Odysseus Cheat Sheet

Full Grammar of PQL

QUERY = (STREAM | VIEW | SOURCE) +
STREAM = STREAM "::=" OPERATOR
VIEW = VIEWNAME "::=" OPERATOR
SOURCE = SOURCENAME "::=" OPERATOR
OPERATOR = QUERY | [OUTPUTPORT ":="] OPERATORTYPE
            "(" (PARAMETERLIST [":="] OPERATORLIST )")"
OPERATORLIST = [ OPERATOR (""," OPERATOR)+ ]
PARAMETERLIST = 
                "=" (PARAMETER (""," PARAMETER)* ""
PARAMETER = NAME "=" PARAMETERVALUE
PARAMETERVALUE = LONG | DOUBLE | STRING | PREDICATE |
LIST | MAP
MAP = "{" MAPENTRY ("," MAPENTRY*) ""
MAPENTRY = PARAMETERVALUE "=" PARAMETERVALUE
PREDICATE = PREDICATETYPE "(" STRING ")"

Advanced Operators

ASSOCIATIVESTORAGE
This operator stores streaming data in an associative storage

SIZES
–
INDEX
–

SUPPRESSPUNCTUATIONS
If set to true, no punctuations will be
delivered from this operator. Default is
false

VALUE
–
DEBUG
Flag, that this operator should be de-
bugged.

STORAGENAME
–
HIERARCHY
–

BUFFEREDFILTER
This operator can be used to reduce data rate. It buffers
incoming elements on port 0 (left) for bufferTime and
evaluates a predicate over the elements on port 1 (right). If
the predicate for the current element evaluates to true, all
elements from port 0 that are younger than
e.startTimeStamp()-bufferTime will be enriched with e and
delivered for deliverTime. Each time the predicate evaluates to
true, the deliverTime will be increased.

DELIVERTIME
–
PREDICATE
–
BUFFERTIME
–
SUPPRESSPUNCTUATIONS
If set to true, no punctuations will be
delivered from this operator. Default is
false

DEBUG
Flag, that this operator should be de-
bugged.

COALESCE
This Operator can be used to combine sequent elements, e.g.
by a set of grouping attributes or with a predicates. In the
attributes case, the elements are merged with also given
aggregations functions, as long as the grouping attributes (e.g.
a sensorid) are the same. When a new group is opened (e.g. a
measurement from a new sensor) the old aggregates values and
the grouping attributes are created as a result. In the
predicate case, the elements are merged as long as the
drainates evaluates to false, i.e. a new tuple is created when
the predicates evaluates to true.

DRAINATDONE
If set to true (default), elements are
not yet written will be written at
done.

FASTGROUPING
Use hash code instead of tuple com-
pare to create group. Potentially un-
safe!

OUTPUTPA
–
CREATEONHEARTBEAT
–
PREDICATE
–
ENDERGROUPOBJECT
DEBUG
Flag, that this operator should be de-
bugged.

HEARTBEATRATE
–
STARTPREDICATE
–
USEROUNDROBINALLOCATION

ATTR
–
NUMBEROFTHREADS
–
DUMPTAVALEUCOUNT
–
AGGREGATIONS
–
SUPPRESSPUNCTUATIONS
–
DRAINATCLOSE
If set to true (default is false), el-
ments are not yet written will be
written at close.

DRAIN
If set to true (default), elements are
not yet written will be written at
done.

MAXBUFFERSIZE
Defines the size of the buffers used
in multithreaded execution

CONVOLUTION
This operator applies a convolution filter, which is often used
in electronic signal processing or in image processing to clean
up wrong values like outliers. The idea behind the convolution
is to correct the current value by looking at its neighbours.
The number of neighbours is the size of the filter. If, for
example, SIZE=3, the filter uses the three values before the
current and three values after the current value to correct the
current value. Therefore, the filter does not deliver any results
for the first SIZE values, because it also needs additionally

SIZE further values after the current one!

ATTRIBUTES
–
GROUP_BY
–
SIZE
–

SUPPRESSPUNCTUATIONS
If set to true, no punctuations will be
delivered from this operator. Default is
false

OPTIONS
–
FUNCTION
–
DEBUG
Flag, that this operator should be de-
bugged.

FASTMEDIAN
Calculate the median for one attribute in the input tuples

PERCENTILES
–
GROUP_BY
–
ATTRIBUTE
–
ROUNDINGFACTOR
–
HISTOGRAM
–
NUMERICAL
–

SUPPRESSPUNCTUATIONS
If set to true, no punctuations will be
delivered from this operator. Default is
false

DEBUG
Flag, that this operator should be de-
bugged.

APPENDGLOBALMEDIAN
If a GROUP_BY element is given, the global media (i.e. median without re-
specting groups) will be annotated to each element.

GENERATOR

EXPRESSIONS
–
GROUP_BY
–
FREQUENCY
–
PREDICATE
–

SUPPRESSPUNCTUATIONS
If set to true, no punctuations will be
delivered from this operator. Default is
false

DEBUG
Flag, that this operator should be de-
bugged.

MULTI
–
TOPK

Calculate the top k elements of the input

**FASTGROUPING**
Use hash code instead of tuple compare to create group. Potentially unsafe!

**GROUP BY**
The number of elements to sort

**SCORINGFUNCTION**
The scoring function for ordering

**TRIGGERBYPUNCTUATION**
If set to true, output is only generated when punctuation arrives.

**DEBUG**
Flag, that this operator should be debuged.

**TIEWITHTIMESTAMP**
If two elements have the same score, this value can be used to define an order by time stamps. (Default is false)

**DESCENDING**
Sort descending (default is true)

**SUPPRESSDUPLICATES**
If set to true (default), output is only generated when a new top k set is available

**TUPLEAGGREGATE**

Select from all elements of a window on with the given method

**ATTRIBUTE**
Attribute on which the method is evaluated

**METHOD**
Method to use (MIN, MAX, LAST, FIRST)

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false

**DEBUG**
Flag, that this operator should be debuged.

**AGGREGATION**

Aggregations on attributes e.g Min, Max, Count, Avg, Sum and grouping.

