

Template.

Reads messages from a local or remote Microsoft MQ queue.

<input type = "MSMQ" *attributes* >

Attribute	Required	Description	Default
queue	X	Name of the queue, e.g. ".\private\$\inqueue", "FormatName:Direct=OS:DXBDJB\private\$\inqueue"	-
createQueue	O	"true" or "false". Whether the queue should be created if it does not already exist.	false
getTimeout	O	Time in milliseconds the listener will wait for a message to appear on each loop	10000
maxMessages	O	The maximum number of messages allowed in the queue before the oldest messages are deleted	10
maxRetry	O	Maximum number of retries to connect to the queue before the messages is undeliverable	
maxWait	O		
name	O	Name of the input/output. For identification in logs only.	
priority	O	Integer. Priority order for input queue. "1" is the highest priority, followed by "2" etc.	-
retryInterval	O		
stylesheet	O	Comma separated list of XSL stylesheets to apply to the received message before further processing	-
undeliverableQueue	O	Name of the local MS MQ queue to send to message to if sending of the message fails	-
xslVersion	O	The XSLT version of the stylesheet. [1.0,2.0 or 3.0]. Used to select the transformer implementation. 1.0 has the least performance impact	1.0
bufferQueueName	-	Local queue name for temporary storage of received messages	
connection	-		
consumerGroup	-	The Kafka consumer group ID	
deleteAfterSend	-	"true" or "false". Whether the source file is deleted after it is sent	false
fileFilter	-	Pattern of the files to watch the creation of	*.*
key	-		-
path	-	Full pathname of the directory to monitor for new files	-
requestURL	-	The local URL	
topic	-	The Kafka Topic	My_topic

Microsoft MQ Input Type.

Reads messages from a local or remote Microsoft MQ queue.

<input type = "MSMQ" *attributes* >

Attribute	Required	Description	Default
queue	X	Name of the queue, e.g. ".\private\$\inqueue", "FormatName:Direct=OS:DXBDJB\private\$\inqueue"	-
createQueue	O	"true" or "false". Whether the queue should be created if it does not already exist.	false
getTimeout	O	Time in milliseconds the listener will wait for a message to appear on each loop	10000
maxRetry	O	Maximum number of retries to connect to the queue before the messages is undeliverable	
maxWait	O		
name	O	Name of the input/output. For identification in logs only.	
priority	O	Integer. Priority order for input queue. "1" is the highest priority, followed by "2" etc.	-
retryInterval	O		
stylesheet	O	Comma separated list of XSL stylesheets to apply to the received message before further processing	-
undeliverableQueue	O	Name of the local MS MQ queue to send to message to if sending of the message fails	-
xslVersion	O	The XSLT version of the stylesheet. [1.0,2.0 or 3.0]. Used to select the transformer implementation. 1.0 has the least performance impact	1.0

IBM MQ Input Type.

Reads messages from an IBM MQ queue.

<input type = "MQ" *attributes* >

Attribute	Required	Description	Default
queue	X	Name of the queue.	-
connection	X	Comma separated string with the connection parameters of the form " qmgr,channel,host,port[,user,password] " Example: "AIEQMGR, AMS.SVRCONN, localhost, 1415"	-
getTimeout	O	Time in milliseconds the listener will wait for a message to appear on each loop	10000
name	O	Name of the input/output. For identification in logs only.	-
priority	O	Integer. Priority order for input queue. "1" is the highest priority, followed by "2" etc.	-
stylesheet	O	Comma separated list of XSL stylesheets to apply to the received message before further processing	-
xslVersion	O	The XSLT version of the stylesheet. [1.0,2.0 or 3.0]. Used to select the transformer implementation. 1.0 has the least performance impact	1.0

IBM MQ OUTPUT Type.

Sends messages to an IBM MQ queue.

<output type = "MQ" *attributes* >

Attribute	Required	Description	Default
queue	X	Name of the queue.	-
connection	X	Comma separated string with the connection parameters of the form " qmgr,channel,host,port[,user,password] " Example: "AIEQMGR, AMS.SVRCONN, localhost, 1415, mqusername, mquserpassword"	-
stylesheet	O	Comma separated list of XSL stylesheets to apply to the received message before further processing	-
xslVersion	O	The XSLT version of the stylesheet. [1.0,2.0 or 3.0]. Used to select the transformer implementation. 1.0 has the least performance impact	1.0

maxMessages	O	Maximum number of messages allowed in the queue. (-1 for no limit)	-1
undeliverableQueue	O	Name of the local MS MQ queue to send to message to if sending of the message fails	-
maxRetry	O	Maximum number of retries to connect to the queue before the messages is undeliverable (-1 for no limit)	10

