



## Red Hat AMQ 7.5

# Release Notes for Red Hat AMQ Broker 7.5

Release Notes for AMQ Broker



# Red Hat AMQ 7.5 Release Notes for Red Hat AMQ Broker 7.5

---

Release Notes for AMQ Broker

## Legal Notice

Copyright © 2020 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, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

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

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

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

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

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

The OpenStack<sup>®</sup> 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 7.5 release.

---

## Table of Contents

<b>CHAPTER 1. FEATURES</b> .....	<b>3</b>
Persistence .....	3
Security .....	3
Performance .....	3
<b>CHAPTER 2. ENHANCEMENTS</b> .....	<b>4</b>
<b>CHAPTER 3. DEPRECATED FEATURES</b> .....	<b>8</b>
<b>CHAPTER 4. TECHNOLOGY PREVIEW</b> .....	<b>9</b>
<b>CHAPTER 5. FIXED ISSUES</b> .....	<b>10</b>
<b>CHAPTER 6. KNOWN ISSUES</b> .....	<b>18</b>
<b>CHAPTER 7. IMPORTANT LINKS</b> .....	<b>22</b>



---

# 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.

### Multi-site fault tolerance

You can use specific broker topologies and the Red Hat Ceph Storage software-defined storage platform to ensure continuity of your messaging system during a data center outage. This type of solution is called a *multi-site, fault-tolerant architecture*. For more information, see [Configuring a multi-site, fault-tolerant messaging system](#).

### 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-831 - Broker should respond to changes in logging.properties at run-time**

AMQ Broker now retains the changes that you make to the **logging.properties** configuration file without restarting the broker.
- **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-945 - Non-persistent messages lost in non-failure scenarios when authorization fails, because delivery mode defaults to asynchronous**

AMQ Broker now enables you to configure notifications for loss of non-persistent messages by clients. Obvious failure conditions such as broker stoppage or disconnection of a storage provider can cause this type of loss. However, the default delivery mode for non-persistent messages, which is asynchronous, can also cause message loss in situations such as failed authorization or use of a non-configured queue. You can now use the **CompletionListener** API to configure notifications for such message-loss events.
- **ENTMQBR-1002 - Enable the pause method for an address**

AMQ Broker 7.5 enables you to pause an address, which pauses all of the queues bound to that address. New queues that you add to this address are also paused until you resume the address. To pause or resume an address, use the Core JMS API to implement the **pause()** or **resume()** methods on an **AddressControl** interface. For more information, see [AddressControl](#) in the Core JMS API documentation.
- **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 anActiveMQServerPlugin 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-1370 - Adduser attribute to queue in XML/JMX**

When a queue is auto-created it is assigned the user name of the connecting client. This user name is now exposed as metadata on the queue and also in JMX and in the AMQ Console. You can also now configure a user on a pre-defined queue by modifying the **broker.xml** file. This enhancement makes it possible to understand and track with whom and how queues originated.
- **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.
- **ENTMQBR-1648 - Don't hardcode openshift.DNS\_PING service name**

AMQ Broker 7.5 enables you to specify a unique value for the OpenShift DNS ping service for each application template deployment that you make in a given OpenShift project. This enhancement enables you to deploy multiple broker clusters in the same OpenShift project. For more information, see the description of the **PING\_SVC\_NAME** parameter in [Application template parameters](#).
- **ENTMQBR-1783 - Add a user and role manager (create/delete/update/list) JMX endpoint in AMQ 7**

AMQ Broker 7.3 supports adding, removing, and updating user information by using the management API when you are using the **PropertiesLoginModule** class. Previously, you could perform these actions only by using the command-line interface.