**DRAINATDONE**
If set to true (default), elements are not yet written will be written at done.

**FASTGROUPING**
Use hash code instead of tuple compare to create group. Potentially unsafe!

**GROUP BY**

**OUTPUTPA**

**DEBUG**
Flag, that this operator should be debuged.

**DUPLICATEELIMINATION**

Removes duplicates (Depending on the time model!)

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false

**DEBUG**
Flag, that this operator should be debuged.

**EXISTENCE**

This operator tests an existence predicate and can be used with the type EXISTS (semi join) and NOT_EXISTS (anti semi join). The predicates can be evaluated against the element from the first input and the second input. Semi join: All elements in the first input for which there are elements in the second input that fulfills the predicate are sent. Semi anti join: All elements in the first input for which there is no element in the second input that fulfills the predicate are sent.

**PREDICATE**

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false

**DEBUG**
Flag, that this operator should be debuged.

**FILTER**

Filters elements of the input stream. If predicate evaluates to true, element will be sent to port 0 else to port 1.

**PREDICATE**

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false

**DEBUG**
Flag, that this operator should be debuged.

**HEARTBEATRATE**

**USEROUNDROBINALLOCATION**
Enables RoundRobin allocation. This is used in multithreaded execution for selecting the specific thread

**NUMBEROFTHREADS**
Use multiple threads for execution (only possible if grouping attributes are set)

**DUMPATVALUECOUNT**

**AGGREGATIONS**

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false

**DEBUG**
Flag, that this operator should be debuged.

**COMAND**

This operator sends commands to other operators.

**TARGETS**
Target operators where commands generated by this operator will be sent

**COMMAND**
The command to be executed. Each listener must be able to provide a command with this name. The command can also be specified by a Tuple/KeyValue-Element named "Command".

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false

**DEBUG**
Flag, that this operator should be debuged.

**UDO**

Calls a user defined operator

**ATTRIBUTES**

**INIT**

**CLASS**

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false

**DEBUG**
Flag, that this operator should be debuged.

**DISTINCT**

This operator removes duplicates.

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false

**DEBUG**
Flag, that this operator should be debuged.

**DEBUG**
Flag, that this operator should be debuged.
INTERSECTION

This operator calculates the intersection between two input sets.

**SUPPRESSPUNCTUATIONS** If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG** Flag, that this operator should be debugged.

JOIN

Operator to combine two datastreams based on the predicate

**ASSUREORDER** If set to false, the operator will not guarantee order in output. Default is true.

**PREDICATE** Predicate to filter combinations

**SweepArea** If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG** Flag, that this operator should be debugged.

**CARD** Type of input streams. For optimization purposes: ONE, ONE, MANY, MANY

LEFTJOIN

Left join: CURRENTLY NOT WORKING CORRECTLY.

**ASSUREORDER** If set to false, the operator will not guarantee order in output. Default is true.

**PREDICATE** Predicate to filter combinations

**SweepArea** Overwrite the sweep area

**SUPPRESSPUNCTUATIONS** If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG** Flag, that this operator should be debugged.

**CARD** Type of input streams. For optimization purposes: ONE, ONE, MANY, MANY

MAP

Performs a mapping of incoming attributes to out-coming attributes using map functions. Odysseus also provides a wide range of mapping functions. Hint: Map is stateless. To used

**MAP**

**ALLOWNULL** If set to true and an error occurs in calculation a null value is added to the element. Else the element is skipped and no output is produced.

**THREADS** Number of threads used to calculate the result.

**EVALUATEONPUNCTUATION** If set to true, map will also create an output (with the last read element) when it receives a punctuation.

**KVEXPRESSIONS** A list of expressions for use with key value objects.

**SUPPRESSERRORS** If set to true calculation errors will not appear in log or console. Could be helpful in scenarios where null values are allowed.

**DEBUG** Flag, that this operator should be debugged.

**EXPRESSIONS** A list of expressions.

**REMOVEATTRIBUTES** A list of attributes to remove. Only for use with key value objects.

**KEEPALLATTRIBUTES** Only for use with key value objects. If set to true, map will keep all attributes - even if not mentioned in kvexpressions.

**SUPPRESSPUNCTUATIONS** If set to true, no punctuations will be delivered from this operator. Default is false.

MERGE

Merge different input streams into one stream with ”first comes first served” semantics.

**SUPPRESSPUNCTUATIONS** If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG** Flag, that this operator should be debugged.

PROJECT

Make a projection on the input object (i.e. filter attributes)

**ATTRIBUTES** A list of attributes that should be used.

**PATHS** A list of attribute to use with keyvalue objects

**SUPPRESSPUNCTUATIONS** If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG** Flag, that this operator should be debugged.

STATEMAP

Performs a mapping of incoming attributes to out-coming attributes using map functions. Odysseus also provides a wide range of mapping functions. Hint: StateMap can use history

**STATEMAP**

**ALLOWNULL** If set to true and an error occurs in calculation a null value is added to the element. Else the element is skipped and no output is produced.

**THREADS** Number of threads used to calculate the result.

**EVALUATEONPUNCTUATION** If set to true, map will also create an output (with the last read element) when it receives a punctuation.