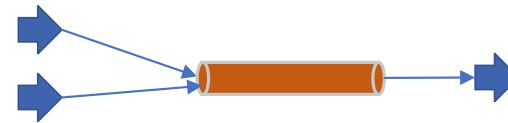
<input type="MSMQ" attributes >

Reads from a local or remote MS MQ queue

Attribute	Require	Description
queue	X	The queue name to read the messages from -



```
<pipe>  
  <input type="MSMQ" queue=".\\private$\\queuein" />  
  <output type="MSMQ" queue=".\\private$\\queueout" />  
</pipe>
```



```
<pipe>  
  <input type="MSMQ" priority=1 queue=".\\private$\\queuein1" />  
  <input type="MSMQ" priority=2 queue=".\\private$\\queuein1" />  
  <output type="MSMQ" queue=".\\private$\\queueout" />  
</pipe>
```



```
<pipe>  
  <input type="MSMQ" queue=".\\private$\\queuein" />  
  <output type="KAFKA" topic="FlightUpdate" connection="localhost:9092" />  
</pipe>
```


Executing and Deployment.

QueueExchange requires .NET version 4.5.2 or higher to execute.

QueueExchange makes use of the TopShelf framework which means it can be executed from the Windows command line or installed as a service. The QueueExchange files can be copied to and run from any directory.

Executing from Command line

C:\QueueExchange\QX.exe

To install as a service:

C:\QueueExchange\QX.exe install start

To uninstall as a service:

C:\QueueExchange\QX.exe uninstall

Multiple instances of QueueExchange can be installed or run by completely copying it to a separate directory for each instance. Each instance should have its own unique configuration

For example:

C:\QueueExchange\Instance1\QX.exe install

C:\QueueExchange\Instance2\QX.exe install

C:\Utility\BridgeQueues\QX.exe install

C:\Services\PublishToKafka\QX.exe install

The definition of the message flow between inputs and outputs is defined in the file **ExchangeConfig.xml** which is in the directory QueueExchange is execute from.

If

Pipes

A pipe defines the connection between a set of inputs and a set of outputs.

In the simplest form, a pipe defines connection between one input and one output. Inputs and outputs can be of any type.

A pipe can have multiple outputs. Outputs can be of the same type or of mixed types.

A pipe can have multiple inputs. Inputs can be of the same or mixed types.

One instance of QueueExchange can have multiple pipes defined. The set of <pipe> elements are defined under a single <pipes> element. Each pipe is defined in a <pipe> elements. The only child elements of <pipe> are <input> and <output>.

<pipe *attributes*>

Attribute	Required	Description	Default
name	O	Name of the pipe for logging purposes	-
randomDistribution	O	Boolean indicating whether a random distribution to the available Outputs should be used	false
roundRobinDistribution	O	Boolean indicating whether a round robin distribution to the available Outputs should be used	false
outputIsolation	O	Boolean. If there are multiple outputs, setting outputIsolation allows the pipe to continues processing without waiting the completion of all the sends	false
maxMessagesPerMinute	O	Integer specifying the maximum number of messages to process through the pip per minute. Leave blank for no limit	-
enableLog	O	Boolean. Enable message logging for this pipe if a logger has been defined	false

Message Prioritization of Input

A pipe can have multiple Input defined. If there are multiple in Inputs, the pipe will prioritize the inputs according the priority set on each Input.

When there are multiple Inputs defined, after the pipe has processed a message, it checks each input in priority order for the next message. This has the effect of processing all the available messages from the highest priority queue before moving on to the next input.

Message Distribution to Outputs

A pipe can distribute messages to outputs in three modes:

1. Distribute to all the outputs (default)

2. Round Robin distribution where the message is sent to only one output. Successive messages are sent to each output in turn
3. Random distribution where the message is sent to only one output selected at random

Pipe Message Throughput Throttling

The throughput of a pipe can be throttled, by specifying the maximum throughput rate

Pipe Specific Logging

Pipe specific logging can be enabled or disabled.

Inputs

<input>

Inputs are a source of messages that can be passed to the pipe.

The available types of inputs are:

- MS MQ queues
- IBM MQ queues
- Kafka Topics
- Rabbit MQ queues
- Files
- HTTP Post
- Test Source

The type of input is specified in the **type** attribute of the <input> element. Other attributes are dependent on the specific type.

Messages received from inputs can be filtered before passing through the pipe. Messages that do not pass the filter are not passed through the pipe.

Received messages can be transformed using XSLT transformation before being passed into the pipeline

Outputs

Outputs are a destination of messages that have been passed through the pipe.

