MDA store functions

This site describes MEP functions available to use a ***Multi Dimensional Array (MDA) store***. Such a MDA is basically a grid, whose borders, and distances between borders can be defined. As an example, the map of a city can be overlayed by a MDA to get equal sized zones or cells.

Functions for MDA stores are the following:

MDADim(Double, Double, Integer)

This function creates a new dimension for any MDA store.

Arguments:

1. The lower border for the dimension
2. The upper border for the dimension
3. The number of borders for the dimension (incl. lower and upper)

The number of cells is than the number of borders minus 1.

Return value: List<Double>

All borders defining the dimension

Example:

0...100	100.01... 200

can be defined by

```
MDADim(0.0, 200.0, ToInteger(3))
```

returning

0	100	200

MDAInit(String, List<List<Double>>)

This function creates a new MDA store.

Arguments:

1. The name for the store
2. A list of the dimensions for the MDA store. Each entry is a list of borders (see MDADim)

Return value: null

Example:

```
MDAInit("MyStore", ToList(MDADim(0.0, 200.0, ToInteger(3))))
```

MDADrop(String)

This function drops an existing MDA store.

Argument: The name for the store

Return value: null

Example:

```
MDADrop("MyStore")
```

MDAIndex(String, Double)

This function retrieves the index of an ⚠️one-dimensional⚠️ MDA store for a given value. In other words, it answers the quesion in which cell the values lies.
Short-cut for MDAIndices with one dimension.

Arguments:

1. The name of the store
2. The value to check

Note that it must be an one-dimensional MDA

Return value: Integer

The cell in which cell the values lies, or -1, if the value is outside the boundaries.

Example:

Using the Nexmark sources:

```
MDAIndex("MyStore", price)
```

MDAIndices(String, List<Double>)

This function retrieves the index of multi-dimensional MDA store for a given value. In other words, it answers the quesion in which cell the values lies.

Arguments:

1. The name of the store
2. The multi-dimensional value to check

Return value: List<Integer>

The cell in which cell the values lies, or -1 for any dimension, if the value is outside the boundaries of that dimension.

Example:

Using the Nexmark sources:

```
MDAIndex("MyStore", ToList(price, ToDouble(initialbid)))
```

MDAAddDim(String, [Integer,] List<Double>)

This function adds a new dimension to an existing MDA store.

Arguments:

1. The name of the store
2. Optional: The index of the dimension. Leave it out to add the dimension as the last dimension
3. A list of borders (see MDADim)

Return value: null

Example:

```
MDAAddDim("MyStore", MDADim(0.0, 200.0, ToInteger(3)))
```

MDARemoveDim(String, Integer)

This function removes an existing dimension from an existing MDA store.

Arguments:

1. The name of the store
2. The index of the dimension to remove

Return value: null

Example:

```
MDARemoveDim("MyStore", ToInteger(0))
```

MDAExchangeDim(String, Integer, List<Double>)

This function replaces en existing dimension of an existing MDA store with a new dimension.

Arguments:

1. The name for the store
2. The index of the dimension to exchange
3. A list of new borders (see MDADim)

Return value: null

Example:

```
MDAExchangeDim("MyDim", ToInteger(0), MDADim(200.0, 400.0, ToInteger(5)))
```