

Using the Memory Adapter with Maps

The Memory Adapter allows to store an entire hash map to memory. The hash map object as a whole can be managed with the Memory adapter the same way as any other objects. Additionally, the Memory adapter allows to add, change, retrieve, and remove single elements from a stored hash map. Therefore, the Memory adapter provides an additional parameter **hashMapKey**. If this parameter is provided, the Memory adapter does not access the entire map, but tries to access the specified element of the map.

In comparison to caching data in a database, this can significantly increase the performance if you e.g. want to cache some data you frequently need to lookup. Using hash maps is much faster than storing single values as you can store all values with one **store** operation call, and you can still access single values using **retrieve** with **hashMapKey** (see [Retrieving Data from the Map](#) below).

Storing a Map to Memory

Using the **store** operation from the Memory adapter, you can store data. Provide the following parameters:

Name	Type	Direction	Description
key	String	in	Set a key that can be used to access the stored map with the retrieve operation.
value	Any	in	Provide the complete map that should be stored to the memory.

If a map with the given key already exists in memory, it will be overwritten by the Memory adapter. In this case, parameter **oldValue** returns the the complete previously stored hash map.

Name	Type	Direction	Description
oldValue	Any	out	If the provided key is already present in the memory, it gets overwritten. In this case, oldValue returns the previous value, means the complete hash map. If no old value is present, oldValue is NULL.

Retrieving Data from the Map

You can retrieve the complete hash map as described on [Storing to and Retrieving Data from Memory](#). If you want to retrieve a dedicated hash map element, provide the following parameters:

Name	Type	Direction	Description
key	String	in	Provide the key of the map that has been stored to memory with store . If no value with this key can be found, the Memory adapter throws error MEMADSM / 9.
hashMapKey	String	in	Provide the key to the hash map element you want to retrieve.

The Memory adapter returns a single map element containing the data from the map. If there is no map element with the given key, the Memory adapter throws an error.

Name	Type	Direction	Description
value	Any	out	Returns the stored map entry.

Related Error Codes

Find a list of all persistent state error codes on [System Errors of the Memory Adapter](#).

Error Code	Description
MEMADSM / 9	The message is not stored here.

Adding or Changing Data in the Map

You can add or change data in the hash map that has been stored to memory. To do this, provide the following parameters:

On this Page:

- [Storing a Map to Memory](#)
- [Retrieving Data from the Map](#)
 - [Related Error Codes](#)
- [Adding or Changing Data in the Map](#)

MemoryAdapter_HashMap_Example



Click the icon to download a simple example model that shows the usage of the Memory adapter with hash maps in **Sc heer PAS Designer**.

Related Pages:

- [Storing to and Retrieving Data from Memory](#)
- [Removing Stored Data](#)
- [Memory Adapter Reference](#)
- [Map Operations](#)
- [System Errors](#)

Name	Type	Direction	Description
key	String	in	Set the key that has been used to store the map to memory.
value	Any	in	Provide the data that should be added to the hash map in memory.
hashMapKey	String	in	Provide the key of the hash map entry you want to add or change.

If there is **no map element with the given key**, the Memory adapter will add this element to the map.
If the **map element with the given key is already existing** in the map, it will be updated to the provided value. In this case, parameter **oldValue** returns the complete previously stored hash map.

Name	Type	Direction	Description
oldValue	Any	out	oldValue returns the complete previously stored hash map if you are changing a hash map entry. If the hash map entry has been added, oldValue is NULL.