

# Unittest for caching

May 8, 2022

## Contents

<b>1</b>	<b>Test Information</b>	<b>3</b>
1.1	Test Candidate Information . . . . .	3
1.2	Unittest Information . . . . .	3
1.3	Test System Information . . . . .	3
<b>2</b>	<b>Statistic</b>	<b>3</b>
2.1	Test-Statistic for testrun with python 3.10.4 (final) . . . . .	3
2.2	Coverage Statistic . . . . .	4
<b>3</b>	<b>Testcases with no corresponding Requirement</b>	<b>5</b>
3.1	Summary for testrun with python 3.10.4 (final) . . . . .	5
3.1.1	caching.property_cache.json: Test cached data (full init) . . . . .	5
3.1.2	caching.property_cache.json: Test cached data (partially init) . . . . .	5
3.1.3	caching.property_cache.json: Test execution of save callback (full init) . . . . .	5
3.1.4	caching.property_cache.json: Test full initialised JSON-Cache-Object . . . . .	6
3.1.5	caching.property_cache.json: Test get from source caused by changed uid (full init) . . . . .	6
3.1.6	caching.property_cache.json: Test get from source caused by changed uid (partially init) . . . . .	6
3.1.7	caching.property_cache.json: Test get from source caused by increased data version (full init) . . . . .	7
3.1.8	caching.property_cache.json: Test get from source caused by increased data version (partially init) . . . . .	7
3.1.9	caching.property_cache.json: Test internal key usage . . . . .	8
3.1.10	caching.property_cache.json: Test partially initialisation of JSON-Cache-Object . . . . .	8
3.1.11	caching.property_cache.pickle: Test cached data (full init) . . . . .	8
3.1.12	caching.property_cache.pickle: Test cached data (partially init) . . . . .	9
3.1.13	caching.property_cache.pickle: Test execution of save callback (full init) . . . . .	9
3.1.14	caching.property_cache.pickle: Test full initialised PICKLE-Cache-Object . . . . .	9
3.1.15	caching.property_cache.pickle: Test get from source caused by changed uid (full init) . . . . .	10
3.1.16	caching.property_cache.pickle: Test get from source caused by changed uid (partially init) . . . . .	10
3.1.17	caching.property_cache.pickle: Test get from source caused by increased data version (full init) . . . . .	11
3.1.18	caching.property_cache.pickle: Test get from source caused by increased data version (partially init) . . . . .	11
3.1.19	caching.property_cache.pickle: Test internal key usage . . . . .	11
3.1.20	caching.property_cache.pickle: Test partially initialised PICKLE-Cache-Object . . . . .	12

<b>A</b>	<b>Trace for testrun with python 3.10.4 (final)</b>	<b>13</b>
A.1	Tests with status Info (20)	13
A.1.1	caching.property_cache_json: Test full initialised JSON-Cache-Object	13
A.1.2	caching.property_cache_json: Test partially initialisation of JSON-Cache-Object	13
A.1.3	caching.property_cache_json: Test cached data (full init)	14
A.1.4	caching.property_cache_json: Test cached data (partially init)	15
A.1.5	caching.property_cache_json: Test get from source caused by increased data version (full init)	15
A.1.6	caching.property_cache_json: Test get from source caused by increased data version (partially init)	16
A.1.7	caching.property_cache_json: Test get from source caused by changed uid (full init)	16
A.1.8	caching.property_cache_json: Test get from source caused by changed uid (partially init)	17
A.1.9	caching.property_cache_json: Test execution of save callback (full init)	17
A.1.10	caching.property_cache_json: Test internal key usage	17
A.1.11	caching.property_cache_pickle: Test full initialised PICKLE-Cache-Object	18
A.1.12	caching.property_cache_pickle: Test partially initialised PICKLE-Cache-Object	19
A.1.13	caching.property_cache_pickle: Test cached data (full init)	19
A.1.14	caching.property_cache_pickle: Test cached data (partially init)	20
A.1.15	caching.property_cache_pickle: Test get from source caused by increased data version (full init)	21
A.1.16	caching.property_cache_pickle: Test get from source caused by increased data version (partially init)	21
A.1.17	caching.property_cache_pickle: Test get from source caused by changed uid (full init)	22
A.1.18	caching.property_cache_pickle: Test get from source caused by changed uid (partially init)	22
A.1.19	caching.property_cache_pickle: Test execution of save callback (full init)	23
A.1.20	caching.property_cache_pickle: Test internal key usage	23
<b>B</b>	<b>Test-Coverage</b>	<b>24</b>
B.1	caching	24
B.1.1	caching.__init__.py	24

# 1 Test Information

## 1.1 Test Candidate Information

The Module `caching` is designed to store information in `json` or `pickle` files to support them much faster then generating them from the original source file. For more Information read the documentation.

---

### Library Information

Name	caching
State	Released
Supported Interpreters	python3
Version	ce99f9454c5612dca5c8ea21cc03e83e

---

### Dependencies

---

## 1.2 Unittest Information

---

### Unittest Information

Version	e113c3759194c81e7df9e29b1debe00d
Testruns with	python 3.10.4 (final)

---

## 1.3 Test System Information

---

### System Information

Architecture	64bit
Distribution	Ubuntu 22.04 Jammy Jellyfish
Hostname	ahorn
Kernel	5.15.0-27-lowlatency (#28-Ubuntu SMP PREEMPT Tue Apr 19 15:27:08 UTC 2022)
Machine	x86_64
Path	/home/dirk/my_repositories/unittest/caching
System	Linux
Username	dirk

---

# 2 Statistic

## 2.1 Test-Statistic for testrun with python 3.10.4 (final)

---

Number of tests	<b>20</b>
Number of successfull tests	<b>20</b>
Number of possibly failed tests	<b>0</b>
Number of failed tests	<b>0</b>

---

Executionlevel	Full Test (all defined tests)
Time consumption	0.022s

---

## 2.2 Coverage Statistic

Module- or Filename	Line-Coverage	Branch-Coverage
caching	99.1%	97.8%
caching.__init__.py	99.1%	

### 3 Testcases with no corresponding Requirement

#### 3.1 Summary for testrun with python 3.10.4 (final)

##### 3.1.1 caching.property\_cache\_json: Test cached data (full init)

###### Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.3!

---

Testrun:	python 3.10.4 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/tests/_init_.py (31)
Start-Time:	2022-05-08 20:50:13,401
Finished-Time:	2022-05-08 20:50:13,402
Time-Consumption	0.001s

---

**Testsummary:**

---

<b>Info</b>	Initialising property_cache_json (load_all_on_init=True).
<b>Info</b>	Collecting data from cache instance.
<b>Success</b>	Cached data is correct (Content {'str': 'string', 'unicode': 'unicode', 'integer': 17, 'float': 3.14159, 'list': [1, 'two', '3', 4], 'dict': {'1': 1, '2': 'two', '3': '3', '4': 4}, 'uncached': 'uncached_data_of_class'} and Type is <class 'dict'>).

---

##### 3.1.2 caching.property\_cache\_json: Test cached data (partially init)

###### Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.4!

