Memory Adapter

The Memory adapter allows to store and retrieve values to respectively from the memory. You can find all data types and operations of the Memory adapter in the service panel at **Base Types/Bridge Base /Base Components/Add Ons/Memory**.

Using the Memory adapter, you can

Task	Adapter Operation	Description	Documentation Reference
Store data to memory	**store**	You can use the Memory adapter to store data to memory.	- Storing to and Retrieving Data from Memory - Using the Memory Adapter with Maps
Retrieve data from memory	**retrieve**	You can use the Memory adapter to retrieve data from memory.	- Storing to and Retrieving Data from Memory - Using the Memory Adapter with Maps
Remove data from memory	**remove**	You can remove data elements from memory that have been stored to a dedicated key.	- Removing Stored Data - Using the Memory Adapter with Maps
Clear the memory	**clear**	You can wipe the complete memory used by the service.	Removing Stored Data

MemoryAdapter_GuessingGame _Example
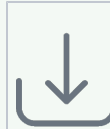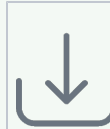
Click the icon to download a simple example model that shows the usage of the Memory adapter in **Scheer PAS** *Designer*.

Adding a Memory Adapter Operation to a Diagram

Expand the path to the Memory adapter in the service panel (**Base Types /Bridge Base /Base Components /Add Ons /Memory**).
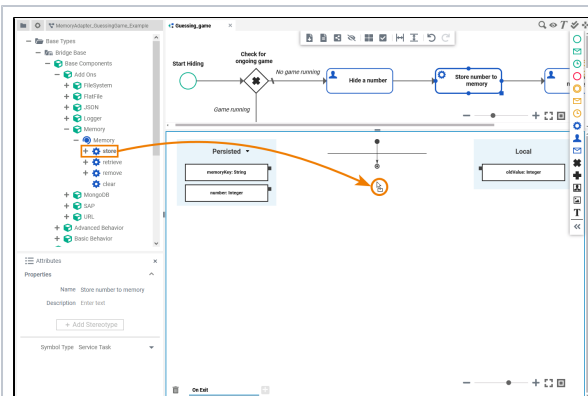
You can drag out operations from the data model to any diagram:

- [BPMN execution diagram](#)
- [mapping diagrams](#)
- [activity diagrams](#)

The example on the left shows how to add a Memory adapter operation to an BPMN execution diagram.

Configuring the Memory Adapter Operation

Once an operation has been added to a diagram, it needs to be configured as a Memory adapter.

Select the newly added Memory adapter operation and switch to the **Attributes** panel. Depending on the diagram type you can see the following information (example BPMN execution diagram):

Attribute	Description	Allowed Values / Example
Name	The name of the adapter operation.	store
Symbol Type	Operations added to a execution diagram are execution steps.	**Execution Step**

All this is predefined and cannot be changed.

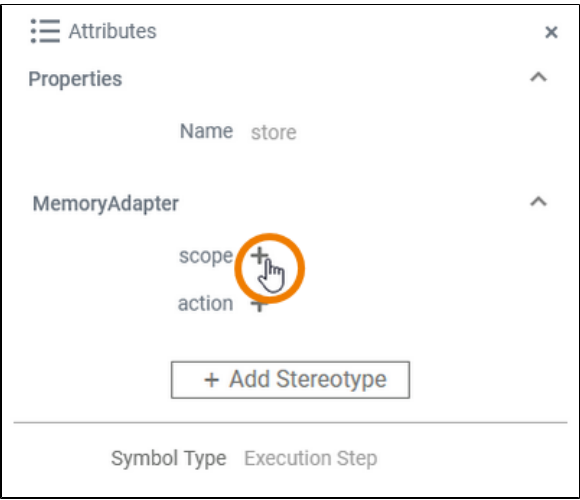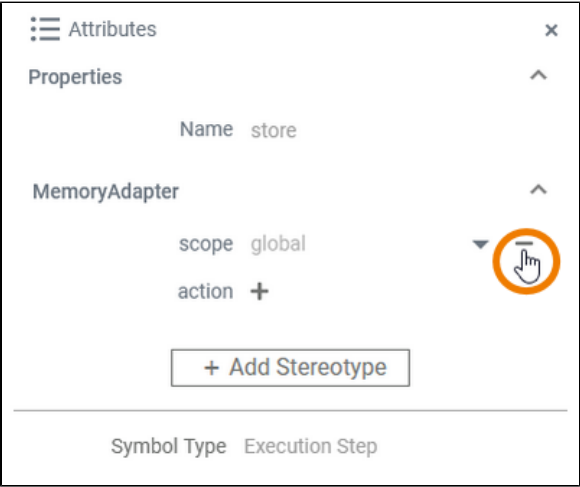|	
Attributes ☰ ✕ Properties ⌃ Name store [+ Add Stereotype] Symbol Type Execution Step	Click **Add Stereotype** to define the selected operation as to be a Memory adapter.
Select Stereotype ☐ File System Adapter ☐ Flat File Adapter ☐ Kafka Producer Adapter ☐ Logger ☑ Memory Adapter ☐ MongoDB Adapter ☐ REST Adapter ☐ S3 ☐ SAP IDoc Composer ☐ SAP IDoc Parser ☐ SAP IDoc Record Composer ☐ SAP IDoc Record Parser ☐ SAP RFC Adapter ☐ SAP TRFC Adapter ☐ SAP TRFC Confirm Transaction ☐ SAP TRFC Create Transaction ☐ SAP XML IDoc Composer ☐ SAP XML IDoc Parser ☐ SQL Adapter ☐ URL Adapter [Save] [Cancel]	Select **Memory Adapter** from the list of available adapter stereotypes. Click **Save**.
Attributes ☰ ✕ Properties ⌃ Name store MemoryAdapter ⌄ [+ Add Stereotype] Symbol Type Execution Step	The **Attributes** panel shows the added adapter stereotype. Now you still need to configure the adapter. Expand the stereotype by clicking the arrow on the right.

		To configure a Memory adapter, you have the following options:

Attributes   ✕
 	
Properties   ⌃
 	
        Name   store
 	
MemoryAdapter   ⌃
 	
       scope   ✛👆
       action   ✚
 	
     + Add Stereotype
 	
―――――――――――――――
      Symbol Type   Execution Step
		scope: The scope the value should be stored to. The default scope is **global**. Configure this value if you want to store to session scope. Refer to section Memory Scope below for more information on the implications of the memory scope. Refer to Memory Adapter Reference for further information. The adapter option **action** derives from the used operation. Do not configure this.

Attributes   ✕
 	
Properties   ⌃
 	
        Name   store
 	
MemoryAdapter   ⌃
 	
       scope   global   ▼   ➖👆
       action   ✚
 	
     + Add Stereotype
 	
―――――――――――――――
      Symbol Type   Execution Step
		Click the ➖ icon to remove unwanted configurations.

Memory Scope

By default, the memory is in scope of the service (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 between requests (service/global scope)**
  To store objects in global 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) only**
  To keep data for the request only, set **scope** to **session** on the Memory adapter. At the end of the request session, the memory will be cleared. Regarding the Designer, the session scope corresponds to the implementation of one execution diagram.

Refer to xUML Runtime Transaction Concepts for more information on what is regarded as a session by the Runtime in general, and about transaction handling.