TestGeneratorTransportHandler

TestGeneratorTransportHandler creates a stream of tuple. The parameters defines the timeintervals and composition of the datastream. A ProtocolHandler is not necessary. DataHandler should be Tuple
Some options are mutually exclusive.

The TransportHandler can create three types of streams.

1. a stream with identical time intervals
2. a stream with two different time intervals
3. a stream with tuples without EndTimestamp

To create a stream with identical time intervals only use **windowsize**. In addition the option **overlapping** can be set.
if overlapping is “not”(default), then the next elements starts when the previous end
if overlapping is “completly”, then all elements has identical start- and endtimestamps
if overlapping is “partially”, then starts the time interval of the next element, while the previous element is valid.
windowsize1 and **windowsize2** are ignored.

To create a stream with two different timeintervals use **windowsize1** and **windowsize2**. **windowsize**, **overlapping** and **infinite** are ignored.
Streams with different timeintervals are always overlapping. In addition the option **identicalstart** and **identicalend** can be set. Only one of them can be true. As the name says, when one of them is true, the start- or the endtimestamp of both elements are identical.

To create a stream with tuples without EndTimestamp set **infinite** *true.* When windowsize is set, infinite elements and elements with time interval are sent in alternation.
windowsize1 and **windowsize2** are ignored.

The options **adjacent**, **outoforder**, **num** and **delay** can be used with every type of streams

In some situations is the Endtimestamp identical with the Starttimestamp of another element. Then are both elements adjacent. When **adjacent** set to false, then is a gap between both elements created.

Schema

The Schema must contains one attribute with datatype “starttimestamp” and one attribute with datatype “endtimestamp”. These attributes contains the generated timestamps
The other attributes can be some datatypes of Odysseus
Each datatype has a generator for testdata (like MAX_VALUE, MIN_VALUE, etc.)
Supported datatypes: double, float, long, integer, short, byte, char, boolean, String, List

Options

windowsize(long) - Time Interval of all elements
overlapping(String) - “not”, “completely” or “partially”. **default: “not”**
windowsize1(long) - First time interval
windowsize2(long) - Second time interval
identicalstart(boolean) - Starttimestamp of both elements are identical. **default: false**
identicalend(boolean) - Endtimestamp of both elements are identical. **default: false**
infinite(boolean) - Elements do not have an Endtimestamp. **default: false**
num(Integer) - sets the number of elements in stream; if num = -1, then is the stream infinite
delay(long) - time between the elements
outoforder(boolean) - Every element in the stream is out of order. **default: false**
adjacent(boolean) - **default: true**

Example

Some example for different option combinations. The element on the top is the first element in the stream

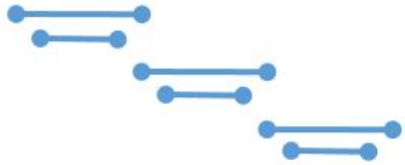
Options:
windowsize = 1000

Options:
windowsize = 1000
adjacent = false

Options:
windowsize = 1000
overlapping = “completely”

Options:
windowsize = 1000
overlapping = “partially”

Options:
windowsize1 = 1000
windowsize2 = 500



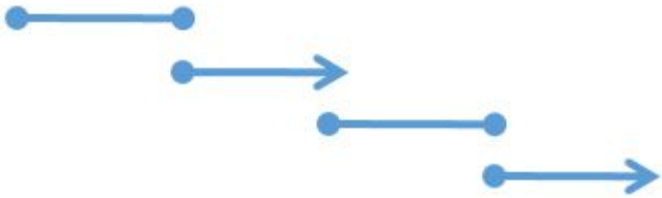
Options:
windowsize1 = 1000
windowsize2 = 500
identicalstart = true

Options:
windowsize1 = 1000
windowsize2 = 500
identicalend = true

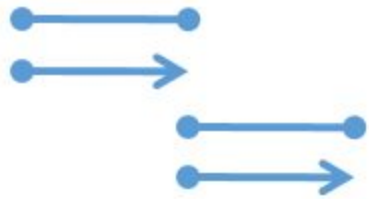
Options:
infinite = true

Options:
infinite = true
identicalstart = true

Options:
infinite = true
windowsize = 1000



Options:
infinite = true
windowsize = 1000
identicalstart = true



Options:
windowsize = 1000
outoforder = true