

# Unittest for caching

January 7, 2021

## Contents

<b>1</b>	<b>Test Information</b>	<b>5</b>
1.1	Test Candidate Information . . . . .	5
1.2	Unittest Information . . . . .	5
1.3	Test System Information . . . . .	5
<b>2</b>	<b>Statistic</b>	<b>5</b>
2.1	Test-Statistic for testrun with python 2.7.18 (final) . . . . .	5
2.2	Test-Statistic for testrun with python 3.8.5 (final) . . . . .	6
2.3	Coverage Statistic . . . . .	6
<b>3</b>	<b>Testcases with no corresponding Requirement</b>	<b>7</b>
3.1	Summary for testrun with python 2.7.18 (final) . . . . .	7
3.1.1	caching.property_cache.json: Test cached data (full init) . . . . .	7
3.1.2	caching.property_cache.json: Test cached data (partially init) . . . . .	7
3.1.3	caching.property_cache.json: Test execution of save callback (full init) . . . . .	7
3.1.4	caching.property_cache.json: Test full initialised JSON-Cache-Object . . . . .	8
3.1.5	caching.property_cache.json: Test get from source caused by changed uid (full init) . . . . .	8
3.1.6	caching.property_cache.json: Test get from source caused by changed uid (partially init) . . . . .	8
3.1.7	caching.property_cache.json: Test get from source caused by increased data version (full init) . . . . .	9
3.1.8	caching.property_cache.json: Test get from source caused by increased data version (partially init) . . . . .	9
3.1.9	caching.property_cache.json: Test internal key usage . . . . .	9
3.1.10	caching.property_cache.json: Test partially initialisation of JSON-Cache-Object . . . . .	10
3.1.11	caching.property_cache.pickle: Test cached data (full init) . . . . .	10
3.1.12	caching.property_cache.pickle: Test cached data (partially init) . . . . .	11
3.1.13	caching.property_cache.pickle: Test execution of save callback (full init) . . . . .	11
3.1.14	caching.property_cache.pickle: Test full initialised PICKLE-Cache-Object . . . . .	11
3.1.15	caching.property_cache.pickle: Test get from source caused by changed uid (full init) . . . . .	12
3.1.16	caching.property_cache.pickle: Test get from source caused by changed uid (partially init) . . . . .	12
3.1.17	caching.property_cache.pickle: Test get from source caused by increased data version (full init) . . . . .	12
3.1.18	caching.property_cache.pickle: Test get from source caused by increased data version (partially init) . . . . .	13

3.1.19	caching.property_cache_pickle: Test internal key usage . . . . .	13
3.1.20	caching.property_cache_pickle: Test partially initialised PICKLE-Cache-Object . . . . .	14
3.2	Summary for testrun with python 3.8.5 (final) . . . . .	14
3.2.1	caching.property_cache_json: Test cached data (full init) . . . . .	14
3.2.2	caching.property_cache_json: Test cached data (partially init) . . . . .	14
3.2.3	caching.property_cache_json: Test execution of save callback (full init) . . . . .	15
3.2.4	caching.property_cache_json: Test full initialised JSON-Cache-Object . . . . .	15
3.2.5	caching.property_cache_json: Test get from source caused by changed uid (full init) . . . . .	15
3.2.6	caching.property_cache_json: Test get from source caused by changed uid (partially init) . . . . .	16
3.2.7	caching.property_cache_json: Test get from source caused by increased data version (full init) . . . . .	16
3.2.8	caching.property_cache_json: Test get from source caused by increased data version (partially init) . . . . .	17
3.2.9	caching.property_cache_json: Test internal key usage . . . . .	17
3.2.10	caching.property_cache_json: Test partially initialisation of JSON-Cache-Object . . . . .	17
3.2.11	caching.property_cache_pickle: Test cached data (full init) . . . . .	18
3.2.12	caching.property_cache_pickle: Test cached data (partially init) . . . . .	18
3.2.13	caching.property_cache_pickle: Test execution of save callback (full init) . . . . .	18
3.2.14	caching.property_cache_pickle: Test full initialised PICKLE-Cache-Object . . . . .	19
3.2.15	caching.property_cache_pickle: Test get from source caused by changed uid (full init) . . . . .	19
3.2.16	caching.property_cache_pickle: Test get from source caused by changed uid (partially init) . . . . .	20
3.2.17	caching.property_cache_pickle: Test get from source caused by increased data version (full init) . . . . .	20
3.2.18	caching.property_cache_pickle: Test get from source caused by increased data version (partially init) . . . . .	20
3.2.19	caching.property_cache_pickle: Test internal key usage . . . . .	21
3.2.20	caching.property_cache_pickle: Test partially initialised PICKLE-Cache-Object . . . . .	21

**A Trace for testrun with python 2.7.18 (final) . . . . . 22**

A.1	Tests with status <b>Warning</b> (4) . . . . .	22
A.1.1	caching.property_cache_json: Test full initialised JSON-Cache-Object . . . . .	22
A.1.2	caching.property_cache_json: Test partially initialisation of JSON-Cache-Object . . . . .	23
A.1.3	caching.property_cache_json: Test cached data (full init) . . . . .	25
A.1.4	caching.property_cache_json: Test cached data (partially init) . . . . .	26

A.2	Tests with status Info (16)	27
A.2.1	caching.property_cache_json: Test get from source caused by increased data version (full init)	27
A.2.2	caching.property_cache_json: Test get from source caused by increased data version (partially init)	28
A.2.3	caching.property_cache_json: Test get from source caused by changed uid (full init)	30
A.2.4	caching.property_cache_json: Test get from source caused by changed uid (partially init)	31
A.2.5	caching.property_cache_json: Test execution of save callback (full init)	32
A.2.6	caching.property_cache_json: Test internal key usage	33
A.2.7	caching.property_cache_pickle: Test full initialised PICKLE-Cache-Object	34
A.2.8	caching.property_cache_pickle: Test partially initialised PICKLE-Cache-Object	34
A.2.9	caching.property_cache_pickle: Test cached data (full init)	35
A.2.10	caching.property_cache_pickle: Test cached data (partially init)	36
A.2.11	caching.property_cache_pickle: Test get from source caused by increased data version (full init)	37
A.2.12	caching.property_cache_pickle: Test get from source caused by increased data version (partially init)	38
A.2.13	caching.property_cache_pickle: Test get from source caused by changed uid (full init)	40
A.2.14	caching.property_cache_pickle: Test get from source caused by changed uid (partially init)	41
A.2.15	caching.property_cache_pickle: Test execution of save callback (full init)	42
A.2.16	caching.property_cache_pickle: Test internal key usage	43
<b>B</b>	<b>Trace for testrun with python 3.8.5 (final)</b>	<b>44</b>
B.1	Tests with status Info (20)	44
B.1.1	caching.property_cache_json: Test full initialised JSON-Cache-Object	44
B.1.2	caching.property_cache_json: Test partially initialisation of JSON-Cache-Object	44
B.1.3	caching.property_cache_json: Test cached data (full init)	45
B.1.4	caching.property_cache_json: Test cached data (partially init)	46
B.1.5	caching.property_cache_json: Test get from source caused by increased data version (full init)	47
B.1.6	caching.property_cache_json: Test get from source caused by increased data version (partially init)	48
B.1.7	caching.property_cache_json: Test get from source caused by changed uid (full init)	50
B.1.8	caching.property_cache_json: Test get from source caused by changed uid (partially init)	51
B.1.9	caching.property_cache_json: Test execution of save callback (full init)	52
B.1.10	caching.property_cache_json: Test internal key usage	53

Unittest for caching

B.1.11	caching.property_cache_pickle: Test full initialised PICKLE-Cache-Object . . . . .	54
B.1.12	caching.property_cache_pickle: Test partially initialised PICKLE-Cache-Object . . . . .	54
B.1.13	caching.property_cache_pickle: Test cached data (full init) . . . . .	55
B.1.14	caching.property_cache_pickle: Test cached data (partially init) . . . . .	56
B.1.15	caching.property_cache_pickle: Test get from source caused by increased data version (full init) .	57
B.1.16	caching.property_cache_pickle: Test get from source caused by increased data version (partially init) . . . . .	58
B.1.17	caching.property_cache_pickle: Test get from source caused by changed uid (full init) . . . . .	60
B.1.18	caching.property_cache_pickle: Test get from source caused by changed uid (partially init) . . . .	61
B.1.19	caching.property_cache_pickle: Test execution of save callback (full init) . . . . .	62
B.1.20	caching.property_cache_pickle: Test internal key usage . . . . .	63
<b>C</b>	<b>Test-Coverage</b>	<b>64</b>
C.1	caching . . . . .	64
C.1.1	caching.__init__.py . . . . .	64

## 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	python2, python3
Version	6c187a02e86ab5fee81eb4f34f61602b

---

#### Dependencies

---

### 1.2 Unittest Information

---

#### Unittest Information

Version	e113c3759194c81e7df9e29b1debe00d
Testruns with	python 2.7.18 (final), python 3.8.5 (final)

---

### 1.3 Test System Information

---

#### System Information

Architecture	64bit
Distribution	Linux Mint 20 ulyana
Hostname	ahorn
Kernel	5.4.0-59-generic (#65-Ubuntu SMP Thu Dec 10 12:01:51 UTC 2020)
Machine	x86_64
Path	/user_data/data/dirk/prj/unittest/caching/unittest
System	Linux
Username	dirk

---

## 2 Statistic

### 2.1 Test-Statistic for testrun with python 2.7.18 (final)

---

Number of tests	<b>20</b>
Number of successfull tests	<b>16</b>
Number of possibly failed tests	<b>4</b>
Number of failed tests	<b>0</b>

---

Executionlevel	Full Test (all defined tests)
Time consumption	0.042s

---

## 2.2 Test-Statistic for testrun with python 3.8.5 (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	0.034s

---

## 2.3 Coverage Statistic

---

<b>Module- or Filename</b>	<b>Line-Coverage</b>	<b>Branch-Coverage</b>
caching	100.0%	100.0%
caching.__init__.py	100.0%	

---

### 3 Testcases with no corresponding Requirement

#### 3.1 Summary for testrun with python 2.7.18 (final)

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

###### Testresult

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

---

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (31)
Start-Time:	2021-01-07 21:59:04,735
Finished-Time:	2021-01-07 21:59:04,737
Time-Consumption	0.002s

---

**Testsummary:**

---

<b>Info</b>	Initialising property_cache_json (load_all_on_init=True).
<b>Info</b>	Collecting data from cache instance.
<b>Warning</b>	Cached data is NOT correct. See detailed log for more information.

