

# Memory Adapter Reference



This page explains the **Memory Adapter** in Bridge context. If you were looking for the same information regarding the [PAS Designer](#), refer to [Memory Adapter](#) in the Designer guide.

## On this Page:

- [Tagged Values \(<<MemoryAdapter>>\)](#)
- [Parameters](#)
  - [store](#)
  - [retrieve](#)
  - [remove](#)
  - [clear](#)

## Related Pages:

- [Memory Adapter](#)
- [Using the Memory Adapter with Maps](#)

## Tagged Values (<<MemoryAdapter>>)

Attribute	Description	Allowed Values	
action	Supply the action you want to perform.	store	Store a value to memory.
		retrieve	Retrieve a value from memory.
		remove	Remove a value from memory.
		clear	Clear the memory.
scope	Supply the scope of the operation.	global	The stored data is accessible globally to the service, and available across service calls (default).
		session	The stored data is accessible within a service call.

## Parameters

### store

Store a value to memory.

Name	Type	Direction	Description
key	String	in	Set a key that can be used to access the stored value with the <a href="#">retrieve</a> operation.
value	Any	in	Provide the data that should be stored to the memory.
hashMapKey	String	in	In case the stored value is a hash map, you can use <b>hashMapKey</b> to provide the key of a hash map entry you want to override. Refer to <a href="#">Using the Memory Adapter with Maps</a> for more information on this.
oldValue	Any	out	If the provided key is already present in the memory, it gets overwritten. In this case, <b>oldValue</b> returns the previous value. If no old value is present, <b>oldValue</b> is NULL. In case <b>value</b> contains a hash map, <b>oldValue</b> returns the complete previously stored hash map.

### retrieve

Retrieve a value from memory.

Name	Type	Direction	Description
key	String	in	Provide the key of the value that has been stored to memory with <a href="#">store</a> . If no value with this key can be found, the Memory adapter throws error MEMADSM / 9.
hashMapKey	String	in	In case the stored value is a hash map, you can use <b>hashMapKey</b> to retrieve a dedicated value from the map. Refer to <a href="#">Using the Memory Adapter with Maps</a> for more information on this.
value	Any	out	Returns the either the stored value, or the map entry.

### remove

Remove a value from memory.

Name	Type	Direction	Description
<b>key</b>	String	in	Provide the key of the value that has been stored to memory with <a href="#">store</a> . If no value with this key can be found, the Memory adapter throws error MEMADSM / 12.
<b>hashMapKey</b>	String	in	In case the stored value is a hash map, you can use <b>hashMapKey</b> to provide the key of a hash map entry you want to remove. Refer to <a href="#">Using the Memory Adapter with Maps</a> for more information on this.
<b>oldValue</b>	Any	out	Returns the value of the memory item that has been removed. In case <b>value</b> contains a hash map, <b>oldValue</b> returns the complete previously stored hash map.

## clear

Clear all stored data. No parameters.