The available types of outputs are:

- MS MQ queues
- IBM MQ queues
- Kafka Topics
- Rabbit MQ queues
- Files
- HTTP Client
- HTTP Get
- Sink

The type of output is specified in the ***type*** attribute of the <output> element. Other attributes are dependent on the specific type.

Messages passed to outputs can be filtered before being sent to the destination. Messages that do not pass the filter are not sent to the destination.

Messages passed to outputs can be transformed using XSLT transformation before being sent to the destination.

Filters

The only valid child element of <input> and <output> is <filter>. The filter element is optional.

Filters can be specified on inputs and outputs. The filter must return “true” for the message to pass. Filters are applied to messages received from inputs before they are passed into the pipe. Filters are applied to outputs before they are sent to their destination.

The filter can define an <altqueue> element. If a filter is not passed, the message will be sent to the destination defined by <altqueue> if it is present. <altqueue> has the same definition as any of the valid <output> types.

The filters that operator on the message data are:

- xpeexists – the XPath specified exists in the message (XML data)
- xpequals – the node at the specified XPath in the messages is equal to the specified value
- xpmatches – the node at the specified XPath in the messages matches specified regex
- dateRange – the date element at the specified XPath falls within the range specified relative to “now”
- contains – the message contains the specified text
- matches – the message matched the specified regex
- length – the message has a minimum specified value
- bool – always returns the specified Boolean value (testing purposes)

```
<filters>
  <contains>
    <value>FlightUpdatedNotification</value>
  </contains>
```

```
</filter>
```

Compound filtering expressions can be created by using Boolean operators. The Boolean operators available for compound filtering expressions are AND, OR, NOT and XOR.

```
<filter>
  <or>
    <contains>
      <value>FlightUpdatedNotification</value>
    </contains>
    <contains>
      <value>FlightCreatedNotification</value>
    </contains>
  </or>
</filter>
```

An alternate destination can be specified if the message does not pass the filter. The alternate queue is specified in the same way as output types, and can have filters applied to it as well

```
<filter>
  <altqueue type="MSMQ" queueName=".\\private$\\altDestinationQueue" />
  <contains>
    <value>FlightUpdatedNotification</value>
  </contains>
</filter>
```

Input Types

Microsoft MQ Input Type.

Reads messages from a local or remote Microsoft MQ queue.

<input type = "MSMQ" *attributes* >

Attribute	Required	Description	Default
queue	X	Name of the queue, e.g. ".\private\$\inqueue", "FormatName:Direct=OS:DXBDJB\private\$\inqueue"	-
createQueue	O	"true" or "false". Whether the queue should be created if it does not already exist.	false
getTimeout	O	Time in milliseconds the listener will wait for a message to appear on each loop	10000
maxRetry	O	Maximum number of retries to connect to the queue before the messages is undeliverable	
name	O	Name of the input/output. For identification in logs only.	
priority	O	Integer. Priority order for input queue. "1" is the highest priority, followed by "2" etc.	-
retryInterval	O		
stylesheet	O	Comma separated list of XSL stylesheets to apply to the received message before further processing	-
undeliverableQueue	O	Name of the local MS MQ queue to send to message to if sending of the message fails	-
xslVersion	O	The XSLT version of the stylesheet. [1.0,2.0 or 3.0]. Used to select the transformer implementation. 1.0 has the least performance impact	1.0

Setting **createQueue** = "true" is only valid for local queue

IBM MQ Input Type.

Reads messages from an IBM MQ queue.

<input type = "MQ" *attributes* >

Attribute	Required	Description	Default
queue	X	Name of the queue.	-
connection	X	Comma separated string with the connection parameters of the form "qmgr,channel,host,port[,user,password] " Example: "AIEQMGR, AMS.SVRCONN, localhost, 1415"	-
getTimeout	O	Time in milliseconds the listener will wait for a message to appear on each loop	10000
name	O	Name of the input/output. For identification in logs only.	-
priority	O	Integer. Priority order for input queue. "1" is the highest priority, followed by "2" etc.	-
stylesheet	O	Comma separated list of XSL stylesheets to apply to the received message before further processing	-
xslVersion	O	The XSLT version of the stylesheet. [1.0,2.0 or 3.0]. Used to select the transformer implementation. 1.0 has the least performance impact	1.0

HTTP Input Type.

Reads messages POSTED to a HTTP Listener.

