auth_retries is exceeded

Raises netssh2.exceptions.Exception if any other issue happens when authenticating

Returns True

Return type bool

clean_shell_header()

Gets rid of interactive shell header. :raises: netssh2.exceptions.NetSsh2Timeout in case of timeout when reading the channel.

Returns True

Return type bool

close_channel (*keep_alive=False*)

Closes channel and returns exist status. :param keep_alive: Should we keep the channel open and just get exist status? default False :type keep_alive: bool :return: channel exit status :rtype: int

configure_channel ()

Does any channel configuration, for example invoking interactive shell if invoke_shell=True. :return: True :rtype: bool

configure_session ()

Sets up nonblocking mode, which allows to wait for socket to be ready and more control over channel. :return: True :rtype: bool

create_session ()

Connect to a host using ssh :return: True :rtype: bool

disconnect ()

Disconnect from a ssh session :return: True :rtype: bool

open_channel ()

Opens channel on ssh2 session. :raises: netssh2.exceptions.NetSsh2ChannelException if the channel is not available for some reason.

Returns True

Return type bool

read_chan (*buff_size=None, stderr=False*)

Read channel output in non blocking way. :param buff_size: Buffer size to read from the channel :type buff_size: int

Parameters **stderr** (*bool*) – Do we want to read STDERR (True) instead of STDOUT (False)?

Raises netssh2.exceptions.NetSsh2Timeout in case of timeout when reading channel.

Returns (size, buffer) size and payload read from the channel

Return type tuple(int, str)

read_output ()

Reads output of the command we sent. :return: True :rtype: bool

read_stderr ()

Reads 1024 chars from channel STDERR. :return: size of buffer that was read :rtype: int

read_stdout ()

Reads 1024 chars from channel STDOUT. :return: size of buffer that was read :rtype: int

read_until_prompt ()

Keep reading channel until we get prompt. :return: True :rtype: bool

run_cmd (*cmd, return_output=False*)

Run a command to a specific ssh session :param cmd: command to run over ssh :type cmd: string

Parameters **return_output** (*bool*) – Should we return also output

Returns exit_status of command or (exit_status, output) in case of return_output=True

Return type int or tuple(int, str)

send_cmd (*cmd*)

Sends command over ssh2 channel. :param cmd: command to send :type cmd: string

Raises netssh2.exceptions.NetSsh2Timeout in case of timeout

Returns True

Return type bool

send_scp (*source_file*, *destination_path*)

Send file over SCP. :param source_file: Path to source file :type source_file: string :param destination_path: Path on server to copy file to :type destination_path: string :return: exist status :rtype: string

set_prompt (*new_prompt*)

Setting different prompt with opened session. :param new_prompt: new string to wait for after command with invoke_shell :type new_prompt: string

Returns True

Return type bool

set_timeout (*new_timeout*)

Setting different timeout with opened session. :param new_timeout: New timeout value in milliseconds. :type new_timeout: int

Returns True

Return type bool

4.2 netssh2.exceptions

Module for defining netssh2 exceptions

exception netssh2.exceptions.**NetSsh2AuthenticationError**

Base exception for any authentication errors

exception netssh2.exceptions.**NetSsh2ChannelException**

Base exception for any channel errors

exception netssh2.exceptions.**NetSsh2HostError**

Base exception for any host errors

exception netssh2.exceptions.**NetSsh2Timeout**

Base exception for any timeout errors

exception netssh2.exceptions.**NetSsh2TooManyRetriesException**

Base exception when ran out of retries.

4.3 netssh2.globals

File for storing global variables

netssh2.globals.**LIBSSH2_ERROR_EAGAIN** = -37

SSH2 return value of blocked socket

5.1 0.1.7 (2019-04-26)

- Make sure command timeout works.
- Add tests for logging.
- Allow to change logging level on Session init plus some more debug.
- Move away from unittest to pure py.test.
- Allow to use key authentication.
- Wrap channel.write for nonblocking transfer.
- Add scp_send.
- Split command to multiple if it is longer than target buffer.
- Add toggle for sending newline when getting rid of the header.