---

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

###### Testresult

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

---

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (32)
Start-Time:	2021-01-07 21:59:04,737
Finished-Time:	2021-01-07 21:59:04,739
Time-Consumption	0.002s

---

**Testsummary:**

---

<b>Info</b>	Initialising property_cache_json (load_all_on_init=True).
<b>Info</b>	Collecting data from cache instance.
<b>Warning</b>	Cached data is NOT correct. See detailed log for more information.

---

##### 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.2.5!

---

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (37)
Start-Time:	2021-01-07 21:59:04,751
Finished-Time:	2021-01-07 21:59:04,752
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 <type 'bool'>).

---

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

**Testresult**

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

---

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (29)
Start-Time:	2021-01-07 21:59:04,731
Finished-Time:	2021-01-07 21:59:04,733
Time-Consumption	0.002s

---

**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>Warning</b>	Cache object is NOT correct. See detailed log for more information.

---

**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.2.3!

---

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (35)
Start-Time:	2021-01-07 21:59:04,746
Finished-Time:	2021-01-07 21:59:04,748
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 {'unicode': u'__unicode__', 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '3': 'three', '2': 2, '4': '4'}, 'str': '__string__', 'integer': 34} and Type is <type '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.2.4!

---

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (36)
Start-Time:	2021-01-07 21:59:04,748
Finished-Time:	2021-01-07 21:59:04,751

---

Time-Consumption 0.003s

---

**Testsummary:**

**Info** Initialising property\_cache.json (load\_all\_on\_init=True).  
**Success** Instance data after changing uid is correct (Content {'unicode': u'\_\_unicode\_\_', 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '3': 'three', '2': 2, '4': '4'}, 'str': '\_\_string\_\_', 'integer': 34} and Type is <type '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.2.1!

---

Testrun: python 2.7.18 (final)  
 Caller: /user\_data/data/dirk/prj/unittest/caching/unittest/src/tests/\_\_\_init\_\_\_py (33)  
 Start-Time: 2021-01-07 21:59:04,739  
 Finished-Time: 2021-01-07 21:59:04,741  
 Time-Consumption 0.002s

---

**Testsummary:**

**Info** Initialising property\_cache.json (load\_all\_on\_init=True).  
**Success** Instance data after increasing data\_version is correct (Content {'unicode': u'\_\_unicode\_\_', 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '3': 'three', '2': 2, '4': '4'}, 'str': '\_\_string\_\_', 'integer': 34} and Type is <type '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.2.2!

---

Testrun: python 2.7.18 (final)  
 Caller: /user\_data/data/dirk/prj/unittest/caching/unittest/src/tests/\_\_\_init\_\_\_py (34)  
 Start-Time: 2021-01-07 21:59:04,741  
 Finished-Time: 2021-01-07 21:59:04,746  
 Time-Consumption 0.005s

---

**Testsummary:**

**Info** Initialising property\_cache.json (load\_all\_on\_init=True).  
**Success** Instance data after increasing data\_version is correct (Content {'unicode': u'\_\_unicode\_\_', 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '3': 'three', '2': 2, '4': '4'}, 'str': '\_\_string\_\_', 'integer': 34} and Type is <type '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.2.6!

---

Testrun: python 2.7.18 (final)  
 Caller: /user\_data/data/dirk/prj/unittest/caching/unittest/src/tests/\_init\_.py (38)  
 Start-Time: 2021-01-07 21:59:04,752  
 Finished-Time: 2021-01-07 21:59:04,753  
 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 is correct (Content {u'\_\_property\_cache\_data\_version\_': u'no data version', u'\_\_property\_cache\_data\_version\_': u'no second data version', u'\_\_property\_cache\_uid\_': u'no uid', u'\_\_property\_cache\_uid\_': u'no second uid'}) and Type is <type 'dict'>).  
**Success** Keyfilter returnvalue for 5 (<type 'int'>) is correct (Content 5 and Type is <type 'int'>).

---

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

**Testresult**

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

---

Testrun: python 2.7.18 (final)  
 Caller: /user\_data/data/dirk/prj/unittest/caching/unittest/src/tests/\_init\_.py (30)  
 Start-Time: 2021-01-07 21:59:04,733  
 Finished-Time: 2021-01-07 21:59:04,735  
 Time-Consumption 0.002s

---

**Testsummary:**

---

**Info** Initialising property\_cache.json (load\_all\_on\_init=False).  
**Info** Partially initialising cache object by requesting some information.  
**Info** Extracting storage object from property\_cache.json for comparison.  
**Warning** Cache object is NOT correct. See detailed log for more information.

---

### 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.2.9!

---

Testrun: python 2.7.18 (final)  
 Caller: /user\_data/data/dirk/prj/unittest/caching/unittest/src/tests/\_init\_.py (44)  
 Start-Time: 2021-01-07 21:59:04,759  
 Finished-Time: 2021-01-07 21:59:04,762  
 Time-Consumption 0.003s

---

**Testsummary:**

---

**Info** Initialising property\_cache.pickle (load\_all\_on\_init=True).  
**Info** Collecting data from cache instance.

---

**Success** Cached data is correct (Content {'uncached': 'uncached\_data\_of\_class', 'unicode': u'unicode', 'float': 3.14159, 'list': [1, 'two', '3', 4], 'dict': {'1': 1, '3': '3', '2': 'two', '4': 4}, 'str': 'string', 'integer': 17} and Type is <type '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.2.10!

---

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (45)
Start-Time:	2021-01-07 21:59:04,762
Finished-Time:	2021-01-07 21:59:04,764
Time-Consumption	0.003s

---

#### 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 {'uncached': 'uncached_data_of_class', 'unicode': u'unicode', 'float': 3.14159, 'list': [1, 'two', '3', 4], 'dict': {'1': 1, '3': '3', '2': 'two', '4': 4}, 'str': 'string', 'integer': 17} and Type is <type '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.2.15!

---

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (50)
Start-Time:	2021-01-07 21:59:04,775
Finished-Time:	2021-01-07 21:59:04,775
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 <type '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.2.7!

---

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (42)
Start-Time:	2021-01-07 21:59:04,753
Finished-Time:	2021-01-07 21:59:04,755

Time-Consumption 0.002s

---

**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', 'float': 3.14159, 'list': [1, 'two', '3', 4], 'dict': {'1': 1, '3': '3', '2': 'two', '4': 4}, 'unicode': u'unicode', 'integer': 17} and Type is <type '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.2.13!

---

Testrun: python 2.7.18 (final)  
 Caller: /user\_data/data/dirk/prj/unittest/caching/unittest/src/tests/\_init....py (48)  
 Start-Time: 2021-01-07 21:59:04,770  
 Finished-Time: 2021-01-07 21:59:04,772  
 Time-Consumption 0.002s

---

**Testsummary:**

**Info** Initialising property\_cache\_pickle (load\_all\_on\_init=True).  
**Success** Instance data after changing uid is correct (Content {'unicode': u'\_\_unicode\_\_', 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '3': 'three', '2': 2, '4': '4'}, 'str': '\_\_string\_\_', 'integer': 34} and Type is <type '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.2.14!

---

Testrun: python 2.7.18 (final)  
 Caller: /user\_data/data/dirk/prj/unittest/caching/unittest/src/tests/\_init....py (49)  
 Start-Time: 2021-01-07 21:59:04,772  
 Finished-Time: 2021-01-07 21:59:04,774  
 Time-Consumption 0.003s

---

**Testsummary:**

**Info** Initialising property\_cache\_pickle (load\_all\_on\_init=True).  
**Success** Instance data after changing uid is correct (Content {'unicode': u'\_\_unicode\_\_', 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '3': 'three', '2': 2, '4': '4'}, 'str': '\_\_string\_\_', 'integer': 34} and Type is <type '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.2.11!

---

Testrun: python 2.7.18 (final)  
 Caller: /user\_data/data/dirk/prj/unittest/caching/unittest/src/tests/\_init....py (46)  
 Start-Time: 2021-01-07 21:59:04,765  
 Finished-Time: 2021-01-07 21:59:04,767  
 Time-Consumption 0.002s

---

**Testsummary:**

---

**Info** Initialising property\_cache\_pickle (load\_all\_on\_init=True).  
**Success** Instance data after increasing data\_version is correct (Content {'unicode': u'\_\_unicode\_\_', 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '3': 'three', '2': 2, '4': '4'}, 'str': '\_\_string\_\_', 'integer': 34} and Type is <type '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.2.12!

---

Testrun: python 2.7.18 (final)  
 Caller: /user\_data/data/dirk/prj/unittest/caching/unittest/src/tests/\_init....py (47)  
 Start-Time: 2021-01-07 21:59:04,767  
 Finished-Time: 2021-01-07 21:59:04,770  
 Time-Consumption 0.003s

---

**Testsummary:**

---

**Info** Initialising property\_cache\_pickle (load\_all\_on\_init=True).  
**Success** Instance data after increasing data\_version is correct (Content {'unicode': u'\_\_unicode\_\_', 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '3': 'three', '2': 2, '4': '4'}, 'str': '\_\_string\_\_', 'integer': 34} and Type is <type '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.2.16!

