

Unittest for caching

September 15, 2024

Contents

1	Test Information	3
1.1	Test Candidate Information	3
1.2	Unittest Information	3
1.3	Test System Information	3
2	Statistic	3
2.1	Test-Statistic for testrun with python 3.11.2 (final)	3
2.2	Coverage Statistic	4
3	Testcases with no corresponding Requirement	5
3.1	Summary for testrun with python 3.11.2 (final)	5
3.1.1	caching.property_cache.json: Test cached data (full init)	5
3.1.2	caching.property_cache.json: Test cached data (partially init)	5
3.1.3	caching.property_cache.json: Test execution of save callback (full init)	5
3.1.4	caching.property_cache.json: Test full initialised JSON-Cache-Object	6
3.1.5	caching.property_cache.json: Test get from source caused by changed uid (full init)	6
3.1.6	caching.property_cache.json: Test get from source caused by changed uid (partially init)	6
3.1.7	caching.property_cache.json: Test get from source caused by increased data version (full init)	7
3.1.8	caching.property_cache.json: Test get from source caused by increased data version (partially init)	7
3.1.9	caching.property_cache.json: Test internal key usage	8
3.1.10	caching.property_cache.json: Test partially initialisation of JSON-Cache-Object	8
3.1.11	caching.property_cache.pickle: Test cached data (full init)	8
3.1.12	caching.property_cache.pickle: Test cached data (partially init)	9
3.1.13	caching.property_cache.pickle: Test execution of save callback (full init)	9
3.1.14	caching.property_cache.pickle: Test full initialised PICKLE-Cache-Object	9
3.1.15	caching.property_cache.pickle: Test get from source caused by changed uid (full init)	10
3.1.16	caching.property_cache.pickle: Test get from source caused by changed uid (partially init)	10
3.1.17	caching.property_cache.pickle: Test get from source caused by increased data version (full init)	11
3.1.18	caching.property_cache.pickle: Test get from source caused by increased data version (partially init)	11
3.1.19	caching.property_cache.pickle: Test internal key usage	11
3.1.20	caching.property_cache.pickle: Test partially initialised PICKLE-Cache-Object	12

A	Trace for testrun with python 3.11.2 (final)	13
A.1	Tests with status Info (20)	13
A.1.1	caching.property_cache_json: Test full initialised JSON-Cache-Object	13
A.1.2	caching.property_cache_json: Test partially initialisation of JSON-Cache-Object	13
A.1.3	caching.property_cache_json: Test cached data (full init)	14
A.1.4	caching.property_cache_json: Test cached data (partially init)	15
A.1.5	caching.property_cache_json: Test get from source caused by increased data version (full init)	15
A.1.6	caching.property_cache_json: Test get from source caused by increased data version (partially init)	16
A.1.7	caching.property_cache_json: Test get from source caused by changed uid (full init)	16
A.1.8	caching.property_cache_json: Test get from source caused by changed uid (partially init)	17
A.1.9	caching.property_cache_json: Test execution of save callback (full init)	17
A.1.10	caching.property_cache_json: Test internal key usage	17
A.1.11	caching.property_cache_pickle: Test full initialised PICKLE-Cache-Object	18
A.1.12	caching.property_cache_pickle: Test partially initialised PICKLE-Cache-Object	19
A.1.13	caching.property_cache_pickle: Test cached data (full init)	19
A.1.14	caching.property_cache_pickle: Test cached data (partially init)	20
A.1.15	caching.property_cache_pickle: Test get from source caused by increased data version (full init)	20
A.1.16	caching.property_cache_pickle: Test get from source caused by increased data version (partially init)	21
A.1.17	caching.property_cache_pickle: Test get from source caused by changed uid (full init)	21
A.1.18	caching.property_cache_pickle: Test get from source caused by changed uid (partially init)	22
A.1.19	caching.property_cache_pickle: Test execution of save callback (full init)	22
A.1.20	caching.property_cache_pickle: Test internal key usage	23
B	Test-Coverage	23
B.1	caching	23
B.1.1	caching.__init__.py	23

1 Test Information

1.1 Test Candidate Information

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

Library Information

Name	caching
State	Released
Supported Interpreters	python3
Version	03683b9fee192b005ed723f0f14eecd5

Dependencies

1.2 Unittest Information

Unittest Information

Version	6120114abb148e96dd3b8bd41a7af82a
Testruns with	python 3.11.2 (final)