- **ENTMQBR-1918 - [AMQ7, critical-analyzer] add extra detail to output for analyzing cause of timeout**  
The logs for the Critical Analyzer now provide extra information about a slow-performing broker when you enable tracing on the package named **org.apache.activemq.artemis.utils.critical**.
- **ENTMQBR-1932 - Support AMQ 7 on Java 11**  
AMQ Broker 7.4 adds runtime support for JDK 11 Java Virtual Machines (JVMs).
- **ENTMQBR-1974 - Support masked passwords in management.xml**  
AMQ Broker now supports masking the values of the **key-store-password** and **trust-store-password** settings in the **management.xml** file. These values are used when you secure remote JMX access with SSL.
- **ENTMQBR-2020 - Track routed and unrouted messages to an address**  
Address metrics now include information about how many messages the address has received. This number is divided between routed and unrouted messages. Prior to this enhancement, it was impossible to determine if an address had received a message which was not routed, for example, in the case of a JMS topic with no subscriptions.
- **ENTMQBR-2025 - Allow users to define custom last value queue keys**  
AMQ Broker 7.4 supports last value queues with custom last value key names. Previously, last value queues used only the default last value key name of **\_AMQ\_LVQ\_NAME**.
- **ENTMQBR-2139 - Support MSSQL for database message persistence**  
AMQ Broker 7.4 supports Microsoft SQL Server 2016 for JDBC-based persistence.
- **ENTMQBR-2173 - Support a Prometheus plugin or agent**  
AMQ Broker 7.4 includes a pre-configured plugin for Prometheus. Prometheus is software built to monitor large, scalable systems and store historical runtime data over an extended time period. The AMQ Broker Prometheus plugin exports broker runtime metrics to Prometheus format, enabling you to use Prometheus itself to visualize and run queries on the data.
- **ENTMQBR-2299 - An option to reject messages without a JMSUserID**  
Starting in AMQ Broker 7.4, you can track and log the origins of messages using the **\_AMQ\_VALIDATED\_USER** message key. You can also configure the broker to reject a message that does not have a user name already populated by the client when it sends the message.
- **ENTMQBR-2345 - Enhancements to the databaseStoreType**  
In AMQ Broker 7.4, you can add an encrypted user name and password when configuring a **database-store** element in your **broker.xml** configuration file.
- **ENTMQBR-2687 - Add new ActiveMQArtemis CRD version to reflect actual configuration schema**  
In AMQ Broker 7.5, the Custom Resource Definition (CRD) files used by the AMQ Broker Operator have been redesigned. This enhancement makes the CRDs more consistent with the **broker.xml** configuration file that you use to configure on-premise broker deployments. For more information, see [Overview of Custom Resource Definitions](#).
- **ENTMQBR-2846 - Operator-based deployments larger than one broker are clustered by default**  
In the 7.5 version of the AMQ Broker Operator, a deployment of two or more brokers is

clustered by default. In the 7.4 version of the Operator, you needed to explicitly include a **clusterConfig** configuration element in the Custom Resource (CR) that you created for your deployment.

- **ENTMQBR-2921 - Scaledown controller no longer explicitly deployed via Custom Resource**  
In the 7.5 version of the AMQ Broker Operator, if you intend to enable message migration in your broker deployment, you no longer need to deploy a dedicated Custom Resource (CR) instance for the scaledown controller. Now, if you set **messageMigration** to **true** in your main broker CR, the Operator automatically instantiates and starts the scaledown controller to prepare for message migration. If you set **messageMigration** to **false**, the Operator does not instantiate the scaledown controller. If you do not explicitly set a value for **messageMigration**, the Operator sets a default value of **true**. In this case, the Operator also automatically instantiates and starts the scaledown controller.
- **ENTMQBR-3126 - Added AMQ Broker support for running within OpenShift Container Platform on IBM Z**  
AMQ Broker 7.5 adds broker and Operator container images that support RHEL 8 on IBM Z. This added support enables you to create broker deployments within OpenShift Container Platform on IBM Z. For more information, see [Deploying AMQ Broker on OpenShift Container Platform Using an Operator](#).