---

Testrun: python 2.7.18 (final)  
 Caller: /user\_data/data/dirk/prj/unittest/caching/unittest/src/tests/\_init....py (51)  
 Start-Time: 2021-01-07 21:59:04,775  
 Finished-Time: 2021-01-07 21:59:04,776  
 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 is correct (Content {'\_\_property\_cache\_data\_version\_': u'no data version', '\_\_property\_cache\_data\_version\_': u'no second data version', '\_\_property\_cache\_uid\_': u'no uid', '\_\_property\_cache\_uid\_': u'no second uid'}) and Type is <type 'dict'>).  
**Success** Keyfilter returnvalue for 5 (<type 'int'>) is correct (Content 5 and Type is <type '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.2.8!

---

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (43)
Start-Time:	2021-01-07 21:59:04,755
Finished-Time:	2021-01-07 21:59:04,758
Time-Consumption	0.003s

---

**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 {'float': 3.14159, 'list': [1, 'two', '3', 4], 'str': 'string'} and Type is <type 'dict'>).

---

## 3.2 Summary for testrun with python 3.8.5 (final)

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

#### Testresult

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

---

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (31)
Start-Time:	2021-01-07 21:59:05,187
Finished-Time:	2021-01-07 21:59:05,188
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'>).

---

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

#### Testresult

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

---

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (32)
Start-Time:	2021-01-07 21:59:05,188
Finished-Time:	2021-01-07 21:59:05,189

Time-Consumption 0.001s

---

**Testsummary:**

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

---

### 3.2.3 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 B.1.9!

---

Testrun: python 3.8.5 (final)  
 Caller: /user\_data/data/dirk/prj/unittest/caching/unittest/src/tests/\_init\_.py (37)  
 Start-Time: 2021-01-07 21:59:05,201  
 Finished-Time: 2021-01-07 21:59:05,202  
 Time-Consumption 0.000s

---

**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'>).

---

### 3.2.4 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 B.1.1!

---

Testrun: python 3.8.5 (final)  
 Caller: /user\_data/data/dirk/prj/unittest/caching/unittest/src/tests/\_init\_.py (29)  
 Start-Time: 2021-01-07 21:59:05,184  
 Finished-Time: 2021-01-07 21:59:05,185  
 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'>).

---

### 3.2.5 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 B.1.7!



---

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (35)
Start-Time:	2021-01-07 21:59:05,197
Finished-Time:	2021-01-07 21:59:05,198
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'>).

---

### 3.2.6 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 B.1.8!

---

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (36)
Start-Time:	2021-01-07 21:59:05,199
Finished-Time:	2021-01-07 21:59:05,201
Time-Consumption	0.003s

---

**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'>).

---

### 3.2.7 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 B.1.5!

---

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (33)
Start-Time:	2021-01-07 21:59:05,189
Finished-Time:	2021-01-07 21:59:05,191
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'>).

---

### 3.2.8 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 B.1.6!

---

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (34)
Start-Time:	2021-01-07 21:59:05,192
Finished-Time:	2021-01-07 21:59:05,197
Time-Consumption	0.005s

---

**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'>).

---

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

#### Testresult

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

---

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (38)
Start-Time:	2021-01-07 21:59:05,202
Finished-Time:	2021-01-07 21:59:05,203
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'>).

---

### 3.2.10 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 B.1.2!

---

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (30)
Start-Time:	2021-01-07 21:59:05,185
Finished-Time:	2021-01-07 21:59:05,187
Time-Consumption	0.002s

---

**Testsummary:**

---

**Info** Initialising property\_cache.json (load\_all\_on\_init=False).  
**Info** Partially initialising cache object by requesting some information.  
**Info** Extracting storage object from property\_cache.json for comparison.  
**Success** Cache object is correct (Content {'integer': 17, 'str': 'string', 'unicode': 'unicode'} and Type is <class 'dict'>).

---

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

#### Testresult

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

---

Testrun: python 3.8.5 (final)  
 Caller: /user\_data/data/dirk/prj/unittest/caching/unittest/src/tests/\_init\_.py (44)  
 Start-Time: 2021-01-07 21:59:05,205  
 Finished-Time: 2021-01-07 21:59:05,207  
 Time-Consumption 0.002s

---

#### Testsummary:

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

---

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

#### Testresult

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

---

Testrun: python 3.8.5 (final)  
 Caller: /user\_data/data/dirk/prj/unittest/caching/unittest/src/tests/\_init\_.py (45)  
 Start-Time: 2021-01-07 21:59:05,207  
 Finished-Time: 2021-01-07 21:59:05,210  
 Time-Consumption 0.002s

---

#### Testsummary:

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

---

### 3.2.13 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 B.1.19!

---

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (50)
Start-Time:	2021-01-07 21:59:05,219
Finished-Time:	2021-01-07 21:59:05,219
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'>).

---

### 3.2.14 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 B.1.11!

---

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (42)
Start-Time:	2021-01-07 21:59:05,203
Finished-Time:	2021-01-07 21:59:05,204
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.
<b>Success</b>	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'>).

---

### 3.2.15 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 B.1.17!

---

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (48)
Start-Time:	2021-01-07 21:59:05,216
Finished-Time:	2021-01-07 21:59:05,217
Time-Consumption	0.001s

---

**Testsummary:**

---

<b>Info</b>	Initialising property_cache_pickle (load_all_on_init=True).
<b>Success</b>	Instance data after changing uid is correct (Content {'_string_': '_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'>).

---

### 3.2.16 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 B.1.18!

---

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init....py (49)
Start-Time:	2021-01-07 21:59:05,217
Finished-Time:	2021-01-07 21:59:05,219
Time-Consumption	0.002s

---

**Testsummary:**

---

<b>Info</b>	Initialising property_cache_pickle (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'>).

---

### 3.2.17 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 B.1.15!

---

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init....py (46)
Start-Time:	2021-01-07 21:59:05,210
Finished-Time:	2021-01-07 21:59:05,213
Time-Consumption	0.003s

---

**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'>).

---

### 3.2.18 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 B.1.16!

---

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init....py (47)
Start-Time:	2021-01-07 21:59:05,213
Finished-Time:	2021-01-07 21:59:05,215
Time-Consumption	0.003s

---

**Testsummary:**

---

<b>Info</b>	Initialising property_cache_pickle (load_all_on_init=True).
-------------	---

**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'>).

---

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

#### Testresult

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

---

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (51)
Start-Time:	2021-01-07 21:59:05,219
Finished-Time:	2021-01-07 21:59:05,220
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.
<b>Success</b>	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'>).
<b>Success</b>	Keyfilter returnvalue for 5 (<class 'int'>) is correct (Content 5 and Type is <class 'int'>).

---

### 3.2.20 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 B.1.12!

---

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/caching/unittest/src/tests/_init_.py (43)
Start-Time:	2021-01-07 21:59:05,204
Finished-Time:	2021-01-07 21:59:05,205
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 2.7.18 (final)

### A.1 Tests with status **Warning** (4)

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

##### Testresult

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

---

**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

↪ /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/load\_on\_init.json as cache  
 ↪ file.

JsonCache: Cache file does not exists (yet).

JsonCache: Loading all data from source - ['dict', 'unicode', 'str', 'integer', 'float',  
 ↪ 'list']

JsonCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/load\_on\_init.json)

---

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

---

JsonCache: Loading properties from cache

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/load\_on\_init.json)

Using storage object of cache class for comparison: {u'str': u'string',

↪ u'\_property\_cache\_uid': u'my\_unique\_id', u'float': 3.14159, u'list': [1, u'two', u'3',  
 ↪ 4], u'dict': {u'1': 1, u'3': u'3', u'2': u'two', u'4': 4}, u'unicode': u'unicode',  
 ↪ u'\_property\_cache\_data\_version\_': 1, u'integer': 17}

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

---

**Warning** Cache object is NOT correct. See detailed log for more information.

---

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

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

```
Type <type 'unicode'> is NOT <type 'str'> for result.str (u'string'). Check for known json
↳ storage type deviation <str> -> <unicode>.
```

```
Type <type 'unicode'> is NOT <type 'str'> for test_variable (u'two').
```

```
Content u'two' is incorrect for test_variable.
```

```
Type <type 'unicode'> is NOT <type 'str'> for test_variable (u'two').
```

```
Content u'two' is incorrect for test_variable.
```

```
Type <type 'unicode'> is NOT <type 'int'> for test_variable (u'two').
```

```
Type <type 'unicode'> is NOT <type 'str'> for result.list[1] (u'two'). Check for known json
↳ storage type deviation <str> -> <unicode>.
```

```
Type <type 'unicode'> is NOT <type 'str'> for test_variable (u'3').
```

```
Content u'3' is incorrect for test_variable.
```

```
Type <type 'unicode'> is NOT <type 'int'> for test_variable (u'3').
```

```
Type <type 'unicode'> is NOT <type 'str'> for result.list[2] (u'3'). Check for known json
↳ storage type deviation <str> -> <unicode>.
```

```
Type <type 'unicode'> is NOT <type 'str'> for result.dict.3 (u'3'). Check for known json
↳ storage type deviation <str> -> <unicode>.
```

```
Type <type 'unicode'> is NOT <type 'str'> for result.dict.2 (u'two'). Check for known json
↳ storage type deviation <str> -> <unicode>.
```

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

#### Testresult

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

---

**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

```
↳ /user_data/data/dirk/prj/unittest/caching/unittest/output_data/no_load_on_init.json as
↳ cache file.
```

```
JsonCache: Cache file does not exists (yet).
```

JsonCache: cache-file stored

```
↳ (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/no_load_on_init.json)
```

