

---

**k2hdkc**

***Release 1.0.0***

**Hiroataka Wakabayashi, Takeshi Nakatani**

**Mar 04, 2022**



**CONTENTS:**

<b>1</b>	<b>k2hdkc_python</b>	<b>1</b>
1.1	Overview . . . . .	1
1.2	Install . . . . .	2
1.3	Usage . . . . .	2
1.4	Development . . . . .	2
1.5	Documents . . . . .	2
1.6	Packages . . . . .	3
1.7	License . . . . .	3
1.8	AntPickax . . . . .	3
<b>2</b>	<b>k2hdkc</b>	<b>5</b>
2.1	k2hdkc package . . . . .	5
<b>3</b>	<b>Credits</b>	<b>9</b>
3.1	Development Lead . . . . .	9
3.2	Contributors . . . . .	9
<b>4</b>	<b>History</b>	<b>11</b>
4.1	1.0.0 (2022-02-07) . . . . .	11
<b>5</b>	<b>Indices and tables</b>	<b>13</b>
	<b>Python Module Index</b>	<b>15</b>
	<b>Index</b>	<b>17</b>



## K2HDKC\_PYTHON

### 1.1 Overview

k2hdkc\_python is an official python driver for [k2hdkc](#).



- 🗄️ **Multiplex**  
Auto **Scaling**  
Auto **Merging**  
Replication
- ⚙️ **SubKey**  
Encrypt / Expiration  
Queue / CAS
- 💾 **Memory** or **File**

## 1.2 Install

Let's install k2hdkc using pip:

```
pip install k2hdkc
```

## 1.3 Usage

Firstly you must install the k2hdkc shared library:

```
$ curl -o- https://raw.githubusercontent.com/yahoojapan/k2hdkc_python/master/cluster/start_server.  
↪ sh | bash
```

Then, Let's try to set a key and get it:

```
import k2hdkc  
  
k = k2hdkc.K2hdkc('slave.yaml')  
k.set('hello', 'world')  
v = k.get('hello')  
print(v)      // world
```

## 1.4 Development

Clone this repository and go into the directory, then run the following command:

```
$ python3 -m pip install --upgrade build  
$ python3 -m build
```

## 1.5 Documents

Here are documents including other components.

[Document top page](#)

[About K2HDKC](#)

[About AntPickax](#)

## 1.6 Packages

Here are packages including other components.

k2hdkc(python packages)

## 1.7 License

MIT License. See the LICENSE file.

## 1.8 AntPickax

**k2hdkc\_python** is a project by [AntPickax](#), which is an open source team in [Yahoo Japan Corporation](#).





## 2.1 k2hdkc package

### 2.1.1 Submodules

### 2.1.2 k2hdkc.k2hdkc module

K2hdkc Python Driver

```
class k2hdkc.k2hdkc.K2hdkc(conf_file, port=8031, cuk=None, rejoin=True, rejoin_forever=True,  
                             clear_backup=True)
```

Bases: object

K2hdkc class provides methods to handle key/value pairs in k2hdkc hash database.

**K2H\_INVALID\_HANDLE = 0**

```
add_subkey(key, subkey, subval, check_attr=True, password=None, expire_duration=None,  
            time_unit=TimeUnit.SECONDS)
```

Adds a new subkey to a current subkey.

```
cas_decrement(key, password=None, expire_duration=None)
```

Decrements a variable in a cluster by using a CAS operation.

```
cas_get(key, data_type, password=None, expire_duration=None)
```

Gets a variable from a cluster using a CAS operation.

```
cas_increment(key, password=None, expire_duration=None)
```

Increments a variable in a cluster by using a CAS operation.

```
cas_init(key, val=None, password=None, expire_duration=None)
```

Initializes a variable in a cluster by using a CAS operation.

```
cas_set(key, old_val, new_val, password=None, expire_duration=None)
```

Sets a value in a cluster by using a CAS operation.

```
clear_subkeys(key)
```

Clears subkeys of a key. Another subkeys that a subkey has will be removed recursively.

```
close()
```

Closes the handle

```
get(key, password=None)
```

Gets the value

```
get_attributes(key, use_str=True)
```

Retrieves attributes of a key.

**get\_subkeys**(*key, use\_str=True*)

Retrieves subkeys of a key.

**keyqueue\_get**(*prefix, is\_fifo=True, password=None, expire\_duration=None*)

Gets a new key/value element from queue.

