



Red Hat AMQ 7.2

AMQ Broker 7.2 Release Notes

Release Notes for AMQ Broker

Red Hat AMQ 7.2 AMQ Broker 7.2 Release Notes

Release Notes for AMQ Broker

Legal Notice

Copyright © 2018 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 7.2 release.

Table of Contents

CHAPTER 1. FEATURES	3
Persistence	3
Security	3
Performance	3
CHAPTER 2. ENHANCEMENTS	4
CHAPTER 3. RESOLVED ISSUES	6
CHAPTER 4. KNOWN ISSUES	8
CHAPTER 5. IMPORTANT LINKS	11

CHAPTER 1. FEATURES

This section outlines the major features added in AMQ 7.1 and later.

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-428 - Implement SASL external**

AMQ Broker can now validate AMQP connections by using the identity of the connection's SSL certificate rather than using traditional credentials such as a username and password. This benefits deployments where secure access is managed through SSL certificates.

- **ENTMQBR-893 - Masking of password does not work with `login.config`**

You can now mask passwords in the JAAS configuration file `login.config`. Previously, password masking was not supported for `login.config`, which meant that the passwords in the file were stored in plain text.

- **ENTMQBR-908 - Support masked passwords in `bootstrap.xml`**

You can now mask passwords in the `broker.xml` configuration file. Previously, password masking was not supported for `broker.xml`, which meant that the passwords in the file were stored in plain text.

- **ENTMQBR-935 - Should have Ability to split `broker.xml` into multiple files and import into `broker.xml`**

AMQ Broker now supports XML inclusions in the `broker.xml` configuration file, which enables you to break `broker.xml` into separate files. For example, if you have multiple brokers that share the same set of addresses and address settings, you can define the addresses in a separate file, and then include that file in each broker's `broker.xml` configuration file.

- **ENTMQBR-1056 - Add Support for DB2 v11.latest**

The 7.2 version of AMQ Broker adds support for the database DB2, version 11.

- **ENTMQBR-1063 - Improve DB2 compatibility**

AMQ Broker now supports IBM DB2 versions 10.5 and 11.1 for message storage. The following enhancements have been made:

- **AUTO_INCREMENT** is now supported on DB2.
- You can now configure the maximum allowed time difference between the system time and DB2 time. This makes high availability more reliable, because the broker will be shut down automatically if the time difference exceeds the configured value.
- Database tables can now be created externally. Concurrent table creation checking has been improved.

- **ENTMQBR-1271 - Allow custom `etc` and data directories to be used**

AMQ Broker 7.2 provides greater flexibility in storing a broker instance's configuration files and data. Previously, a broker instance's configuration files and data could only be stored in the `etc/` and `data/` directories within the broker instance's directory. Now, you can store these files in any custom directory, including locations outside of the broker instance's directory.

- **ENTMQBR-1282 - Support for exclusive consumers**

AMQ Broker now supports exclusive queues. An exclusive queue routes messages to a single consumer even if multiple consumers are present. In addition, the consumer that receives the messages will receive them with guaranteed message ordering so that they can be consumed in the same order in which they were produced.

- **ENTMQBR-1308 - Create an `ActiveMQServerPlugin` for logging various broker events**

Starting in AMQ Broker 7.2, logging has been improved through a new `LoggingActiveMQServerPlugin`. By using this plugin, you can configure highly-informative

logging about many different types of events that would otherwise only be available through debug logs.

- **ENTMQBR-1367 - Support SSL Certificate Revocation List**

Starting in AMQ Broker 7.2, you can now specify an SSL certificate revocation list on an acceptor's URL. If you are using SSL for client communication, this means that you can specify a list of certificates that have been revoked by the issuing certificate authority (CA) before their scheduled expiration date and should no longer be trusted.

- **ENTMQBR-1385 - Add support for virtual topic consumers**

AMQ Broker now supports migrating existing OpenWire clients that use virtual topic subscriptions (an ActiveMQ 5.x-only feature) without requiring any changes to the client. This functionality maps the virtual topic concept to an Artemis fully qualified queue name (FQQN). By using a FQQN, you can create and access a standard Artemis topic subscriber queue.

- **ENTMQBR-1403 - Support referrals in LDAP login module**

AMQ Broker now supports LDAP referrals. If you use LDAP referrals in your LDAP infrastructure, you can now configure the broker to use these referrals to authenticate and authorize users. You can use the broker's JAAS LDAP login module to control how the broker handles LDAP referrals.