## CHAPTER 3. DEPRECATED FEATURES

- Starting in 7.3, AMQ Broker no longer ships with the Hawtio dispatch console plugin, **dispatch-hawtio-console.war**. Previously, the dispatch console was used to manage AMQ Interconnect. However, AMQ Interconnect now uses its own, standalone web console.
- Starting in 7.5, network pinging is a deprecated feature. Network pinging cannot protect a broker cluster from network isolation issues that can lead to irrecoverable message loss. This feature will be removed in a future release. Red Hat continues to support existing AMQ Broker deployments that use network pinging. However, Red Hat no longer recommends use of network pinging in new deployments. For guidance on configuring a broker cluster for high availability and to avoid network isolation issues, see [Implementing high availability](#).

## CHAPTER 4. TECHNOLOGY PREVIEW



### IMPORTANT

Technology Preview features are not supported with Red Hat production service-level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them for production. For more information, see [Red Hat Technology Preview Features Support Scope](#).

- Support for federated queues and addresses is available as a Technology Preview feature only. Federation enables you to transmit messages between brokers without the brokers being part of a cluster. For more information, see [Federation](#) in the Apache ActiveMQ Artemis documentation.

## CHAPTER 5. FIXED 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-781 - [Error Codes] same Error Code(s) defined in different places with different reasons**  
In previous releases of AMQ Broker, logging error codes were not unique for all broker log messages. The duplicate and conflicting logging error codes are now corrected.
- **ENTMQBR-882 - Standby slave does not announce replication to master when primary slave is down**  
This issue occurred in past releases when multiple backup brokers, also referred to as slaves, served a single live (master) broker. In this scenario, if a primary backup broker failed, the secondary backup tried to replicate. But that operation failed, the secondary backup could not take over for the primary backup, and as a result, high-availability was lost. This issue is now resolved.
- **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-956 - [AMQ7, AMQP, JMX, DeliveringCount] JMX operation listDeliveringMessages() returns empty string whendeliveringCount > 0**  
In previous releases, the JMX operation `listDeliveringMessages()` did not return messages that were consumed by an AMQ JMS client in a transaction. The transactional messages are now returned correctly when the JMX operation `listDeliveringMessages()` is called.
- **ENTMQBR-965 - [AMQ7, openwire, exclusive consumer] AMQ6 OpenWire clients using exclusive consumer do not behave the same with an AMQ 7 broker**  
In AMQ 7, setting the maximum number of consumers for a queue to one did not work the same as for an *exclusive consumer* in AMQ 6. In AMQ 6, using an exclusive consumer means that one consumer will receive all of 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.

Previously in AMQ 7, if you set **max-consumers = 1** for a queue, when this limit is reached, the broker returned a **AMQ119200** error and did not allow any new consumers to join the queue. This error no longer occurs.

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

Previously, when an OpenWire client (ActiveMQ) disconnected cleanly, the broker showed a warning message similar to the following:

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

This issue is now resolved.
- **ENTMQBR-1061 - Exception when using LegacyLDAPSecuritySettingsPlugin**

Previously, when using LegacyLDAPSecuritySettingsPlugin, if you specified authorization settings with non-standard, but still valid syntax, you saw an exception. This is because the broker interpreted LDAP search results based on hard-coded key names, not on order. This issue is corrected.
- **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-1169 - [AMQ 7, Hawtio, large message] Sending very large messages through Hawtio results in an exception**  
Previously, if you used the Hawtio console to send a message exceeding the broker's internal buffer size, you saw an exception. This issue is corrected. The broker now automatically converts this type of message to a large message and sends it successfully.
- **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-1699 - Cannot set the message content when using/artemis producer**  
Earlier releases of AMQ Broker 7.x did not contain an option to specify the message content when using the **./artemis** producer, as the A-MQ 6.x product did. Now, the **--message** parameter is available and you can use it to specify the content of the message.
- **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.
- **ENTMQBR-1816 - Queue#MessageCount negative and messages stop flowing to AMQP consumer**



Previously, an AMQP client could send a disposition for a delivered message indicating that the client received the message but didn't consume it. This behavior could cause the message count on the broker to have a negative value. This issue is now resolved.

