

---

# **hiera-py Documentation**

***Release 0.0.1***

**Thomas Van Doren**

April 03, 2013



# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Installation . . . . .	3
1.2	Getting Started . . . . .	3
<b>2</b>	<b>Reference</b>	<b>5</b>
2.1	Exceptions . . . . .	6
<b>3</b>	<b>Development</b>	<b>7</b>
3.1	Testing . . . . .	7
<b>4</b>	<b>TODO</b>	<b>9</b>
<b>5</b>	<b>License</b>	<b>11</b>
<b>6</b>	<b>Authors</b>	<b>13</b>
	<b>Python Module Index</b>	<b>15</b>



Python interface for the hiera hierarchical database.

See the [documentation on puppetlabs.com](#) for more detail.

Contents:



# INTRODUCTION

## 1.1 Installation

```
pip install hiera-py
```

```
# Or, if you prefer easy_install:  
easy_install hiera-py
```

```
# Or, if you prefer to install from source:  
python setup.py install
```

## 1.2 Getting Started

```
>>> import hiera  
>>> hiera_client = hiera.HieraClient('/etc/hiera.yml', environment='dev')  
>>> hiera_client.get('my_key')  
'my_value'  
>>> hiera_client.get('nonexistent_key')  
Traceback (most recent call last):  
...  
hiera.exc.HieraError: Failed to retrieve key nonexistent_key. ...
```





# REFERENCE

Python client for Hierarchical database.

**class** `hier.client.HieraClient` (*config\_filename*, *hier\_binary=u'hiera'*, *\*\*kwargs*)  
 Python client for Hierarchical database.

**\_\_init\_\_** (*config\_filename*, *hier\_binary=u'hiera'*, *\*\*kwargs*)  
 Create a new instance with the given settings.

Key value params passed into this will be added to the environment when running the hiera client. For example, (environment='developer', osfamily='Debian') as keyword args to **\_\_init\_\_** would result in hiera calls like this:

```
hiera --config <config_filename> <key> environment=developer osfamily=Debian
```

## Parameters

- **config\_filename** – Path to the hiera configuration file.
- **hier\_binary** – Path to the hiera binary. Defaults to 'hiera'.

**get** (*key\_name*)

Request the given key from hiera.

Returns the string version of the key when successful.

Raises `hier.exc.HieraError` if the key does not exist or there was an error invoking hiera. Raises `hier.exc.HieraNotFoundError` if the hiera CLI binary could not be found.

**Parameters** **key\_name** – string key

**Return type** str value for key or None

**class** `hier.HieraClient` (*config\_filename*, *hier\_binary=u'hiera'*, *\*\*kwargs*)  
 Python client for Hierarchical database.

**\_\_init\_\_** (*config\_filename*, *hier\_binary=u'hiera'*, *\*\*kwargs*)  
 Create a new instance with the given settings.

Key value params passed into this will be added to the environment when running the hiera client. For example, (environment='developer', osfamily='Debian') as keyword args to **\_\_init\_\_** would result in hiera calls like this:

```
hiera --config <config_filename> <key> environment=developer osfamily=Debian
```

## Parameters

- **config\_filename** – Path to the hiera configuration file.

- **hiera\_binary** – Path to the hiera binary. Defaults to 'hiera'.

**get** (*key\_name*)

Request the given key from hiera.

Returns the string version of the key when successful.

Raises `hiera.exc.HieraError` if the key does not exist or there was an error invoking hiera. Raises `hiera.exc.HieraNotFoundError` if the hiera CLI binary could not be found.

**Parameters** **key\_name** – string key

**Return type** str value for key or None

## 2.1 Exceptions

Exceptions for specific hiera issues.

**class** `hiera.exc.HieraError` (*message, returncode=None, output=None*)

Generic Hiera error.

**\_\_init\_\_** (*message, returncode=None, output=None*)

Override instance init so that return code and console output can be added to error.

**class** `hiera.exc.HieraNotFoundError`

Hiera error indicating the hiera CLI could not be found.

# DEVELOPMENT

## 3.1 Testing

Tox is used to run the tests. Travis automatically verifies commits. It uses python 2.7 and pep8.

```
# Run the tests against python 2.7.  
tox
```

```
# Run the tests against python 2.7 with code coverage.  
tox -e cover
```

```
# Run the tests against a bunch of python versions.  
tox -e py25,py26,py27,py31,py32,py33,py34
```



# TODO

Things I would like to do, eventually...

- Add additional accessors for array and hash types.
- Add support for scope files (either yaml or json format).
- Support more python versions



# LICENSE

BSD





# AUTHORS

Thomas Van Doren



# PYTHON MODULE INDEX

## h

`hiera.client`, 5  
`hiera.exc`, 6