**KVEXPRESSIONS** A list of expressions for use with key value objects.

**SUPPRESSERRORS** If set to true calculation errors will not appear in log or console. Could be helpful in scenarios where null values are allowed.

**DEBUG** Flag, that this operator should be debugged.

**EXPRESSIONS** A list of expressions.

**REMOVEATTRIBUTES** A list of attributes to remove. Only for use with key value objects.

**KEEPALLATTRIBUTES** Only for use with key value objects. If set to true, map will keep all attributes - even if not mentioned in kvexpressions.

**SUPPRESSPUNCTUATIONS** If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG** Flag, that this operator should be debugged.

**EXPR** A list of expressions for use with key value objects.

**PROJECT**

Make a projection on the input object (i.e. filter attributes)

**ATTRIBUTES** A list of attributes that should be used.

**PATHS** A list of attribute to use with keyvalue objects

**SUPPRESSPUNCTUATIONS** If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG** Flag, that this operator should be debugged.

**RENAME**

Names the attributes

**ISNOOP** A flag to avoid removing this operator even if nothing in the schema is changed.

**NOOP** A flag to avoid removing this operator even if nothing in the schema is changed.

**ALIASES** The new list of attributes. Must be exactly the same length as in the input schema.

**SUPPRESSPUNCTUATIONS** If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG** Flag, that this operator should be debugged.

**TYPE**

The new type name of the output schema.

**PAIRS**

If set to true, aliases will be interpreted as pairs oldAttribute, new Attribute.

**SELECT**

The select operator filters the incoming data stream according to the given predicate.

**PREDICATE**

**SUPPRESSPUNCTUATIONS** If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG** Flag, that this operator should be debugged.

**HEARTBEATRATE**

**SETSYSTEMTIME**

The SetSystemTime operator sets the system time to the timestamp of incoming elements when the difference is too big.

**THRESHOLD** Max allowed difference between system time and element time stamp before system time is set.

**SUPPRESSPUNCTUATIONS** If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG** Flag, that this operator should be debugged.

**SORT**

The sort operator filters the incoming data stream according to a schema.

**ATTRIBUTES** A list of attributes that should be used.

**ASCENDING** The sort of each attribute

**SUPPRESSPUNCTUATIONS** If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG** Flag, that this operator should be debugged.
This Operator can be used to update the timestamp information in the meta data part. Be careful because this may lead undefined semantics.

**DATEFORMAT**
If using a string for date information, use this format to parse the date (in Java syntax).

**MONTH**
The name of the attribute for the month part of the start timestamp for application time

**HOUR**
The name of the attribute for the hour part of the start timestamp for application time

**FACTOR**
A multiplication factor for a single attributed timestamp to calc milliseconds (e.g. if input is seconds, use 1000 here)

**DEBUG**
Flag, that this operator should be debuged.

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false

**DRAINATCLOSE**
If set to true (default is false), this buffer will be emptied when calling close. Remark: Could lead to longer termination time!

**DEBUG**
Flag, that this operator should be debuged.

**Benchmark Operators**

**CALCLATENCY**
Odysseus has some features to measure the latency of single stream elements. This latency information is modeled as an interval. An operator in Odysseus can modify the start point of this interval. This operator sets the endpoint and determines the place in the query plan, where the latency measurement finds place. There can be multiple operators in the plan, to measure latency at different places.

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false

**DEBUG**
Flag, that this operator should be debuged.

**LATENCYTOPAYLOAD (Deprecated)**
Deprecated: You Latency.start, Latency.end, Latency.latency etc. directly as attributes! Adds attributes with the current latency information (start, end, latency, max_start, max_latency) to each tuple.

**SMALL**

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false

**DEBUG**
Flag, that this operator should be debuged.

**Enrich Operators**

**ENRICH**
This operator enriches tuples with data that is cached, e.g. to enrich a stream with a list of categories. The first input stream, therefore, should be only stream limited data to avoid buffer overflows. The second input is the data stream that should be enriched.

**MINIMUMSIZE**
Blocks all until there are at least minimumSize elements in the cache

**PREDICATE**
Predicate to filter combinations

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false

**DEBUG**
Flag, that this operator should be debuged.
Order Operators

ASSUREORDER

Operator which ensures the order of tuples

SUPPRESSPUNCTUATIONS If set to true, no punctuations will be delivered from this operator. Default is false

DEBUG Flag, that this operator should be debuged.

Pattern Operators

CHANGECORRELATE

This operator can reduce traffic. It lets an event pass if its different than the last event, if specified, numeric values can have a tolerance band (relative or absolute defined) e.i. only if the new values lies outside this band, it is send (aka known as deadband or histerese band)

GROUP BY

SUPPRESSCOUNTATTRIBUTE

USEBASEVALUE If this is set to true, the actual value is compared to the base value instead to the last value. Default is false. Does not work with 'useWindow'.

TOLERANCE

DEBUG Flag, that this operator should be debuged.

SENDLASTOFSAMEOBJECTS If set to false (default), in a group of same objects, the first is send. If set to true, the last one is send.

HEARTBEATRATE –

ATTR

BASEVALUE If 'useBaseValue' is true, the actual value is compared to the base value instead to the last value.

DELIVERFIRSTELEMENT

USEWINDOW If this is set to true, the operator compares not to the last value (or base value), but instead to the elements in the window. Therefore the difference to the minimum and maximum value to the new value is calculated. Default is false.

SUPPRESSPUNCTUATIONS If set to true, no punctuations will be delivered from this operator. Default is false

RELATIVETOLERANCE –

SASE

This operator can parse a query in SASE+ syntax.