- **ENTMQBR-1878** - [AMQ7, broker shutdown, JMX connector] JVM does not exit if connector configured in management-context

In previous releases, sometimes the broker did not properly close the management context when stopping the broker. The broker would appear to be stopped, but the JVM process never fully exited. Now there are checks to ensure that the management context is always closed when stopping the broker and the JVM process exits, even when a management context **connector** is configured.
- **ENTMQBR-1880** - Hawtio Artemis tab disappears if we use IE

The AMQ Console Hawtio plug-in employs the Javascript arrow function, which is not supported by Internet Explorer. When using the AMQ Console, if you click the back or forward arrows in the browser window, the Artemis tab cannot be viewed in the console. To workaround this issue, use a different browser or use the navigation tools in the console application itself.
- **ENTMQBR-1958** - Queue preference changes to display columns not persistent through page refresh

Refreshing the AMQ Console caused changes to the default column lists of tabs to be lost. To maintain your settings, you had to refrain from refreshing the browser window. Now column list preferences are stored and reused each time you access a page and refreshing the browser does not cause settings to be lost.
- **ENTMQBR-1995** - Enabling TRACE-level logging in the broker leads to **ConcurrentModificationException** and failed operations

In previous releases, multiple threads could concurrently modify a message's properties. This resulted in **WARN** messages in the log when TRACE-level logging was enabled. Now, concurrent access to message properties is done safely and the **WARN** messages in the log related to this issue no longer occur.
- **ENTMQBR-2011** - Consumer of store-forward internal queues get dropped when syncing large messages

Using temporary destinations in a clustered environment caused messages to get dropped from store and forward queues when clustering messages between brokers. As a result, the broker was not able to distribute messages across cluster nodes and needed to be restarted. Now you can safely use temporary destinations in a clustered environment.
- **ENTMQBR-2014** - AMQ7:io.netty.handler.codec.DecoderException: java.lang.NullPointerException in broker log

When a client connected by using an unrecognized protocol, a **NullPointerException** with full stack-trace was logged. Now the AMQ Broker detects and handles unrecognized protocols more gracefully and a proper **ERROR** message is logged.
- **ENTMQBR-2024** - Broker ignores client requests for advisory consumers when advisory messages are disabled

In prior releases, when advisory messages were disabled on the broker, the broker ignored client requests to create advisory consumers and made no record of these consumers. If clients attempted to query these consumers, you saw an exception. This issue is corrected. The broker now creates records of advisory consumers when clients request them, and subsequent operations are successful.
- **ENTMQBR-2052** - Message load balancing off, unable to send messages to queue on other nodes instead of subscribed consumer

In releases prior to AMQ Broker 7.2.3, a recursive call would result in a **StackOverflowError**

error message in certain conditions when message load balancing was set to **OFF**. As a result, the broker would not receive messages sent from clients. In AMQ Broker 7.2.3, a verification step has been added to ensure that a recursive call will not become infinite. Now, brokers receive any message sent by a client even when message load balancing is set to **OFF**.