**keyqueue\_put**(*prefix, key, val, is\_fifo=True, is\_check\_attr=True, password=None, expire\_duration=None*)

Adds a new key/value pair element to a queue.

**property libc**

returns libc handle

**property libk2hdkc**

returns libk2hkc handle

**queue\_get**(*prefix, is\_fifo=True, password=None, expire\_duration=None*)

Gets a new element to a queue.

**queue\_put**(*prefix, val, is\_fifo=True, is\_check\_attr=True, password=None, expire\_duration=None*)

Adds a new element to a queue.

**remove**(*key*)

Removes a key from a cluster.

**remove\_subkeys**(*key, subkeys, nested=False*)

Removes a subkey from the current subkeys.

**rename**(*key, newkey, parent\_key=None, is\_check\_attr=True, password=None, expire\_duration=None*)

Renames a key in a cluster.

**set**(*key, val, clear\_subkeys=False, subkeys=None, password=None, expire\_duration=None, time\_unit=TimeUnit.SECONDS*)

Sets a key/value pair

**set\_subkeys**(*key, subkeys*)

Replaces current subkeys with new one.

## 2.1.3 Module contents

k2hdkc package

**class** k2hdkc.**K2hdkc**(*conf\_file, port=8031, cuk=None, rejoin=True, rejoin\_forever=True, clear\_backup=True*)

Bases: object

K2hdkc class provides methods to handle key/value pairs in k2hdkc hash database.

**K2H\_INVALID\_HANDLE = 0**

**add\_subkey**(*key, subkey, subval, check\_attr=True, password=None, expire\_duration=None, time\_unit=TimeUnit.SECONDS*)

Adds a new subkey to a current subkey.

**cas\_decrement**(*key, password=None, expire\_duration=None*)

Decrements a variable in a cluster by using a CAS operation.

**cas\_get**(*key, data\_type, password=None, expire\_duration=None*)

Gets a variable from a cluster using a CAS operation.

**cas\_increment**(*key, password=None, expire\_duration=None*)

Increments a variable in a cluster by using a CAS operation.

**cas\_init**(*key, val=None, password=None, expire\_duration=None*)

Initializes a variable in a cluster by using a CAS operation.

**cas\_set**(*key, old\_val, new\_val, password=None, expire\_duration=None*)  
Sets a value in a cluster by using a CAS operation.

**clear\_subkeys**(*key*)  
Clears subkeys of a key. Another subkeys that a subkey has will be removed recursively.

**close**()  
Closes the handle

**get**(*key, password=None*)  
Gets the value

**get\_attributes**(*key, use\_str=True*)  
Retrieves attributes of a key.

**get\_subkeys**(*key, use\_str=True*)  
Retrieves subkeys of a key.

**keyqueue\_get**(*prefix, is\_fifo=True, password=None, expire\_duration=None*)  
Gets a new key/value element from queue.

**keyqueue\_put**(*prefix, key, val, is\_fifo=True, is\_check\_attr=True, password=None, expire\_duration=None*)  
Adds a new key/value pair element to a queue.

**property libc**  
returns libc handle

**property libk2hdkc**  
returns libk2hkc handle

**queue\_get**(*prefix, is\_fifo=True, password=None, expire\_duration=None*)  
Gets a new element to a queue.

**queue\_put**(*prefix, val, is\_fifo=True, is\_check\_attr=True, password=None, expire\_duration=None*)  
Adds a new element to a queue.

**remove**(*key*)  
Removes a key from a cluster.

**remove\_subkeys**(*key, subkeys, nested=False*)  
Removes a subkey from the current subkeys.

**rename**(*key, newkey, parent\_key=None, is\_check\_attr=True, password=None, expire\_duration=None*)  
Renames a key in a cluster.

**set**(*key, val, clear\_subkeys=False, subkeys=None, password=None, expire\_duration=None, time\_unit=TimeUnit.SECONDS*)  
Sets a key/value pair

**set\_subkeys**(*key, subkeys*)  
Replaces current subkeys with new one.



## CREDITS

### 3.1 Development Lead

- Hirotaka Wakabayashi <[hiwakaba@yahoo-corp.jp](mailto:hiwakaba@yahoo-corp.jp)>

### 3.2 Contributors

- Takeshi Nakatani <[ggtakec@gmail.com](mailto:ggtakec@gmail.com)>



## HISTORY

### 4.1 1.0.0 (2022-02-07)

- First release on PyPI.