---

Testrun:	python 3.10.4 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/tests/_init_.py (32)
Start-Time:	2022-05-08 20:50:13,402
Finished-Time:	2022-05-08 20:50:13,403
Time-Consumption	0.001s

---

**Testsummary:**

---

<b>Info</b>	Initialising property_cache_json (load_all_on_init=True).
<b>Info</b>	Collecting data from cache instance.
<b>Success</b>	Cached data is correct (Content {'str': 'string', 'unicode': 'unicode', 'integer': 17, 'float': 3.14159, 'list': [1, 'two', '3', 4], 'dict': {'1': 1, '2': 'two', '3': '3', '4': 4}, 'uncached': 'uncached_data_of_class'} and Type is <class 'dict'>).

---

##### 3.1.3 caching.property\_cache\_json: Test execution of save callback (full init)

###### Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.9!

---

Testrun:	python 3.10.4 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/tests/_init_.py (37)

Start-Time: 2022-05-08 20:50:13,410  
 Finished-Time: 2022-05-08 20:50:13,411  
 Time-Consumption 0.001s

**Testsummary:**

**Info** Installing save\_callback, which sets a variable to True on execution.  
**Success** Save callback execution variable is correct (Content True and Type is <class 'bool'>).

**3.1.4 caching.property\_cache\_json: Test full initialised JSON-Cache-Object****Testresult**

This test was passed with the state: **Success**. See also full trace in section A.1.1!

Testrun: python 3.10.4 (final)  
 Caller: /home/dirk/my\_repositories/unittest/caching/unittest/src/tests/\_init\_.py (29)  
 Start-Time: 2022-05-08 20:50:13,397  
 Finished-Time: 2022-05-08 20:50:13,398  
 Time-Consumption 0.001s

**Testsummary:**

**Info** Initialising property\_cache\_json (load\_all\_on\_init=True).  
**Info** Extracting storage object from property\_cache\_json for comparison.  
**Success** Cache object is correct (Content {'dict': {'1': 1, '2': 'two', '3': '3', '4': 4}, 'float': 3.14159, 'integer': 17, 'list': [1, 'two', '3', 4], 'str': 'string', 'unicode': 'unicode'} and Type is <class 'dict'>).

**3.1.5 caching.property\_cache\_json: Test get from source caused by changed uid (full init)****Testresult**

This test was passed with the state: **Success**. See also full trace in section A.1.7!

Testrun: python 3.10.4 (final)  
 Caller: /home/dirk/my\_repositories/unittest/caching/unittest/src/tests/\_init\_.py (35)  
 Start-Time: 2022-05-08 20:50:13,407  
 Finished-Time: 2022-05-08 20:50:13,408  
 Time-Consumption 0.001s

**Testsummary:**

**Info** Initialising property\_cache\_json (load\_all\_on\_init=True).  
**Success** Instance data after changing uid is correct (Content {'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '2': 2, '3': 'three', '4': '4'}} and Type is <class 'dict'>).

**3.1.6 caching.property\_cache\_json: Test get from source caused by changed uid (partially init)****Testresult**

This test was passed with the state: **Success**. See also full trace in section A.1.8!

---

Testrun:	python 3.10.4 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/tests/_init_.py (36)
Start-Time:	2022-05-08 20:50:13,408
Finished-Time:	2022-05-08 20:50:13,410
Time-Consumption	0.002s

---

**Testsummary:**

---

<b>Info</b>	Initialising property_cache.json (load_all_on_init=True).
<b>Success</b>	Instance data after changing uid is correct (Content {'str': '__string__', 'unicode': '__unicode__', 'integer': 34, 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '2': 2, '3': 'three', '4': '4'}} and Type is <class 'dict'>).

---

### 3.1.7 caching.property\_cache.json: Test get from source caused by increased data version (full init)

#### Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.5!

---

Testrun:	python 3.10.4 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/tests/_init_.py (33)
Start-Time:	2022-05-08 20:50:13,403
Finished-Time:	2022-05-08 20:50:13,404
Time-Consumption	0.001s

---

**Testsummary:**

---

<b>Info</b>	Initialising property_cache.json (load_all_on_init=True).
<b>Success</b>	Instance data after increasing data_version is correct (Content {'str': '__string__', 'unicode': '__unicode__', 'integer': 34, 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '2': 2, '3': 'three', '4': '4'}} and Type is <class 'dict'>).

---

### 3.1.8 caching.property\_cache.json: Test get from source caused by increased data version (partially init)

#### Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.6!

---

Testrun:	python 3.10.4 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/tests/_init_.py (34)
Start-Time:	2022-05-08 20:50:13,404
Finished-Time:	2022-05-08 20:50:13,406
Time-Consumption	0.002s

---

**Testsummary:**

---

<b>Info</b>	Initialising property_cache.json (load_all_on_init=True).
<b>Success</b>	Instance data after increasing data_version is correct (Content {'str': '__string__', 'unicode': '__unicode__', 'integer': 34, 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '2': 2, '3': 'three', '4': '4'}} and Type is <class 'dict'>).

---



### 3.1.9 caching.property\_cache\_json: Test internal key usage

#### Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.10!

---

Testrun:	python 3.10.4 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/tests/_init_.py (38)
Start-Time:	2022-05-08 20:50:13,411
Finished-Time:	2022-05-08 20:50:13,412
Time-Consumption	0.001s

---

**Testsummary:**

---

<b>Info</b>	Initialising property_cache_json (load_all_on_init=True).
<b>Info</b>	Extracting storage object from property_cache_json for comparison.
<b>Success</b>	Cache is correct (Content {'__property_cache_data_version_': 'no second data version', '__property_cache_uid_': 'no second uid', '_property_cache_data_version_': 'no data version', '_property_cache_uid_': 'no uid'} and Type is <class 'dict'>).
<b>Success</b>	Keyfilter returnvalue for 5 (<class 'int'>) is correct (Content 5 and Type is <class 'int'>).

---

### 3.1.10 caching.property\_cache\_json: Test partially initialisation of JSON-Cache-Object

#### Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.2!

---

Testrun:	python 3.10.4 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/tests/_init_.py (30)
Start-Time:	2022-05-08 20:50:13,398
Finished-Time:	2022-05-08 20:50:13,401
Time-Consumption	0.002s

---

**Testsummary:**

---

<b>Info</b>	Initialising property_cache_json (load_all_on_init=False).
<b>Info</b>	Partially initialising cache object by requesting some information.
<b>Info</b>	Extracting storage object from property_cache_json for comparison.
<b>Success</b>	Cache object is correct (Content {'integer': 17, 'str': 'string', 'unicode': 'unicode'} and Type is <class 'dict'>).

---

### 3.1.11 caching.property\_cache\_pickle: Test cached data (full init)

#### Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.13!

---

Testrun:	python 3.10.4 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/tests/_init_.py (44)
Start-Time:	2022-05-08 20:50:13,414
Finished-Time:	2022-05-08 20:50:13,415
Time-Consumption	0.001s

---

**Testsummary:**

---

<b>Info</b>	Initialising property_cache_pickle (load_all_on_init=True).
<b>Info</b>	Collecting data from cache instance.
<b>Success</b>	Cached data is correct (Content {'str': 'string', 'unicode': 'unicode', 'integer': 17, 'float': 3.14159, 'list': [1, 'two', '3', 4], 'dict': {'1': 1, '2': 'two', '3': '3', '4': 4}, 'uncached': 'uncached_data_of_class'} and Type is <class 'dict'>).

