

Configuring AMQP clients to interact with applications that put to or get from MQ queues

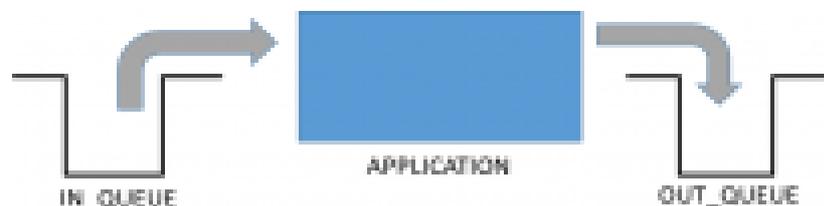
[FSWU Chris Matthewson](#)

Published on 18/12/2018

The IBM MQ AMQP implementation supports publish/subscribe only. AMQP clients cannot put messages to, or get messages from, MQ queues directly – they are limited to publishing and subscribing to topics in a queue manager’s topic tree. If you have existing applications that either get messages from an existing queue for processing, or put messages to a queue, it is not possible for AMQP clients to directly interact with the queues used by that application.

In order to allow publishing AMQP clients to send messages to an existing queue, or subscribing AMQP clients to receive messages that were put to a queue, some additional objects must be created. With these administrative changes, the AMQP clients can be integrated with an existing queue-based application.

For instance, for an existing application that picks up messages from a queue ‘IN_QUEUE’, and puts messages to a queue ‘OUT_QUEUE’:



it is possible for AMQP clients to put messages to ‘IN_QUEUE’, and get messages from ‘OUT_QUEUE’, with some limited administrative changes. There are no changes required to the application itself.

Sending messages to the application:

The application is already picking up messages from ‘IN_QUEUE’ – we want an AMQP client to be able to publish messages so that they go to this queue to be processed by the application. To do this, we can create a new administrative subscription, where the topic string this subscription receives messages from is the topic string that the AMQP client publishes to, and the destination queue of this subscription is the input queue for the application, ‘IN_QUEUE’. So, any messages that are published to the defined topic string for that administrative subscription will be routed to the defined destination – in this case ‘IN_QUEUE’.

Assuming the AMQP client will be publishing to a topic string `'/application/in'`, we can then create an **administrative subscription 'APP_IN'** with the MQSC command:

```
DEF SUB(APP_IN) TOPICSTR('/application/in') DEST('IN_QUEUE')
```

With this object defined, all messages published to `'/application/in'` will be routed to the destination `'IN_QUEUE'`, where they will be picked up by the application in the same way as any other messages put to this queue by other applications.

Getting messages from the application

The application is putting messages to `'OUT_QUEUE'`, where they can be picked up and processed by other clients. These could be standard MQ clients, or a JMS activation specification for delivery to a message-driven bean – however in this case we want the messages to be delivered to an AMQP client instead. But, AMQP clients only use `publish/subscribe`, and cannot pick messages up directly from a queue – so some new administrative changes will have to be made to route these messages to a subscribing AMQP client. Note that these changes involve replacing the clients previously receiving messages with the subscribing AMQP client – to use a combination of AMQP and other clients to get messages, more extensive changes would be required.

In this case, there are a couple of objects that will need to be created. First a topic object, defined for the topic string that the AMQP client is subscribed to. Any messages delivered to this topic object will be delivered to the AMQP client subscribing to the same topic string. Assuming the AMQP client will be subscribing to a topic string `'/application/out'`, we can define the **topic object 'APP_OUT'** with the MQSC command:

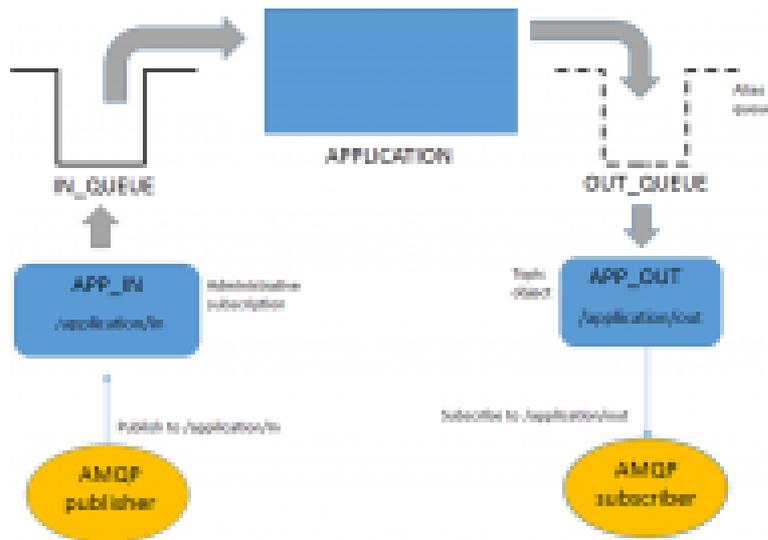
```
DEF TOPIC(APP_OUT) TOPICSTR('/application/out')
```

We then need to ensure that messages put to `'OUT_QUEUE'` by the application are delivered to this new topic object, so they will be sent to the subscribing client. To do this, **replace the existing queue 'OUT_QUEUE' with an alias queue of the same name**, with a target type of the topic object just created. The MQSC command to create this alias queue is:

```
DEF QALIAS(OUT_QUEUE) TARGTYPE(TOPIC) TARGET(APP_OUT)
```

Now messages put by the application to `'OUT_QUEUE'` will not wait on the queue to be picked up – instead they will be delivered to the target of this alias queue: the new topic object `'APP_OUT'`. The AMQP client which is subscribed to the topic string represented by this topic object `'/application/out'` will then receive any messages sent to this topic object from the alias queue.

So, with these new objects created, and AMQP clients either publishing or subscribing to these new objects, the diagram above with the application getting from `IN_QUEUE` and putting to `OUT_QUEUE` now looks like:



Summary

While AMQP clients can only use publish/subscribe with MQ, it is still possible for these clients to put messages to, or get messages from, existing queues, with some simple administrative changes. No changes to application code are required.

For an AMQP publisher to put a message to a queue: create an administrative subscription for the topic string the AMQP client is publishing to, with a destination of the intended queue.

For an AMQP subscriber to get messages from a queue: replace the queue with an alias queue of the same name, with a target of a topic object representing the topic string the AMQP client is subscribed to.