

Unittest for caching

September 23, 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	5b76bef0032d8f8905a892a334f0f5b8

Dependencies	
--------------	--

1.2 Unittest Information

Unittest Information	
Version	ac6e9667753d32025048abc5366ddb10
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	0.025s

2.2 Coverage Statistic

Module- or Filename	Line-Coverage	Branch-Coverage
caching	89.5%	78.3%
caching.__init__.py	89.5%	

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:	2024-09-23 07:59:55,184
Finished-Time:	2024-09-23 07:59:55,185
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:	2024-09-23 07:59:55,185
Finished-Time:	2024-09-23 07:59:55,187
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: 2024-09-23 07:59:55,195
 Finished-Time: 2024-09-23 07:59:55,195
 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'>).

3.1.4 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: 2024-09-23 07:59:55,181
 Finished-Time: 2024-09-23 07:59:55,182
 Time-Consumption 0.001s

Testsummary:

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

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

Testresult

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

Testrun: python 3.11.2 (final)
 Caller: /home/dirk/my_repositories/unittest/caching/unittest/src/report/_init_.py (323)
 Start-Time: 2024-09-23 07:59:55,191
 Finished-Time: 2024-09-23 07:59:55,192
 Time-Consumption 0.002s

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:	2024-09-23 07:59:55,193
Finished-Time:	2024-09-23 07:59:55,195
Time-Consumption	0.002s

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:	2024-09-23 07:59:55,187
Finished-Time:	2024-09-23 07:59:55,188
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:	2024-09-23 07:59:55,188
Finished-Time:	2024-09-23 07:59:55,191
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:	2024-09-23 07:59:55,195
Finished-Time:	2024-09-23 07:59:55,196
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:	2024-09-23 07:59:55,182
Finished-Time:	2024-09-23 07:59:55,184
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'>).

3.1.11 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:	2024-09-23 07:59:55,198
Finished-Time:	2024-09-23 07:59:55,199
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:	2024-09-23 07:59:55,199
Finished-Time:	2024-09-23 07:59:55,200
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:	2024-09-23 07:59:55,207
Finished-Time:	2024-09-23 07:59:55,207
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: 2024-09-23 07:59:55,196
 Finished-Time: 2024-09-23 07:59:55,197
 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: 2024-09-23 07:59:55,204
 Finished-Time: 2024-09-23 07:59:55,205
 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: 2024-09-23 07:59:55,205
 Finished-Time: 2024-09-23 07:59:55,206
 Time-Consumption 0.001s

Testsummary:

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

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

Testresult

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

Testrun:	python 3.11.2 (final)
Caller:	/home/dirk/my_repositories/unittest/caching/unittest/src/report/_init....py (323)
Start-Time:	2024-09-23 07:59:55,201
Finished-Time:	2024-09-23 07:59:55,202
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:	2024-09-23 07:59:55,202
Finished-Time:	2024-09-23 07:59:55,204
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:	2024-09-23 07:59:55,207
Finished-Time:	2024-09-23 07:59:55,208
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:	2024-09-23 07:59:55,197
Finished-Time:	2024-09-23 07:59:55,198
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 {'unicode': 'unicode', 'str': 'string', 'integer': 17}) 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: {'unicode': 'unicode', 'str': 'string',
↳ 'integer': 17}
```

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

```
Result (Cache object): { 'unicode': 'unicode', 'str': 'string', 'integer': 17 } (<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 89.5%

The branch coverage for caching was 78.3%

B.1.1 caching.__init__.py

The line coverage for caching.__init__.py was 89.5%

The branch coverage for caching.__init__.py was 78.3%

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 time
32
33 try:
34     from config import APP_NAME as ROOT_LOGGER_NAME
35 except ImportError:
36     ROOT_LOGGER_NAME = 'root'
37 logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__)
38
39 __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.
40 For more Information read the documentation.""" % __name__.replace('-', '\\-')
41 """The Module Description"""
42 __INTERPRETER__ = (3, )
43 """The Tested Interpreter-Versions"""
44
45
46 class property_cache_pickle(object):
47     """
48     This class caches the data from a given `source_instance`. It takes the data from the cache instead of generating the data from the `source_instance`, if the conditions for the cache usage are given.
49
50     .. admonition:: Required properties for the `source_instance`
51
52         * **uid():** returns the unique id of the source's source or None, if you don't want to use the unique id.
53         * **keys():** returns a list of all available keys.
54         * **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).
```

Unittest for caching

