



Red Hat JBoss AMQ 7.0

AMQ Broker 7.0 Release Notes

Release Notes for AMQ Broker

Red Hat JBoss AMQ 7.0 AMQ Broker 7.0 Release Notes

Release Notes for AMQ Broker

Legal Notice

Copyright © 2017 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution-Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

These release notes contain the latest information about new features, enhancements, fixes, and issues contained in the AMQ Broker 7.0 release.

Table of Contents

CHAPTER 1. FEATURES	3
Persistence	3
Security	3
Performance	3
CHAPTER 2. ENHANCEMENTS	4
CHAPTER 3. RESOLVED ISSUES	5
CHAPTER 4. KNOWN ISSUES	6

CHAPTER 1. FEATURES

The features added in this release, and that were not in previous releases of the AMQ broker, are outlined below.

Persistence

Failover

Brokers can be fully redundant, allowing a backup broker to take over connections from a primary broker in the event of a failure. Optionally, a shared volume can be configured between the primary and backup broker.

Delivery guarantee

Some messages that a developer sends from his application require a guarantee of delivery guaranteeing that messages get from the application sending them to the application consuming and processing them. If a network link goes down or the broker experiences a power outage while the message is in transit the message broker must resend undelivered messages once it returns online or the network connection is restored.

Zero persistence

Brokers can be configured to not persist any messages for scenarios in which no guarantee is required, but high performance is.

Security

Basic authentication

Brokers can authenticate clients using basic username and password credentials.

Role-based authentication

Brokers can authenticate and authorize clients to specific queues or message addresses based on roles assigned to the client.

SSL-encrypted connections

Connections between clients and broker or a broker and another broker are secure and utilize SSL to encrypt the connections.

Certificate-based authentication

Brokers can authenticate a client connection or a connection from another broker using certificates.

Performance

Dead connection detection

Brokers can detect dead connections when a client has exceeded specific lifespan that can be set either on the client's connection factory or globally on the server.

Slow consumer detection

Brokers can detect slow consumers and disconnect them from the server when this feature is enabled.

Refer to the ActiveMQ Artemis project for additional information:

<http://activemq.apache.org/artemis/index.html>

CHAPTER 2. ENHANCEMENTS

- **ENTMQBR-512 - Console should display message body**

As of this release, the message body is displayed when messages are browsed by using the AMQ Console.

- **ENTMQBR-327 - Design and implement an addressing system that can be used cross-protocol**

Previously, there were issues observed when sending/receiving messages between a core JMS client and clients using other protocols. This was because the core JMS client used prefixes for destinations (namely, `jms.topic.` and `jms.queue`) that were hidden. The new addressing model in AMQ 7 enables customers to configure destinations that independent of JMS conventions.

- **ENTMQBR-312 - Support 'anonymous' senders**

Previously, the broker did not support AMQP clients creating sender links with a null target address and then identifying the destination of each message by using its 'to' field.

For the AMQP JMS client, this meant that any 'anonymous' MessageProducers (those created without a Destination) needed to open and close a sending link for each message that was sent. Similarly, the same scenario occurred for every message sent by using JmsProducer instances, as these were also all 'anonymous'. Due to both of these scenarios, performance was negatively impacted. Support for 'anonymous' senders is added in this release.

CHAPTER 3. RESOLVED ISSUES

- **ENTMQBR-785 - Changing a queue attribute can result in lost messages**
Previously, using the `artemis queue update` command to modify queue attributes resulted in messages being lost or removed when the broker was restarted. This issue occurred because the ID of the queue was not preserved when the broker was restarted. This issue is resolved in the 7.0.3 release.
- **ENTMQBR-782 - Queue JMX properties not matching up**
When running a version of AMQ Broker earlier than 7.0.3, it was possible under high load that the queue JMX attributes `Message count`, `Messages added`, and `Message acknowledged` could become out of sync. As a result, sometimes in AMQ Console, higher values for `Messages added` than `Message acknowledged` were displayed, while `Message count` was 0 and all messages were correctly consumed.