---

**3.1.12 caching.property\_cache\_pickle: Test cached data (partially init)**

**Testresult**

This test was passed with the state: **Success**. See also full trace in section A.1.14!

---

Testrun:	python 3.10.4 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/tests/_init....py (45)
Start-Time:	2022-05-08 20:50:13,415
Finished-Time:	2022-05-08 20:50:13,416
Time-Consumption	0.001s

---

**Testsummary:**

---

<b>Info</b>	Initialising property_cache_pickle (load_all_on_init=True).
<b>Info</b>	Collecting data from cache instance.
<b>Success</b>	Cached data is correct (Content {'str': 'string', 'unicode': 'unicode', 'integer': 17, 'float': 3.14159, 'list': [1, 'two', '3', 4], 'dict': {'1': 1, '2': 'two', '3': '3', '4': 4}, 'uncached': 'uncached_data_of_class'} and Type is <class 'dict'>).

---

**3.1.13 caching.property\_cache\_pickle: Test execution of save callback (full init)**

**Testresult**

This test was passed with the state: **Success**. See also full trace in section A.1.19!

---

Testrun:	python 3.10.4 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/tests/_init....py (50)
Start-Time:	2022-05-08 20:50:13,421
Finished-Time:	2022-05-08 20:50:13,421
Time-Consumption	0.000s

---

**Testsummary:**

---

<b>Info</b>	Installing save_callback, which sets a variable to True on execution.
<b>Success</b>	Save callback execution variable is correct (Content True and Type is <class 'bool'>).

---

**3.1.14 caching.property\_cache\_pickle: Test full initialised PICKLE-Cache-Object**

**Testresult**

This test was passed with the state: **Success**. See also full trace in section A.1.11!

---

Testrun:	python 3.10.4 (final)
----------	-----------------------

Caller: /home/dirk/my\_repositories/unittest/caching/unittest/src/tests/\_init\_.py (42)  
 Start-Time: 2022-05-08 20:50:13,412  
 Finished-Time: 2022-05-08 20:50:13,413  
 Time-Consumption 0.001s

**Testsummary:**

**Info** Initialising property\_cache\_pickle (load\_all\_on\_init=True).  
**Info** Extracting storage object from property\_cache\_pickle for comparison.  
**Success** Cache object is correct (Content {'str': 'string', 'unicode': 'unicode', 'integer': 17, 'float': 3.14159, 'list': [1, 'two', '3', 4], 'dict': {'1': 1, '2': 'two', '3': '3', '4': 4}} and Type is <class 'dict'>).

**3.1.15 caching.property\_cache\_pickle: Test get from source caused by changed uid (full init)**

**Testresult**

This test was passed with the state: **Success**. See also full trace in section A.1.17!

Testrun: python 3.10.4 (final)  
 Caller: /home/dirk/my\_repositories/unittest/caching/unittest/src/tests/\_init\_.py (48)  
 Start-Time: 2022-05-08 20:50:13,419  
 Finished-Time: 2022-05-08 20:50:13,419  
 Time-Consumption 0.001s

**Testsummary:**

**Info** Initialising property\_cache\_pickle (load\_all\_on\_init=True).  
**Success** Instance data after changing uid is correct (Content {'str': 'string', 'unicode': 'unicode', 'integer': 34, 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '2': 2, '3': 'three', '4': '4'}} and Type is <class 'dict'>).

**3.1.16 caching.property\_cache\_pickle: Test get from source caused by changed uid (partially init)**

**Testresult**

This test was passed with the state: **Success**. See also full trace in section A.1.18!

Testrun: python 3.10.4 (final)  
 Caller: /home/dirk/my\_repositories/unittest/caching/unittest/src/tests/\_init\_.py (49)  
 Start-Time: 2022-05-08 20:50:13,420  
 Finished-Time: 2022-05-08 20:50:13,421  
 Time-Consumption 0.001s

**Testsummary:**

**Info** Initialising property\_cache\_pickle (load\_all\_on\_init=True).  
**Success** Instance data after changing uid is correct (Content {'str': 'string', 'unicode': 'unicode', 'integer': 34, 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '2': 2, '3': 'three', '4': '4'}} and Type is <class 'dict'>).

### 3.1.17 caching.property\_cache\_pickle: Test get from source caused by increased data version (full init)

#### Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.15!

---

Testrun:	python 3.10.4 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/tests/...init...py (46)
Start-Time:	2022-05-08 20:50:13,416
Finished-Time:	2022-05-08 20:50:13,417
Time-Consumption	0.001s

---

**Testsummary:**

---

<b>Info</b>	Initialising property_cache_pickle (load_all_on_init=True).
<b>Success</b>	Instance data after increasing data_version is correct (Content {'str': '...string...', 'unicode': '...unicode...', 'integer': 34, 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '2': 2, '3': 'three', '4': '4'}} and Type is <class 'dict'>).

---

### 3.1.18 caching.property\_cache\_pickle: Test get from source caused by increased data version (partially init)

#### Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.16!

---

Testrun:	python 3.10.4 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/tests/...init...py (47)
Start-Time:	2022-05-08 20:50:13,417
Finished-Time:	2022-05-08 20:50:13,418
Time-Consumption	0.001s

---

**Testsummary:**

---

<b>Info</b>	Initialising property_cache_pickle (load_all_on_init=True).
<b>Success</b>	Instance data after increasing data_version is correct (Content {'str': '...string...', 'unicode': '...unicode...', 'integer': 34, 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '2': 2, '3': 'three', '4': '4'}} and Type is <class 'dict'>).

---

### 3.1.19 caching.property\_cache\_pickle: Test internal key usage

#### Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.20!

---

Testrun:	python 3.10.4 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/tests/...init...py (51)
Start-Time:	2022-05-08 20:50:13,422
Finished-Time:	2022-05-08 20:50:13,422
Time-Consumption	0.001s

---

**Testsummary:**

---

<b>Info</b>	Initialising property_cache_pickle (load_all_on_init=True).
<b>Info</b>	Extracting storage object from property_cache_pickle for comparison.

**Success** Cache is correct (Content {'\_\_property\_cache\_uid\_': 'no uid', '\_\_property\_cache\_uid\_': 'no second uid', '\_\_property\_cache\_data\_version\_': 'no data version', '\_\_property\_cache\_data\_version\_': 'no second data version'} and Type is <class 'dict'>).

**Success** Keyfilter returnvalue for 5 (<class 'int'>) is correct (Content 5 and Type is <class 'int'>).

---

### 3.1.20 caching.property\_cache\_pickle: Test partially initialised PICKLE-Cache-Object

#### Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.12!

---

Testrun:	python 3.10.4 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/tests/_init_.py (43)
Start-Time:	2022-05-08 20:50:13,413
Finished-Time:	2022-05-08 20:50:13,414
Time-Consumption	0.001s

---

#### Testsummary:

---

<b>Info</b>	Initialising property_cache_pickle (load_all_on_init=False).
<b>Info</b>	Partially initialising cache object by requesting some information.
<b>Info</b>	Extracting storage object from property_cache_pickle for comparison.
<b>Success</b>	Cache object is correct (Content {'str': 'string', 'integer': 17, 'unicode': 'unicode'} and Type is <class 'dict'>).

---

## A Trace for testrun with python 3.10.4 (final)

### A.1 Tests with status Info (20)

#### A.1.1 caching.property\_cache\_json: Test full initialised JSON-Cache-Object

##### Testresult

This test was passed with the state: **Success**.

