

---

# Easy SNMP

*Release*

Nov 11, 2017



---

## Contents

---

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Why Another Library?</b>	<b>5</b>
<b>3</b>	<b>Installation</b>	<b>7</b>
<b>4</b>	<b>Quick Start</b>	<b>9</b>
<b>5</b>	<b>API</b>	<b>11</b>
5.1	Session API . . . . .	11
5.2	Easy API . . . . .	14
5.3	Exceptions . . . . .	15
	<b>Python Module Index</b>	<b>17</b>



*A blazingly fast and Pythonic SNMP library based on the official Net-SNMP bindings*



# CHAPTER 1

---

## Introduction

---

This is a fork of the official [Net-SNMP Python Bindings](#) but attempts to bring a more Pythonic interface to the library. Check out the [Net-SNMP website](#) for more information about SNMP.

This module provides a full featured SNMP client API supporting all dialects of the SNMP protocol.





---

### Why Another Library?

---

- The [original Net-SNMP Python library](#) is a great starting point but is quite un-Pythonic and lacks proper unit tests and documentation.
- [PySNMP](#) is written in pure Python and therefore has a huge performance hit. In some brief tests, I estimate that both the Net-SNMP Python bindings and Easy SNMP are more than 4 times faster. Further to this, PySNMP has an even less Pythonic interface than the official Net-SNMP bindings.
- Many other libraries like [Snimpy](#) are sadly based on PySNMP and so they suffer the same performance penalty.



## CHAPTER 3

---

### Installation

---

Easy SNMP has been tested and is supported on systems running Net-SNMP 5.7.x.

If your OS ships with this version of Net-SNMP, then you can install it without compiling it via your package manager:

On RHEL / CentOS systems:

```
sudo yum install net-snmp-devel
```

On Debian / Ubuntu systems:

```
sudo apt-get install libsnmp-dev snmp-mibs-downloader
```

If your OS doesn't ship with Net-SNMP 5.7.x, please follow instructions provided on the [Net-SNMP install page](<http://www.net-snmp.org/docs/INSTALL.html>) to build and install Net-SNMP 5.7.x on your system.

You'll also need to ensure that you have the following packages installed so that Easy SNMP installs correctly:

On RHEL / CentOS systems:

```
sudo yum install gcc python-devel
```

On Debian / Ubuntu systems:

```
sudo apt-get install gcc python-dev
```

Install Easy SNMP via pip as follows:

```
pip install easysnmp
```



## CHAPTER 4

---

### Quick Start

---

There are primarily two ways you can use the Easy SNMP library.

The first is with the use of a Session object which is most suitable when you are planning on requesting multiple pieces of SNMP data from a source.

```
from easysnmp import Session

# Create an SNMP session to be used for all our requests
session = Session(hostname='localhost', community='public', version=2)

# You may retrieve an individual OID using an SNMP GET
location = session.get('sysLocation.0')

# You may also specify the OID as a tuple (name, index)
# Note: the index is specified as a string as it can be of other types than
# just a regular integer
contact = session.get(('sysContact', '0'))

# And of course, you may use the numeric OID too
description = session.get('.1.3.6.1.2.1.1.1.0')

# Set a variable using an SNMP SET
session.set('sysLocation.0', 'The SNMP Lab')

# Perform an SNMP walk
system_items = session.walk('system')

# Each returned item can be used normally as its related type (str or int)
# but also has several extended attributes with SNMP-specific information
for item in system_items:
    print '{oid}.{oid_index} {snmp_type} = {value}'.format(
        oid=item.oid,
        oid_index=item.oid_index,
        snmp_type=item.snmp_type,
        value=item.value
```

```
)
```

You may also use Easy SNMP via its simple interface which is intended for one-off operations where you wish to specify all details in the request:

```
from easysnmp import snmp_get, snmp_set, snmp_walk

# Grab a single piece of information using an SNMP GET
snmp_get('sysDescr.0', hostname='localhost', community='public', version=1)

# Perform an SNMP SET to update data
snmp_set(
    'sysLocation.0', 'My Cool Place',
    hostname='localhost', community='public', version=1
)

# Perform an SNMP walk
snmp_walk('system', hostname='localhost', community='public', version=1)
```

## 5.1 Session API

```
class easysnmp.Session(hostname=u'localhost', version=3, community=u'public', timeout=1, re-
tries=3, remote_port=0, local_port=0, security_level=u'no_auth_or_privacy',
security_username=u'initial', privacy_protocol=u'DEFAULT', pri-
vacy_password=u'', auth_protocol=u'DEFAULT', auth_password=u'',
context_engine_id=u'', security_engine_id=u'', context=u'', en-
gine_boots=0, engine_time=0, our_identity=u'', their_identity=u'',
their_hostname=u'', trust_cert=u'', use_long_names=False,
use_numeric=False, use_sprint_value=False, use_enums=False, best_guess=0,
retry_no_such=False, abort_on_nonexistent=False)
```

A Net-SNMP session which may be setup once and then used to query and manipulate SNMP data.