SCHEMA

QUERY

SUPPRESSPUNCTUATIONS If set to true, no punctuations will be delivered from this operator. Default is false

ONEMATCHPERINSTANCE –

DEBUG Flag, that this operator should be debuged.

TYPE

HEARTBEATRATE –

Plan Operators

PLANMODIFICATIONACTION

Executes plan modifications based on receiving tuple data

QUERYIDATTRIBUTE Attribute to read the query id to execute the commands on

SUPPRESSPUNCTUATIONS If set to true, no punctuations will be delivered from this operator. Default is false

DEBUG Flag, that this operator should be debuged.

COMMANDATTRIBUTE Attribute to read the plan modification commands

Processing Operators

ADWIN (Deprecated)

Change detection window operator.

ATTRIBUTE –

DELTA –

SUPPRESSPUNCTUATIONS If set to true, no punctuations will be delivered from this operator. Default is false

DEBUG Flag, that this operator should be debuged.

ASSUREORDER

Operator which ensures the order of tuples

SUPPRESSPUNCTUATIONS If set to true, no punctuations will be delivered from this operator. Default is false

DEBUG Flag, that this operator should be debuged.

ASSUREHEARTBEAT

This operator assures that every n time elements there will be a heartbeat on the garantees, that no element (heartbeat or streamobject) is send, that is older than the last send hearbeat (i.e. the generated heartbeats are in order and indicate time progress). Heartbeats can be send periodically (sendAlwaysHeartbeats = true) or only if no other stream elements indicate time progress (e.g. in out of order scenarios)
independent if a new element has been received or not.

ALLOWOUTOFORDER –
SENDALWAYSAFTERTIME –
STARTATCURRENTTIME –
REALTIMEDELAY –
SUPPRESSPUNCTUATIONS If set to true, no punctuations will be delivered from this operator. Default is false
STARTAFTERFIRSTELEMENT –
DEBUG Flag, that this operator should be debuged.

BLOOMFILTER
Filter incoming streams using a Bloom filter
ATTRIBUTES –
FPP –
SUPPRESSPUNCTUATIONS If set to true, no punctuations will be delivered from this operator. Default is false
DEBUG Flag, that this operator should be debuged.
INSERTIONS –

BUFFER
Typically, Odysseus provides a buffer placement strategy to place buffers in the query plan. This operator allows adding buffers by hand. Buffers receives data stream elements and stores them in an internal element buffer. The scheduler stops the execution here for now. Later, the scheduler resumes to execution (e.g. with an another thread).
THREADED If set to true, this buffer will not be scheduled by the scheduler, but uses an own thread. Handle with care!
SUPPRESSPUNCTUATIONS If set to true, no punctuations will be delivered from this operator. Default is false
DRAINATCLOSE If set to true (default is false), this buffer will be emptied when calling close. Remark: Could lead to longer termination time!
MAXBUFFERSIZE –
DEBUG Flag, that this operator should be debuged.
TYPE –

CACHE
This operator can can some stream elements. At runtime, every time a new operator is connected it will get the cached elements. This can be usefull when reading from a csv file and multiple parts of a query need this information.
MAXELEMENTS –
SUPPRESSPUNCTUATIONS If set to true, no punctuations will be delivered from this operator. Default is false
DEBUG Flag, that this operator should be debuged.

COMBINE
Takes values of attributes from the input operators and combines them in one tuple
SUPPRESSPUNCTUATIONS If set to true, no punctuations will be delivered from this operator. Default is false
BUFFERNEWINPUTELEMENTS If WaitForAllChanged is set, specifies, if new Input should be buffered or overrides older Input that hast not been transfered yet
DEBUG Flag, that this operator should be debuged.
WAITFORALLCHANGED If true, there is only output when there has been input on all ports

CONSISTENTHASHFRAGMENT
Can be used to fragment incoming streams
ATTRIBUTES –
SUPPRESSPUNCTUATIONS If set to true, no punctuations will be delivered from this operator. Default is false
DEBUG Flag, that this operator should be debuged.
HEARTBEATREATE Send heartbeats to all other ports. Default is 10

CREATENEWFILENAMEPUNCTUATION
Depending on a predicate and a name: Create NewFilenamePunctuations
PREDICATE If expression evaluates to true, a NewFilenamePunctuations is created from the filename attribute value
SUPPRESSPUNCTUATIONS If set to true, no punctuations will be delivered from this operator. Default is false
FILENAME The expression to create the output filename
DEBUG Flag, that this operator should be debuged.

CREATENEWFILENAMEPUNCTUATION
Depending on a predicate and a name: Create NewFilenamePunctuations
PREDICATE If expression evaluates to true, a NewFilenamePunctuations is created from the filename attribute value
SUPPRESSPUNCTUATIONS If set to true, no punctuations will be delivered from this operator. Default is false
FILENAME The expression to create the output filename
DEBUG Flag, that this operator should be debuged.

HASHFRAGMENT
Can be used to fragment incoming streams
ATTRIBUTES –
OPTIMIZEDISTRIBUTION –
FRAGMENTS –
SUPPRESSPUNCTUATIONS If set to true, no punctuations will be delivered from this operator. Default is false
DEBUG Flag, that this operator should be debuged.
HEARTBEATREATE Send heartbeats to all other ports. Default is 10

LISTFRAGMENT
Can be used to fragment incoming streams
SUPPRESSPUNCTUATIONS If set to true, no punctuations will be delivered from this operator. Default is false
DEBUG Flag, that this operator should be debuged.