---

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

---



## Unittest for caching

```
JsonCache: Loading property for 'float' from source instance (3.14159)
```

```
JsonCache: cache-file stored
```

```
↪ (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/no_load_on_init.json)
```

```
JsonCache: Loading property for 'list' from source instance ([1, 'two', '3', 4])
```

```
JsonCache: cache-file stored
```

```
↪ (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/no_load_on_init.json)
```

```
JsonCache: Loading property for 'str' from source instance ('string')
```

```
JsonCache: cache-file stored
```

```
↪ (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/no_load_on_init.json)
```

---

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

---

```
JsonCache: Loading properties from cache
```

```
↪ (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/no_load_on_init.json)
```

```
Using storage object of cache class for comparison: {u'_property_cache_data_version': 1,  
↪ u'float': 3.14159, u'list': [1, u'two', u'3', 4], u'str': u'string',  
↪ u'_property_cache_uid_': u'my_unique_id'}
```

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

---

**Warning** Cache object is NOT correct. See detailed log for more information.

---

```
Result (Cache object): { u'float': 3.14159, u'list': [ 1, u'two', u'3', 4 ], u'str':  
↪ u'string' } (<type 'dict'>)
```

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

```
Type <type 'unicode'> is NOT <type 'str'> for test_variable (u'two').
```

```
Content u'two' is incorrect for test_variable.
```

```
Type <type 'unicode'> is NOT <type 'str'> for test_variable (u'two').
```

```
Content u'two' is incorrect for test_variable.
```

```
Type <type 'unicode'> is NOT <type 'int'> for test_variable (u'two').
```

```
Type <type 'unicode'> is NOT <type 'str'> for result.list[1] (u'two'). Check for known json  
↪ storage type deviation <str> -> <unicode>.
```

```
Type <type 'unicode'> is NOT <type 'str'> for test_variable (u'3').
```

```
Content u'3' is incorrect for test_variable.
```

```
Type <type 'unicode'> is NOT <type 'int'> for test_variable (u'3').
```

```
Type <type 'unicode'> is NOT <type 'str'> for result.list[2] (u'3'). Check for known json  
↪ storage type deviation <str> -> <unicode>.
```

```
Type <type 'unicode'> is NOT <type 'str'> for result.str (u'string'). Check for known json  
↪ storage type deviation <str> -> <unicode>.
```

### A.1.3 caching.property\_cache\_json: Test cached data (full init)

#### Testresult

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

---

**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 /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ /cache\_data\_test\_load\_on\_init.json as cache  
 ↵ file.

JsonCache: Cache file does not exists (yet).

JsonCache: Loading all data from source - ['dict', 'unicode', 'str', 'integer', 'float',  
 ↵ 'list']

JsonCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ cache\_data\_test\_load\_on\_init.json)

---

**Info** Collecting data from cache instance.

---

JsonCache: Key 'uncached' is not in cached\_keys. Uncached data will be returned.

JsonCache: Loading properties from cache (/user\_data/data/dirk/prj/unittest/caching/unittest/ ↵  
 ↵ output\_data/cache\_data\_test\_load\_on\_init.json)

JsonCache: Providing property for 'dict' from cache ({u'1': 1, u'3': u'3', u'2': u'two',  
 ↵ u'4': 4})

JsonCache: Providing property for 'unicode' from cache (u'unicode')

JsonCache: Providing property for 'str' from cache (u'string')

JsonCache: Providing property for 'integer' from cache (17)

JsonCache: Providing property for 'float' from cache (3.14159)

JsonCache: Providing property for 'list' from cache ([1, u'two', u'3', 4])

---

**Warning** Cached data is NOT correct. See detailed log for more information.

---

```

Result (Cached data): { 'uncached': 'uncached_data_of_class', 'unicode': u'unicode', 'float':
↳ 3.14159, 'list': [ 1, u'two', u'3', 4 ], 'dict': { u'1': 1, u'3': u'3', u'2': u'two',
↳ u'4': 4 }, 'str': u'string', 'integer': 17 } (<type 'dict'>)
Expectation (Cached data): result = { 'uncached': 'uncached_data_of_class', 'dict': { '1': 1,
↳ '3': '3', '2': 'two', '4': 4 }, 'unicode': u'unicode', 'str': 'string', 'integer': 17,
↳ 'float': 3.14159, 'list': [ 1, 'two', '3', 4 ] } (<type 'dict'>)
Type <type 'unicode'> is NOT <type 'str'> for test_variable (u'two').
Content u'two' is incorrect for test_variable.
Type <type 'unicode'> is NOT <type 'str'> for test_variable (u'two').
Content u'two' is incorrect for test_variable.
Type <type 'unicode'> is NOT <type 'int'> for test_variable (u'two').
Type <type 'unicode'> is NOT <type 'str'> for result.list[1] (u'two'). Check for known json
↳ storage type deviation <str> -> <unicode>.
Type <type 'unicode'> is NOT <type 'str'> for test_variable (u'3').
Content u'3' is incorrect for test_variable.
Type <type 'unicode'> is NOT <type 'int'> for test_variable (u'3').
Type <type 'unicode'> is NOT <type 'str'> for result.list[2] (u'3'). Check for known json
↳ storage type deviation <str> -> <unicode>.
Type <type 'unicode'> is NOT <type 'str'> for result.dict.3 (u'3'). Check for known json
↳ storage type deviation <str> -> <unicode>.
Type <type 'unicode'> is NOT <type 'str'> for result.dict.2 (u'two'). Check for known json
↳ storage type deviation <str> -> <unicode>.
Type <type 'unicode'> is NOT <type 'str'> for result.str (u'string'). Check for known json
↳ storage type deviation <str> -> <unicode>.

```

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

##### Testresult

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

---

**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 /user_data/data/dirk/prj/unittest/caching/unittest/output_data_
↳ /cache_data_test_no_load_on_init.json as cache
↳ file.

```

```

JsonCache: Cache file does not exists (yet).

```

```

JsonCache: Loading all data from source - ['dict', 'unicode', 'str', 'integer', 'float',
↳ 'list']

```

```

JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/_
↳ cache_data_test_no_load_on_init.json)

```

---

**Info** Collecting data from cache instance.

---

```

JsonCache: Key 'uncached' is not in cached_keys. Uncached data will be returned.
JsonCache: Loading properties from cache (/user_data/data/dirk/prj/unittest/caching/unittest/
↪ output_data/cache_data_test_no_load_on_init.json)
JsonCache: Providing property for 'dict' from cache ({u'1': 1, u'3': u'3', u'2': u'two',
↪ u'4': 4})
JsonCache: Providing property for 'unicode' from cache (u'unicode')
JsonCache: Providing property for 'str' from cache (u'string')
JsonCache: Providing property for 'integer' from cache (17)
JsonCache: Providing property for 'float' from cache (3.14159)
JsonCache: Providing property for 'list' from cache ([1, u'two', u'3', 4])

```

---

**Warning** Cached data is NOT correct. See detailed log for more information.

---

```

Result (Cached data): { 'uncached': 'uncached_data_of_class', 'unicode': u'unicode', 'float':
↪ 3.14159, 'list': [ 1, u'two', u'3', 4 ], 'dict': { u'1': 1, u'3': u'3', u'2': u'two',
↪ u'4': 4 }, 'str': u'string', 'integer': 17 } (<type 'dict'>)
Expectation (Cached data): result = { 'uncached': 'uncached_data_of_class', 'dict': { '1': 1,
↪ '3': '3', '2': 'two', '4': 4 }, 'unicode': u'unicode', 'str': 'string', 'integer': 17,
↪ 'float': 3.14159, 'list': [ 1, 'two', '3', 4 ] } (<type 'dict'>)
Type <type 'unicode'> is NOT <type 'str'> for test_variable (u'two').
Content u'two' is incorrect for test_variable.
Type <type 'unicode'> is NOT <type 'str'> for test_variable (u'two').
Content u'two' is incorrect for test_variable.
Type <type 'unicode'> is NOT <type 'int'> for test_variable (u'two').
Type <type 'unicode'> is NOT <type 'str'> for result.list[1] (u'two'). Check for known json
↪ storage type deviation <str> -> <unicode>.
Type <type 'unicode'> is NOT <type 'str'> for test_variable (u'3').
Content u'3' is incorrect for test_variable.
Type <type 'unicode'> is NOT <type 'int'> for test_variable (u'3').
Type <type 'unicode'> is NOT <type 'str'> for result.list[2] (u'3'). Check for known json
↪ storage type deviation <str> -> <unicode>.
Type <type 'unicode'> is NOT <type 'str'> for result.dict.3 (u'3'). Check for known json
↪ storage type deviation <str> -> <unicode>.
Type <type 'unicode'> is NOT <type 'str'> for result.dict.2 (u'two'). Check for known json
↪ storage type deviation <str> -> <unicode>.
Type <type 'unicode'> is NOT <type 'str'> for result.str (u'string'). Check for known json
↪ storage type deviation <str> -> <unicode>.

```

## A.2 Tests with status Info (16)

### A.2.1 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 /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ /data\_version\_test\_load\_on\_init.json as cache  
 ↵ file.

JsonCache: Cache file does not exists (yet).

JsonCache: Loading all data from source - ['dict', 'unicode', 'str', 'integer', 'float',  
 ↵ 'list']

JsonCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ data\_version\_test\_load\_on\_init.json)

---

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

---

JsonCache: Loading properties from cache (/user\_data/data/dirk/prj/unittest/caching/unittest/ ↵  
 ↵ output\_data/data\_version\_test\_load\_on\_init.json)

JsonCache: Data version increased, ignoring previous cache data

JsonCache: Loading all data from source - ['dict', 'unicode', 'str', 'integer', 'float',  
 ↵ 'list']

JsonCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ data\_version\_test\_load\_on\_init.json)

JsonCache: Providing property for 'dict' from cache ({'1': '1', '3': 'three', '2': 2, '4':  
 ↵ '4'})

JsonCache: Providing property for 'unicode' from cache (u'\_\_unicode\_\_')

JsonCache: Providing property for 'str' from cache ('\_\_string\_\_')

JsonCache: Providing property for 'integer' from cache (34)

JsonCache: Providing property for 'float' from cache (2.71828)

JsonCache: Providing property for 'list' from cache (['one', 2, 3, '4'])

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

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

## A.2.2 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).

---

## Unittest for caching

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