### Parameters

- **hostname** – hostname or IP address of SNMP agent
- **version** – the SNMP version to use; 1, 2 (equivalent to 2c) or 3
- **community** – SNMP community string (used for both R/W) (v1 & v2)
- **timeout** – seconds before retry
- **retries** – retries before failure
- **remote\_port** – allow remote UDP port to be overridden (this will communicate on port 161 at its default setting)
- **local\_port** – allow overriding of the local SNMP port
- **security\_level** – security level (no\_auth\_or\_privacy, auth\_without\_privacy or auth\_with\_privacy) (v3)
- **security\_username** – security name (v3)
- **privacy\_protocol** – privacy protocol (v3)

- **privacy\_password** – privacy passphrase (v3)
- **auth\_protocol** – authentication protocol (MD5 or SHA) (v3)
- **auth\_password** – authentication passphrase (v3)
- **context\_engine\_id** – context engine ID, will be probed if not supplied (v3)
- **security\_engine\_id** – security engine ID, will be probed if not supplied (v3)
- **context** – context name (v3)
- **engine\_boots** – the number of times the SNMP engine has re-booted/re-initialized since SNMP engine ID was last configured (v3)
- **engine\_time** – the number of seconds since the engine\_boots counter was last incremented (v3)
- **our\_identity** – the fingerprint or file name for the local X.509 certificate to use for our identity (run net-snmp-cert to create and manage certificates) (v3 TLS / DTLS)
- **their\_identity** – the fingerprint or file name for the local X.509 certificate to use for their identity (v3 TLS / DTLS)
- **their\_hostname** – their hostname to expect; either their\_hostname or a trusted certificate plus a hostname is needed to validate the server is the proper server (v3 TLS / DTLS)
- **trust\_cert** – a trusted certificate to use for validating certificates; typically this would be a CA certificate (v3 TLS / DTLS)
- **use\_long\_names** – set to True to have <tags> for getnext methods generated preferring longer Mib name convention (e.g., system.sysDescr vs just sysDescr)
- **use\_numeric** – set to True to have <tags> returned by the get methods untranslated (i.e. dotted-decimal). Setting the use\_long\_names value for the session is highly recommended
- **use\_sprint\_value** – set to True to have return values for get and getnext methods formatted with the libraries sprint\_value function. This will result in certain data types being returned in non-canonical format Note: values returned with this option set may not be appropriate for set operations
- **use\_enums** – set to True to have integer return values converted to enumeration identifiers if possible, these values will also be acceptable when supplied to set operations
- **best\_guess** – this setting controls how oids are parsed; setting to 0 causes a regular lookup. setting to 1 causes a regular expression match (defined as -lb in snmpcmd); setting to 2 causes a random access lookup (defined as -IR in snmpcmd).
- **retry\_no\_such** – if enabled NOSUCH errors in get pdus will be repaired, removing the SNMP variable in error, and resent; undef will be returned for all NOSUCH SNMP variables, when set to False this feature is disabled and the entire get request will fail on any NOSUCH error (applies to v1 only)
- **abort\_on\_nonexistent** – raise an exception if no object or no instance is found for the given oid and oid index

**bulkwalk** (*oids=u'.1.3.6.1.2.1', non\_repeaters=0, max\_repetitions=10*)

Uses SNMP GETBULK operation using the prepared session to automatically retrieve multiple pieces of information in an OID :param oids: you may pass in a single item (multiple values currently

experimental) which may be a string representing the entire OID (e.g. 'sysDescr.0') or may be a tuple containing the name as its first item and index as its second (e.g. ('sysDescr', 0))

**Returns** a list of SNMPVariable objects containing the values that were retrieved via SNMP



**get** (*oids*)

Perform an SNMP GET operation using the prepared session to retrieve a particular piece of information.