METADATA
Change the current meta data
SUPPRESSPUNCTUATIONS If set to true, no punctuations will be delivered from this operator. Default is false
METAATTRIBUTE This overwrites the current meta data. Existing values will not be overwritten.
DEBUG Flag, that this operator should be debuged.

RANGEFRAGMENT
Can be used to fragment incoming streams
FRAGMENTS –
RANGES –
ATTRIBUTE –
SUPPRESSPUNCTUATIONS If set to true, no punctuations will be delivered from this operator. Default is false
DEBUG Flag, that this operator should be debuged.
HEARTBEATREATE Send heartbeats to all other ports. Default is 10

REPLICATIONMERGE
Merge input from semantically equal queries
SUPPRESSPUNCTUATIONS If set to true, no punctuations will be delivered from this operator. Default is false
DEBUG Flag, that this operator should be debuged.

ROUTE
This operator can be used to route the elements in the stream to different further processing operators, depending on the
predicate.

**SENDINGHEARTBEATS**
If an element is routed to an output, heartbeats will be send to all other outputs.

**PREDICATES** –
Evaluate all (true) or only until first true predicate (false), i.e. deliver to all ports where predicate is true or only to first.

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG**
Flag, that this operator should be debugged.

### RRFRAGMENT

Can be used to fragment incoming streams

**FRAGMENTS**
**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG**
Flag, that this operator should be debugged.

**HEARTBEATREATE**
Send heartbeats to all other ports. Default is 10

### REPLACEMENT

This operator can be used if a value is expected but was not delivered timely. Different methods to determine the missing value are available.

**INTERVAL**
Size of the intervals

**REPLACEMENTMETHOD**
The replacement method for missing value.

**VALUEATTRIBUTE**
The attribute with the value attribute.

**TIMESTAMPATTRIBUTE**
The attribute with the timestamp attribute that should be updated.

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false

**DEBUG**
Flag, that this operator should be debugged.

**QUALITYATTRIBUTE**
The attribute with the quality attribute that should be updated.

### SAMPLE

This operator can reduce load by throwing away tuples.

**SAMPLERATE** –

**TIMEVALUE** –

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG**
Flag, that this operator should be debugged.

### SHUFFLEFRAGMENT

Can be used to fragment incoming streams

**FRAGMENTS**

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG**
Flag, that this operator should be debugged.

**HEARTBEATREATE**
Send heartbeats to all other ports. Default is 10

### SYNCWITHSYSTEMTIME

This operator tries to delay elements so that they are not faster than realtime.

**APPLICATIONTIMEFACTOR**
Factor to calculate milliseconds from application time

**APPLICATIONTIMEUNIT**
Unit of application timestamps

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG**
Flag, that this operator should be debugged.

### TIMESHIFT

Shifts the timestamp(s) a given time

**SHIFT**

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG**
Flag, that this operator should be debugged.

### TEMPORDERVERVALDATE

Assure that all elements are ordered by start timestamp and eliminate out of order elements.

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG**
Flag, that this operator should be debugged.

### Beneficial Operators

#### DISTRIBUTION (Deprecated)

Assign a distribution to the given attributes

**ATTRIBUTES**
The attributes holding the expected value.

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false.

**VARIANCE**
The attribute holding the variance of the distribution.

**CONTINUOUS**
The distribution is continuous or discrete.

**DEBUG**
Flag, that this operator should be debugged.

#### EM (Deprecated)

Estimate the distribution of the given attributes using a Gaussian mixture model

**MIXTURES**
The number of mixture components.

**ATTRIBUTES**
The attributes to fit a distribution to

**THRESHOLD**
The threshold for the loglikelihood to terminate the fitting process (default: 10E-5).

**ITERATIONS**
The number of iterations (default: 1000).

**PREDICATE**
The predicate to run a new fitting process.

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG**
Flag, that this operator should be debugged.

**INCREMENTAL**
Reuse the existing model in each fitting process.

#### EXISTENCETOPAYLOAD (Deprecated)

The input object gets one new field with tuple existence.

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG**
Flag, that this operator should be debugged.

#### KALMAN (Deprecated)

Kalman filter operator

**ATTRIBUTES**

**PROCESSNOISE**

**DEBUG**
Flag, that this operator should be debugged.

**INITIALERROR**

**TRANSITION**

**MEASUREMENTNOISE**

**VARIABLES**

**MEASUREMENT**

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false.

**INCREMENTAL**

**INITIALSTATE**

**CONTROL**

#### KDE (Deprecated)

Estimate the distribution of the given attributes using a Gaussian mixture model

**ATTRIBUTES**
The attributes to fit a distribution to

**SUPPRESSPUNCTUATIONS**
If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG**
Flag, that this operator should be debugged.
**Set Operators**

**DIFFERENCE**
This operator calculates the difference between two input sets.

**EXISTENCE**
This operator tests an existence predicate and can be used with the type EXISTS (semi join) and NOT_EXISTS (anti semi join). The predicates can be evaluated against the element from the first input and the second input. Semi join: All elements in the first input for which there are elements in the second input that fulfill the predicate are sent. Anti semi join: All elements in the first input for which there are elements in the second input that fulfill the predicate are sent.

**UNION**
Merges different input streams. (Typically preserves input order. Depending on the processing model)

**CSV Operators**

**CSVFILESINK**
Allows to write tp a csv based file

**CONSOLESINK**
Print input to standard out.

**FILESINK (Deprecated)**
The operator can be used to dump the results of an operator to a file.

**Sink Operators**

**GROUPSPLITFILEWRITER**
GroupSplitFileWriter

**SENDER**
This operator can be used to publish processing results to multiple endpoints using different transport and application protocols.

