

JMS 2.0 shared subscription support

The JMS 2.0 specification adds support for shared subscriptions. QPIDJMS-220 outlines how the Qpid JMS client maps the JMS semantics to AMQP terms. This document outlines how the broker responds to those AMQP performatives.

Connection Open

"SHARED-SUB" capability should be set on offered capabilities.

Link Attach

Assumptions:

- If source is set the address is specified as "exchange/binding" and exchange with name "exchange" exists. If exchange with name "exchange" does not exist the link is refused with error "not found".
- Link does not exist. for Link reattachment see below.
- Terminus durability : NO means 0, YES means 1 or 2.
- When creating a queue its life time policy will be derived from source expiry policy as follow:
 - "link-detach" becomes "delete on no outbound link"
 - "session-end" becomes "delete on session end"
 - "connection-close" becomes "delete on connection close"
 - "never" becomes "permanent"
- SHARED and GLOBAL are derived from Source capabilities if Source is set and from Link capabilities otherwise

Not Null Source	S H A R E D	G L O B A L	TERMINUS DURABILITY	outcome
YES	NO	NO	NO	Create a non-durable queue with random name and exclusivity policy "link". Bind queue to exchange "exchange" using binding key "binding". Broker replies with attach having capability "topic" on a source and attach offered capability "SHARED-SUB".
YES	NO	NO	YES	Create a durable queue with name derived from link name, container id and source durability. Exclusivity policy is set to "link". Bind queue to exchange. Broker replies with attach having capability "topic" on a source and attach offered capability "SHARED-SUB". If queue already exists the link should be refused with error "resource locked"
YES	NO	YES	NO	The same as for SHARED=NO, GLOBAL=NO and DURABLE=NO
YES	NO	YES	YES	Create a durable queue with name derived from link name, global name space and source durability. Exclusivity policy is set to "link". Bind queue to exchange. Broker replies with attach having capabilities "topic" and "global" on a source and attach offered capability "SHARED-SUB". If queue already exists the link should be refused with error "resource locked"
YES	YES	NO	NO	Create a non-durable queue with name derived from link name (up to " "), container id and source durability. Exclusivity policy is set to "shared". Bind queue to exchange. Broker replies with attach having capabilities "topic" and "shared" on a source and attach offered capability "SHARED_SUB". If queue already exists and filters and/or binding key are not the same as derived from attach source, then: <ul style="list-style-type: none"> ◦ if there is/are existing consumer(s), refuse the link with error "resource locked" ◦ if there is no consumer, delete queue and bind new queue using new filters and/or binding key If existing queue exclusivity policy is not "shared", then link creation is refused with error "resource locked". If existing queue lifetime and/or durability are different from expected, then link creation is allowed but the source fields in the reply should be modified accordingly.
YES	YES	NO	YES	Create a durable queue with name derived from link name (up to " "), container id and source durability. Exclusivity policy is set to "shared". Bind queue to exchange. Broker replies with attach having capabilities "topic" and "shared" on a source and attach offered capability "SHARED_SUB". See above, if queue exists
YES	YES	YES	NO	Create a non-durable queue with name derived from link name (up to " "), global namespace and source durability. Exclusivity policy is set to "shared". Bind queue to exchange. Broker replies with attach having capabilities "topic", "shared" and "global" on a source and attach offered capability "SHARED_SUB". See above, if queue exists
YES	YES	YES	YES	Create a durable queue with name derived from link name (up to " "), global name space and source durability. Exclusivity policy is set to "shared". Bind queue to exchange. Broker replies with attach having capabilities "topic", "shared" and "global" on a source and attach offered capability "SHARED_SUB". See above, if queue exists

NO	NO	NO	N/A	<p>Look for queue with name derived from link name, container id and "true" durability.</p> <p>If queue exists and there is no consumer, Attach the link with the Source set.</p> <p>If queue does not exist, refuse the link with error "not found".</p>
NO	NO	YES	N/A	<p>Look for queue with name derived from link name, global name space and "true" durability.</p> <p>If queue exists and there is no consumer, Attach the link with the Source set.</p> <p>If queue does not exist, refuse the link with error "not found"</p>
NO	YES	NO	N/A	<p>Look for queue with name derived from link name (up to " "), container id and "true" durability.</p> <p>If queue exists and there is no consumer, Attach the link with the Source set.</p> <p>If queue does not exist, refuse the link with error "not found".</p>
NO	YES	YES	N/A	<p>Look for queue with name derived from link name (up to " "), global name space and "true" durability.</p> <p>If queue exists and there is no consumer, Attach the link with the Source set.</p> <p>If queue does not exist, refuse the link with error "not found".</p>

Link re-Attach

If link exists and the queue exists see the instructions for the existing queue for SHARED=YES.
If link exists and the queue does not exist, recreate a queue as described in corresponding instructions above.

Link Detach

If the Detach has close=true

- If Link Source has capability "topic" and queue has no consumers, the queue is deleted and the Link is Detached
- If Link Source has capability "topic" and queue has consumers, the Link is detached with error(resource-locked)
- if Link Source does not have capability "topic", the Link is detached

Pseudo Code

The above described behaviour results in the following pseudo code. Where the below differs from the above the above is considered normative:

```

if Link does not exists
  if Source is set
    if Source has SHARED capability
      subscription_name = Link name up to (excluding) "|"
      exclusivity_policy = "SHARED"
    else
      subscription_name = Link name
      exclusivity_policy = "LINK"
    if Source has GLOBAL capability
      name_space = global name space
    else
      name_space = name space derived from container-id
  queue_name = get_queue_name(
    name_space,
    subscription_name,
    Source Durability)
  if queue exists
    if has same exclusivity_policy
      if different binding key or filter
        if no consumer
          rebind queue
        else
          error(resource-locked)
      else
        error(resource-locked)
    else
      create_queue(
        queue_name
        exclusivity_policy,
        Source Durability,
        Source Expiry Policy)
      bind queue
      send Attach
  else
    // name_space and subscription_name are derived as above just from Link capabilities instead of
    Source capabilities
    queue_name = get_queue_name(
      name_space,
      subscription_name,
      TRUE_DURABILITY)
    if queue exists
      Attach Link with correct Source set
    else
      error(not-found)
else
  queue_name = get_queue_name(
    name_space,
    subscription_name,
    TRUE_DURABILITY)
  if queue exists
    if has same exclusivity_policy
      if different binding key or filter
        if no consumer
          rebind queue
        else
          error(resource-locked)
      else
        error(resource-locked)
    else
      create_queue(
        queue_name
        exclusivity_policy,
        Source Durability,
        Source Expiry Policy)
      bind queue
      send Attach

```