This issue is resolved in the 7.0.3 release. For more information, see [Solution #3244211](#).
- **ENTMQBR-610 - Feature Last-Value Queue is not working with AMQP protocol**
The Last Value Queue feature did not work with the AMQP protocol. This issue is resolved in the 7.0.3 release.
- **ENTMQBR-643 - Unable to log in to Hawtio console when the broker runs as a Windows service**
When installing and running the broker as a Windows service, users were unable to log in to the management console. This issue is resolved in the 7.0.2 release.
- **ENTMQBR-506 - Hawtio console stops working after backup node goes from live to backup.**
In a previous Beta release, a bug that prevented the HawtIO console from being usable on a back-up server existed. The issue occurred only if that server had been live before being relegated to a back-up. This issue is resolved in the 7.0.0 release.
- **ENTMQBR-498 - slow consumer: consumer is killed even though it receives at a rate higher than the threshold**
In a previous Beta release, brokers using the AMQP `slow-consumer - *` parameters could incorrectly close consumers that met the configured thresholds after the first scan interval. The issue is resolved in the 7.0.0 release.
- **ENTMQBR-77 - Expand error tree list**
In a previous Beta release, attempting to 'expand all nodes' or the reverse in the left hand tree of the console produced no result. Users had to step through the tree manually. This issue is resolved in the 7.0.0 release.

For additional details about issues resolved in specific releases, see the following articles:

- [AMQ 7 - 7.0.1 Resolved Issues](#)
- [AMQ 7 - 7.0.2 Resolved Issues](#)
- [AMQ 7 - 7.0.3 Resolved Issues](#)

CHAPTER 4. KNOWN ISSUES

- **ENTMQBR-897 - OpenWire client/protocol issues with special characters in destination name**

Some special characters are currently unsupported in queue and address names with the OpenWire protocol, specifically:

```
, # >
```

In addition, address and queue names cannot contain empty spaces.

- **ENTMQBR-884 - after connecting to hawtio "Artemis" panel is not displayed**

After connecting to AMQ Console, the Artemis panel is not displayed.

To work around this issue, replace the value `hawtio-no-slf4j.war` with `console.war` in the `etc/bootstrap.xml` file.

- **ENTMQBR-827 - AMQ Broker 7.0 permits cross-origin access to AMQ Console**
Currently, AMQ Broker permits cross-origin access to AMQ Console. To restrict access to AMQ Console, you must create an access management file that defines the permitted origin URLs, and then add a system property to apply the restrictions.

1. In the `BROKER_INSTANCE_DIR/etc/` directory, create a new `jolokia-access.xml` file.

2. Add a `<cors>` section:

```
<cors>
...
</cors>
```

3. Within the `<cors>` section, add an `<allow-origins>` section for each origin URL that you want to allow. For example:

```
<cors>
  <!-- Allow cross origin access from www.jolokia.org ... -->
  <allow-origin>http://www.jolokia.org</allow-origin>
  <!-- ... and all servers from jmx4perl.org with any protocol -->
  <allow-origin>*://*.jmx4perl.org</allow-origin>
  <!-- optionally allow access to web console from localhost -->
  <allow-origin>http://localhost:8181/*</allow-origin>
  <!-- Check for the proper origin on the server side, too -->
  <strict-checking/>
</cors>
```

4. Add a system property to apply the security restrictions you defined in the access management file:

```
$ export JAVA_ARGS="-
Djolokia.policyLocation=file:BROKER_INSTANCE_DIR/etc/jolokia-
access.xml"
```

- **ENTMQBR-803 - AMQ 7 ships with an example that is not supported**

AMQ Broker 7.0 ships with an example that is not supported. The `artemis-ra-rar` example is not supported because AMQ 7 does not currently include a RAR archive.

- **ENTMQBR-655 - [AMQP] Unable to send message when populate-validated-user is enabled**

The security feature, `populate-validated-user`, is not supported for messages produced using the AMQP protocol, as it would result in a violation of the AMQP specification due to the `user-id` being part of the immutable bare message.