CHAPTER 3. RESOLVED ISSUES

- **ENTMQBR-742 - JMS Queues are not being auto-deleted for Openwire and AMQP clients**
Previously, addresses and queues with `<auto-delete-addresses>` and `<auto-delete-queues>` set to `true` were not deleted when AMQ OpenWire JMS clients disconnected from them.
- **ENTMQBR-914 - [AMQ7, broker startup] AMQ224000: Failure in initialization: java.lang.IllegalStateException: Cannot find queue with id XXXX**
In previous versions of AMQ Broker, the broker would fail to start if you had an XA transaction in the prepared state, then deleted the queue that this transaction used, and then attempted to restart the broker.
- **ENTMQBR-929 - LDAPLoginModule cannot process referrals**
Previously, the LDAP JAAS login module was unable to handle LDAP referrals, which caused authentication and authorization failures.
- **ENTMQBR-930 - Unable to login with multiple LDAP modules configured**
The commit operation of the LDAP JAAS login module would always return `null`, resulting in unexpected behavior when multiple instances of the module were configured in the same domain.
- **ENTMQBR-943 - [AMQ7, Openwire, Compression] consuming Openwire compressed bytemessage throws java.util.zip.DataFormatException: incorrect header check**
Previously, when using the OpenWire protocol to send small, compressed ByteMessages that have JMS properties set for a queue, an exception was thrown on the consumer side when it attempted to decompress the message. See the Knowledge Base article on the Red Hat Customer Portal for more details: <https://access.redhat.com/solutions/3269061>.
- **ENTMQBR-966 - Unsettled AMQP messages are lost when Receiver Link is opened on remote cluster member**
An issue causing message loss has been fixed in this release. Previously, if messages were sent to a broker using the AMQP address, and the address was not set on the messages, then some of the messages could be lost if they were redistributed.
- **ENTMQBR-967 - [AMQ 7.1.0 CR1.1] Limit non-ssl connection, handshake-timeout not configurable**
Previously, the broker did not disconnect unauthenticated clients. With AMQ Broker 7.2, you can use the configuration parameter `handshake-timeout` to limit the amount of time that an unauthenticated client can remain connected.
- **ENTMQBR-973 - Incorrect message priority displayed in hawtio console**
When viewing messages in AMQ Console, the message priority is now correct. Before, the message priority incorrectly defaulted to `4`.
- **ENTMQBR-1016 - [AMQ7,Hawtio]AMQ 7 hawtio console store users password in browser's local cache after user get logout**
A security issue has been fixed for AMQ Console. Before, if you logged into AMQ Console, the value of the `Password` field was visible from local storage in Google Chrome Developer tools.
- **ENTMQBR-1018 - When live-slave fails-back to master, it turns off everything down, even its console**
In high-availability configurations, AMQ Console is now accessible when a slave broker returns control to the master broker. Previously, AMQ Console would become unavailable for the slave broker when it gave control back to the master broker.

- **ENTMQBR-1030 - Restrict directory listings of Hawtio within the web server configuration**
AMQ Console no longer permits access to restricted directory listings.
- **ENTMQBR-1130 - Destinations undeployed when master recovers from outage**
When adding destinations to a broker's configuration file (**broker.xml**) at runtime, the destinations are now preserved in the configuration file and reloaded if the broker is restarted. Previously, if you added destinations to a broker's configuration file, the destinations would not be reloaded when the broker was restarted.
- **ENTMQBR-1184 - LargeMessage Produced by AMQP Protocol Can Not Be Consumed By AMQP Protocol**
In previous releases, if the size of an AMQP JMS Object Message was greater than the value specified for the maximum journal record size, an exception was thrown on the broker and the consumer was not able to receive the message. This issue was caused by a problem in the AMQP large message to core message conversion process.

This issue is fixed and AMQP large messages can be sent and received as usual.
- **ENTMQBR-1461 - AMQP: IndexOutOfBoundsException when dispatching ObjectMessage that was handled as a Large Message**
Previously, if you were using the AMQP protocol with the Qpid JMS client, and you sent a JMS **ObjectMessage** that was also a large message (larger than the **min-large-message-size**), an error would occur when the message was consumed. This error no longer occurs.
- **ENTMQBR-1466 - [3 HA pairs] Slave does not become live after master is killed and isolates itself**
The quorum voting protocol has been corrected. Previously, in high-availability configurations consisting of three high-availability pairs with replication, this issue occasionally prevented slave brokers from taking over during a failover event. Instead, the slave broker would become isolated from the broker cluster.
- **ENTMQBR-1500 - Jolokia read request does not fetch all attributes**
Using Jolokia, it is now possible to request all of the attributes for the broker MBean.
- **ENTMQBR-1548 - Implementation of AMQP interceptor is passing a null RemotingConnection reference**
Previously, if you were using the interceptor API with the AMQP protocol, and you implemented the following method, the connection parameter was always null:


```
public boolean intercept(AMQPMessage message, RemotingConnection connection)
```


Now, the connection parameter is properly set.
- **ENTMQBR-1757 - AMQ Broker throws ERROR if we update the address ANYCAST to MULTICAST or vice-versa in broker.xml**
Previously, if the broker was stopped and the routing type for an existing address with queues was changed in the **broker.xml** file, the broker would fail to restart. The relevant code for updating the configuration has been modified so that such a configuration change is possible, and even if there is an error deploying the address or queue, the broker will log the error and still start.