**SINK**
Represents a view for s sink.
CSVFILESOURCE

Allows to read input from a csv based file

**DATEFORMAT**

The date format used.

**SCHEMA2**

The output schema for port 2.

**SCHEMA1**

The output schema for port 1.

**SCHEMA3**

The output schema for port 3.

**TEXTDELIMITER**

Delimiter for Strings. No default.

**TRIM**

If set to true, for each element leading and trailing whitespaces are removed. Default false.

**OPTIONS**

Additional options.

**METAATTRIBUTE**

If set, this value overwrites the meta data created from this source.

**DEBUG**

Flag, that this operator should be debuged.

**FILENAME**

If set, this value overwrites the meta data created from this source.

**MAXTIMETOWAITFORNEWEVENTMS**

For access. Max time to wait for a new element before calling done. Typically used when the input stream has an end

**READMETADATA**

If the source provides meta data, use this flag to enable reading of meta data.

**SCHEMA**

The output schema.

**INPUTSCHEMA**

A list of data types describing the input format. Must be compatible with output schema!

**SOURCE**

The name of the sourcetype to create.

**SUPPRESSPUNCTUATIONS**

If set to true, no punctuations will be delivered from this operator. Default is false

**REQUEST**

Flag, that this operator should be debuged.

**RECEIVE**

Generic operator to connect to an input that sends data (i.e. pushed from source).

**DATEFORMAT**

The date format used.

**SCHEMA2**

The output schema for port 2.

**SCHEMA1**

The output schema for port 1.

**SCHEMA3**

The output schema for port 3.

**PROTOCOL**

The name of the protocol handler to use, e.g. Csv or SizeByteBuffer.

**OPTIONS**

Additional options.

**METAATTRIBUTE**

If set, this value overwrites the meta data created from this source.

**DEBUG**

Flag, that this operator should be debuged.

**TRANSPORT**

The name of the transport handler to use, e.g. File or TcpServer.

**NAF**

Enable or disable new access framework

**MAXTIMETOWAITFORNEWEVENTMS**

For access. Max time to wait for a new element before calling done. Typically used when the input stream has an end

**READMETADATA**

If the source provides meta data, use this flag to enable reading of meta data.

**SCHEMA**

The output schema.

**INPUTSCHEMA**

A list of data types describing the input format. Must be compatible with output schema!

**SOURCE**

The name of the sourcetype to create.

**SUPPRESSPUNCTUATIONS**

If set to true, no punctuations will be delivered from this operator. Default is false

**REQUEST**

Flag, that this operator should be debuged.

**QUERYSOURCE**

Attach a named query as source

**OPERATOR**

The name of the query that should deliver data or a tuple with queryname and operatorname

**PORT**

The number of the output port of the operator in the query that should be connect to.

**SUPPRESSPUNCTUATIONS**

If set to true, no punctuations will be delivered from this operator. Default is false

**REQUEST**

Flag, that this operator should be debuged.

**READMETADATA**

If the source provides meta data, use this flag to enable reading of meta data.

**SCHEMA**

The output schema.

**INPUTSCHEMA**

A list of data types describing the input format. Must be compatible with output schema!

**SOURCE**

The name of the sourcetype to create.

**SUPPRESSPUNCTUATIONS**

If set to true, no punctuations will be delivered from this operator. Default is false

**REQUEST**

Flag, that this operator should be debuged.
Generic operator to connect to an input which input must be retrieved (i.e. pulled from source).

**DATEFORMAT**
- The date format used.

**SCHEMA2**
- The output schema for port 2.

**SCHEMA1**
- The output schema for port 1.

**SCHEMA**
- The output schema for port 3.

**PROTOCOL**
- The name of the protocol handler to use, e.g. Csv or SizeByteBuffer.

**OPTIONS**
- Additional options.
- If set, this value overwrites the meta data created from this source.

**DEBUG**
- Flag, that this operator should be debuged.

**NAF**
- Enable or disable new access framework.

**TRANSPORT**
- The name of the transport handler to use, e.g. File or TcpServer.

**MAXTIMETOWAITFORNEWEVENTMS**
- For access. Max time to wait for a new element before calling done. Typically used when the input stream has an end.
- If the source provides meta data, use this flag to enable reading of meta data.

**READMETADATA**
- Enable or disable new access framework.

**SCHEMA**
- The output schema.

**INPUTSCHEMA**
- A list of data types describing the input format. Must be compatible with output schema!

**SOURCE**
- The name of the sourcetype to create.

**SUPPRESSPUNCTUATIONS**
- If set to true, no punctuations will be delivered from this operator. Default is false.

**DATABATCH**
- The name of the datahandler to use, e.g. Tuple or Document.

**STREAM**

**Integrate a view.**

**SOURCE**
- The output schema.

**SCHEMA**
- The name of the protocol handler to use, e.g. Csv or SizeByteBuffer.

**DEBUG**
- Flag, that this operator should be debuged.

**SUPPRESSPUNCTUATIONS**
- If set to true, no punctuations will be delivered from this operator. Default is false.

**DATAHANDLE**
- The name of the transport handler to use, e.g. File or TcpServer.

**DEBUG**
- Flag, that this operator should be debuged.

**SUPPRESSPUNCTUATIONS**
- If set to true, no punctuations will be delivered from this operator. Default is false.

**INPUTDATAHANDLE**
- Datahandler to use as input (e.g. format delievered from preceeding operator).

**OUTPUTDATAHANDLE**
- Datahandler to use for creation of elements.

**KEYVALUETOTUPLE**
- Translates a key-value/json object to a tuple

**SCHEMA**
- The output schema of this operator

**SUPPRESSPUNCTUATIONS**
- If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG**
- Flag, that this operator should be debuged.