1.3 Test System Information

System Information

Architecture	64bit
Distribution	Debian GNU/Linux 12 bookworm
Hostname	ahorn
Kernel	6.1.0-17-amd64 (#1 SMP PREEMPT_DYNAMIC Debian 6.1.69-1 (2023-12-30))
Machine	x86_64
Path	/home/dirk/my_repositories/unittest/caching
System	Linux
Username	dirk

2 Statistic

2.1 Test-Statistic for testrun with python 3.11.2 (final)

Number of tests	20
Number of successfull tests	20
Number of possibly failed tests	0
Number of failed tests	0

Executionlevel	Full Test (all defined tests)
Time consumption	0.028s

2.2 Coverage Statistic

Module- or Filename	Line-Coverage	Branch-Coverage
caching	92.7%	88.0%
caching.__init__.py	92.7%	

3 Testcases with no corresponding Requirement

3.1 Summary for testrun with python 3.11.2 (final)

3.1.1 caching.property_cache_json: Test cached data (full init)

Testresult

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

Testrun:	python 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/report/_init....py (323)
Start-Time:	2024-09-15 16:36:11,638
Finished-Time:	2024-09-15 16:36:11,639
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'>).

3.1.2 caching.property_cache_json: Test cached data (partially init)

Testresult

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

Testrun:	python 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/report/_init....py (323)
Start-Time:	2024-09-15 16:36:11,639
Finished-Time:	2024-09-15 16:36:11,640
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'>).

3.1.3 caching.property_cache_json: Test execution of save callback (full init)

Testresult

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

Testrun:	python 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/report/_init....py (323)

Start-Time: 2024-09-15 16:36:11,650
 Finished-Time: 2024-09-15 16:36:11,650
 Time-Consumption 0.001s

Testsummary:

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

3.1.4 caching.property_cache_json: Test full initialised JSON-Cache-Object

Testresult

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

Testrun: python 3.11.2 (final)
 Caller: /home/dirk/my_repositories/unittest/caching/unittest/src/report/_init_.py (323)
 Start-Time: 2024-09-15 16:36:11,633
 Finished-Time: 2024-09-15 16:36:11,635
 Time-Consumption 0.002s

Testsummary:

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

3.1.5 caching.property_cache_json: Test get from source caused by changed uid (full init)

Testresult

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

Testrun: python 3.11.2 (final)
 Caller: /home/dirk/my_repositories/unittest/caching/unittest/src/report/_init_.py (323)
 Start-Time: 2024-09-15 16:36:11,645
 Finished-Time: 2024-09-15 16:36:11,647
 Time-Consumption 0.001s

Testsummary:

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

3.1.6 caching.property_cache_json: Test get from source caused by changed uid (partially init)

Testresult

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

Testrun:	python 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/report/_init_.py (323)
Start-Time:	2024-09-15 16:36:11,647
Finished-Time:	2024-09-15 16:36:11,649
Time-Consumption	0.003s

Testsummary:

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

3.1.7 caching.property_cache.json: Test get from source caused by increased data version (full init)

Testresult

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

Testrun:	python 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/report/_init_.py (323)
Start-Time:	2024-09-15 16:36:11,641
Finished-Time:	2024-09-15 16:36:11,642
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 {'str': '__string__', 'unicode': '__unicode__', 'integer': 34, 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '2': 2, '3': 'three', '4': '4'}} and Type is <class 'dict'>).

3.1.8 caching.property_cache.json: Test get from source caused by increased data version (partially init)

Testresult

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

Testrun:	python 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/report/_init_.py (323)
Start-Time:	2024-09-15 16:36:11,642
Finished-Time:	2024-09-15 16:36:11,645
Time-Consumption	0.003s

Testsummary:

Info	Initialising property_cache.json (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'>).

3.1.9 caching.property_cache_json: Test internal key usage

Testresult

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

Testrun:	python 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/report/_init_.py (323)
Start-Time:	2024-09-15 16:36:11,650
Finished-Time:	2024-09-15 16:36:11,651
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 {'_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'>).
Success	Keyfilter returnvalue for 5 (<class 'int'>) is correct (Content 5 and Type is <class 'int'>).

3.1.10 caching.property_cache_json: Test partially initialisation of JSON-Cache-Object

Testresult

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

Testrun:	python 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/report/_init_.py (323)
Start-Time:	2024-09-15 16:36:11,635
Finished-Time:	2024-09-15 16:36:11,638
Time-Consumption	0.003s

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'>).