---

**Info** Initialising property\_cache\_json (load\_all\_on\_init=True).

---

Deleting cache file from filesystem to ensure identical conditions for each test run.

Initialising cached class with

```
↪ /home/dirk/my_repositories/unittest/caching/unittest/output_data/load_on_init.json as
↪ cache file.
```

---

**Info** Extracting storage object from property\_cache\_json for comparison.

---

```
Using storage object of cache class for comparison: {'_property_cache_data_version_': 1,
↪ '_property_cache_uid_': 'my_unique_id', 'dict': {'1': 1, '2': 'two', '3': '3', '4': 4},
↪ 'float': 3.14159, 'integer': 17, 'list': [1, 'two', '3', 4], 'str': 'string', 'unicode':
↪ 'unicode'}
```

Deleting overhead keys: \_property\_cache\_data\_version\_, \_property\_cache\_uid\_

---

**Success** Cache object is correct (Content {'dict': {'1': 1, '2': 'two', '3': '3', '4': 4}, 'float': 3.14159, 'integer': 17, 'list': [1, 'two', '3', 4], 'str': 'string', 'unicode': 'unicode'} and Type is <class 'dict'>).

---

```
Result (Cache object): { 'dict': { '1': 1, '2': 'two', '3': '3', '4': 4 }, 'float': 3.14159,
↪ 'integer': 17, 'list': [ 1, 'two', '3', 4 ], 'str': 'string', 'unicode': 'unicode' }
↪ (<class 'dict'>)
```

```
Expectation (Cache object): result = { 'str': 'string', 'unicode': 'unicode', 'integer': 17,
↪ 'float': 3.14159, 'list': [ 1, 'two', '3', 4 ], 'dict': { '1': 1, '2': 'two', '3': '3',
↪ '4': 4 } } (<class 'dict'>)
```

#### A.1.2 caching.property\_cache\_json: Test partial initialisation of JSON-Cache-Object

##### Testresult

This test was passed with the state: **Success**.

---

**Info** Initialising property\_cache\_json (load\_all\_on\_init=False).

---

Deleting cache file from filesystem to ensure identical conditions for each test run.

Initialising cached class with

```
↪ /home/dirk/my_repositories/unittest/caching/unittest/output_data/no_load_on_init.json as
↪ cache file.
```

---

**Info** Partially initialising cache object by requesting some information.

---

**Info** Extracting storage object from `property_cache_json` for comparison.

---

```
Using storage object of cache class for comparison: {'_property_cache_data_version': 1,
↪ '_property_cache_uid': 'my_unique_id', 'integer': 17, 'str': 'string', 'unicode':
↪ 'unicode'}
```

```
Deleting overhead keys: _property_cache_data_version_, _property_cache_uid_
```

---

**Success** Cache object is correct (Content `{'integer': 17, 'str': 'string', 'unicode': 'unicode'}` and Type is `<class 'dict'>`).

---

```
Result (Cache object): { 'integer': 17, 'str': 'string', 'unicode': 'unicode' } (<class
↪ 'dict'>)
```

```
Expectation (Cache object): result = { 'str': 'string', 'unicode': 'unicode', 'integer': 17 }
↪ (<class 'dict'>)
```

### A.1.3 `caching.property_cache_json`: Test cached data (full init)

#### Testresult

This test was passed with the state: **Success**.

---

**Info** Initialising `property_cache_json` (`load_all_on_init=True`).

---

```
Deleting cache file from filesystem to ensure identical conditions for each test run.
```

```
Initialising cached class with /home/dirk/my_repositories/unittest/caching/unittest/output_da_
↪ ta/cache_data_test_load_on_init.json as cache
↪ file.
```

---

**Info** Collecting data from cache instance.

---

**Success** Cached data is correct (Content `{'str': 'string', 'unicode': 'unicode', 'integer': 17, 'float': 3.14159, 'list': [1, 'two', '3', 4], 'dict': {'1': 1, '2': 'two', '3': '3', '4': 4}, 'uncached': 'uncached_data_of_class'}` and Type is `<class 'dict'>`).

---

```
Result (Cached data): { 'str': 'string', 'unicode': 'unicode', 'integer': 17, 'float':
↪ 3.14159, 'list': [ 1, 'two', '3', 4 ], 'dict': { '1': 1, '2': 'two', '3': '3', '4': 4 },
↪ 'uncached': 'uncached_data_of_class' } (<class 'dict'>)
```

```
Expectation (Cached data): result = { 'str': 'string', 'unicode': 'unicode', 'integer': 17,
↪ 'float': 3.14159, 'list': [ 1, 'two', '3', 4 ], 'dict': { '1': 1, '2': 'two', '3': '3',
↪ '4': 4 }, 'uncached': 'uncached_data_of_class' } (<class 'dict'>)
```

#### A.1.4 caching.property\_cache.json: Test cached data (partially init)

##### Testresult

This test was passed with the state: **Success**.

---

**Info** Initialising property\_cache.json (load\_all\_on\_init=True).

---

Deleting cache file from filesystem to ensure identical conditions for each test run.

Initialising cached class with /home/dirk/my\_repositories/unittest/caching/unittest/output\_data/cache\_data\_test\_no\_load\_on\_init.json as cache  
 ↪ file.

---

**Info** Collecting data from cache instance.

---

**Success** Cached data is correct (Content {'str': 'string', 'unicode': 'unicode', 'integer': 17, 'float': 3.14159, 'list': [1, 'two', '3', 4], 'dict': {'1': 1, '2': 'two', '3': '3', '4': 4}, 'uncached': 'uncached\_data\_of\_class'} and Type is <class 'dict'>).

---

Result (Cached data): { 'str': 'string', 'unicode': 'unicode', 'integer': 17, 'float': 3.14159, 'list': [ 1, 'two', '3', 4 ], 'dict': { '1': 1, '2': 'two', '3': '3', '4': 4 }, 'uncached': 'uncached\_data\_of\_class' } (<class 'dict'>)

Expectation (Cached data): result = { 'str': 'string', 'unicode': 'unicode', 'integer': 17, 'float': 3.14159, 'list': [ 1, 'two', '3', 4 ], 'dict': { '1': 1, '2': 'two', '3': '3', '4': 4 }, 'uncached': 'uncached\_data\_of\_class' } (<class 'dict'>)

#### A.1.5 caching.property\_cache.json: Test get from source caused by increased data version (full init)

##### Testresult

This test was passed with the state: **Success**.

---

**Info** Initialising property\_cache.json (load\_all\_on\_init=True).

---

Deleting cache file from filesystem to ensure identical conditions for each test run.

Initialising cached class with /home/dirk/my\_repositories/unittest/caching/unittest/output\_data/data\_version\_test\_load\_on\_init.json as cache  
 ↪ file.

---

**Success** Instance data after increasing data\_version is correct (Content {'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '2': 2, '3': 'three', '4': '4'}} and Type is <class 'dict'>).

---

Result (Instance data after increasing data\_version): { 'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ], 'dict': { '1': '1', '2': 2, '3': 'three', '4': '4' } } (<class 'dict'>)

Expectation (Instance data after increasing data\_version): result = { 'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ], 'dict': { '1': '1', '2': 2, '3': 'three', '4': '4' } } (<class 'dict'>)



### A.1.6 caching.property\_cache.json: Test get from source caused by increased data version (partially init)

#### Testresult

This test was passed with the state: **Success**.

