

Key Value Store Feature

Odysseus provides a simply feature for storing, retrieving and removing of values from a key value store.

Out of the box, odysseus provides

- **MemoryStore**: A key value store that keep the values in memory only
- **FileStore**: A key value store that keeps the data in main memory and stores it additionally to disk
- With a plugin you could write to a [Redis-Store](#).

Creating a store

A store can be created with the [Odysseus Script](#) command: #CREATE_KV_STORE name type (PARAM_1_KEY=PARAM_1_VALUE,..., PARAM_N_KEY=PARAM_N_VALUE)

To create a Memory Store simply use:

```
#CREATE_KV_STORE test MemoryStore
```

Some stores have parameters, e.g. the FileStore (with the paramter filename):

```
#CREATE_KV_STORE test FileStore (filename=c:/temp/tmp.store)
```

For all stores, the normal behavior is to allow creation only if there exists no store with the desired name. But with the parameter "createOnlyIfNotExists", one can change that behavior:

```
#CREATE_KV_STORE test MemoryStore(createOnlyIfNotExists=false) /// default is true
```

Access a store

[MEP](#) functions can be used to read an write from the store. So this can be used in all operators that contain expressions or predicates (e.g. [Map operator](#) or [Select operator](#))

- **kvwrite**(String StoreName, String KeyName, Object Value): (Overwrites) a value in the give store and key
- **kvread**(String StoreName, String KeyName): reads the value from the store with the key
- **kvremove**(String StoreName, String KeyName): remove the value from the store

The following code block shows an example

```
#PARSER PQL
#RUNQUERY
timer = TIMER({
    period = 1000,
    source = 'timersource'
})

Map1 = MAP({EXPRESSIONS = ['time', 'kvwrite("test2", "Time", time)', 'kvread("test2", "Time)']}, timer)
```

You could also use an operator to write to an existing store:

StoreWriter

```
#PARSER PQL
#RUNQUERY
timer = TIMER({
    period = 1000,
    source = 'timersource'
})

out = StoreWriter({IdAttribute='time', store="test2"}, timer)
```