If, however, this feature is enabled, the broker will currently fail to enqueue AMQP messages sent without a message **Application Properties** section, causing the message to be rejected.

- **ENTMQBR-652 - List of known amq-jon-plugin bugs**

This version of `amq-jon-plugin` has known issues with broker and queue MBeans.

Issues with the broker MBean:

- Closing a connection throws `java.net.SocketTimeoutException` exception
- `listSessions()` throws `java.lang.ClassCastException`
- Adding address settings throws `java.lang.IllegalArgumentException`
- `getConnectorServices()` operation cannot be found
- `listConsumersAsJSON()` operation cannot be found
- `getDivertNames()` operation cannot be found
- Listing network topology throws `IllegalArgumentException`
- Remove address settings has wrong parameter name

Issues with the queue MBean:

- `expireMessage()` throws `argument type mismatch` exception
 - `listDeliveringMessages()` throws `IllegalArgumentException`
 - `listMessages()` throws `java.lang.Exception`
 - `moveMessages()` throws `IllegalArgumentException` with error message `argument type mismatch`
 - `removeMessage()` throws `IllegalArgumentException` with error message `argument type mismatch`
 - `removeMessages()` throws exception with error `Can't find operation removeMessage with 2 arguments`
 - `retryMessage()` throws `argument type mismatch IllegalArgumentException`
 - **ENTMQBR-648 - JMS OpenWire client is unable to send messages to queue with defined `purgeOnNoConsumer` or queue filter**
- Using AMQ6 JMS client to send messages to an address with an appropriately configured queue with `purgeOnNoConsumer=true` fails if the queue has no consumers. It is recommended that users do not use the `purgeOnNoConsumer` functionality when used with JMS clients.

- **ENTMQBR-636 - Journal breaks, causing `JavaNullPointerException` under high load**
Issues can occur doing IO when the broker is under high load and if certain JVM tunings have not been set correctly. To avoid these issues, ensure that you allocate enough memory and heap space for the broker. See the section titled "Tuning the VM" in the Performance Tuning chapter of the ActiveMQ Artemis documentation at <https://activemq.apache.org/artemis/docs/latest/perf-tuning.html>.
- **ENTMQBR-617 - Cert based authentication does not work with AMQP and OpenWire clients**
Authentication based on SSL certificates is not supported for AMQP and OpenWire clients.
- **ENTMQBR-599 - Define truststore and keystore by artemis CLI**
Using the `--ssl-key`, `--ssl-key-password`, `--ssl-trust`, and `--ssl-trust-password` parameters from the command-line interface when creating an instance of the broker, do not work. To work around this problem, set the corresponding properties manually in the `bootstrap.xml` file.
- **ENTMQBR-569 - Conversion of IDs from OpenWire to AMQP results in sending IDs as binary**
When communicating cross-protocol from an A-MQ 6 OpenWire client to an AMQP client, additional information is encoded in the application message properties. This is benign information used internally by the broker and can be ignored.
- **ENTMQBR-533 - Large message support is documented, but supported only with the Core protocol**
Although the product guide titled *Using AMQ Broker* contains a chapter about working with large messages, this feature is not yet supported for all protocols. Specifically, large messages are supported only with the core protocol at this time.
- **ENTMQBR-520 - Receiving from address named same as queue bound to other address should not be allowed**
A queue with a name that matches any address should only ever be assigned to the matching address. Creating a queue with the same name as an existing address, but bound to an address with a different name, is an invalid configuration. The result of doing so can result in incorrect messages being routed to the queue.
- **ENTMQBR-463 - Attributes in clustering settings has order restrictions, would be nice to either have better error message or simply ignore the order**
Currently the sequence of the elements in the cluster connection configuration has to be in a specific order. The workaround is to adhere to the order in the configuration schema.
- **ENTMQBR-17 - AMQ222117: unable to start cluster connection**
In some environments that support IPv6, a broker cluster may not form properly due to a `SocketException` as indicated by a message "Can't assign requested address." To work around this issue, set the `java.net.preferIPv4Stack` system property to `true`.

Revised on 2017-12-08 15:17:08 EST