3.1.11 caching.property_cache_pickle: Test cached data (full init)

Testresult

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

Testrun:	python 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/report/_init_.py (323)
Start-Time:	2024-09-15 16:36:11,654
Finished-Time:	2024-09-15 16:36:11,655
Time-Consumption	0.001s

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'>).

3.1.12 caching.property_cache_pickle: Test cached data (partially init)

Testresult

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

Testrun:	python 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/report/_init_.py (323)
Start-Time:	2024-09-15 16:36:11,655
Finished-Time:	2024-09-15 16:36:11,656
Time-Consumption	0.001s

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'>).

3.1.13 caching.property_cache_pickle: Test execution of save callback (full init)

Testresult

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

Testrun:	python 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/report/_init_.py (323)
Start-Time:	2024-09-15 16:36:11,662
Finished-Time:	2024-09-15 16:36:11,663
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'>).

3.1.14 caching.property_cache_pickle: Test full initialised PICKLE-Cache-Object

Testresult

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

Testrun:	python 3.11.2 (final)
----------	-----------------------

Caller: /home/dirk/my_repositories/unittest/caching/unittest/src/report/_init....py (323)
 Start-Time: 2024-09-15 16:36:11,652
 Finished-Time: 2024-09-15 16:36:11,653
 Time-Consumption 0.001s

Testsummary:

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

3.1.15 caching.property_cache_pickle: Test get from source caused by changed uid (full init)

Testresult

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

Testrun: python 3.11.2 (final)
 Caller: /home/dirk/my_repositories/unittest/caching/unittest/src/report/_init....py (323)
 Start-Time: 2024-09-15 16:36:11,659
 Finished-Time: 2024-09-15 16:36:11,660
 Time-Consumption 0.001s

Testsummary:

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

3.1.16 caching.property_cache_pickle: Test get from source caused by changed uid (partially init)

Testresult

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

Testrun: python 3.11.2 (final)
 Caller: /home/dirk/my_repositories/unittest/caching/unittest/src/report/_init....py (323)
 Start-Time: 2024-09-15 16:36:11,660
 Finished-Time: 2024-09-15 16:36:11,662
 Time-Consumption 0.002s

Testsummary:

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

3.1.17 caching.property_cache_pickle: Test get from source caused by increased data version (full init)

Testresult

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

Testrun:	python 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/report/_init....py (323)
Start-Time:	2024-09-15 16:36:11,656
Finished-Time:	2024-09-15 16:36:11,657
Time-Consumption	0.001s

Testsummary:

Info	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'>).

3.1.18 caching.property_cache_pickle: Test get from source caused by increased data version (partially init)

Testresult

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

Testrun:	python 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/report/_init....py (323)
Start-Time:	2024-09-15 16:36:11,657
Finished-Time:	2024-09-15 16:36:11,659
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 {'str': 'string_', 'unicode': 'unicode_', 'integer': 34, 'float': 2.71828, 'list': ['one', 2, 3, '4'], 'dict': {'1': '1', '2': 2, '3': 'three', '4': '4'}} and Type is <class 'dict'>).

3.1.19 caching.property_cache_pickle: Test internal key usage

Testresult

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

Testrun:	python 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/report/_init....py (323)
Start-Time:	2024-09-15 16:36:11,663
Finished-Time:	2024-09-15 16:36:11,664
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_uid_': 'no uid', '__property_cache_uid_': 'no second uid', '_property_cache_data_version_': 'no data version', '__property_cache_data_version_': 'no second data version'}) and Type is <class 'dict'>).

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

3.1.20 caching.property_cache_pickle: Test partially initialised PICKLE-Cache-Object

Testresult

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

Testrun:	python 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/report/_init_.py (323)
Start-Time:	2024-09-15 16:36:11,653
Finished-Time:	2024-09-15 16:36:11,654
Time-Consumption	0.001s

Testsummary:

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

A Trace for testrun with python 3.11.2 (final)

A.1 Tests with status Info (20)

A.1.1 caching.property_cache_json: Test full initialised JSON-Cache-Object

Testresult

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

Info Initialising property_cache_json (load_all_on_init=True).

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

Initialising cached class with

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

Info Extracting storage object from property_cache_json for comparison.

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

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

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

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

A.1.2 caching.property_cache_json: Test 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

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

Info Partially initialising cache object by requesting some information.