---

**Info** Initialising property\_cache.json (load\_all\_on\_init=True).

---

Deleting cache file from filesystem to ensure identical conditions for each test run.

Initialising cached class with /home/dirk/my\_repositories/unittest/caching/unittest/output\_data/data\_version\_test\_no\_load\_on\_init.json as cache  
 ↳ file.

---

**Success** Instance data after increasing data\_version is correct (Content {'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '2': 2, '3': 'three', '4': '4'}} and Type is <class 'dict'>).

---

Result (Instance data after increasing data\_version): { 'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ], 'dict': { '1': '1', '2': 2, '3': 'three', '4': '4' } } (<class 'dict'>)

Expectation (Instance data after increasing data\_version): result = { 'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ], 'dict': { '1': '1', '2': 2, '3': 'three', '4': '4' } } (<class 'dict'>)

### A.1.7 caching.property\_cache.json: Test get from source caused by changed uid (full init)

#### Testresult

This test was passed with the state: **Success**.

---

**Info** Initialising property\_cache.json (load\_all\_on\_init=True).

---

Deleting cache file from filesystem to ensure identical conditions for each test run.

Initialising cached class with /home/dirk/my\_repositories/unittest/caching/unittest/output\_data/uid\_test\_load\_on\_init.json as cache  
 ↳ file.

---

**Success** Instance data after changing uid is correct (Content {'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '2': 2, '3': 'three', '4': '4'}} and Type is <class 'dict'>).

---

Result (Instance data after changing uid): { 'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ], 'dict': { '1': '1', '2': 2, '3': 'three', '4': '4' } } (<class 'dict'>)

Expectation (Instance data after changing uid): result = { 'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ], 'dict': { '1': '1', '2': 2, '3': 'three', '4': '4' } } (<class 'dict'>)

### A.1.8 caching.property\_cache.json: Test get from source caused by changed uid (partially init)

#### Testresult

This test was passed with the state: **Success**.

---

**Info** Initialising property\_cache.json (load\_all\_on\_init=True).

---

Deleting cache file from filesystem to ensure identical conditions for each test run.

Initialising cached class with /home/dirk/my\_repositories/unittest/caching/unittest/output\_data/uid\_test\_no\_load\_on\_init.json as cache  
 ↪ file.

---

**Success** Instance data after changing uid is correct (Content {'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '2': 2, '3': 'three', '4': '4'}} and Type is <class 'dict'>).

---

Result (Instance data after changing uid): { 'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_',  
 ↪ 'integer': 34, 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ], 'dict': { '1': '1', '2':  
 ↪ 2, '3': 'three', '4': '4' } } (<class 'dict'>)

Expectation (Instance data after changing uid): result = { 'str': '\_\_string\_\_', 'unicode':  
 ↪ '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ], 'dict': {  
 ↪ '1': '1', '2': 2, '3': 'three', '4': '4' } } (<class 'dict'>)

### A.1.9 caching.property\_cache.json: Test execution of save callback (full init)

#### Testresult

This test was passed with the state: **Success**.

---

**Info** Installing save\_callback, which sets a variable to True on execution.

---



---

**Success** Save callback execution variable is correct (Content True and Type is <class 'bool'>).

---

Result (Save callback execution variable): True (<class 'bool'>)

Expectation (Save callback execution variable): result = True (<class 'bool'>)

### A.1.10 caching.property\_cache.json: Test internal key usage

#### Testresult

This test was passed with the state: **Success**.

---

**Info** Initialising property\_cache.json (load\_all\_on\_init=True).

---

Deleting cache file from filesystem to ensure identical conditions for each test run.

Initialising cached class with  
 ↪ /home/dirk/my\_repositories/unittest/caching/unittest/output\_data/internal\_keys\_test.json  
 ↪ as cache file.

---

**Info** Extracting storage object from property\_cache.json for comparison.

---

Using storage object of cache class for comparison: {'\_\_property\_cache\_data\_version\_': 'no  
 ↪ second data version', '\_\_property\_cache\_uid\_': 'no second uid',  
 ↪ '\_\_property\_cache\_data\_version\_': 'no data version', '\_\_property\_cache\_uid\_': 'no uid',  
 ↪ '\_property\_cache\_data\_version\_': 1, '\_property\_cache\_uid\_': 'my\_unique\_id'}

Deleting overhead keys: \_property\_cache\_data\_version\_, \_property\_cache\_uid\_

---

**Success** Cache is correct (Content {'\_\_property\_cache\_data\_version\_': 'no second data version',  
 '\_\_property\_cache\_uid\_': 'no second uid', '\_property\_cache\_data\_version\_': 'no data version',  
 '\_property\_cache\_uid\_': 'no uid'} and Type is <class 'dict'>).

---

Result (Cache): { '\_\_property\_cache\_data\_version\_': 'no second data version',  
 ↪ '\_\_property\_cache\_uid\_': 'no second uid', '\_\_property\_cache\_data\_version\_': 'no data  
 ↪ version', '\_\_property\_cache\_uid\_': 'no uid' } (<class 'dict'>)

Expectation (Cache): result = { '\_\_property\_cache\_uid\_': 'no uid', '\_\_property\_cache\_uid\_':  
 ↪ 'no second uid', '\_property\_cache\_data\_version\_': 'no data version',  
 ↪ '\_\_property\_cache\_data\_version\_': 'no second data version' } (<class 'dict'>)

---

**Success** Keyfilter returnvalue for 5 (<class 'int'>) is correct (Content 5 and Type is <class 'int'>).

---

Result (Keyfilter returnvalue for 5 (<class 'int'>)): 5 (<class 'int'>)

Expectation (Keyfilter returnvalue for 5 (<class 'int'>)): result = 5 (<class 'int'>)

---

### A.1.11 caching.property\_cache\_pickle: Test full initialised PICKLE-Cache-Object

#### Testresult

This test was passed with the state: **Success**.

---

**Info** Initialising property\_cache\_pickle (load\_all\_on\_init=True).

---

Deleting cache file from filesystem to ensure identical conditions for each test run.

Initialising cached class with

↪ /home/dirk/my\_repositories/unittest/caching/unittest/output\_data/load\_on\_init.pkl as  
 ↪ cache file.

---

**Info** Extracting storage object from property\_cache\_pickle for comparison.

---

Using storage object of cache class for comparison: {'str': 'string', 'unicode': 'unicode',  
 ↪ 'integer': 17, 'float': 3.14159, 'list': [1, 'two', '3', 4], 'dict': {'1': 1, '2': 'two',  
 ↪ '3': '3', '4': 4}, '\_property\_cache\_uid\_': 'my\_unique\_id',  
 ↪ '\_property\_cache\_data\_version\_': 1}

Deleting overhead keys: \_property\_cache\_data\_version\_, \_property\_cache\_uid\_

---

**Success** Cache object is correct (Content {'str': 'string', 'unicode': 'unicode', 'integer': 17, 'float': 3.14159, 'list': [1, 'two', '3', 4], 'dict': {'1': 1, '2': 'two', '3': '3', '4': 4}} and Type is <class 'dict'>).