`<input type = "HTTP" attributes >`

Attribute	Required	Description	Default
requestURL	X	The URL on the local system that the HTTP listener will be deployed on ,eg, "localhost:8080/input/". Note: The URL must end with a "/"	-
bufferQueueName	X	Name of a local MS MQ for buffering purposes. (Will be created if it does not already exist)	-
name	O	Name of the input/output. For identification in logs only.	-
priority	O	Integer. Priority order for input queue. "1" is the highest priority, followed by "2" etc.	-
stylesheet	O	Comma separated list of XSL stylesheets to apply to the received message before further processing	-
xslVersion	O	The XSLT version of the stylesheet. [1.0,2.0 or 3.0]. Used to select the transformer implementation. 1.0 has the least performance impact	1.0

When this input type is defined a HTTP listener is started and listens for "POST" request on the defined URL. The body of the POST request is extracted and passed into the pipe.

KAFKA Input Type.

Reads messages from a KAFKA Topic.

<input type = "KAFKA" *attributes* >

Attribute	Required	Description	Default
topic	X	The Kafka defined topic that is listened to.	-
connection	O	The hostname:port of the Kafka server to connect to	localhost:9092
consumerGroup	O	The Kafka consumer group for the listener	QueueExchange
bufferQueueName	X	Name of a local MS MQ for buffering purposes. (Will be created if it does not already exist)	
name	O	Name of the input/output. For identification in logs only.	-
priority	O	Integer. Priority order for input queue. "1" is the highest priority, followed by "2" etc.	-
stylesheet	O	Comma separated list of XSL stylesheets to apply to the received message before further processing	-
xslVersion	O	The XSLT version of the stylesheet. [1.0,2.0 or 3.0]. Used to select the transformer implementation. 1.0 has the least performance impact	1.0

This input listens for messages on the defined Kafka topic and passes them to the pipe for processing

RABBIT Input Type.

Reads messages from a Rabbit MQ queue on the default exchange.

<input type = "RABBITDEFEX" *attributes* >

Attribute	Required	Description	Default
queue	X	The Rabbit MQ queue to read from.	-
connection	X	The hostname:port of the Rabbit server to connect to	-
bufferQueueName	X	Name of a local MS MQ for buffering purposes. (Will be created if it does not already exist)	-
name	O	Name of the input/output. For identification in logs only.	-
priority	O	Integer. Priority order for input queue. "1" is the highest priority, followed by "2" etc.	-
stylesheet	O	Comma separated list of XSL stylesheets to apply to the received message before further processing	-
xslVersion	O	The XSLT version of the stylesheet. [1.0,2.0 or 3.0]. Used to select the transformer implementation. 1.0 has the least performance impact	1.0

FILE Input Type.

Watches the specified directory for creation of new files and passes those new files to the pipeline

<input type = "FILE" *attributes* >

Attribute	Required	Description	Default
path	X	The absolute pathname of the directory to monitor	-
fileFilter	O	File name pattern of files to watch for.	*,*
deleteAfterSend	O	Boolean. Indicates whether to delete the file after it has been passes to the pipeline	false
name	O	Name of the input/output. For identification in logs only.	-

bufferQueueName	X	Name of a local MS MQ for buffering purposes. (Will be created if it does not already exist)	-
priority	O	Integer. Priority order for input queue. "1" is the highest priority, followed by "2" etc.	-
stylesheet	O	Comma separated list of XSL stylesheets to apply to the received message before further processing	-
xslVersion	O	The XSLT version of the stylesheet. [1.0,2.0 or 3.0]. Used to select the transformer implementation. 1.0 has the least performance impact	1.0

Sample – One to One

```
<pipes>
  <pipe>
    <input type="MSMQ" queue=".\private$\appOutputQueue" />
    <output type="MQ" queue="QUEUE.IN" connection="QXQMGR, GLASSFISH.SVRCONN, localhost, 1414" />
  </pipe>
</pipes>
```

Sample – One to Many

```
<pipes>
  <pipe>
    <input type="MSMQ" queue=".\private$\app1OutputQueue" />
    <output type="MQ" queue="QUEUE.IN" connection="QXQMGR, GLASSFISH.SVRCONN, localhost, 1414" />
    <output type="MSMQ" queue="FormatName:Direct=OS:DXBDJB\private$\appOutputQueue" />
  </pipe>
</pipes>
```

```
<pipes>
  <pipe>
    <input type="MSMQ" queue=".\private$\app1OutputQueue" />
    <output type="MQ" queue="QUEUE.IN" connection="QXQMGR, GLASSFISH.SVRCONN, localhost, 1414" />
    <output type="MSMQ" queue="FormatName:Direct=OS:DXBDJB\private$\appOutputQueue" >
      <filter>
        <xpexists>
          <value>//FlightID/Registration</value>
        </xpexists>
      </filter>
    </output>
  </pipe>
</pipes>
```