Info Extracting storage object from property_cache_json for comparison.

Using storage object of cache class for comparison: {'integer': 17, 'str': 'string',
 ↪ 'unicode': 'unicode'}

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

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

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

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

Testresult

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

Info Initialising property_cache_json (load_all_on_init=True).

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

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

Info Collecting data from cache instance.

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

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

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

A.1.4 caching.property_cache.json: Test cached data (partially init)

Testresult

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

Info Initialising property_cache.json (load_all_on_init=True).

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

Initialising cached class with /home/dirk/my_repositories/unittest/caching/unittest/output_data/cache_data_test_no_load_on_init.json as cache
 ↪ file.

Info Collecting data from cache instance.

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

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

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

A.1.5 caching.property_cache.json: Test get from source caused by increased data version (full init)

Testresult

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

Info Initialising property_cache.json (load_all_on_init=True).

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

Initialising cached class with /home/dirk/my_repositories/unittest/caching/unittest/output_data/data_version_test_load_on_init.json as cache
 ↪ file.

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

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

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

A.1.6 caching.property_cache.json: Test get from source caused by increased data version (partially init)

Testresult

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

Info Initialising property_cache.json (load_all_on_init=True).

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

Initialising cached class with /home/dirk/my_repositories/unittest/caching/unittest/output_data/data_version_test_no_load_on_init.json as cache
 ↳ file.

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

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

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

A.1.7 caching.property_cache.json: Test get from source caused by changed uid (full init)

Testresult

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

Info Initialising property_cache.json (load_all_on_init=True).

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

Initialising cached class with /home/dirk/my_repositories/unittest/caching/unittest/output_data/uid_test_load_on_init.json as cache
 ↳ file.

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

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

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

A.1.8 caching.property_cache_json: Test get from source caused by changed uid (partially init)

Testresult

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

Info Initialising property_cache_json (load_all_on_init=True).

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

Initialising cached class with /home/dirk/my_repositories/unittest/caching/unittest/output_data/uid_test_no_load_on_init.json as cache
 ↪ file.

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

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

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

A.1.9 caching.property_cache_json: Test execution of save callback (full init)

Testresult

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

Info Installing save_callback, which sets a variable to True on execution.

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

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

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

A.1.10 caching.property_cache_json: Test internal key usage

Testresult

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

Info Initialising property_cache_json (load_all_on_init=True).

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

Initialising cached class with
 ↪ /home/dirk/my_repositories/unittest/caching/unittest/output_data/internal_keys_test.json
 ↪ as cache file.

Info Extracting storage object from property_cache_json for comparison.

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

Success Cache is correct (Content {'__property_cache_data_version_': 'no second data version', '__property_cache_uid_': 'no second uid', '_property_cache_data_version_': 'no data version', '_property_cache_uid_': 'no uid'} and Type is <class 'dict'>).

```
Result (Cache): { '__property_cache_data_version_': 'no second data version',
↪ '__property_cache_uid_': 'no second uid', '_property_cache_data_version_': 'no data
↪ version', '_property_cache_uid_': 'no uid' } (<class 'dict'>)
```

```
Expectation (Cache): result = { '_property_cache_uid_': 'no uid', '__property_cache_uid_':
↪ 'no second uid', '_property_cache_data_version_': 'no data version',
↪ '__property_cache_data_version_': 'no second data version' } (<class 'dict'>)
```

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

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

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

A.1.11 caching.property_cache_pickle: Test full initialised PICKLE-Cache-Object

Testresult

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

Info Initialising property_cache_pickle (load_all_on_init=True).

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

```
Initialising cached class with
```

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

Info Extracting storage object from property_cache_pickle for comparison.

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

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

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

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

A.1.12 caching.property_cache_pickle: Test partially initialised PICKLE-Cache-Object

Testresult

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

Info Initialising property_cache_pickle (load_all_on_init=False).

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

Initialising cached class with

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

Info Partially initialising cache object by requesting some information.

Info Extracting storage object from property_cache_pickle for comparison.

Using storage object of cache class for comparison: {'integer': 17, 'unicode': 'unicode',
↪ 'str': 'string'}

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

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

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

A.1.13 caching.property_cache_pickle: Test cached data (full init)

Testresult

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

Info Initialising property_cache_pickle (load_all_on_init=True).

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

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

Info Collecting data from cache instance.