---

```
Result (Cache object): { 'str': 'string', 'unicode': 'unicode', 'integer': 17, 'float':
↪ 3.14159, 'list': [ 1, 'two', '3', 4 ], 'dict': { '1': 1, '2': 'two', '3': '3', '4': 4 } }
↪ (<class 'dict'>)
```

```
Expectation (Cache object): result = { 'str': 'string', 'unicode': 'unicode', 'integer': 17,
↪ 'float': 3.14159, 'list': [ 1, 'two', '3', 4 ], 'dict': { '1': 1, '2': 'two', '3': '3',
↪ '4': 4 } } (<class 'dict'>)
```

### A.1.12 caching.property\_cache\_pickle: Test partially initialised PICKLE-Cache-Object

#### Testresult

This test was passed with the state: **Success**.

---

**Info** Initialising property\_cache\_pickle (load\_all\_on\_init=False).

---

Deleting cache file from filesystem to ensure identical conditions for each test run.

Initialising cached class with

```
↪ /home/dirk/my_repositories/unittest/caching/unittest/output_data/no_load_on_init.pkl as
↪ cache file.
```

---

**Info** Partially initialising cache object by requesting some information.

---



---

**Info** Extracting storage object from property\_cache\_pickle for comparison.

---

```
Using storage object of cache class for comparison: {'_property_cache_uid': 'my_unique_id',
↪ '_property_cache_data_version': 1, 'str': 'string', 'integer': 17, 'unicode': 'unicode'}
```

Deleting overhead keys: \_property\_cache\_data\_version\_, \_property\_cache\_uid\_

---

**Success** Cache object is correct (Content {'str': 'string', 'integer': 17, 'unicode': 'unicode'} and Type is <class 'dict'>).

---

```
Result (Cache object): { 'str': 'string', 'integer': 17, 'unicode': 'unicode' } (<class
↪ 'dict'>)
```

```
Expectation (Cache object): result = { 'str': 'string', 'unicode': 'unicode', 'integer': 17 }
↪ (<class 'dict'>)
```

### A.1.13 caching.property\_cache\_pickle: Test cached data (full init)

#### Testresult

This test was passed with the state: **Success**.

---

**Info** Initialising property\_cache\_pickle (load\_all\_on\_init=True).

---

Deleting cache file from filesystem to ensure identical conditions for each test run.

```
Initialising cached class with /home/dirk/my_repositories/unittest/caching/unittest/output_da
↪ ta/cache_data_test_load_on_init.pkl as cache
↪ file.
```

---

**Info** Collecting data from cache instance.

---

**Success** Cached data is correct (Content {'str': 'string', 'unicode': 'unicode', 'integer': 17, 'float': 3.14159, 'list': [1, 'two', '3', 4], 'dict': {'1': 1, '2': 'two', '3': '3', '4': 4}, 'uncached': 'uncached\_data\_of\_class'} and Type is <class 'dict'>).

---

```
Result (Cached data): { 'str': 'string', 'unicode': 'unicode', 'integer': 17, 'float':
↪ 3.14159, 'list': [ 1, 'two', '3', 4 ], 'dict': { '1': 1, '2': 'two', '3': '3', '4': 4 },
↪ 'uncached': 'uncached_data_of_class' } (<class 'dict'>)
```

```
Expectation (Cached data): result = { 'str': 'string', 'unicode': 'unicode', 'integer': 17,
↪ 'float': 3.14159, 'list': [ 1, 'two', '3', 4 ], 'dict': { '1': 1, '2': 'two', '3': '3',
↪ '4': 4 }, 'uncached': 'uncached_data_of_class' } (<class 'dict'>)
```

#### A.1.14 caching.property\_cache\_pickle: Test cached data (partially init)

##### Testresult

This test was passed with the state: **Success**.

---

**Info** Initialising property\_cache\_pickle (load\_all\_on\_init=True).

---

```
Deleting cache file from filesystem to ensure identical conditions for each test run.
```

```
Initialising cached class with /home/dirk/my_repositories/unittest/caching/unittest/output_da
↪ ta/cache_data_test_no_load_on_init.pkl as cache
↪ file.
```

---

**Info** Collecting data from cache instance.

---

**Success** Cached data is correct (Content {'str': 'string', 'unicode': 'unicode', 'integer': 17, 'float': 3.14159, 'list': [1, 'two', '3', 4], 'dict': {'1': 1, '2': 'two', '3': '3', '4': 4}, 'uncached': 'uncached\_data\_of\_class'} and Type is <class 'dict'>).

---

```
Result (Cached data): { 'str': 'string', 'unicode': 'unicode', 'integer': 17, 'float':
↪ 3.14159, 'list': [ 1, 'two', '3', 4 ], 'dict': { '1': 1, '2': 'two', '3': '3', '4': 4 },
↪ 'uncached': 'uncached_data_of_class' } (<class 'dict'>)
```

```
Expectation (Cached data): result = { 'str': 'string', 'unicode': 'unicode', 'integer': 17,
↪ 'float': 3.14159, 'list': [ 1, 'two', '3', 4 ], 'dict': { '1': 1, '2': 'two', '3': '3',
↪ '4': 4 }, 'uncached': 'uncached_data_of_class' } (<class 'dict'>)
```

**A.1.15 caching.property\_cache\_pickle: Test get from source caused by increased data version (full init)**

**Testresult**

This test was passed with the state: **Success**.

---

**Info** Initialising property\_cache\_pickle (load\_all\_on\_init=True).

---

Deleting cache file from filesystem to ensure identical conditions for each test run.

Initialising cached class with /home/dirk/my\_repositories/unittest/caching/unittest/output\_data/data\_version\_test\_load\_on\_init.pkl as cache file.

---

**Success** Instance data after increasing data\_version is correct (Content {'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '2': 2, '3': 'three', '4': '4'}} and Type is <class 'dict'>).

---

Result (Instance data after increasing data\_version): { 'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ], 'dict': { '1': '1', '2': 2, '3': 'three', '4': '4' } } (<class 'dict'>)

Expectation (Instance data after increasing data\_version): result = { 'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ], 'dict': { '1': '1', '2': 2, '3': 'three', '4': '4' } } (<class 'dict'>)

**A.1.16 caching.property\_cache\_pickle: Test get from source caused by increased data version (partially init)**

**Testresult**

This test was passed with the state: **Success**.

---

**Info** Initialising property\_cache\_pickle (load\_all\_on\_init=True).

---

Deleting cache file from filesystem to ensure identical conditions for each test run.

Initialising cached class with /home/dirk/my\_repositories/unittest/caching/unittest/output\_data/data\_version\_test\_no\_load\_on\_init.pkl as cache file.

---

**Success** Instance data after increasing data\_version is correct (Content {'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '2': 2, '3': 'three', '4': '4'}} and Type is <class 'dict'>).

---

Result (Instance data after increasing data\_version): { 'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ], 'dict': { '1': '1', '2': 2, '3': 'three', '4': '4' } } (<class 'dict'>)

Expectation (Instance data after increasing data\_version): result = { 'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ], 'dict': { '1': '1', '2': 2, '3': 'three', '4': '4' } } (<class 'dict'>)

### A.1.17 caching.property\_cache\_pickle: Test get from source caused by changed uid (full init)

#### Testresult

This test was passed with the state: **Success**.

---

**Info** Initialising property\_cache\_pickle (load\_all\_on\_init=True).

---

Deleting cache file from filesystem to ensure identical conditions for each test run.

