

MQTT Transport Handler

This transport handler allows to communicate with an MQTT broker (e.g. [Mosquito](#))

The library used here for processing is [PAHO](#).

To use this transport handler you need to install the MQTT Wrapper Feature.

Options

- topic: The topic to send or receive from
- broker: The address of the broker
- client_id: The name of the client. Important: If you use Odysseus for sending and receiving, there must be different client_ids!

Example

```
// Receiving elements

#PARSER PQL
#RUNQUERY
in = RECEIVE({
    transport = 'MQTT',
    protocol = 'simplecsv',
    source = 'mqttsource',
    datahandler = 'Tuple',
    options = [
        ['topic','Test'],
        ['Broker','tcp://localhost:1883'],
        ['Client_ID','OdysseusReceiver']
    ],
    schema = [['Message', 'String']]
})

// Sending elements
#PARSER PQL
#RUNQUERY
ping = TIMER({PERIOD = 1000, SOURCE = 'Timer'})

input = MAP({EXPRESSIONS = ["Hallo " + toString(counter())]}, ping)

out = SENDER({
    transport = 'MQTT',
    protocol = 'simplecsv',
    sink = 'mqttsink',
    WRAPPER = 'GenericPush',
    datahandler = 'Tuple',
    options = [
        ['topic','Test'],
        ['Broker','tcp://localhost:1883'],
        ['Client_ID','OdysseusSender']
    ]
}, input)
```

Remark: Protocol and data handler are just examples. MQTT can be used with any other protocol and data handler (e.g. KeyValue):

```

#PARSER PQL
#RUNQUERY
in = RECEIVE({
    transport = 'MQTT',
    protocol = 'JSON',
    source = 'mqttsource',
    datahandler = 'KeyValueObject',
    options = [
        ['topic', 'Test'],
        ['Broker', 'tcp://localhost:1883'],
        ['Client_ID', 'OdysseusReceiverKV']
    ]
})

#PARSER PQL
#RUNQUERY
ping = TIMER({PERIOD = 1000, SOURCE = 'Timer'})

input = MAP({EXPRESSIONS = [['Hallo " + toString(counter())', 'message']], ping)

kv_input = TUPLETOKEYVALUE(input)

out = SENDER({
    transport = 'MQTT',
    protocol = 'JSON',
    sink = 'mqttsink',
    WRAPPER = 'GenericPush',
    datahandler = 'KeyValueObject',
    options = [
        ['topic', 'Test'],
        ['Broker', 'tcp://localhost:1883'],
        ['Client_ID', 'OdysseusSenderKV']
    ]
}, kv_input)

```

Change Topic based on content of stream

Sometimes, you want to change the topic based on some stream content. As this is a transport handler it does nothing know about content, so we cannot provide e.g. an option to use an attribute as topic. But there is another way, using the [Command Operator](#)

```

#PARSER PQL
#ADDQUERY
timer = TIMER({
    id = 'timer1',
    period = 1000,
    starttime = 0 ,
    source = 'source'
})

filter = SELECT({
    predicate = 'time % 2 == 0',
    heartbeatrate = 1
},
timer
)

/// Use updateTransportOption as function
/// first parameter is the name of the sink ("mqtttsink")+".transport" to state, that an option of the transport
handler should be changed
/// second: The option that should be changed ("TOPIC" is the only value, that is changeable)
/// third: A MEP-Expression that is evaluated and interpreted as String as the new value for the topic.
/// Remark: The function is called in the moment the operator processes the input, i.e. the topic is changed in
that moment. If there is e.g.
/// a buffer after this operator, then it is not really clear, which elements will be send to which topic, so
it is typically the
/// best to put the command operator directly before the sink
command = COMMAND({
    COMMANDEXPRESSION = 'updateTransportOption("mqtttsink.transport","TOPIC","TEST"+toString(time %
2))'
},
filter
)

out = SENDER({
    transport = 'MQTT',
    protocol = 'csv',
    sink = 'mqtttsink',
    WRAPPER = 'GenericPush',
    datahandler = 'tuple',
    options = [
        ['topic','TEST'],
        ['Broker','tcp://192.168.2.34:1888'],
        ['Client_ID','OdysseusSender']
    ]
}, command
)

```