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

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

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

A.1.14 caching.property_cache_pickle: Test cached data (partially init)

Testresult

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

Info Initialising property_cache_pickle (load_all_on_init=True).

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

Initialising cached class with /home/dirk/my_repositories/unittest/caching/unittest/output_data/cache_data_test_no_load_on_init.pkl as cache file.

Info Collecting data from cache instance.

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

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

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

A.1.15 caching.property_cache_pickle: Test get from source caused by increased data version (full init)

Testresult

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

Info Initialising property_cache_pickle (load_all_on_init=True).

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

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

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

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

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

A.1.16 caching.property_cache_pickle: Test get from source caused by increased data version (partially init)

Testresult

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

Info Initialising property_cache_pickle (load_all_on_init=True).

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

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

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

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

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

A.1.17 caching.property_cache_pickle: Test get from source caused by changed uid (full init)

Testresult

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

Info Initialising property_cache_pickle (load_all_on_init=True).

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

```
Initialising cached class with /home/dirk/my_repositories/unittest/caching/unittest/output_da_
↳ ta/uid_test_load_on_init.pkl as cache
↳ file.
```

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

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

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

A.1.18 caching.property_cache_pickle: Test get from source caused by changed uid (partially init)

Testresult

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

Info Initialising property_cache_pickle (load_all_on_init=True).

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

```
Initialising cached class with /home/dirk/my_repositories/unittest/caching/unittest/output_da_
↳ ta/uid_test_no_load_on_init.pkl as cache
↳ file.
```

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

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

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

A.1.19 caching.property_cache_pickle: Test execution of save callback (full init)

Testresult

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

Info Installing save_callback, which sets a variable to True on execution.

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

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

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

A.1.20 caching.property_cache_pickle: Test internal key usage

Testresult

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

Info Initialising property_cache_pickle (load_all_on_init=True).

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

Initialising cached class with

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

Info Extracting storage object from property_cache_pickle for comparison.

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

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 ver-
 sion'} and Type is <class 'dict'>).

Result (Cache): { '_property_cache_uid_': 'no uid', '__property_cache_uid_': 'no second uid',
 ↪ '_property_cache_data_version_': 'no data version', '__property_cache_data_version_': 'no
 ↪ second data version' } (<class 'dict'>)

Expectation (Cache): result = { '_property_cache_uid_': 'no uid', '__property_cache_uid_':
 ↪ 'no second uid', '_property_cache_data_version_': 'no data version',
 ↪ '__property_cache_data_version_': 'no second data version' } (<class 'dict'>)

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

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

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

B Test-Coverage

B.1 caching

The line coverage for caching was 92.7%

The branch coverage for caching was 88.0%

B.1.1 caching.__init__.py

The line coverage for caching.__init__.py was 92.7%

The branch coverage for caching.__init__.py was 88.0%

Unittest for caching

```
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 json
28 import logging
29 import os
30 import pickle
31 import sys
32 import time
33
34 try:
35     from config import APP_NAME as ROOT_LOGGER_NAME
36 except ImportError:
37     ROOT_LOGGER_NAME = 'root'
38 logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__)
39
40 __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.
41 For more Information read the documentation.""" % __name__.replace('-', '\\-')
42 """The Module Description"""
43 __INTERPRETER__ = (3, )
44 """The Tested Interpreter-Versions"""
45
46
47 class property_cache_pickle(object):
48     """
49     Class to cache properties, which take longer on initialising than reading a file in pickle format.
50
51     :param source_instance: The source instance holding the data
52     :type source_instance: instance
53     :param cache_filename: File name, where the properties are stored as cache
54     :type cache_filename: str
55     :param load_all_on_init: Optionally init behaviour control parameter. True will load all available properties from source on init, False not.
56
57     .. note:: source_instance needs to have at least the following methods: uid(), keys(), data_version(), get()
```