**Parameters** **oids** – you may pass in a list of OIDs or single item; each item may be a string representing the entire OID (e.g. 'sysDescr.0') or may be a tuple containing the name as its first item and index as its second (e.g. ('sysDescr', 0))

**Returns** an SNMPVariable object containing the value that was retrieved or a list of objects when you send in a list of OIDs

**get\_bulk** (*oids, non\_repeaters=0, max\_repetitions=10*)

Performs a bulk SNMP GET operation using the prepared session to retrieve multiple pieces of information in a single packet.

**Parameters**

- **oids** – you may pass in a list of OIDs or single item; each item may be a string representing the entire OID (e.g. 'sysDescr.0') or may be a tuple containing the name as its first item and index as its second (e.g. ('sysDescr', 0))
- **non\_repeaters** – the number of objects that are only expected to return a single GET-NEXT instance, not multiple instances
- **max\_repetitions** – the number of objects that should be returned for all the repeating OIDs

**Returns** a list of SNMPVariable objects containing the values that were retrieved via SNMP

**get\_next** (*oids*)

Uses an SNMP GETNEXT operation using the prepared session to retrieve the next variable after the chosen item.

**Parameters** **oids** – you may pass in a list of OIDs or single item; each item may be a string representing the entire OID (e.g. 'sysDescr.0') or may be a tuple containing the name as its first item and index as its second (e.g. ('sysDescr', 0))

**Returns** an SNMPVariable object containing the value that was retrieved or a list of objects when you send in a list of OIDs

**set** (*oid, value, snmp\_type=None*)

Perform an SNMP SET operation using the prepared session.

**Parameters**

- **oid** – the OID that you wish to set which may be a string representing the entire OID (e.g. 'sysDescr.0') or may be a tuple containing the name as its first item and index as its second (e.g. ('sysDescr', 0))
- **value** – the value to set the OID to
- **snmp\_type** – if a numeric OID is used and the object is not in the parsed MIB, a type must be explicitly supplied

**Returns** a boolean indicating the success of the operation

**set\_multiple** (*oid\_values*)

Perform an SNMP SET operation on multiple OIDs with multiple values using the prepared session.

**Parameters** **oid\_values** – a list of tuples whereby each tuple contains a (oid, value) or an (oid, value, snmp\_type)

**Returns** a list of SNMPVariable objects containing the values that were retrieved via SNMP

**walk** (*oids=u'.1.3.6.1.2.1'*)

Uses SNMP GETNEXT operation using the prepared session to automatically retrieve multiple pieces of information in an OID.

**Parameters** **oids** – you may pass in a single item (multiple values currently experimental) which may be a string representing the entire OID (e.g. 'sysDescr.0') or may be a tuple containing the name as its first item and index as its second (e.g. ('sysDescr', 0))

**Returns** a list of SNMPVariable objects containing the values that were retrieved via SNMP

## 5.2 Easy API

`easysnmp.snmp_get` (*oids, \*\*session\_kargs*)

Perform an SNMP GET operation to retrieve a particular piece of information.

### Parameters

- **oids** – you may pass in a list of OIDs or single item; each item may be a string representing the entire OID (e.g. 'sysDescr.0') or may be a tuple containing the name as its first item and index as its second (e.g. ('sysDescr', 0))
- **session\_kargs** – keyword arguments which will be sent used when constructing the session for this operation; all parameters in the Session class are supported

`easysnmp.snmp_set` (*oid, value, type=None, \*\*session\_kargs*)

Perform an SNMP SET operation to update a particular piece of information.

### Parameters

- **oid** – the OID that you wish to set which may be a string representing the entire OID (e.g. 'sysDescr.0') or may be a tuple containing the name as its first item and index as its second (e.g. ('sysDescr', 0))
- **value** – the value to set the OID to
- **snmp\_type** – if a numeric OID is used and the object is not in the parsed MIB, a type must be explicitly supplied
- **session\_kargs** – keyword arguments which will be sent used when constructing the session for this operation; all parameters in the Session class are supported

`easysnmp.snmp_set_multiple` (*oid\_values, \*\*session\_kargs*)

Perform multiple SNMP SET operations to update various pieces of information at the same time.

### Parameters

- **oid\_values** – a list of tuples whereby each tuple contains a (oid, value) or an (oid, value, snmp\_type)
- **session\_kargs** – keyword arguments which will be sent used when constructing the session for this operation; all parameters in the Session class are supported

`easysnmp.snmp_get_next` (*oids, \*\*session\_kargs*)

Uses an SNMP GETNEXT operation to retrieve the next variable after the chosen item.