5.2 0.1.6 (2019-04-12)

- Allow session with no pass.
- Allow to specify read buffer size.
- Add message level to log message.
- No need to merge STDOUT and STDERR with invoke shell.
- Add lots of debug messages for easier debugging.
- Add debug on SocketRecvError.
- Wrap encoding and decoding of strings to functions.
- Do not retry socket connection if we got timeout.

- Print output to debug on timeout when waiting for prompt.

5.3 0.1.5 (2019-04-02)

- Allow changing logging level on the run.
- Fix getting output from when running without interactive shell.

5.4 0.1.4 (2019-03-27)

- Upload to PyPi if pipeline is passing.
- Run CI build job every time (has build checks).
- Fix tag message.
- Specify long_description_content_type in setup.py to avoid warns.
- Move pushing to PyPi to CI and add wheels.
- Add setup.cfg with wheel specs.

5.5 0.1.3 (2019-03-26)

- Push to pypi only when twine check passes.
- Use only subject of commit in changelog.
- Push also tags when bumping version.

5.6 0.1.2 (2019-03-26)

- Add automatic MANIFEST.in updating to bump_version.sh.
- Add manifest file to include everything to built packages.
- Add script for automated version bumping.
- Add .gitignore.

5.7 0.1.1 (2019-03-26)

- Fix date of 0.0.1 release.
- Fix docs link in README.
- Add changelog to documentation.
- Add license metadata setup.py.
- Add changelog to setup.py long description (pypi).
- Have only GPL v3 on PyPi.

5.8 0.1.0 (2019-03-26)

- Add changelog.
- Improve README.
- Add pypi identifiers to setup.py.
- Add RST docs to be generated with sphinx using ‘make html’.
- Fix python envs in tox.
- Add code coverage to tox test.
- Change command prompt too if changing prompt of existing session.
- Add basic tests for tox.
- Cleanup when disconnecting.
- Add messages to assertions.
- Catch error when given invalid hostname.
- Add linting CI.

5.9 0.0.1 (2019-03-21)

- Basic implementation of netssh2 to work with paralel SSH and interactive shell.
- Add license.
- Initial commit.

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`

n

`netssh2.exceptions`, [11](#)
`netssh2.globals`, [11](#)
`netssh2.session`, [9](#)

A

`authenticate_session()`
(*netssh2.session.Session method*), 9

C

`clean_shell_header()` (*netssh2.session.Session method*), 9

`close_channel()` (*netssh2.session.Session method*), 9

`configure_channel()` (*netssh2.session.Session method*), 10

`configure_session()` (*netssh2.session.Session method*), 10

`create_session()` (*netssh2.session.Session method*), 10

D

`disconnect()` (*netssh2.session.Session method*), 10

L

`LIBSSH2_ERROR_EAGAIN` (in *module netssh2.globals*), 11

N

`netssh2.exceptions` (*module*), 11

`netssh2.globals` (*module*), 11

`netssh2.session` (*module*), 9

`NetSsh2AuthenticationError`, 11

`NetSsh2ChannelException`, 11

`NetSsh2HostError`, 11

`NetSsh2Timeout`, 11

`NetSsh2TooManyRetriesException`, 11

O

`open_channel()` (*netssh2.session.Session method*), 10

R

`read_chan()` (*netssh2.session.Session method*), 10

`read_output()` (*netssh2.session.Session method*), 10

`read_stderr()` (*netssh2.session.Session method*), 10

`read_stdout()` (*netssh2.session.Session method*), 10

`read_until_prompt()` (*netssh2.session.Session method*), 10

`run_cmd()` (*netssh2.session.Session method*), 10

S

`send_cmd()` (*netssh2.session.Session method*), 10

`send_scp()` (*netssh2.session.Session method*), 11

`Session` (*class in netssh2.session*), 9

`set_prompt()` (*netssh2.session.Session method*), 11

`set_timeout()` (*netssh2.session.Session method*), 11