Unittest for caching

```
58
59     * uid(): returns the unique id of the source.
60     * keys(): returns a list of all available keys.
61     * data_version(): returns a version number of the current data (it should be
increased, if the get method of the source instance returns improved values or the data
structure had been changed).
62     * get(key, default): returns the property for a key. If key does not exists,
default will be returned.
63
64 Reasons for updating the complete data set:
65
66 * UID of source_instance has changed (in comparison to the cached value).
67 * data_version is increased
68
69 **Example:**
70
71 .. literalinclude:: caching/_examples_/property_cache_pickle.py
72
73 Will result on the first execution to the following output (with a long execution time):
74
75 .. literalinclude:: caching/_examples_/property_cache_pickle_1.log
76
77 With every following execution (slow for getting "two" which is not cached – see
implementation):
78
79 .. literalinclude:: caching/_examples_/property_cache_pickle_2.log
80 """
81 LOG_PREFIX = 'PickCache: '
82 DATA_VERSION_TAG = '_property_cache_data_version_'
83 STORAGE_VERSION_TAG = '_storage_version_'
84 UID_TAG = '_property_cache_uid_'
85 DATA_TAG = '_data_'
86 AGE_TAG = '_age_'
87 #
88 STORAGE_VERSION = 1
89
90 def __init__(self, source_instance, cache_filename, load_all_on_init=False,
callback_on_data_storage=None, max_age=None, store_on_get=True):
91     self._source_instance = source_instance
92     self._cache_filename = cache_filename
93     self._load_all_on_init = load_all_on_init
94     self._callback_on_data_storage = callback_on_data_storage
95     self._max_age = max_age
96     self._store_on_get = store_on_get
97     self._cached_props = None
98
99 def get(self, key, default=None):
100     """
101     Method to get the cached property. If key does not exists in cache, the property will be
loaded from source_instance and stored in cache (file).
102
103     :param key: key for value to get.
104     :param default: value to be returned, if key does not exists.
105     :returns: value for a given key or default value.
106     """
107     if key in self.keys():
108         if self._cached_props is None:
109             self._init_cache()
110         if self._max_age is None:
111             cache_old = False
112     else:
```

Unittest for caching

```

113         cache_old = time.time() - self._cached_props[self.AGE_TAG].get(self._key_filter(
114             key), 0) > self._max_age
115         if cache_old:
116             logger.debug("The cached value is old, cached value will be ignored")
117         if self._key_filter(key) not in self._cached_props[self.DATA_TAG] or cache_old:
118             val = self._source_instance.get(key, None)
119             logger.debug("%s Loading property for '%s' from source instance (%s)", self.
120                 LOG_PREFIX, key, repr(val))
121             if self._store_on_get:
122                 tm = int(time.time())
123                 logger.debug("Storing value=%s with timestamp=%d to cache", val, tm)
124                 self._cached_props[self.DATA_TAG][self._key_filter(key)] = val
125                 self._cached_props[self.AGE_TAG][self._key_filter(key)] = tm
126                 self._save_cache()
127             else:
128                 return val
129         else:
130             logger.debug("%s Providing property for '%s' from cache (%s)", self.LOG_PREFIX,
131                 key, repr(self._cached_props[self.DATA_TAG].get(self._key_filter(key)
132                 ), default)))
133             return self._cached_props[self.DATA_TAG].get(self._key_filter(key), default)
134         else:
135             logger.info("%s Key '%s' is not in cached_keys. Uncached data will be returned.",
136                 self.LOG_PREFIX, key)
137             return self._source_instance.get(key, default)
138
139     def full_update(self):
140         self._load_source()
141         self._save_cache()
142
143     def keys(self):
144         """
145         Method to get the available keys (from :data:`source_instance`).
146         """
147         return self._source_instance.keys()
148
149     def _data_version(self):
150         if self._cached_props is None:
151             return None
152         else:
153             return self._cached_props.get(self.DATA_VERSION_TAG, None)
154
155     def _storage_version(self):
156         if self._cached_props is None:
157             return None
158         else:
159             return self._cached_props.get(self.STORAGE_VERSION_TAG, None)
160
161     def _init_cache(self):
162         load_cache = self._load_cache()
163         uid = self._source_instance.uid() != self._uid()
164         try:
165             data_version = self._source_instance.data_version() > self._data_version()
166         except TypeError:
167             data_version = True
168         try:
169             storage_version = self._storage_version() != self.STORAGE_VERSION
170         except TypeError:
171             storage_version = True
172
173     #