```
Initialising cached class with /user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ /data_version_test_no_load_on_init.json as cache
↳ file.
```

```
JsonCache: Cache file does not exists (yet).
```

```
JsonCache: Loading all data from source - ['dict', 'unicode', 'str', 'integer', 'float',
↳ 'list']
```

```
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.json)
```

---

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

---

```
JsonCache: Loading properties from cache (/user_data/data/dirk/prj/unittest/caching/unittest/
↳ output_data/data_version_test_no_load_on_init.json)
```

```
JsonCache: Data version increased, ignoring previous cache data
```

```
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.json)
```

```
JsonCache: Loading property for 'dict' from source instance ({'1': '1', '3': 'three', '2': 2,
↳ '4': '4'})
```

```
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.json)
```

```
JsonCache: Loading property for 'unicode' from source instance (u'__unicode__')
```

```
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.json)
```

```
JsonCache: Loading property for 'str' from source instance ('__string__')
```

```
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.json)
```

```
JsonCache: Loading property for 'integer' from source instance (34)
```

```
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.json)
```

```
JsonCache: Loading property for 'float' from source instance (2.71828)
```

```
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.json)
```

```
JsonCache: Loading property for 'list' from source instance (['one', 2, 3, '4'])
```

```
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.json)
```

```
Result (Instance data after increasing data_version): { 'unicode': u'__unicode__', 'float':
↳ 2.71828, 'list': [ 'one', 2, 3, '4' ], 'dict': { '1': '1', '3': 'three', '2': 2, '4': '4'
↳ }, 'str': '__string__', 'integer': 34 } (<type 'dict'>)
```

```
Expectation (Instance data after increasing data_version): result = { 'dict': { '1': '1',
↳ '3': 'three', '2': 2, '4': '4' }, 'unicode': u'__unicode__', 'str': '__string__',
↳ 'integer': 34, 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ] } (<type 'dict'>)
```

### A.2.3 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

↪ /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/uid\_test\_load\_on\_init.json  
 ↪ as cache file.

JsonCache: Cache file does not exists (yet).

JsonCache: Loading all data from source - ['dict', 'unicode', 'str', 'integer', 'float',  
 ↪ 'list']

JsonCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/  
 ↪ uid\_test\_load\_on\_init.json)

---

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

---

JsonCache: Loading properties from cache (/user\_data/data/dirk/prj/unittest/caching/unittest/  
 ↪ output\_data/uid\_test\_load\_on\_init.json)

JsonCache: Source uid changed, ignoring previous cache data

JsonCache: Loading all data from source - ['dict', 'unicode', 'str', 'integer', 'float',  
 ↪ 'list']

JsonCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/  
 ↪ uid\_test\_load\_on\_init.json)

JsonCache: Providing property for 'dict' from cache ({'1': '1', '3': 'three', '2': 2, '4':  
 ↪ '4'})

JsonCache: Providing property for 'unicode' from cache (u'\_\_unicode\_\_')

JsonCache: Providing property for 'str' from cache ('\_\_string\_\_')

JsonCache: Providing property for 'integer' from cache (34)

JsonCache: Providing property for 'float' from cache (2.71828)

JsonCache: Providing property for 'list' from cache (['one', 2, 3, '4'])

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

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

#### A.2.4 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 /user_data/data/dirk/prj/unittest/caching/unittest/output_data_
↪ /uid_test_no_load_on_init.json as cache
↪ file.
```

```
JsonCache: Cache file does not exists (yet).
```

```
JsonCache: Loading all data from source - ['dict', 'unicode', 'str', 'integer', 'float',
↪ 'list']
```

```
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/_
↪ uid_test_no_load_on_init.json)
```

---

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

---



```

JsonCache: Loading properties from cache (/user_data/data/dirk/prj/unittest/caching/unittest/
↳ output_data/uid_test_no_load_on_init.json)
JsonCache: Source uid changed, ignoring previous cache data
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.json)
JsonCache: Loading property for 'dict' from source instance ({'1': '1', '3': 'three', '2': 2,
↳ '4': '4'})
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.json)
JsonCache: Loading property for 'unicode' from source instance (u'__unicode__')
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.json)
JsonCache: Loading property for 'str' from source instance ('__string__')
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.json)
JsonCache: Loading property for 'integer' from source instance (34)
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.json)
JsonCache: Loading property for 'float' from source instance (2.71828)
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.json)
JsonCache: Loading property for 'list' from source instance (['one', 2, 3, '4'])
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.json)
Result (Instance data after changing uid): { 'unicode': u'__unicode__', 'float': 2.71828,
↳ 'list': [ 'one', 2, 3, '4' ], 'dict': { '1': '1', '3': 'three', '2': 2, '4': '4' },
↳ 'str': '__string__', 'integer': 34 } (<type 'dict'>)
Expectation (Instance data after changing uid): result = { 'dict': { '1': '1', '3': 'three',
↳ '2': 2, '4': '4' }, 'unicode': u'__unicode__', 'str': '__string__', 'integer': 34,
↳ 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ] } (<type 'dict'>)

```

### A.2.5 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 <type 'bool'>).

---

```

JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ save_callback_load_on_init.json)
Result (Save callback execution variable): True (<type 'bool'>)
Expectation (Save callback execution variable): result = True (<type 'bool'>)

```

## A.2.6 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

↪ /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/internal\_keys\_test.json as  
 ↪ cache file.

JsonCache: Cache file does not exists (yet).

JsonCache: Loading all data from source - ['\_\_property\_cache\_data\_version',  
 ↪ '\_property\_cache\_data\_version', '\_\_property\_cache\_uid', '\_property\_cache\_uid']

JsonCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/internal\_keys\_test.json)

---

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

---

JsonCache: Loading properties from cache

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/internal\_keys\_test.json)

Using storage object of cache class for comparison: {u'\_\_property\_cache\_data\_version': u'no

↪ data version', u'\_\_\_property\_cache\_data\_version': u'no second data version',  
 ↪ u'\_\_property\_cache\_uid': u'no uid', u'\_property\_cache\_uid': u'my\_unique\_id',  
 ↪ u'\_property\_cache\_data\_version': 1, u'\_\_\_property\_cache\_uid': u'no second uid'}

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

---

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

---

Result (Cache): { u'\_\_property\_cache\_data\_version': u'no data version',

↪ u'\_\_\_property\_cache\_data\_version': u'no second data version', u'\_\_property\_cache\_uid':  
 ↪ u'no uid', u'\_\_\_property\_cache\_uid': u'no second uid' } (<type 'dict'>)

Expectation (Cache): result = { '\_\_property\_cache\_data\_version': u'no data version',

↪ '\_\_\_property\_cache\_data\_version': u'no second data version', '\_\_property\_cache\_uid':  
 ↪ u'no uid', '\_\_\_property\_cache\_uid': u'no second uid' } (<type 'dict'>)

---

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

---

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

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

### A.2.7 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

↪ /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/load\_on\_init.pkl as cache  
 ↪ file.

PickCache: Cache file does not exists (yet).

PickCache: Loading all data from source - ['dict', 'unicode', 'str', 'integer', 'float',  
 ↪ 'list']

PickCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/load\_on\_init.pkl)

---

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

---

PickCache: Loading properties from cache

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/load\_on\_init.pkl)

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

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

---

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

---

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

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

---

### A.2.8 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

↪ /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/no\_load\_on\_init.pkl as  
 ↪ cache file.

PickCache: Cache file does not exists (yet).

PickCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/no\_load\_on\_init.pkl)

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

PickCache: Loading property for 'float' from source instance (3.14159)

PickCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/no\_load\_on\_init.pkl)

PickCache: Loading property for 'list' from source instance ([1, 'two', '3', 4])

PickCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/no\_load\_on\_init.pkl)

PickCache: Loading property for 'str' from source instance ('string')

PickCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/no\_load\_on\_init.pkl)

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

PickCache: Loading properties from cache

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/no\_load\_on\_init.pkl)

Using storage object of cache class for comparison: {'\_property\_cache\_data\_version': 1,

↪ 'float': 3.14159, 'list': [1, 'two', '3', 4], 'str': 'string', '\_property\_cache\_uid':  
 ↪ 'my\_unique\_id'}

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

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

Result (Cache object): { 'float': 3.14159, 'list': [ 1, 'two', '3', 4 ], 'str': 'string' }

↪ (<type 'dict'>)

Expectation (Cache object): result = { 'str': 'string', 'float': 3.14159, 'list': [ 1, 'two',

↪ '3', 4 ] } (<type 'dict'>)

### A.2.9 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 /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ /cache\_data\_test\_load\_on\_init.pkl as cache  
 ↵ file.

PickCache: Cache file does not exists (yet).

PickCache: Loading all data from source - ['dict', 'unicode', 'str', 'integer', 'float',  
 ↵ 'list']

PickCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ cache\_data\_test\_load\_on\_init.pkl)

---

**Info** Collecting data from cache instance.

---

PickCache: Key 'uncached' is not in cached\_keys. Uncached data will be returned.

PickCache: Loading properties from cache (/user\_data/data/dirk/prj/unittest/caching/unittest/ ↵  
 ↵ output\_data/cache\_data\_test\_load\_on\_init.pkl)

PickCache: Providing property for 'dict' from cache ({'1': 1, '3': '3', '2': 'two', '4': 4})

PickCache: Providing property for 'unicode' from cache (u'unicode')

PickCache: Providing property for 'str' from cache ('string')

PickCache: Providing property for 'integer' from cache (17)

PickCache: Providing property for 'float' from cache (3.14159)

PickCache: Providing property for 'list' from cache ([1, 'two', '3', 4])

---

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

---

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

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

### A.2.10 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 /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ /cache\_data\_test\_no\_load\_on\_init.pkl as cache  
 ↵ file.