- **ENTMQBR-2055 - [Hawtio] console display blank ClientID for the core client**  
When using releases prior to AMQ Broker 7.2.3 with AMQ Console, the field named **Client ID** was not populated when viewing a AMQ Core Protocol JMS Client connection for a producer or consumer. Now, the value of Client ID is visible in AMQ Console for AMQ Core Protocol JMS Client connections.
- **ENTMQBR-2063 - Broker does not send sasl error when user attempts anonymous sasl and GuestLoginModule is not configured**  
In prior releases, if a client attempted an anonymous login to a broker that did not have guest access enabled, the broker produced an exception. However, this exception was not recognizable to the client as an authentication error. This issue is now resolved.
- **ENTMQBR-2074 - Retry from DLQ and Expiry Queue is broken on web console (Hawtio)**  
Previously, the broker ignored messages retried for delivery from the DLQ. These messages instead stayed on the DLQ. This issue is corrected. The broker now correctly adds redelivered messages from the DLQ to the original queue.
- **ENTMQBR-2085 - [AMQ7, JMSXGroupSeq] Support JMSXGroupSeq -1 to close/reset Groups**  
Previously, the broker did not remove message groups as expected when a client set JMSXGroupSeq to -1, per ActiveMQ 5. As a result, you could not change a message group once you pinned it to a consumer and broker. This issue is resolved.
- **ENTMQBR-2086 - [AMQ 7, JMS Selectors] JMS Selectors broken in some cases**  
With each supported protocol, fields such as **JMSXGroupID** or **AMQGroupID** were not properly parsed by the filter parser. As a result, you could not create a filter using any query that used those field names. To correct this issue, wrappers to parse these pseudo-fields were implemented and now filters correctly parse the input.
- **ENTMQBR-2100 - AMQ 7.2.1: under heavy load: org.apache.activemq.artemis.core.paging.cursor.NonExistentPage: Invalid messageNumber passed**  
While a receiving transaction is committed to a paging operation, if the page is complete, a transaction operation deletes it. Previously, the transaction operation might delete the page before the receiving transaction could access it. This caused the receiving transaction to fail to find a message. In this case, you saw an exception. This issue is now resolved.
- **ENTMQBR-2117 - [AMQ7,Hawtio] Unable to delete queue with single quote from Hawtio console**  
In prior releases, if you used the Hawtio console to delete a message containing a single quote, the broker did not successfully delete the message. This issue is now corrected.
- **ENTMQBR-2147 - (7.2.z) Backup doesn't activate after shared store is reconnected**  
Previously, if you had a live-backup broker pair configured for high availability using shared store, activation of the backup broker upon shutdown of the live broker could fail. Specifically, this situation occurred if the shared store had previously been disconnected and reconnected, before shutdown of the live broker. This issue is now resolved.
- **ENTMQBR-2177 - Allow deleting temporary destination when originating session is closed**

Previously, if you used the JMS API to delete a temporary destination when the originating session was already closed, the operation seemed to be successful, when it was not. This issue is now corrected.

- **ENTMQBR-2325 - JMSBridge fails to stop after throwing an error**  
 Previously, a JMS bridge between JBoss EAP and a broker not using Artemis Core Protocol produced an exception while stopping if the broker was already stopped. As a result of this exception, neither the EAP server nor the JMS bridge could successfully stop. This issue is now resolved.
- **ENTMQBR-2326 - JMSBridge fails to stop**  
 Previously, a JMS bridge between JBoss EAP and a broker not using Artemis Core Protocol produced an exception while stopping. As a result of this exception, neither the EAP server nor the JMS bridge could successfully stop. This issue is now resolved.
- **ENTMQBR-2333 - Temporary Queue Address Leak In Clustered Configuration**  
 Previously, if a temporary queue was replicated across cluster nodes, closing connections to the temporary queue on a given node did not purge the server resources associated with the queue. In other words, the queue had an address leak. This issue is resolved. You can now use the **address-queue-scan-period** parameter to specify how often the broker scans queues for potential deletion.
- **ENTMQBR-2337 - Failover or similar option for AMQ Core Protocol JMS Client**  
 In prior releases, an AMQ Core Protocol JMS Client could make repeated attempts to connect to an unavailable broker server, even if alternative broker servers were specified in the client's **ConnectionFactory** configuration. Specifically, this situation occurred if you set the **initialConnectAttempts** option to **-1**. This type of configuration is shown in the example below.

```
connectionFactory.ConnectionFactory=(tcp://host1:61616,tcp://host2:61716)?
initialConnectAttempts=-1
```