```

Unittest for caching

```
169     if not load_cache or uid or data_version or storage_version:
170         if load_cache:
171             if self._uid() is not None and uid:
172                 logger.debug("%s Source uid changed, ignoring previous cache data", self.
LOG_PREFIX)
173             if self._data_version() is not None and data_version:
174                 logger.debug("%s Data version increased, ignoring previous cache data", self.
LOG_PREFIX)
175                 if storage_version:
176                     logger.debug("%s Storage version changed, ignoring previous cache data", self
.LOG_PREFIX)
177                 self._cached_props = {self.AGE_TAG: {}, self.DATA_TAG: {}}
178                 if self._load_all_on_init:
179                     self._load_source()
180                 self._cached_props[self.UID_TAG] = self._source_instance.uid()
181                 self._cached_props[self.DATA_VERSION_TAG] = self._source_instance.data_version()
182                 self._cached_props[self.STORAGE_VERSION_TAG] = self.STORAGE_VERSION
183                 self._save_cache()
184
185     def _load_cache(self):
186         if os.path.exists(self._cache_filename):
187             with open(self._cache_filename, 'rb') as fh:
188                 self._cached_props = pickle.load(fh)
189                 logger.info('%s Loading properties from cache (%s)', self.LOG_PREFIX, self.
_cache_filename)
190             return True
191         else:
192             logger.debug('%s Cache file does not exists (yet).', self.LOG_PREFIX)
193             return False
194
195     def _key_filter(self, key):
196         return key
197
198     def _load_source(self):
199         if self._cached_props is None:
200             self._init_cache()
201             logger.debug('%s Loading all data from source - %s', self.LOG_PREFIX, repr(self.
_source_instance.keys()))
202             for key in self._source_instance.keys():
203                 self._cached_props[self.DATA_TAG][self._key_filter(key)] = self._source_instance.get(
key)
204                 self._cached_props[self.AGE_TAG][self._key_filter(key)] = int(time.time())
205
206     def _save_cache(self):
207         with open(self._cache_filename, 'wb') as fh:
208             pickle.dump(self._cached_props, fh)
209             logger.info('%s cache-file stored (%s)', self.LOG_PREFIX, self._cache_filename)
210         if self._callback_on_data_storage is not None:
211             self._callback_on_data_storage()
212
213     def _uid(self):
214         if self._cached_props is None:
215             return None
216         else:
217             return self._cached_props.get(self.UID_TAG, None)
218
219
220 class property_cache_json(property_cache_pickle):
```

Unittest for caching

```
221 """
222 Class to cache properties, which take longer on initialising than reading a file in json
223 format. See also parent :py:class:`property_cache_pickle`
224
225 :param source_instance: The source instance holding the data
226 :type source_instance: instance
227 :param cache_filename: File name, where the properties are stored as cache
228 :type cache_filename: str
229 :param load_all_on_init: Optionally init behaviour control parameter. True will load all
230 available properties from source on init, False not.
231
232 .. warning::
233     * This class uses json. You should only use keys of type string!
234     * Unicode types are transferred to strings
235
236 .. note:: source_instance needs to have at least the following methods: uid(), keys(),
237 data_version(), get()
238
239     * uid(): returns the unique id of the source.
240     * keys(): returns a list of all available keys.
241     * data_version(): returns a version number of the current data (it should be
242 increased, if the get method of the source instance returns improved values or the data
243 structure had been changed).
244     * get(key, default): returns the property for a key. If key does not exists,
245 default will be returned.
246
247 Reasons for updating the complete data set:
248
249 * UID of source_instance has changed (in comparison to the cached value).
250 * data_version is increased
251
252 Example:
253
254 .. literalinclude:: caching/_examples_/property_cache_json.py
255
256 Will result on the first execution to the following output (with a long execution time):
257
258 .. literalinclude:: caching/_examples_/property_cache_json_1.log
259
260 With every following execution (slow for getting "two" which is not cached – see
261 implementation):
262
263 .. literalinclude:: caching/_examples_/property_cache_json_2.log
264 """
265 LOG_PREFIX = 'JsonCache: '
266
267 def _load_cache(self):
268     if os.path.exists(self._cache_filename):
269         with open(self._cache_filename, 'r') as fh:
270             self._cached_props = json.load(fh)
271             logger.info('%s Loading properties from cache (%s)', self.LOG_PREFIX, self._cache_filename)
272             return True
273     else:
274         logger.debug('%s Cache file does not exists (yet).', self.LOG_PREFIX)
275         return False
276
277 def _save_cache(self):
278     with open(self._cache_filename, 'w') as fh:
279         json.dump(self._cached_props, fh, sort_keys=True, indent=4)
280         logger.info('%s cache-file stored (%s)', self.LOG_PREFIX, self._cache_filename)
281     if self._callback_on_data_storage is not None:
282         self._callback_on_data_storage()
```