**KEEPINPUT**
- –

**TYPE**
- –

**TIMESTAMPTOPAYLOAD (Deprecated)**
- Depracated: Use Map and TimeInterval.Start and TimeInterval.End directly. This operator is needed before data is send to another system (e.g. via a socket sink) to keep the time meta information (i.e. start and end time stamp). The input object gets two new fields with start and end timestamp. If this output is read again by (another) Odysseus instance, the following needs to be attached to the schema: ['Start', 'StartTimestamp'], ['End', 'EndTimestamp']

**ATTRIBUTES**
- Names of the attributes for the start and endtimestamp (default meta_validStart and meta_validEnd).

**SUPPRESSPUNCTUATIONS**
- If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG**
- Flag, that this operator should be debuged.

**TUPLETOKEYVALUE**
- Converts a tuple to a key-value/JSON object

**SUPPRESSPUNCTUATIONS**
- If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG**
- Flag, that this operator should be debuged.
The UnNest operator unpacks incoming tuple with a multi value attribute to create multiple tuples.

**RECALCULATE**

**ATTRIBUTE**

**SUPPRESSPUNCTUATIONS** If set to true, no punctuations will be delivered from this operator. Default is false.

**DEBUG** Flag, that this operator should be debugged.

### Aggregates

- **AMEN**
- **AMENI2**
- **AVG**
- **COMPLETEEN**
- **COVA**
- **COV**
- **FIRST**
- **LAST**

### Functions

**Bit**

- `adler(BitVector) → BitVector`
- `crc(BitVector) → BitVector`
- `subset(BitVector, Integer, Integer) → BitVector`
- `toBinary(UnsignedInt16) → BitVector`
- `toBinary(Byte) → BitVector`
- `toBinary(String) → BitVector`
- `toLong(BitVector) → Long`

**Bool**

- `toBoolean(Object) → Boolean`
- `toByte(BitVector) → Byte`
- `toInteger(BitVector) → Integer`
- `xor(Boolean, Boolean) → Boolean`

**Compare**

- `strlike(String, String) → Boolean`

**Crypt**

- `MD4withRSA(Sign)(Simple Type, String, String) → Boolean`
- `MD5withRSA(Sign)(Simple Type, String, String) → Boolean`
- `RSA(Number) → List, String`
- `SHA1(String) → String`
- `SHA1withRSA(Sign)(Simple Type, String, String) → Boolean`
- `SHA1withECDSA(Sign)(Simple Type, String, String) → Boolean`
- `SHA224(String) → String`
- `SHA256(String) → String`
- `SHA256withECDSA(Sign)(Simple Type, String, String) → Boolean`
- `SHA384(String) → String`
- `SHA384withECDSA(Sign)(Simple Type, String, String) → Boolean`

**Functions**

- `burn(Double) → Double`
- `counter(Byte, Byte) → Long`
- `count() → Long`
- `counter(Boolean) → Long`
- `eif(Boolean, Object, Object) → Object`
- `eval(String) → Object`
- `hash(String) → Integer`
- `isNull(String) → Boolean`
- `isNull(Number) → Boolean`
- `load() → Double`
- `mem() → Long`
- `random(Byte, Integer) → Integer`
- `read(String) → String`
- `REMOVED_timeliness(Number) → Double`
- `rnd() → Double`
- `sleep(Double) → Double`
- `SMAX(Object, Double) → Double`
- `SMIN(Object, Double) → Double`
- `Split(String, String, Long) → List, String`
- `split(String, String, List) → String`
- `storedLine(String, Matrix, Matrix) → Matrix`
- `storedValue(String, Matrix, Matrix) → Double`
- `uptime() → Long`
- `uuid() → String`

**Hex**

- `toHex(String) → HexString`
- `toHexDiscrete(Number) → HexString`
- `toHex(Double) → HexString`

**List**

- `contains(Simple Type, List) → Boolean`
- `filter(List, String) → List`
- `foreach(List, String) → List`
- `indexOf(List, Simple Type) → Integer`
- `isEmpty(List) → Boolean`
- `listProject(List, String) → List`
- `max(List, Integer) → Tuple`
- `max(List, Integer) → Tuple`

**Number**

- `max(List, Integer) → Tuple`
- `max(List, Integer) → Tuple`
- `size(List) → Integer`
- `toList(Object, Object, Object, Object, Object, Object) → List`
- `toList(Object, Object, Object, Object, Object, Object) → List`
- `toList(Object, Object, Object, Object, Object, Object) → List`
- `toList(Object, Object, Object, Object, Object, Object) → List`
- `toList(Object, Object, Object, Object, Object, Object) → List`
- `toList(Object, Object, Object, Object, Object, Object) → List`
- `toList(Object, Object, Object, Object, Object, Object) → List`
toList(Object, Object, Object, Object, Object, Object, Object, Object) → List
toList(Object) → List