For additional details about issues resolved in maintenance releases, see the following article:

- [AMQ 7 - 7.2.1 Resolved Issues](#)

CHAPTER 4. KNOWN ISSUES

- **ENTMQBR-17 - AMQ222117: Unable to start cluster connection**

A broker cluster may fail to initialize properly in environments that support IPv6. The failure is due to a `SocketException` that is indicated by the log message **Can't assign requested address**. To work around this issue, set the `java.net.preferIPv4Stack` system property to `true`.

- **ENTMQBR-463 - Attributes in clustering settings have 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-520 - Receiving from address named the same as a queue bound to another address should not be allowed**

A queue with the same name as an address must only be assigned to 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. Doing so can result in incorrect messages being routed to the queue.

- **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-599 - Define truststore and keystore by Artemis cli**

Creating a broker instance by using the `--ssl-key`, `--ssl-key-password`, `--ssl-trust`, and `--ssl-trust-password` parameters does not work. To work around this issue, set the corresponding properties manually in `bootstrap.xml` after creating the broker.

- **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-636 - Journal breaks, causing `JavaNullPointerException`, under perf load (mpt)**

To prevent IO-related issues from occurring when the broker is managing heavy loads, verify that the JVM is allocated with enough memory and heap space. See the section titled "Tuning the VM" in the [Performance Tuning](#) chapter of the ActiveMQ Artemis documentation.

- **ENTMQBR-648 - JMS Openwire client is unable to send messages to queue with defined `purgeOnNoConsumer` or queue filter**

Using an A-MQ 6 JMS client to send messages to an address that has a queue with `purgeOnNoConsumer` set to `true` fails if the queue has no consumers. It is recommended that you do not set the `purgeOnNoConsumer` option when using A-MQ 6 JMS clients.

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

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

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-655 - [AMQP] Unable to send message when populate-validated-user is enabled**
The configuration option `populate-validated-user` is not supported for messages produced using the AMQP protocol.
- **ENTMQBR-781 - [Error Codes] same Error Code(s) defined in different places with different reasons**
Logging error codes are not unique for all broker log messages. Use the logger name, `org.apache.activemq.artemis.core.server` for example, to identify uniqueness.
- **ENTMQBR-882 - Standby slave does not announce replication to master when primary slave is down**
This issue occurs when multiple backup brokers, also referred to as slaves, are serving a single live (master) broker. If a primary backup broker fails, the secondary backup tries to replicate. But that operation fails, the secondary backup cannot take over for the primary backup, and as a result, high-availability is lost.
- **ENTMQBR-897 - Openwire client/protocol issues with special characters in destination name**
Currently AMQ OpenWire JMS clients cannot access queues and addresses that include the following characters in their name: comma (','), hash ('#'), greater than ('>'), and whitespace.
- **ENTMQBR-944 - [A-MQ7, Hawtio, RBAC] User gets no feedback if operation access was denied by RBAC**

The console can indicate that an operation attempted by an unauthorized user was successful when it was not.

- **ENTMQBR-945 - Non-persistent messages lost in non-failure scenarios when authorization fails, because delivery mode defaults to asynchronous**

Clients may not notice the loss of non-persistent messages. Message loss is often associated with an obvious failure, such as the broker stopping or the storage provider disconnecting. However, the asynchronous behavior of the default delivery mode can lead to the loss of non-persistent messages in other situations too, such as failing authorization or trying to use a queue that is not configured.

- **ENTMQBR-956 - [AMQ7, AMQP, JMX, DeliveringCount] JMX operation `listDeliveringMessages()` returns empty string when `deliveringCount > 0`**

The JMX operation `listDeliveringMessages()` does not return messages that were consumed in a transaction by an AMQ JMS client.

- **ENTMQBR-965 - [AMQ7, openwire, exclusive consumer] AMQ6 openwire clients using exclusive consumer, do not behave the same with an AMQ7 broker**

In AMQ 7, setting the maximum number of consumers for a queue to one does not work the same as an "exclusive consumer" in AMQ 6. In AMQ 6, an exclusive consumer means that one consumer will get all the messages. Any other consumers are still registered, just not consuming messages. When the current exclusive consumer leaves the queue, another consumer becomes the exclusive consumer.

In AMQ 7, if you set `max-consumers = 1` for a queue, when this limit is reached, the broker returns a **AMQ119200** error and does not allow any new consumers to join the queue.

- **ENTMQBR-1045 - "Client connection failed" message in broker log, with OpenWire client**

Currently, when an OpenWire client (ActiveMQ) disconnects cleanly, the broker shows a warning message similar to the following:

```
Client connection failed, clearing up resources for session
ID:localhost-45795-1527065581471-5:1:1
```

This warning can be safely ignored.

CHAPTER 5. IMPORTANT LINKS

- [Red Hat AMQ 7 Supported Configurations](#)
- [Red Hat AMQ 7 Component Details](#)
- [Red Hat JBoss AMQ 7.1 Release Notes](#)

Revised on 2018-10-12 18:24:23 UTC