This issue is now resolved. If the first listed server that the client tries is unavailable, the client immediately tries the next, and so on. The client repeats this loop until it establishes a broker connection.
- **ENTMQBR-2386 - Slow consumers not disconnected if producer msg/s rate is low**  
 Previously, when checking for slow consumers, the broker calculated only messages that a producer added during the check period. The broker did not take into account existing messages in the queue at the start of the check period. This could cause the broker to miss legitimate cases of slow message consumption. This situation could arise, for example, if a producer previously sent large numbers of messages to the queue in bursts, but was idle during the slow consumer check period.
- **ENTMQBR-2464 - Qpid JMS client throws incorrect exception when sending message to activemq.management**  
 In prior releases, when you used the AMQP Qpid JMS client to send a message to the internal queue **activemq.management**, you saw an exception. This issue is now resolved.
- **ENTMQBR-2467 - [AMQ7, large header] message with very large header shuts broker down with "AMQ222010: Critical IO Error, shutting down the server.."**  
 In prior releases, if a client sent a message with a header larger than the value of the **journal-buffer-size** parameter, you saw an exception that caused the broker to shut down. This issue is resolved. The broker now rejects messages with a header larger than the value of the **journal-buffer-size** parameter.
- **ENTMQBR-2468 - java.lang.NullPointerException with message replication**

When the values of the **config-delete-addresses** and **config-delete-queues** parameters of your broker configuration are set to **force**, the broker reloads the **broker.xml** configuration file after a change. When the broker reloads the configuration, any auto-created, clustered queues are deleted. In prior releases, this deletion caused a failure that prevented the broker from starting. This issue is now resolved. For any queues that are deleted, the broker catches an exception and logs a warning, but starts as expected.

- **ENTMQBR-2476 - Live server does not shutdown when using vote-on-replication-failure**

Previously, in a cluster of three or more live-backup groups that was using the replication high availability (HA) policy, the live broker might not shut down when its replication connection failed. Specifically, this issue occurred when all of the following were true:

- The **vote-on-replication-failure** parameter in the HA policy configuration was set to **true**
- The live broker failed to receive a majority vote during quorum voting
- The backup broker became live before quorum voting took place

This issue is now resolved.

- **ENTMQBR-2498 - Message Grouping may have messages stuck if a consumer is busy during the delivery**

Previously, if you used message grouping, a message might not be delivered if the consumer was busy. In these cases, messages remained stuck on the queue until the broker was restarted. This issue is now resolved.

- **ENTMQBR-2526 - Clustering does not move messages to another node, when there is a divert**

Previously, in a broker cluster configured with diverts, messages were not successfully redistributed from an anycast queue on one broker to an anycast queue with active consumers at the same address on another broker. Instead, messages were dropped. This issue is now resolved.

- **ENTMQBR-2528 - MQTT Consumer gets java.io.EOFException when subscribed with QOS 2 for an address with custom security settings**

Previously, an MQTT consumer saw an exception when subscribed with quality of service (QoS) level 2 to an address with custom security settings. This issue is now resolved.

- **ENTMQBR-2575 - [AMQ 7.3.0] Hawtio - Can't send Message with Headers**

Previously, you could not use the Hawtio console to send messages with headers. This issue is now resolved.

- **ENTMQBR-2579 - Masked password does not work in LDAP security-setting-plugin**

In prior releases, if you used the **artemis mask** command to encrypt a password for inclusion in the LDAP **<security-setting-plugin>** configuration, the encrypted password did not work, even if the **mask-password** attribute in your **broker.xml** configuration file was set to **true**. As a result of this issue, you saw an LDAP authentication error. This issue is now resolved.

- **ENTMQBR-2601 - Statically created addresses may be auto-deleted**

Previously, the reaper thread, which deletes auto-created queues, could also delete queues that you statically defined in the **broker.xml** configuration file. This issue is now resolved. The reaper thread deletes only auto-created queues.