```
56         * get(key, default): returns the property for a key. If key does not exists ,
          default will be returned.
57
58     :param source_instance: The source instance holding the data
59     :type source_instance: instance
60     :param cache_filename: File name, where the properties are stored as cache
61     :type cache_filename: str
62     :param load_all_on_init: True will load all data from the source instance , when the cache
          will be initialised the first time.
63     :type load_all_on_init: bool
64     :param callback_on_data_storage: The callback will be executed every time when the cache file
          is stored. It will be executed with the instance of this class as first argument.
65     :type callback_on_data_storage: method
66     :param max_age: The maximum age of the cached data in seconds or None for no maximum age.
67     :type max_age: int or None
68     :param store_on_get: False will prevent cache storage with execution of the get(key,
          default) method. You need to store the cache somewhere else.
69     :type store_on_get: bool
70
71     .. admonition:: The cache will be used, if all following conditions are given
72
73         * The key is in the list returned by keys() method of the source_instance
74         * The key is not in the list of keys added by the add_source_get_keys() method
75
76         * The cache age is less then the given max_age parameter or the given max_age is
          None.
77
78         * The uid of the source instance (e.g. a checksum or unique id of the source) is
          identically to to uid stored in the cache.
79
80         * The data version of the source_instance is  $\leq$  the data version stored in the
          cache.
81
82         * The value is available in the previous stored information
83
84     Example:
85
86     .. literalinclude:: caching/_examples_/property_cache_pickle.py
87
88     Will result on the first execution to the following output (with a long execution time):
89
90     .. literalinclude:: caching/_examples_/property_cache_pickle_1.log
91
92     With every following execution the time consumption my by much smaller:
93
94     .. literalinclude:: caching/_examples_/property_cache_pickle_2.log
95     """
96     DATA_VERSION_TAG = '_property_cache_data_version_'
97     STORAGE_VERSION_TAG = '_storage_version_'
98     UID_TAG = '_property_cache_uid_'
99     DATA_TAG = '_data_'
100    AGE_TAG = '_age_'
101
102    #
103    STORAGE_VERSION = 1
104
105    def __init__(self, source_instance, cache_filename, load_all_on_init=False,
106                callback_on_data_storage=None, max_age=None, store_on_get=True):
107        self._source_instance = source_instance
108        self._cache_filename = cache_filename
109        self._load_all_on_init = load_all_on_init
110        self._callback_on_data_storage = callback_on_data_storage
111        self._max_age = max_age
112        self._store_on_get = store_on_get
113
114    #
```

Unittest for caching

```
108     self._source_get_keys = []
109     self._cached_props = None
110
111     def add_source_get_keys(self, keys):
112         """
113         This will add one or more keys to a list of keys which will always be provided by the `
114         source_instance` instead of the cache.
115
116         :param keys: The key or keys to be added
117         :type keys: list, tuple, str
118         """
119         if type(keys) in [list, tuple]:
120             self._source_get_keys.extend(keys)
121         else:
122             self._source_get_keys.append(keys)
123
124     def full_update(self, sleep_between_keys=0):
125         """
126         With the execution of this method, the complete source data which needs to be cached,
127         will be read from the source instance
128         and the resulting cache will be stored to the given file.
129
130         :param sleep_between_keys: Time to sleep between each source data generation
131         :type sleep_between_keys: float, int
132
133         .. hint:: Use this method, if you initialised the class with `store_on_get=False`
134         """
135         self._load_source(sleep_between_keys=sleep_between_keys)
136         self._save_cache()
137
138     def get(self, key, default=None):
139         """
140         Method to get the cached property. If the key does not exists in the cache or `
141         source_instance`, `default` will be returned.
142
143         :param key: key for value to get.
144         :param default: value to be returned, if key does not exists.
145         :returns: value for a given key or default value.
146         """
147         if key in self._source_instance.keys() and key not in self._source_get_keys:
148             if self._cached_props is None:
149                 self._init_cache()
150             if self._max_age is None:
151                 cache_old = False
152             else:
153                 cache_old = time.time() - self._cached_props[self.AGE_TAG].get(self._key_filter(
154                     key), 0) > self._max_age
155                 if cache_old:
156                     logger.debug("The cached value is old, cached value will be ignored")
157             if self._key_filter(key) not in self._cached_props[self.DATA_TAG] or cache_old:
158                 logger.debug("Loading property for key='%s' from source instance", key)
159                 val = self._source_instance.get(key, None)
160                 if self._store_on_get:
161                     tm = int(time.time())
162                     logger.debug("Adding key=%s, value=%s with timestamp=%d to chache", key, val,
163                                 tm)
164                     self._cached_props[self.DATA_TAG][self._key_filter(key)] = val
165                     self._cached_props[self.AGE_TAG][self._key_filter(key)] = tm
166                     self._save_cache()
167             else:
168                 return val
169         else:
170             return default
```

Unittest for caching

