

Requirement Specification for Module caching

August 12, 2025

Contents

1	SEC-0001: Cache generation (json /pickle)	2
1.1	REQ-0003: Data generation from source instance, if no cache is available	2
1.2	REQ-0001: Create complete cache from the given data instance	2
1.3	REQ-0005: Create cache partially from a given data instance by get method	2
1.4	REQ-0015: Ignore corrupt cache file	2
2	SEC-0002: Load spreading for full update	2
2.1	REQ-0004: Full update with delay between each data generation for the cache	2
2.2	REQ-0002: No cache generation if disabled	3
3	SEC-0003: Dump cache conditions	3
3.1	REQ-0006: Dump cache if time is expired	3
3.2	REQ-0007: Dump cache if data version increases	3
3.3	REQ-0008: Dump cache if data uid is changed	3
3.4	REQ-0009: Dump cache if storage version is changed	3
3.5	REQ-0014: Dump cache if stored value is 'None'	4
4	SEC-0004: Definition of uncached data	4
4.1	REQ-0010: Define uncached data	4
5	SEC-0005: Callback on data storage	4
5.1	REQ-0011: If no data is changed, no callback will be executed	4
5.2	REQ-0012: Callback execution in case of a full update	4
5.3	REQ-0013: Callback execution in case of get function	5

1 SEC-0001: Cache generation (json /pickle)

1.1 REQ-0003: Data generation from source instance, if no cache is available

If the cache is not available, the data shall be generated from the source instance.

<i>Reason</i>	There shall be the possibility to create the cache on demand, so the fallback is to generate the data from the source instance.
<i>Fitcriterion</i>	Caching is called without previous cache generation and the data from the source instance is completely available.

1.2 REQ-0001: Create complete cache from the given data instance

There shall be a method caching all information from the given instance.

<i>Reason</i>	Independent usage of data generation and data usage (e.g. the user requesting the data is not able to create the data).
<i>Fitcriterion</i>	Caching is called twice with different data instances and the cached data from the first call is completely available.

1.3 REQ-0005: Create cache partially from a given data instance by get method

On getting data from the cached instance, the information shall be stored in the cache file.

<i>Reason</i>	There shall be the possibility to create the cache on demand, so the fallback is to generate the data from the source instance.
<i>Fitcriterion</i>	Caching is called twice with different data instances and the cached data from the first call is available for all keys cached on the first run.

1.4 REQ-0015: Ignore corrupt cache file

Ignore corrupt cachefile, while loading cache.

<i>Reason</i>	Suppress exceptions while caching.
<i>Fitcriterion</i>	Loading cache results in no exception, when cache file is empty.

2 SEC-0002: Load spreading for full update

2.1 REQ-0004: Full update with delay between each data generation for the cache

The full update method shall pause for a given time between every cached item.

<i>Reason</i>	Load spreading in case of cyclic called <code>.full_update()</code> .
<i>Fitcriterion</i>	The time consumption of the method <code>.full_update(<sleep_time>)</code> shall consume n times the given <code>sleep_time</code> . Where n is the number of items which will be cached from the source instance.

2.2 REQ-0002: No cache generation if disabled

The cache shall be generated by the `.get()` method, only if the cache instance parameter `store_on_get` is set to `True`.

<i>Reason</i>	Independent usage of data generation and data usage (e.g. the user requesting the data is not able to create the data).
<i>Fitcriterion</i>	Create a caching instance with <code>store_on_get</code> set to <code>False</code> . Get every item of the source instance with the <code>.get()</code> method and check that no cache file exists.

3 SEC-0003: Dump cache conditions

3.1 REQ-0006: Dump cache if time is expired

Dump the cached item, if this item is older then the given expiry time.

<i>Reason</i>	Ensure, that the cache is updated from time to time. For example for items which do not change very often.
<i>Fitcriterion</i>	Create a cache instance, cache some data. Intialise a second caching instance with a different source instance and a expire time. Wait for longer than the given expiry time and check that the items from the second source instance are returned.

3.2 REQ-0007: Dump cache if data version increases

Dump the complete cache, if the *data version* of the source instance is increased.

<i>Reason</i>	The data version is part of the source instance. Increasing the data version indicates, that the source instance generates the data in another way or the structure of the data is changed. In that condition, the cache needs to be ignored.
<i>Fitcriterion</i>	Create a cached instance and cache some items. Generate a second cached instance with different source data and a increased data version. Ensure, that the cache instance returns the values from the second source. It is required to set <code>load_all_on_init</code> to <code>False</code> and <code>store_on_get</code> to <code>True</code> .

3.3 REQ-0008: Dump cache if data uid is changed

Dump the complete cache, if the *data uid* of the source instance is changed.

<i>Reason</i>	The data uid is part of the source instance. Changing the data uid indicates, that the source of the data created by the source instance is changed (e.g. the uid of a file or folder) and the cache needs to be ignored.
<i>Fitcriterion</i>	Create a cached instance and cache some items. Generate a second cached instance with different source data and a changed data uid. Ensure, that the cache instance returns the values from the second source. It is required to set <code>load_all_on_init</code> to <code>False</code> and <code>store_on_get</code> to <code>True</code> .

3.4 REQ-0009: Dump cache if storage version is changed

Dump the complete cache, if the *storage version* of the caching class is changed.

<i>Reason</i>	The storage version is part of the caching class. Changing the storage version indicates, that the previously stored cache is not compatible due to new data storage and the cache needs to be ignored.
<i>Fitcriterion</i>	Create a cached instance and cache some items. Generate a second cached instance with different source data and a changed storage version. Ensure, that the cache instance returns the values from the second source. It is required to set <code>load_all_on_init</code> to <code>False</code> and <code>store_on_get</code> to <code>True</code> .

3.5 REQ-0014: Dump cache if stored value is 'None'

Dump the cached item, if the stored value is `None`.

<i>Reason</i>	If no information is stored in the cache, the data shall be generated by the source instance.
<i>Fitcriterion</i>	Create a cached instance and cache some items. One needs to have <code>None</code> as value. Generate a second cached instance with different source data (especially, the previous item with value <code>None</code> needs to have a not <code>None</code> value. Ensure, that the caching instance returns not <code>None</code> from the second source.

4 SEC-0004: Definition of uncached data

4.1 REQ-0010: Define uncached data

It shall be possible to define items which are not cached.

<i>Reason</i>	If there is dynamic changed data in the source instance, it shall be possible to define these items as non cached to get them always from the source instance.
<i>Fitcriterion</i>	Create a cached instance and cache some items. Generate a second cached instance with different source data and set at least one item as source item. This item should be previously cached. Ensure, that the source item is the one from the second source instance.

5 SEC-0005: Callback on data storage

5.1 REQ-0011: If no data is changed, no callback will be executed

The store callback shall not be executed, if no cache is stored.

<i>Reason</i>	Do actions, if cache data is stored to disk.
<i>Fitcriterion</i>	Initialise the cache instance without storing cache data. Ensure, that the callback is never executed.

5.2 REQ-0012: Callback execution in case of a full update

The storage callback shall be called once on every `full_update()`.

<i>Reason</i>	Do actions, if cache data is stored to disk.
<i>Fitcriterion</i>	Initialise the cache instance and ensure, that the callback is executed as often as the <code>.full_update()</code> method is executed.

5.3 REQ-0013: Callback execution in case of get function

The storage callback, shall be called once on every `.get()`, if `storage_on_get` is set to `True`.

<i>Reason</i>	Do actions, if cache data is stored to disk.
<i>Fitterion</i>	Initialise the cache instance and ensure, that the callback is executed as often as the <code>.get()</code> method is executed.
