

Memory Adapter



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.

The memory add-on allows to store and retrieve values to respectively from the memory.

Example File (Builder project Add-ons/System):



```
<your example path>\Add-ons\System\uml\memory.xml  
<your example path>\Add-ons\System\uml\memoryInSessionScope.xml
```

On this Page:

- [Memory Scope](#)
- [Storing Data to Memory](#)
- [Retrieving Data From Memory](#)
 - [Related Error Codes](#)
- [Removing Stored Data](#)
 - [Related Error Codes](#)
- [Clearing the Memory](#)

Related Pages:

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

Memory Scope

By default, the memory is in scope of the service composite (server process), but it is also possible to store values in a request session context. Basically, the memory adapter is a thread-safe hash list allowing to

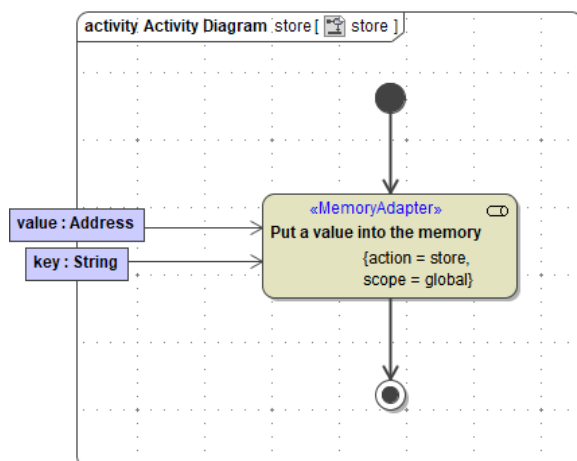
- **share objects among requests (service scope, also called global scope)**
To store objects in service scope and to share them between requests to the same service, set **scope** to **global** on the Memory adapter. In this case, you need to propagate the memory key between requests, or use a static key.
- **keep data for one request (session scope)**
To keep data within the request, set **scope** to **session** on the Memory adapter. At the end of the request session, the memory will be cleared.

Storing Data to Memory

To store data to memory, provide the Memory adapter with the following:

- the data in parameter **value**
- a **key**, that can be used to access the data in memory
- the Memory Adapter action **store**

Figure: Storing a Data to memory



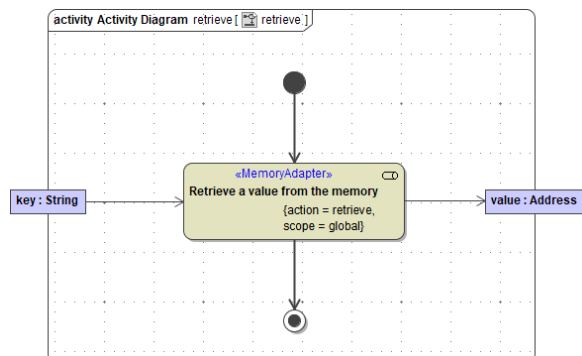
If an element with the given key already exists in memory, it will be overwritten by the Memory Adapter. In this case, parameter **oldValue** returns the value that got overwritten.

Retrieving Data From Memory

To retrieve data that has been stored to memory, provide the Memory adapter with the following:

- the **key** that has been used to store the data to memory
- the Memory Adapter action **retrieve**

Figure: Retrieving Data from Memory



If no data with the given key can be found, the Memory adapter throws an error.

Related Error Codes

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

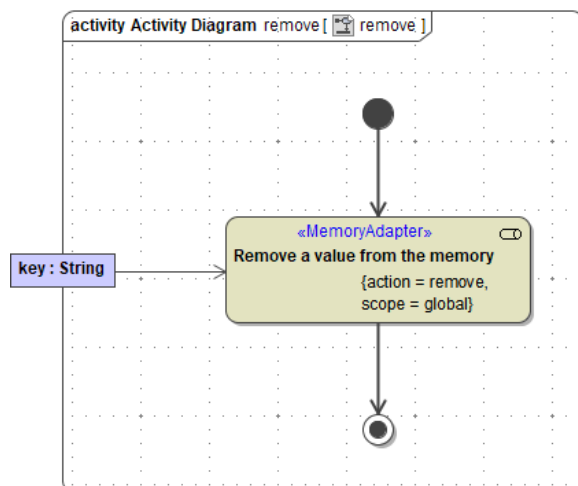
Error Code	Description
MEMADSM/9	The message is not stored here.

Removing Stored Data

To remove data from memory, provide the Memory adapter with the following:

- the **key** that has been used to store the data to memory
- the Memory Adapter action **remove**

Figure: Removing Data from Memory



Parameter **oldValue** returns the value that has been removed. If no data with the given key can be found, the Memory adapter throws an error (MEMADSM/12, see also [the list of system errors](#)).

Related Error Codes

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

Error Code	Description
MEMADSM/12	The message is not stored here.

Clearing the Memory

To clear the complete used memory, provide the Memory adapter with the following:

- the Memory Adapter action **clear**

Figure: Clearing the Memory