Initialising cached class with /home/dirk/my\_repositories/unittest/caching/unittest/output\_data/uid\_test\_load\_on\_init.pkl as cache  
 ↳ file.

---

**Success** Instance data after changing uid is correct (Content {'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '2': 2, '3': 'three', '4': '4'}} and Type is <class 'dict'>).

---

Result (Instance data after changing uid): { 'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_',  
 ↳ 'integer': 34, 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ], 'dict': { '1': '1', '2':  
 ↳ 2, '3': 'three', '4': '4' } } (<class 'dict'>)

Expectation (Instance data after changing uid): result = { 'str': '\_\_string\_\_', 'unicode':  
 ↳ '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ], 'dict': {  
 ↳ '1': '1', '2': 2, '3': 'three', '4': '4' } } (<class 'dict'>)

### A.1.18 caching.property\_cache\_pickle: Test get from source caused by changed uid (partially init)

#### Testresult

This test was passed with the state: **Success**.

---

**Info** Initialising property\_cache\_pickle (load\_all\_on\_init=True).

---

Deleting cache file from filesystem to ensure identical conditions for each test run.

Initialising cached class with /home/dirk/my\_repositories/unittest/caching/unittest/output\_data/uid\_test\_no\_load\_on\_init.pkl as cache  
 ↳ file.

---

**Success** Instance data after changing uid is correct (Content {'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '2': 2, '3': 'three', '4': '4'}} and Type is <class 'dict'>).

---

Result (Instance data after changing uid): { 'str': '\_\_string\_\_', 'unicode': '\_\_unicode\_\_',  
 ↳ 'integer': 34, 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ], 'dict': { '1': '1', '2':  
 ↳ 2, '3': 'three', '4': '4' } } (<class 'dict'>)

Expectation (Instance data after changing uid): result = { 'str': '\_\_string\_\_', 'unicode':  
 ↳ '\_\_unicode\_\_', 'integer': 34, 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ], 'dict': {  
 ↳ '1': '1', '2': 2, '3': 'three', '4': '4' } } (<class 'dict'>)

### A.1.19 caching.property\_cache\_pickle: Test execution of save callback (full init)

#### Testresult

This test was passed with the state: **Success**.

---

**Info** Installing save\_callback, which sets a variable to True on execution.

---

**Success** Save callback execution variable is correct (Content True and Type is <class 'bool'>).

---

Result (Save callback execution variable): True (<class 'bool'>)

Expectation (Save callback execution variable): result = True (<class 'bool'>)

### A.1.20 caching.property\_cache\_pickle: Test internal key usage

#### Testresult

This test was passed with the state: **Success**.

---

**Info** Initialising property\_cache\_pickle (load\_all\_on\_init=True).

---

Deleting cache file from filesystem to ensure identical conditions for each test run.

Initialising cached class with

```
↪ /home/dirk/my_repositories/unittest/caching/unittest/output_data/internal_keys_test.pkl
↪ as cache file.
```

---

**Info** Extracting storage object from property\_cache\_pickle for comparison.

---

```
Using storage object of cache class for comparison: {'__property_cache_uid_': 'no uid',
↪ '___property_cache_uid_': 'no second uid', '__property_cache_data_version_': 'no data
↪ version', '___property_cache_data_version_': 'no second data version',
↪ '_property_cache_uid_': 'my_unique_id', '_property_cache_data_version_': 1}
```

Deleting overhead keys: \_property\_cache\_data\_version\_, \_property\_cache\_uid\_

---

**Success** Cache is correct (Content {'\_\_property\_cache\_uid\_': 'no uid', '\_\_\_property\_cache\_uid\_': 'no second uid', '\_property\_cache\_data\_version\_': 'no data version', '\_\_\_property\_cache\_data\_version\_': 'no second data version'} and Type is <class 'dict'>).

---

Result (Cache): { '\_\_property\_cache\_uid\_': 'no uid', '\_\_\_property\_cache\_uid\_': 'no second uid', '\_property\_cache\_data\_version\_': 'no data version', '\_\_\_property\_cache\_data\_version\_': 'no second data version' } (<class 'dict'>)

Expectation (Cache): result = { '\_\_property\_cache\_uid\_': 'no uid', '\_\_\_property\_cache\_uid\_': 'no second uid', '\_property\_cache\_data\_version\_': 'no data version', '\_\_\_property\_cache\_data\_version\_': 'no second data version' } (<class 'dict'>)

---

**Success** Keyfilter returnvalue for 5 (<class 'int'>) is correct (Content 5 and Type is <class 'int'>).

---

Result (Keyfilter returnvalue for 5 (<class 'int'>)): 5 (<class 'int'>)

Expectation (Keyfilter returnvalue for 5 (<class 'int'>)): result = 5 (<class 'int'>)



## B Test-Coverage

### B.1 caching

The line coverage for caching was 99.1%

The branch coverage for caching was 97.8%

#### B.1.1 caching.\_\_init\_\_.py

The line coverage for caching.\_\_init\_\_.py was 99.1%

The branch coverage for caching.\_\_init\_\_.py was 97.8%

```

1 #!/usr/bin/env python
2 # -*- coding: utf-8 -*-
3 #
4 """
5 caching (Caching Module)
6 =====
7
8 **Author:**
9
10 * Dirk Alders <sudo-dirk@mount-mockery.de>
11
12 **Description:**
13
14     This Module supports functions and classes for caching e.g. properties of other instances.
15
16 **Submodules:**
17
18 * :class:`caching.property_cache_json`
19 * :class:`caching.property_cache_pickle`
20
21 **Unittest:**
22
23     See also the :download:`unittest <caching/_testresults_/unittest.pdf>` documentation.
24 """
25 __DEPENDENCIES__ = []
26
27 import hashlib
28 import hmac
29 import json
30 import logging
31 import os
32 import pickle
33 import sys
34
35 try:
36     from config import APP_NAME as ROOT_LOGGER_NAME
37 except ImportError:
38     ROOT_LOGGER_NAME = 'root'
39 logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__)
40
41 __DESCRIPTION__ = """The Module {\\tt %s} is designed to store information in {\\tt json} or {\\tt pickle} files to support them much faster then generating them from the original source file.
42 For more Information read the documentation.""" % __name__.replace('-', '\\-')
43 """The Module Description"""
44 __INTERPRETER__ = (3, )
45 """The Tested Interpreter-Versions"""

