

Map Operations

A **hash table** or **hash map** is a data structure that uses a hash function to map keys (e.g. a person's name), to their associated values (e.g. their address). Thus, a hash map implements an associative array. A hash function is used to transform the key into the index of an array element where the corresponding value is stored.

In many situations, the lookup of elements within hash tables turns out to be more efficient than within arrays - even if the hash map building time is taken into account. In a well-dimensioned hash table, the average lookup time is independent of the number of elements stored in the table.

The following operations are used to manipulate and access maps:

<code>buildMap()</code>	builds a map from an array.
<code>setMapValue()</code>	updates the value of a specific key.
<code>getMapValue()</code>	gets the value of a specific key.
<code>getMapEntries()</code>	gets all map entries to an array.
<code>removeMapValue()</code>	removes the entry corresponding to a specific key.


The following sections describe the above operations in more detail.

Using Maps with the Memory Adapter

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 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.

This can significantly increase the performance, if you e.g. want to cache some data you frequently need to lookup, in comparison to caching this data in a database.

Example File (Builder project E2E Action Language/Map):



<your example path>\E2E Action Language\map\uml\mapMemoryAdapter.xml

For more information on handling maps with the Memory Adapter, continue reading on [Using the Memory Adapter with Maps](#).

Related Pages:

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