## INDICES AND TABLES

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



## PYTHON MODULE INDEX

### k

`k2hdkc`, 6

`k2hdkc.k2hdkc`, 5



## A

`add_subkey()` (*k2hdkc.K2hdkc method*), 6  
`add_subkey()` (*k2hdkc.k2hdkc.K2hdkc method*), 5

## C

`cas_decrement()` (*k2hdkc.K2hdkc method*), 6  
`cas_decrement()` (*k2hdkc.k2hdkc.K2hdkc method*), 5  
`cas_get()` (*k2hdkc.K2hdkc method*), 6  
`cas_get()` (*k2hdkc.k2hdkc.K2hdkc method*), 5  
`cas_increment()` (*k2hdkc.K2hdkc method*), 6  
`cas_increment()` (*k2hdkc.k2hdkc.K2hdkc method*), 5  
`cas_init()` (*k2hdkc.K2hdkc method*), 6  
`cas_init()` (*k2hdkc.k2hdkc.K2hdkc method*), 5  
`cas_set()` (*k2hdkc.K2hdkc method*), 7  
`cas_set()` (*k2hdkc.k2hdkc.K2hdkc method*), 5  
`clear_subkeys()` (*k2hdkc.K2hdkc method*), 7  
`clear_subkeys()` (*k2hdkc.k2hdkc.K2hdkc method*), 5  
`close()` (*k2hdkc.K2hdkc method*), 7  
`close()` (*k2hdkc.k2hdkc.K2hdkc method*), 5

## G

`get()` (*k2hdkc.K2hdkc method*), 7  
`get()` (*k2hdkc.k2hdkc.K2hdkc method*), 5  
`get_attributes()` (*k2hdkc.K2hdkc method*), 7  
`get_attributes()` (*k2hdkc.k2hdkc.K2hdkc method*), 5  
`get_subkeys()` (*k2hdkc.K2hdkc method*), 7  
`get_subkeys()` (*k2hdkc.k2hdkc.K2hdkc method*), 5

## K

`K2H_INVALID_HANDLE` (*k2hdkc.K2hdkc attribute*), 6  
`K2H_INVALID_HANDLE` (*k2hdkc.k2hdkc.K2hdkc attribute*), 5  
`k2hdkc`  
    *module*, 6  
`K2hdkc` (*class in k2hdkc*), 6  
`K2hdkc` (*class in k2hdkc.k2hdkc*), 5  
`k2hdkc.k2hdkc`  
    *module*, 5  
`keyqueue_get()` (*k2hdkc.K2hdkc method*), 7  
`keyqueue_get()` (*k2hdkc.k2hdkc.K2hdkc method*), 6  
`keyqueue_put()` (*k2hdkc.K2hdkc method*), 7  
`keyqueue_put()` (*k2hdkc.k2hdkc.K2hdkc method*), 6

## L

`libc` (*k2hdkc.K2hdkc property*), 7  
`libc` (*k2hdkc.k2hdkc.K2hdkc property*), 6  
`libk2hdkc` (*k2hdkc.K2hdkc property*), 7  
`libk2hdkc` (*k2hdkc.k2hdkc.K2hdkc property*), 6

## M

*module*  
    *k2hdkc*, 6  
    *k2hdkc.k2hdkc*, 5

## Q

`queue_get()` (*k2hdkc.K2hdkc method*), 7  
`queue_get()` (*k2hdkc.k2hdkc.K2hdkc method*), 6  
`queue_put()` (*k2hdkc.K2hdkc method*), 7  
`queue_put()` (*k2hdkc.k2hdkc.K2hdkc method*), 6

## R

`remove()` (*k2hdkc.K2hdkc method*), 7  
`remove()` (*k2hdkc.k2hdkc.K2hdkc method*), 6  
`remove_subkeys()` (*k2hdkc.K2hdkc method*), 7  
`remove_subkeys()` (*k2hdkc.k2hdkc.K2hdkc method*), 6  
`rename()` (*k2hdkc.K2hdkc method*), 7  
`rename()` (*k2hdkc.k2hdkc.K2hdkc method*), 6

## S

`set()` (*k2hdkc.K2hdkc method*), 7  
`set()` (*k2hdkc.k2hdkc.K2hdkc method*), 6  
`set_subkeys()` (*k2hdkc.K2hdkc method*), 7  
`set_subkeys()` (*k2hdkc.k2hdkc.K2hdkc method*), 6