```

## Unittest for caching

```
46
47
48 class property_cache_pickle(object):
49     """
50     Class to cache properties, which take longer on initialising than reading a file in pickle
51     format.
52
53     :param source_instance: The source instance holding the data
54     :type source_instance: instance
55     :param cache_filename: File name, where the properties are stored as cache
56     :type cache_filename: str
57     :param load_all_on_init: Optionally init behaviour control parameter. True will load all
58     available properties from source on init, False not.
59
60     .. note:: source_instance needs to have at least the following methods: uid(), keys(),
61     data_version(), get()
62
63         * uid(): returns the unique id of the source.
64         * keys(): returns a list of all available keys.
65         * data_version(): returns a version number of the current data (it should be
66     increased, if the get method of the source instance returns improved values or the data
67     structure had been changed).
68         * get(key, default): returns the property for a key. If key does not exists,
69     default will be returned.
70
71     Reasons for updating the complete data set:
72
73     * UID of source_instance has changed (in comparison to the cached value).
74     * data_version is increased
75
76     **Example:**
77
78     .. literalinclude:: caching/_examples_/property_cache_pickle.py
79
80     Will result on the first execution to the following output (with a long execution time):
81
82     .. literalinclude:: caching/_examples_/property_cache_pickle_1.log
83
84     With every following execution (slow for getting "two" which is not cached – see
85     implementation):
86
87     .. literalinclude:: caching/_examples_/property_cache_pickle_2.log
88     """
89     LOG_PREFIX = 'PickCache: '
90     DATA_VERSION_TAG = '_property_cache_data_version_'
91     UID_TAG = '_property_cache_uid_'
92
93     def __init__(self, source_instance, cache_filename, load_all_on_init=False,
94     callback_on_data_storage=None):
95         self._source_instance = source_instance
96         self._cache_filename = cache_filename
97         self._load_all_on_init = load_all_on_init
98         self._callback_on_data_storage = callback_on_data_storage
99         self._cached_props = None
100
101     def get(self, key, default=None):
102         """
103         Method to get the cached property. If key does not exists in cache, the property will be
104         loaded from source_instance and stored in cache (file).
105
106         :param key: key for value to get.
107         """
```

## Unittest for caching

```
98     :param default: value to be returned, if key does not exists.
99     :returns: value for a given key or default value.
100     """
101     if key in self.keys():
102         if self._cached_props is None:
103             self._init_cache()
104         if self._key_filter(key) not in self._cached_props:
105             val = self._source_instance.get(key, None)
106             logger.debug("%s Loading property for '%s' from source instance (%s)", self.
LOG_PREFIX, key, repr(val))
107             self._cached_props[self._key_filter(key)] = val
108             self._save_cache()
109         else:
110             logger.debug("%s Providing property for '%s' from cache (%s)", self.LOG_PREFIX,
key, repr(self._cached_props.get(self._key_filter(key), default)))
111             return self._cached_props.get(self._key_filter(key), default)
112         else:
113             logger.info("%s Key '%s' is not in cached_keys. Uncached data will be returned.",
self.LOG_PREFIX, key)
114             return self._source_instance.get(key, default)
115
116     def keys(self):
117         """
118         Method to get the available keys (from :data:`source_instance`).
119         """
120         return self._source_instance.keys()
121
122     def _data_version(self):
123         if self._cached_props is None:
124             return None
125         else:
126             return self._cached_props.get(self.DATA_VERSION_TAG, None)
127
128     def _init_cache(self):
129         if not self._load_cache() or self._source_instance.uid() != self._uid() or self.
_source_instance.data_version() > self._data_version():
130             if self._uid() is not None and self._source_instance.uid() != self._uid():
131                 logger.debug("%s Source uid changed, ignoring previous cache data", self.
LOG_PREFIX)
132             if self._data_version() is not None and self._source_instance.data_version() > self.
_data_version():
133                 logger.debug("%s Data version increased, ignoring previous cache data", self.
LOG_PREFIX)
134             self._cached_props = dict()
135             if self._load_all_on_init:
136                 self._load_source()
137             self._cached_props[self.UID_TAG] = self._source_instance.uid()
138             self._cached_props[self.DATA_VERSION_TAG] = self._source_instance.data_version()
139             self._save_cache()
140
141     def _load_cache(self):
142         if os.path.exists(self._cache_filename):
143             with open(self._cache_filename, 'rb') as fh:
144                 self._cached_props = pickle.load(fh)
145             logger.info('%s Loading properties from cache (%s)', self.LOG_PREFIX, self.
_cache_filename)
146             return True
147         else:
148             logger.debug('%s Cache file does not exists (yet).', self.LOG_PREFIX)
149             return False
150
151     def _key_filter(self, key):
```

## Unittest for caching

```
152     if sys.version_info >= (3, 0):
153         tps = [str]
154     else:
155         tps = [str, unicode]
156     if type(key) in tps:
157         if key.endswith(self.DATA_VERSION_TAG) or key.endswith(self.UID_TAG):
158             return '_' + key
159     return key
160
161     def _load_source(self):
162         logger.debug('%s Loading all data from source - %s', self.LOG_PREFIX, repr(self._source_instance.keys()))
163         for key in self._source_instance.keys():
164             val = self._source_instance.get(key)
165             self._cached_props[self._key_filter(key)] = val
166
167     def _save_cache(self):
168         with open(self._cache_filename, 'wb') as fh:
169             pickle.dump(self._cached_props, fh)
170             logger.info('%s cache-file stored (%s)', self.LOG_PREFIX, self._cache_filename)
171         if self._callback_on_data_storage is not None:
172             self._callback_on_data_storage()
173
174     def _uid(self):
175         if self._cached_props is None:
176             return None
177         else:
178             return self._cached_props.get(self.UID_TAG, None)
179
180
181 class property_cache_json(property_cache_pickle):
182     """
183     Class to cache properties, which take longer on initialising than reading a file in json
184     format. See also parent :py:class:`property_cache_pickle`
185
186     :param source_instance: The source instance holding the data
187     :type source_instance: instance
188     :param cache_filename: File name, where the properties are stored as cache
189     :type cache_filename: str
190     :param load_all_on_init: Optionally init behaviour control parameter. True will load all
191     available properties from source on init, False not.
192
193     .. warning::
194         * This class uses json. You should only use keys of type string!
195         * Unicode types are transferred to strings
196
197     .. note:: source_instance needs to have at least the following methods: uid(), keys(),
198     data_version(), get()
199
200         * uid(): returns the unique id of the source.
201         * keys(): returns a list of all available keys.
202         * data_version(): returns a version number of the current data (it should be
203     increased, if the get method of the source instance returns improved values or the data
204     structure had been changed).
205         * get(key, default): returns the property for a key. If key does not exists,
206     default will be returned.
207
208     Reasons for updating the complete data set:
209
210     * UID of source_instance has changed (in comparison to the cached value).
211     * data_version is increased
```

## Unittest for caching

```
207 **Example:**
208
209 .. literalinclude:: caching/_examples_/property_cache_json.py
210
211 Will result on the first execution to the following output (with a long execution time):
212
213 .. literalinclude:: caching/_examples_/property_cache_json_1.log
214
215 With every following execution (slow for getting "two" which is not cached – see
216 implementation):
217
218 .. literalinclude:: caching/_examples_/property_cache_json_2.log
219 """
220 LOG.PREFIX = 'JsonCache: '
221
222 def _load_cache(self):
223     if os.path.exists(self._cache_filename):
224         with open(self._cache_filename, 'r') as fh:
225             self._cached_props = json.load(fh)
226             logger.info('%s Loading properties from cache (%s)', self.LOG_PREFIX, self._
227                 _cache_filename)
228             return True
229     else:
230         logger.debug('%s Cache file does not exists (yet).', self.LOG_PREFIX)
231         return False
232
233 def _save_cache(self):
234     with open(self._cache_filename, 'w') as fh:
235         json.dump(self._cached_props, fh, sort_keys=True, indent=4)
236         logger.info('%s cache-file stored (%s)', self.LOG_PREFIX, self._cache_filename)
237     if self._callback_on_data_storage is not None:
238         self._callback_on_data_storage()
```