### Parameters

- **oids** – you may pass in a list of OIDs or single item; each item may be a string representing the entire OID (e.g. 'sysDescr.0') or may be a tuple containing the name as its first item and index as its second (e.g. ('sysDescr', 0))

- **session\_kargs** – keyword arguments which will be sent used when constructing the session for this operation; all parameters in the Session class are supported

`easysnmp.snmp_get_bulk (oids, non_repeaters=0, max_repetitions=10, **session_kargs)`

Performs a bulk SNMP GET operation to retrieve multiple pieces of information in a single packet.

#### Parameters

- **oids** – you may pass in a list of OIDs or single item; each item may be a string representing the entire OID (e.g. 'sysDescr.0') or may be a tuple containing the name as its first item and index as its second (e.g. ('sysDescr', 0))
- **non\_repeaters** – the number of objects that are only expected to return a single GET-NEXT instance, not multiple instances
- **max\_repetitions** – the number of objects that should be returned for all the repeating OIDs
- **session\_kargs** – keyword arguments which will be sent used when constructing the session for this operation; all parameters in the Session class are supported

`easysnmp.snmp_walk (oids=u'.1.3.6.1.2.1', **session_kargs)`

Uses SNMP GETNEXT operation to automatically retrieve multiple pieces of information in an OID for you.

#### Parameters

- **oids** – you may pass in a single item (multiple values currently experimental) which may be a string representing the entire OID (e.g. 'sysDescr.0') or may be a tuple containing the name as its first item and index as its second (e.g. ('sysDescr', 0))
- **session\_kargs** – keyword arguments which will be sent used when constructing the session for this operation; all parameters in the Session class are supported

`easysnmp.snmp_bulkwalk (oids=u'.1.3.6.1.2.1', non_repeaters=0, max_repetitions=10, **session_kargs)`

Uses SNMP GETBULK operation using the prepared session to automatically retrieve multiple pieces of information in an OID

#### Parameters

- **oids** – you may pass in a single item \* string representing the entire OID (e.g. 'sysDescr.0') \* tuple (name, index) (e.g. ('sysDescr', 0)) \* list of OIDs
- **non\_repeaters** – the number of objects that are only expected to return a single GET-NEXT instance, not multiple instances
- **max\_repetitions** – the number of objects that should be returned for all the repeating OIDs

**Returns** a list of SNMPVariable objects containing the values that were retrieved via SNMP

## 5.3 Exceptions

**class** `easysnmp.EasySNMPError`

The base Easy SNMP exception which covers all exceptions raised.

**class** `easysnmp.EasySNMPConnectionError`

Indicates a problem connecting to the remote host.

**class** `easysnmp.EasySNMPTimeoutError`

Raised when an SNMP request times out.

**class** `easysnmp.EasySNMPUnknownObjectIDError`

Raised when a nonexistent OID is requested.

**class** `easysnmp.EasySNMPNoSuchObjectError`

Raised when an OID is requested which may have some form of existence but is an invalid object name.

**class** `easysnmp.EasySNMPNoSuchInstanceError`

Raised when a particular OID index requested from Net-SNMP doesn't exist.

**class** `easysnmp.EasySNMPUndeterminedTypeError`

Raised when the type cannot be determined when setting the value of an OID.

**e**

easysnmp, [11](#)



## B

bulkwalk() (easysnmp.Session method), 12

## E

easysnmp (module), 11, 14, 15

EasySNMPConnectionError (class in easysnmp), 15

EasySNMPError (class in easysnmp), 15

EasySNMPNoSuchInstanceError (class in easysnmp), 16

EasySNMPNoSuchObjectError (class in easysnmp), 16

EasySNMPTimeoutError (class in easysnmp), 15

EasySNMPUndeterminedTypeError (class in easysnmp),  
16

EasySNMPUnknownObjectIDError (class in easysnmp),  
15

## G

get() (easysnmp.Session method), 13

get\_bulk() (easysnmp.Session method), 13

get\_next() (easysnmp.Session method), 13

## S

Session (class in easysnmp), 11

set() (easysnmp.Session method), 13

set\_multiple() (easysnmp.Session method), 13

snmp\_bulkwalk() (in module easysnmp), 15

snmp\_get() (in module easysnmp), 14

snmp\_get\_bulk() (in module easysnmp), 15

snmp\_get\_next() (in module easysnmp), 14

snmp\_set() (in module easysnmp), 14

snmp\_set\_multiple() (in module easysnmp), 14

snmp\_walk() (in module easysnmp), 15

## W

walk() (easysnmp.Session method), 13