```
165         logger.debug("Providing property for '%s' from cache", key)
166         return self._cached_props[self.DATA_TAG].get(self._key_filter(key), default)
167     else:
168         if key not in self._source_instance.keys():
169             logger.debug("Key '%s' is not in cached.keys. Uncached data will be returned.",
170 key)
171         elif key in self._source_get_keys:
172             logger.debug("Key '%s' is excluded by .add_source_get_keys(). Uncached data will
173 be returned.", key)
174         return self._source_instance.get(key, default)
175
176 def _data_version(self):
177     if self._cached_props is None:
178         return None
179     else:
180         return self._cached_props.get(self.DATA_VERSION_TAG, None)
181
182 def _storage_version(self):
183     if self._cached_props is None:
184         return None
185     else:
186         return self._cached_props.get(self.STORAGE_VERSION_TAG, None)
187
188 def _init_cache(self):
189     load_cache = self._load_cache()
190     uid = self._source_instance.uid() != self._uid()
191     try:
192         data_version = self._source_instance.data_version() > self._data_version()
193     except TypeError:
194         data_version = True
195     try:
196         storage_version = self._storage_version() != self.STORAGE_VERSION
197     except TypeError:
198         storage_version = True
199
200     #
201     if not load_cache or uid or data_version or storage_version:
202         if load_cache:
203             if self._uid() is not None and uid:
204                 logger.debug("Source uid changed, ignoring previous cache data")
205             if self._data_version() is not None and data_version:
206                 logger.debug("Data version increased, ignoring previous cache data")
207             if storage_version:
208                 logger.debug("Storage version changed, ignoring previous cache data")
209         self._cached_props = {self.AGE_TAG: {}, self.DATA_TAG: {}}
210         if self._load_all_on_init:
211             self._load_source()
212         self._cached_props[self.UID_TAG] = self._source_instance.uid()
213         self._cached_props[self.DATA_VERSION_TAG] = self._source_instance.data_version()
214         self._cached_props[self.STORAGE_VERSION_TAG] = self.STORAGE_VERSION
215         if self._store_on_get:
216             self._save_cache()
217
218 def _load_cache(self):
219     if os.path.exists(self._cache_filename):
220         with open(self._cache_filename, 'rb') as fh:
221             self._cached_props = pickle.load(fh)
222         logger.debug('Loading properties from cache (%s)', self._cache_filename)
223         return True
224     else:
225         logger.debug('Cache file does not exists (yet).')
226         return False
```

Unittest for caching

```
225 def _key_filter(self, key):
226     return key
227
228 def _load_source(self, sleep_between_keys=0):
229     if self._cached_props is None:
230         self._init_cache()
231         logger.debug('Loading all data from source - %s', repr(self._source_instance.keys()))
232         for key in self._source_instance.keys():
233             if key not in self._source_get_keys:
234                 self._cached_props[self.DATA_TAG][self._key_filter(key)] = self._source_instance.
get(key)
235                 self._cached_props[self.AGE_TAG][self._key_filter(key)] = int(time.time())
236                 time.sleep(sleep_between_keys)
237
238 def _save_cache(self):
239     with open(self._cache_filename, 'wb') as fh:
240         pickle.dump(self._cached_props, fh)
241         logger.debug('cache-file stored (%s)', self._cache_filename)
242     if self._callback_on_data_storage is not None:
243         self._callback_on_data_storage(self)
244
245 def _uid(self):
246     if self._cached_props is None:
247         return None
248     else:
249         return self._cached_props.get(self.UID_TAG, None)
250
251
252 class property_cache_json(property_cache_pickle):
253     """
254     See also parent :py:class:`property_cache_pickle` for detailed information.
255
256     .. important::
257         * This class uses json. You should only use keys of type string!
258         * Unicode types are transfered to strings
259
260         See limitations of json.
261
262     Example:
263
264     .. literalinclude:: caching/_examples_/property_cache_json.py
265
266     Will result on the first execution to the following output (with a long execution time):
267
268     .. literalinclude:: caching/_examples_/property_cache_json_1.log
269
270     With every following execution the time cosumption my by much smaller:
271
272     .. literalinclude:: caching/_examples_/property_cache_json_2.log
273     """
274     def _load_cache(self):
275         if os.path.exists(self._cache_filename):
276             with open(self._cache_filename, 'r') as fh:
277                 self._cached_props = json.load(fh)
278                 logger.debug('Loading properties from cache (%s)', self._cache_filename)
279             return True
280         else:
281             logger.debug('Cache file does not exists (yet).')
282         return False
283
284     def _save_cache(self):
285         with open(self._cache_filename, 'w') as fh:
286             json.dump(self._cached_props, fh, sort_keys=True, indent=4)
287             logger.debug('cache-file stored (%s)', self._cache_filename)
288         if self._callback_on_data_storage is not None:
289             self._callback_on_data_storage(self)
```