PickCache: Cache file does not exists (yet).

PickCache: Loading all data from source - ['dict', 'unicode', 'str', 'integer', 'float',  
 ↵ 'list']

PickCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ cache\_data\_test\_no\_load\_on\_init.pkl)

---

**Info** Collecting data from cache instance.

---

PickCache: Key 'uncached' is not in cached\_keys. Uncached data will be returned.

PickCache: Loading properties from cache (/user\_data/data/dirk/prj/unittest/caching/unittest/ ↵  
 ↵ output\_data/cache\_data\_test\_no\_load\_on\_init.pkl)

PickCache: Providing property for 'dict' from cache ({'1': 1, '3': '3', '2': 'two', '4': 4})

PickCache: Providing property for 'unicode' from cache (u'unicode')

PickCache: Providing property for 'str' from cache ('string')

PickCache: Providing property for 'integer' from cache (17)

PickCache: Providing property for 'float' from cache (3.14159)

PickCache: Providing property for 'list' from cache ([1, 'two', '3', 4])

---

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

---

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

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

### A.2.11 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 /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ /data\_version\_test\_load\_on\_init.pkl as cache  
 ↵ file.

PickCache: Cache file does not exists (yet).

PickCache: Loading all data from source - ['dict', 'unicode', 'str', 'integer', 'float',  
 ↵ 'list']

PickCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ data\_version\_test\_load\_on\_init.pkl)

---

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

---

PickCache: Loading properties from cache (/user\_data/data/dirk/prj/unittest/caching/unittest/ ↵  
 ↵ output\_data/data\_version\_test\_load\_on\_init.pkl)

PickCache: Data version increased, ignoring previous cache data

PickCache: Loading all data from source - ['dict', 'unicode', 'str', 'integer', 'float',  
 ↵ 'list']

PickCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ data\_version\_test\_load\_on\_init.pkl)

PickCache: Providing property for 'dict' from cache ({'1': '1', '3': 'three', '2': 2, '4':  
 ↵ '4'})

PickCache: Providing property for 'unicode' from cache (u'\_\_unicode\_\_')

PickCache: Providing property for 'str' from cache ('\_\_string\_\_')

PickCache: Providing property for 'integer' from cache (34)

PickCache: Providing property for 'float' from cache (2.71828)

PickCache: Providing property for 'list' from cache (['one', 2, 3, '4'])

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

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

### A.2.12 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).

---

## Unittest for caching

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

```
Initialising cached class with /user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ /data_version_test_no_load_on_init.pkl as cache
↳ file.
```

```
PickCache: Cache file does not exists (yet).
```

```
PickCache: Loading all data from source - ['dict', 'unicode', 'str', 'integer', 'float',
↳ 'list']
```

```
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.pkl)
```

---

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

---

```
PickCache: Loading properties from cache (/user_data/data/dirk/prj/unittest/caching/unittest/
↳ output_data/data_version_test_no_load_on_init.pkl)
```

```
PickCache: Data version increased, ignoring previous cache data
```

```
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.pkl)
```

```
PickCache: Loading property for 'dict' from source instance ({'1': '1', '3': 'three', '2': 2,
↳ '4': '4'})
```

```
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.pkl)
```

```
PickCache: Loading property for 'unicode' from source instance (u'__unicode__')
```

```
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.pkl)
```

```
PickCache: Loading property for 'str' from source instance ('__string__')
```

```
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.pkl)
```

```
PickCache: Loading property for 'integer' from source instance (34)
```

```
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.pkl)
```

```
PickCache: Loading property for 'float' from source instance (2.71828)
```

```
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.pkl)
```

```
PickCache: Loading property for 'list' from source instance (['one', 2, 3, '4'])
```

```
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.pkl)
```

```
Result (Instance data after increasing data_version): { 'unicode': u'__unicode__', 'float':
↳ 2.71828, 'list': [ 'one', 2, 3, '4' ], 'dict': { '1': '1', '3': 'three', '2': 2, '4': '4'
↳ }, 'str': '__string__', 'integer': 34 } (<type 'dict'>)
```

```
Expectation (Instance data after increasing data_version): result = { 'dict': { '1': '1',
↳ '3': 'three', '2': 2, '4': '4' }, 'unicode': u'__unicode__', 'str': '__string__',
↳ 'integer': 34, 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ] } (<type 'dict'>)
```



### A.2.13 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

↪ /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/uid\_test\_load\_on\_init.pkl  
 ↪ as cache file.

PickCache: Cache file does not exists (yet).

PickCache: Loading all data from source - ['dict', 'unicode', 'str', 'integer', 'float',  
 ↪ 'list']

PickCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/uid\_test\_load\_on\_init.pkl)

---

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

---

PickCache: Loading properties from cache

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/uid\_test\_load\_on\_init.pkl)

PickCache: Source uid changed, ignoring previous cache data

PickCache: Loading all data from source - ['dict', 'unicode', 'str', 'integer', 'float',  
 ↪ 'list']

PickCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/uid\_test\_load\_on\_init.pkl)

PickCache: Providing property for 'dict' from cache ({'1': '1', '3': 'three', '2': 2, '4':  
 ↪ '4'})

PickCache: Providing property for 'unicode' from cache (u'\_\_unicode\_\_')

PickCache: Providing property for 'str' from cache ('\_\_string\_\_')

PickCache: Providing property for 'integer' from cache (34)

PickCache: Providing property for 'float' from cache (2.71828)

PickCache: Providing property for 'list' from cache (['one', 2, 3, '4'])

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

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

**A.2.14** `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 `/user_data/data/dirk/prj/unittest/caching/unittest/output_data/uid_test_no_load_on_init.pkl` as cache  
↪ file.

PickCache: Cache file does not exists (yet).

PickCache: Loading all data from source - ['dict', 'unicode', 'str', 'integer', 'float', 'list']

PickCache: cache-file stored (`/user_data/data/dirk/prj/unittest/caching/unittest/output_data/uid_test_no_load_on_init.pkl`)

---

**Success** Instance data after changing uid is correct (Content `{'unicode': u'__unicode__', 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '3': 'three', '2': 2, '4': '4'}, 'str': '.__string__', 'integer': 34}` and Type is `<type 'dict'>`).

---

```

PickCache: Loading properties from cache (/user_data/data/dirk/prj/unittest/caching/unittest/
↳ output_data/uid_test_no_load_on_init.pkl)
PickCache: Source uid changed, ignoring previous cache data
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.pkl)
PickCache: Loading property for 'dict' from source instance ({'1': '1', '3': 'three', '2': 2,
↳ '4': '4'})
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.pkl)
PickCache: Loading property for 'unicode' from source instance (u'__unicode__')
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.pkl)
PickCache: Loading property for 'str' from source instance ('__string__')
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.pkl)
PickCache: Loading property for 'integer' from source instance (34)
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.pkl)
PickCache: Loading property for 'float' from source instance (2.71828)
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.pkl)
PickCache: Loading property for 'list' from source instance (['one', 2, 3, '4'])
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.pkl)
Result (Instance data after changing uid): { 'unicode': u'__unicode__', 'float': 2.71828,
↳ 'list': [ 'one', 2, 3, '4' ], 'dict': { '1': '1', '3': 'three', '2': 2, '4': '4' },
↳ 'str': '__string__', 'integer': 34 } (<type 'dict'>)
Expectation (Instance data after changing uid): result = { 'dict': { '1': '1', '3': 'three',
↳ '2': 2, '4': '4' }, 'unicode': u'__unicode__', 'str': '__string__', 'integer': 34,
↳ 'float': 2.71828, 'list': [ 'one', 2, 3, '4' ] } (<type 'dict'>)

```

### A.2.15 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 <type 'bool'>).

---

```

PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ save_callback_load_on_init.json)
Result (Save callback execution variable): True (<type 'bool'>)
Expectation (Save callback execution variable): result = True (<type 'bool'>)

```

### A.2.16 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

↪ /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/internal\_keys\_test.pkl as  
 ↪ cache file.

PickCache: Cache file does not exists (yet).

PickCache: Loading all data from source - ['\_\_property\_cache\_data\_version',  
 ↪ '\_property\_cache\_data\_version\_', '\_\_property\_cache\_uid\_', '\_property\_cache\_uid\_']

PickCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/internal\_keys\_test.pkl)

---

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

---

PickCache: Loading properties from cache

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/internal\_keys\_test.pkl)

Using storage object of cache class for comparison: {'\_\_property\_cache\_data\_version\_': u'no

↪ data version', '\_\_\_property\_cache\_data\_version\_': u'no second data version',  
 ↪ '\_\_property\_cache\_uid\_': u'no uid', '\_property\_cache\_uid\_': 'my-unique-id',  
 ↪ '\_property\_cache\_data\_version\_': 1, '\_\_\_property\_cache\_uid\_': u'no second uid'}

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

---

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

---

Result (Cache): { '\_\_property\_cache\_data\_version\_': u'no data version',

↪ '\_\_\_property\_cache\_data\_version\_': u'no second data version', '\_property\_cache\_uid\_':  
 ↪ u'no uid', '\_\_\_property\_cache\_uid\_': u'no second uid' } (<type 'dict'>)

Expectation (Cache): result = { '\_\_property\_cache\_data\_version\_': u'no data version',

↪ '\_\_\_property\_cache\_data\_version\_': u'no second data version', '\_property\_cache\_uid\_':  
 ↪ u'no uid', '\_\_\_property\_cache\_uid\_': u'no second uid' } (<type 'dict'>)

---

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

---

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

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

## B Trace for testrun with python 3.8.5 (final)

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

#### B.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

↪ /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/load\_on\_init.json as cache  
↪ file.