Math
abs(Number) → Double
acos(Number) → Double
as2DVector(ProbabilisticDouble, ProbabilisticDouble) → VectorProbabilisticDouble
as3DVector(ProbabilisticDouble, ProbabilisticDouble, ProbabilisticDouble) → VectorProbabilisticDouble
asin(Number) → Double
atan(Number) → Double
atan2(Number | Object, Number | Object) → Double
ceil(Number) → Double
cos(Number) → Double
cosh(Number) → Double
distance(Matrix, Matrix) → Double
distance(VectorProbabilisticDouble, MatrixBoolean) → Double
distance(ProbabilisticDouble, Number) → Double
det(Matrix) → Double
dotProduct(Matrix, Matrix) → Double
dotProduct(Vector, Vector) → Double
eig(Matrix) → Vector
get(Vector, Vector) → Double
identity(Number) → Matrix
ileg(Matrix) → Vector
inv(Matrix) → Matrix
Max(Matrix) → Double
Max(Vector) → Double
Median(Vector) → Double
Median(Matrix) → Double
Min(Vector) → Double
Min(Matrix) → Double
ones(Number, Number) → Matrix
perms(Matrix) → Double
perms(Vector) → Matrix
readMatrix(String) → Matrix
readVector(String) → Vector
StdDev(Vector) → Double
StdDev(Matrix) → Double
sub(Matrix, Number, Number, Number, Number) → Matrix
sub(Vector, Number, Number) → Vector
Sum(Vector) → Double
Sum(Matrix) → Double
svd(Matrix) → Vector
toMatrix(Vector) → Matrix
toString(Matrix) → String
toString(Vector) → String
vector(Vector) → Double
tr(Matrix) → Double
trans(Matrix) → Matrix
Var(Matrix) → Double
Var(Vector) → Double
vectorFromString(String, String) → Vector
vectorFromString(String, String, Discrete Number) → Vector
zeros(Number, Number) → Matrix

Mдаstore
MdaIndices(String, List) → List_Integer
MdaIndices(String, Double, Double) → List_Integer

Mep
assureNumber(Number) → Double

String
concat(Object, Object) → String
length(String) → Integer
lower(String) → String
regex(String, String) → Boolean
startsWith(String, String) → Boolean
contains(String, String) → Boolean
substring(String, Number, Number) → String
substring(String, Number) → String
upper(String) → String

Time
businessDays(Date, Date) → Integer
curdate() → Date
dateInMillis(Date) → Long
day(String, String) → Integer
day(Date) → Integer
dayOfMonth(Date) → Integer
dayOfMonth(String, String) → Integer
dayOfWeek(Date, String) → Integer
format(Date, String) → String
hour(Date) → Integer
hour(String, String) → Integer
hours(Date, Date) → Integer
millisecond(Date) → Long
millisecond(String, String) → Long
milliseconds(Date, Date) → Long
millisecond() → Long
minute(Date) → Integer
minute(String, String) → Integer
minuteOfDay(Date) → Integer
minutes(Date, Date) → Integer
month(Date) → Integer
month(String, String) → Integer
months(Date, Date) → Integer
nanoTime() → Long
second(String, String) → Integer
seconds(Date, Date) → Integer
second(Date) → Integer
seconds(Date, Date) → Integer
seconds(Date, Date) → Integer

Transform
doubleToBoolean(Double) → Boolean
doubleToByte(Double) → Byte
doubleToChar(Double) → Char
doubleToFloat(Double) → Float
doubleToInteger(Double) → Integer
doubleToLong(Double) → Long
doubleToShort(Double) → Short
toString(Object) → String
toByte(Object) → Byte
toChar(Discrete Number) → Char
toChar(String) → Char
toDouble(Object) → Double
toFloat(UnsignedInt16, UnsignedInt16) → Float
toFloat(UnsignedInt16, UnsignedInt16, Boolean) → Float
tuple(Object, Object, Object, Object, Object, Object) → Tuple

tuple(Object, Object, Object, Object, Object, Object, Object) → Tuple

tuple(Object, Object, Object, Object, Object, Object, Object) → Tuple

tuple(Object, Object, Object, Object, Object, Object, Object, Object) → Tuple

Symbols

%!Boolean → Boolean
%!Number → Number
%!Object → Object
%!String → String
!=(ProbabilisticResult, ProbabilisticResult) → ProbabilisticResult
!=(Boolean, Boolean) → Boolean
!=(Number | Object, Number | Object) → Boolean
!=(String, String) → String
![](tuple(Object, Object, Object, Object, Object, Object, Object, Object, Object)) → Tuple

Data Handlers

AVG
PARTIALAGGREGATE
LIST
LONG
BITVECTOR
LIST
SHORT
BOOLEAN
LIST
STRING
BYTE
LIST
DUPLEX
BYTEBUFFER
LONG
COUNT
PARTIAL
AGGREGATE
MATRX
DATE
MULTI
VALUE
DOCUMENT
MV
DOUBLE
NESTEDKEYVALUEOBJECT
END
TIMESTAMP
NTUPLE
FLOAT
PROBABILISTIC
INTEGER
PROBABILISTIC
TUPLE
INTERVAL
DOUBLE
RELATIONALELEMENT
PARTIAL
AGGREGATE
KEYVALUEOBJECT
SHORT
LIST
START
TIMESTAMP
LIST
BOOLEAN
START
TIMESTAMP
LIST
BYTE
STRING
LIST
CHAR
TIMESTAMP
LIST
DATE
TUPLE
LIST
DOUBLE
UNSIGNED
INT
16
LIST
FLOAT
VECTOR
LIST
INTEGER
Sample Odysseus query

```odysseus
#PARSER PQL
#ADDQUERY
input = ACCESS({source='source',
                wrapper='GenericPull',
                transport='File',
                protocol='CSV',
                dataHandler='Tuple',
                metaattribute=['TimeInterval'],
                options=[["filename","example.csv"],
                schema=[["value","Double"]])

output = MAP({expressions = ['value + 3']}, input)
```

Copyright © 2015 ODYSSEUS Team

http://odysseus.informatik.uni-oldenburg.de
Wiki: http://wiki.odysseus.informatik.uni-oldenburg.de
Forum: http://forum.odysseus.informatik.uni-oldenburg.de