JsonCache: Cache file does not exists (yet).

JsonCache: Loading all data from source - ['str', 'unicode', 'integer', 'float', 'list',  
↪ 'dict']

JsonCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/load\_on\_init.json)

---

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

---

JsonCache: Loading properties from cache

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/load\_on\_init.json)

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'>)

---

#### B.1.2 caching.property\_cache.json: Test partially 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

↪ /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/no\_load\_on\_init.json as  
 ↪ cache file.

JsonCache: Cache file does not exists (yet).

JsonCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/no\_load\_on\_init.json)

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

JsonCache: Loading property for 'str' from source instance ('string')

JsonCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/no\_load\_on\_init.json)

JsonCache: Loading property for 'integer' from source instance (17)

JsonCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/no\_load\_on\_init.json)

JsonCache: Loading property for 'unicode' from source instance ('unicode')

JsonCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/no\_load\_on\_init.json)

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

JsonCache: Loading properties from cache

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/no\_load\_on\_init.json)

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'>)

### B.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 /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ /cache\_data\_test\_load\_on\_init.json as cache  
 ↵ file.

JsonCache: Cache file does not exists (yet).

JsonCache: Loading all data from source - ['str', 'unicode', 'integer', 'float', 'list',  
 ↵ 'dict']

JsonCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ cache\_data\_test\_load\_on\_init.json)

---

**Info** Collecting data from cache instance.

---

JsonCache: Loading properties from cache (/user\_data/data/dirk/prj/unittest/caching/unittest/ ↵  
 ↵ output\_data/cache\_data\_test\_load\_on\_init.json)

JsonCache: Providing property for 'str' from cache ('string')

JsonCache: Providing property for 'unicode' from cache ('unicode')

JsonCache: Providing property for 'integer' from cache (17)

JsonCache: Providing property for 'float' from cache (3.14159)

JsonCache: Providing property for 'list' from cache ([1, 'two', '3', 4])

JsonCache: Providing property for 'dict' from cache ({'1': 1, '2': 'two', '3': '3', '4': 4})

JsonCache: Key 'uncached' is not in cached\_keys. Uncached data will be returned.

---

**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'>)

#### B.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 /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ /cache\_data\_test\_no\_load\_on\_init.json as cache  
 ↵ file.

JsonCache: Cache file does not exists (yet).

JsonCache: Loading all data from source - ['str', 'unicode', 'integer', 'float', 'list',  
 ↵ 'dict']

JsonCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ cache\_data\_test\_no\_load\_on\_init.json)

---

**Info** Collecting data from cache instance.

---

JsonCache: Loading properties from cache (/user\_data/data/dirk/prj/unittest/caching/unittest/ ↵  
 ↵ output\_data/cache\_data\_test\_no\_load\_on\_init.json)

JsonCache: Providing property for 'str' from cache ('string')

JsonCache: Providing property for 'unicode' from cache ('unicode')

JsonCache: Providing property for 'integer' from cache (17)

JsonCache: Providing property for 'float' from cache (3.14159)

JsonCache: Providing property for 'list' from cache ([1, 'two', '3', 4])

JsonCache: Providing property for 'dict' from cache ({'1': 1, '2': 'two', '3': '3', '4': 4})

JsonCache: Key 'uncached' is not in cached\_keys. Uncached data will be returned.

---

**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'>)

### B.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 /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ /data\_version\_test\_load\_on\_init.json as cache  
 ↵ file.

JsonCache: Cache file does not exists (yet).

JsonCache: Loading all data from source - ['str', 'unicode', 'integer', 'float', 'list',  
 ↵ 'dict']

JsonCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ data\_version\_test\_load\_on\_init.json)

---

**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'>).

---

JsonCache: Loading properties from cache (/user\_data/data/dirk/prj/unittest/caching/unittest/ ↵  
 ↵ output\_data/data\_version\_test\_load\_on\_init.json)

JsonCache: Data version increased, ignoring previous cache data

JsonCache: Loading all data from source - ['str', 'unicode', 'integer', 'float', 'list',  
 ↵ 'dict']

JsonCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ data\_version\_test\_load\_on\_init.json)

JsonCache: Providing property for 'str' from cache ('\_\_string\_\_')

JsonCache: Providing property for 'unicode' from cache ('\_\_unicode\_\_')

JsonCache: Providing property for 'integer' from cache (34)

JsonCache: Providing property for 'float' from cache (2.71828)

JsonCache: Providing property for 'list' from cache (['one', 2, 3, '4'])

JsonCache: Providing property for 'dict' from cache ({'1': '1', '2': 2, '3': 'three', '4':  
 ↵ '4'})

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'>)

### B.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).

---

## Unittest for caching

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

```
Initialising cached class with /user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ /data_version_test_no_load_on_init.json as cache
↳ file.
```

```
JsonCache: Cache file does not exists (yet).
```

```
JsonCache: Loading all data from source - ['str', 'unicode', 'integer', 'float', 'list',
↳ 'dict']
```

```
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.json)
```

---

**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'>).

---

```
JsonCache: Loading properties from cache (/user_data/data/dirk/prj/unittest/caching/unittest/
↳ output_data/data_version_test_no_load_on_init.json)
```

```
JsonCache: Data version increased, ignoring previous cache data
```

```
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.json)
```

```
JsonCache: Loading property for 'str' from source instance ('__string__')
```

```
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.json)
```

```
JsonCache: Loading property for 'unicode' from source instance ('__unicode__')
```

```
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.json)
```

```
JsonCache: Loading property for 'integer' from source instance (34)
```

```
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.json)
```

```
JsonCache: Loading property for 'float' from source instance (2.71828)
```

```
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.json)
```

```
JsonCache: Loading property for 'list' from source instance (['one', 2, 3, '4'])
```

```
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.json)
```

```
JsonCache: Loading property for 'dict' from source instance ({'1': '1', '2': 2, '3': 'three',
↳ '4': '4'})
```

```
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.json)
```

```
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'>)
```

### B.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

↪ /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/uid\_test\_load\_on\_init.json  
 ↪ as cache file.

JsonCache: Cache file does not exists (yet).

JsonCache: Loading all data from source - ['str', 'unicode', 'integer', 'float', 'list',  
 ↪ 'dict']

JsonCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/  
 ↪ uid\_test\_load\_on\_init.json)

---

**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'>).

---

JsonCache: Loading properties from cache (/user\_data/data/dirk/prj/unittest/caching/unittest/  
 ↪ output\_data/uid\_test\_load\_on\_init.json)

JsonCache: Source uid changed, ignoring previous cache data

JsonCache: Loading all data from source - ['str', 'unicode', 'integer', 'float', 'list',  
 ↪ 'dict']

JsonCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/  
 ↪ uid\_test\_load\_on\_init.json)

JsonCache: Providing property for 'str' from cache ('\_\_string\_\_')

JsonCache: Providing property for 'unicode' from cache ('\_\_unicode\_\_')

JsonCache: Providing property for 'integer' from cache (34)

JsonCache: Providing property for 'float' from cache (2.71828)

JsonCache: Providing property for 'list' from cache (['one', 2, 3, '4'])

JsonCache: Providing property for 'dict' from cache ({'1': '1', '2': 2, '3': 'three', '4':  
 ↪ '4'})

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'>)

### B.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 /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/uid\_test\_no\_load\_on\_init.json as cache  
 ↳ file.

JsonCache: Cache file does not exists (yet).

JsonCache: Loading all data from source - ['str', 'unicode', 'integer', 'float', 'list', 'dict']

JsonCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/uid\_test\_no\_load\_on\_init.json)

---

**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'>).

---

```

JsonCache: Loading properties from cache (/user_data/data/dirk/prj/unittest/caching/unittest/
↳ output_data/uid_test_no_load_on_init.json)
JsonCache: Source uid changed, ignoring previous cache data
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.json)
JsonCache: Loading property for 'str' from source instance ('__string__')
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.json)
JsonCache: Loading property for 'unicode' from source instance ('__unicode__')
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.json)
JsonCache: Loading property for 'integer' from source instance (34)
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.json)
JsonCache: Loading property for 'float' from source instance (2.71828)
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.json)
JsonCache: Loading property for 'list' from source instance (['one', 2, 3, '4'])
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.json)
JsonCache: Loading property for 'dict' from source instance ({'1': '1', '2': 2, '3': 'three',
↳ '4': '4'})
JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.json)
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'>)

```

### B.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'>).

---

```

JsonCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ save_callback_load_on_init.json)
Result (Save callback execution variable): True (<class 'bool'>)
Expectation (Save callback execution variable): result = True (<class 'bool'>)

```

### B.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

↪ /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/internal\_keys\_test.json as  
 ↪ cache file.

JsonCache: Cache file does not exists (yet).

JsonCache: Loading all data from source - ['\_property\_cache\_uid\_', '\_\_property\_cache\_uid\_',  
 ↪ '\_property\_cache\_data\_version\_', '\_\_property\_cache\_data\_version\_']

JsonCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/internal\_keys\_test.json)

---

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

---

JsonCache: Loading properties from cache

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/internal\_keys\_test.json)

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'>)

### B.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

↪ /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/load\_on\_init.pkl as cache  
 ↪ file.

PickCache: Cache file does not exists (yet).

PickCache: Loading all data from source - ['str', 'unicode', 'integer', 'float', 'list',  
 ↪ 'dict']

PickCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/load\_on\_init.pkl)

---

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

---

PickCache: Loading properties from cache

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/load\_on\_init.pkl)

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'>)

---

### B.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

↪ /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/no\_load\_on\_init.pkl as  
 ↪ cache file.

PickCache: Cache file does not exists (yet).

PickCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/no\_load\_on\_init.pkl)

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

PickCache: Loading property for 'str' from source instance ('string')

PickCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/no\_load\_on\_init.pkl)

PickCache: Loading property for 'integer' from source instance (17)

PickCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/no\_load\_on\_init.pkl)

PickCache: Loading property for 'unicode' from source instance ('unicode')

PickCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/no\_load\_on\_init.pkl)

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

PickCache: Loading properties from cache

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/no\_load\_on\_init.pkl)

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'>)

### B.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 /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ /cache\_data\_test\_load\_on\_init.pkl as cache  
 ↵ file.

PickCache: Cache file does not exists (yet).

PickCache: Loading all data from source - ['str', 'unicode', 'integer', 'float', 'list', ↵  
 ↵ 'dict']

PickCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ cache\_data\_test\_load\_on\_init.pkl)

---

**Info** Collecting data from cache instance.

---

PickCache: Loading properties from cache (/user\_data/data/dirk/prj/unittest/caching/unittest/ ↵  
 ↵ output\_data/cache\_data\_test\_load\_on\_init.pkl)

PickCache: Providing property for 'str' from cache ('string')

PickCache: Providing property for 'unicode' from cache ('unicode')

PickCache: Providing property for 'integer' from cache (17)

PickCache: Providing property for 'float' from cache (3.14159)

PickCache: Providing property for 'list' from cache ([1, 'two', '3', 4])

PickCache: Providing property for 'dict' from cache ({'1': 1, '2': 'two', '3': '3', '4': 4})

PickCache: Key 'uncached' is not in cached\_keys. Uncached data will be returned.

---

**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'>)

#### B.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 /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ /cache\_data\_test\_no\_load\_on\_init.pkl as cache  
 ↵ file.

PickCache: Cache file does not exists (yet).

PickCache: Loading all data from source - ['str', 'unicode', 'integer', 'float', 'list',  
 ↵ 'dict']

PickCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ cache\_data\_test\_no\_load\_on\_init.pkl)

---

**Info** Collecting data from cache instance.

---

PickCache: Loading properties from cache (/user\_data/data/dirk/prj/unittest/caching/unittest/ ↵  
 ↵ output\_data/cache\_data\_test\_no\_load\_on\_init.pkl)

PickCache: Providing property for 'str' from cache ('string')

PickCache: Providing property for 'unicode' from cache ('unicode')

PickCache: Providing property for 'integer' from cache (17)

PickCache: Providing property for 'float' from cache (3.14159)

PickCache: Providing property for 'list' from cache ([1, 'two', '3', 4])

PickCache: Providing property for 'dict' from cache ({'1': 1, '2': 'two', '3': '3', '4': 4})

PickCache: Key 'uncached' is not in cached\_keys. Uncached data will be returned.

---

**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'>)

### B.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 /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ /data\_version\_test\_load\_on\_init.pkl as cache  
 ↵ file.

PickCache: Cache file does not exists (yet).

PickCache: Loading all data from source - ['str', 'unicode', 'integer', 'float', 'list',  
 ↵ 'dict']

PickCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ data\_version\_test\_load\_on\_init.pkl)

---

**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'>).

---

PickCache: Loading properties from cache (/user\_data/data/dirk/prj/unittest/caching/unittest/ ↵  
 ↵ output\_data/data\_version\_test\_load\_on\_init.pkl)

PickCache: Data version increased, ignoring previous cache data

PickCache: Loading all data from source - ['str', 'unicode', 'integer', 'float', 'list',  
 ↵ 'dict']

PickCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ ↵  
 ↵ data\_version\_test\_load\_on\_init.pkl)

PickCache: Providing property for 'str' from cache ('\_\_string\_\_')

PickCache: Providing property for 'unicode' from cache ('\_\_unicode\_\_')

PickCache: Providing property for 'integer' from cache (34)

PickCache: Providing property for 'float' from cache (2.71828)

PickCache: Providing property for 'list' from cache (['one', 2, 3, '4'])

PickCache: Providing property for 'dict' from cache ({'1': '1', '2': 2, '3': 'three', '4':  
 ↵ '4'})

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'>)

#### B.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).

---

## Unittest for caching

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

```
Initialising cached class with /user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ /data_version_test_no_load_on_init.pkl as cache
↳ file.
```

```
PickCache: Cache file does not exists (yet).
```

```
PickCache: Loading all data from source - ['str', 'unicode', 'integer', 'float', 'list',
↳ 'dict']
```

```
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.pkl)
```

---

**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'>).

---

```
PickCache: Loading properties from cache (/user_data/data/dirk/prj/unittest/caching/unittest/
↳ output_data/data_version_test_no_load_on_init.pkl)
```

```
PickCache: Data version increased, ignoring previous cache data
```

```
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.pkl)
```

```
PickCache: Loading property for 'str' from source instance ('__string__')
```

```
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.pkl)
```

```
PickCache: Loading property for 'unicode' from source instance ('__unicode__')
```

```
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.pkl)
```

```
PickCache: Loading property for 'integer' from source instance (34)
```

```
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.pkl)
```

```
PickCache: Loading property for 'float' from source instance (2.71828)
```

```
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.pkl)
```

```
PickCache: Loading property for 'list' from source instance (['one', 2, 3, '4'])
```

```
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.pkl)
```

```
PickCache: Loading property for 'dict' from source instance ({'1': '1', '2': 2, '3': 'three',
↳ '4': '4'})
```

```
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ data_version_test_no_load_on_init.pkl)
```

```
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'>)
```

### B.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

↪ /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/uid\_test\_load\_on\_init.pkl  
 ↪ as cache file.

PickCache: Cache file does not exists (yet).

PickCache: Loading all data from source - ['str', 'unicode', 'integer', 'float', 'list',  
 ↪ 'dict']

PickCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/uid\_test\_load\_on\_init.pkl)

---

**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'>).

---

PickCache: Loading properties from cache

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/uid\_test\_load\_on\_init.pkl)

PickCache: Source uid changed, ignoring previous cache data

PickCache: Loading all data from source - ['str', 'unicode', 'integer', 'float', 'list',  
 ↪ 'dict']

PickCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/uid\_test\_load\_on\_init.pkl)

PickCache: Providing property for 'str' from cache ('\_\_string\_\_')

PickCache: Providing property for 'unicode' from cache ('\_\_unicode\_\_')

PickCache: Providing property for 'integer' from cache (34)

PickCache: Providing property for 'float' from cache (2.71828)

PickCache: Providing property for 'list' from cache (['one', 2, 3, '4'])

PickCache: Providing property for 'dict' from cache ({'1': '1', '2': 2, '3': 'three', '4':  
 ↪ '4'})

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'>)

**B.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 /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ uid\_test\_no\_load\_on\_init.pkl as cache  
 ↳ file.

PickCache: Cache file does not exists (yet).

PickCache: Loading all data from source - ['str', 'unicode', 'integer', 'float', 'list', 'dict']

PickCache: cache-file stored (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/ uid\_test\_no\_load\_on\_init.pkl)

---

**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'>).

---

## Unittest for caching

```
PickCache: Loading properties from cache (/user_data/data/dirk/prj/unittest/caching/unittest/
↳ output_data/uid_test_no_load_on_init.pkl)
PickCache: Source uid changed, ignoring previous cache data
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.pkl)
PickCache: Loading property for 'str' from source instance ('__string__')
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.pkl)
PickCache: Loading property for 'unicode' from source instance ('__unicode__')
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.pkl)
PickCache: Loading property for 'integer' from source instance (34)
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.pkl)
PickCache: Loading property for 'float' from source instance (2.71828)
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.pkl)
PickCache: Loading property for 'list' from source instance (['one', 2, 3, '4'])
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.pkl)
PickCache: Loading property for 'dict' from source instance ({'1': '1', '2': 2, '3': 'three',
↳ '4': '4'})
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ uid_test_no_load_on_init.pkl)
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'>)
```

### B.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'>).

---

```
PickCache: cache-file stored (/user_data/data/dirk/prj/unittest/caching/unittest/output_data/
↳ save_callback_load_on_init.json)
```

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

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

### B.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

↪ /user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/internal\_keys\_test.pkl as  
 ↪ cache file.

PickCache: Cache file does not exists (yet).

PickCache: Loading all data from source - ['\_property\_cache\_uid\_', '\_\_property\_cache\_uid\_',  
 ↪ '\_property\_cache\_data\_version\_', '\_\_property\_cache\_data\_version\_']

PickCache: cache-file stored

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/internal\_keys\_test.pkl)

---

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

---

PickCache: Loading properties from cache

↪ (/user\_data/data/dirk/prj/unittest/caching/unittest/output\_data/internal\_keys\_test.pkl)

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'>)



## C Test-Coverage

### C.1 caching

The line coverage for `caching` was 100.0%

The branch coverage for `caching` was 100.0%

#### C.1.1 `caching.__init__.py`

The line coverage for `caching.__init__.py` was 100.0%

The branch coverage for `caching.__init__.py` was 100.0%

```

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__ = (2, 3)

```

## Unittest for caching

```
45 """The Tested Interpreter—Versions"""
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).
```

## Unittest for caching

```
97     :param key: key for value to get.
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
```

## Unittest for caching

```
151 def _key_filter(self, key):
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.
163         _source_instance.keys()))
164     for key in self._source_instance.keys():
165         val = self._source_instance.get(key)
166         self._cached_props[self._key_filter(key)] = val
167
168 def _save_cache(self):
169     with open(self._cache_filename, 'wb') as fh:
170         pickle.dump(self._cached_props, fh)
171         logger.info('%s cache-file stored (%s)', self.LOG_PREFIX, self._cache_filename)
172     if self._callback_on_data_storage is not None:
173         self._callback_on_data_storage()
174
175 def _uid(self):
176     if self._cached_props is None:
177         return None
178     else:
179         return self._cached_props.get(self.UID_TAG, None)
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 transfered 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

```
206
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()
```