- **ENTMQBR-2675 - Attribute group-name ignored in replicated colocated configurations**

Previously, if you configured a cluster of three master-slave broker pairs to use high availability with colocated backups, the value of the **group-name** attribute of each broker was ignored. As

a result, when you started brokers in a new cluster configured this way, the first two master-slave pairs that you started immediately began replicating to each other, while the final master-slave pair that you started did not get a replication partner. The final master-slave pair then polled the cluster indefinitely, looking for a partner. This issue is now resolved.

- [ENTMQBR-2707](#) - Transaction rollback ordering is broken with AMQP

Previously, when rolling back AMQP transactional message data, the broker might fail to preserve the message order. This issue is now resolved.

- [ENTMQBR-2733](#) - JDBC HA shared store does not take credentials from the `jdbc-user` and `jdbc-password` tags

Previously, when you configured shared store high availability using JDBC persistence, the broker did not read the `jdbc-user` and `jdbc-password` values that you specified in the `database-store` configuration. To work around this issue, you could include the credentials directly in the `jdbc-Connection-URL` element. However, when specified in this way, the credentials were not encrypted. This issue is now resolved. The broker reads the encrypted values that you specify for `jdbc-user` and `jdbc-password`.

- [ENTMQBR-2876](#) - OpenWire Consumers are Not Cleaned up When the Session is Closed but Connection Remains Open

Previously, if an OpenWire client closed a session without closing associated consumers, the consumers were not cleaned up by the broker. Even if the client exited, an `OpenWireConnection` object remained on the broker, with a potentially large collection in `OpenWireConnection.consumerExchanges`. Over time, this situation exhausted the broker heap space and the broker entered an out-of-memory (OOM) state. This issue is now resolved.

For information about issues resolved in maintenance releases, see the following articles:

- [AMQ 7 - 7.4.1 Resolved Issues](#)
- [AMQ 7 - 7.2.4 Resolved Issues](#)
- [AMQ 7 - 7.2.3 Resolved Issues](#)
- [AMQ 7 - 7.2.2 Resolved Issues](#)
- [AMQ 7 - 7.2.1 Resolved Issues](#)

## CHAPTER 6. 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-522 - Broker running on windows write problems with remove temp files when shutting down**  
On Windows, the broker does not successfully clean up temporary files when it shuts down. This issue causes the shutdown process to be slow. In addition, temporary files not deleted by the broker accumulate over time.
- **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 `NullPointerException`, 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 `queuefilter`**  
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 wherpopulate-validated-user is enabled**  
The configuration option **populate-validated-user** is not supported for messages produced using the AMQP protocol.
  - **ENTMQBR-738 - Unable to build AMQ 7 examples offline with provided offline repo**  
You cannot build the examples included with AMQ Broker in an offline environment. This issue is caused by missing dependencies in the provided offline Maven repository.
  - **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-1498 - Diagram in management console for HA (replication, sharedstore) does not reflect real topology**  
If you configure a broker cluster with some extra, passive slaves, the cluster diagram in the web console does not show these passive slaves.

- **ENTMQBR-1815 - Hawtio view changes on automated refresh**

When automatic refresh is enabled, the Hawtio console updates the screen every 5 seconds. Automatic refreshes also change the view to the **Attributes** screen, causing you to lose focus from any other screen you are viewing. To work around this issue, select **Preferences** in the top-right corner of the console. In the **Update rate** drop-down list, select **No refreshes**.
- **ENTMQBR-1848 - "javax.jms.JMSEException: Incorrect Routing Type for queue, expecting: ANYCAST" occurs when qpid-jms client consumes a message from a multicast queue as javax.jms.Queue object with FQQN**

Currently, sending a message by using the Qpid JMS client to a multicast queue by using FQQN (fully qualified queue name) to an address that has multiple queues configured generates an error message on the client, and the message cannot be sent. To work around this issue, modify the broker configuration to resolve the error and unblock the client.
- **ENTMQBR-1875 - [AMQ 7, ha, replicated store] backup broker appear not to go "live" or shutdown after - ActiveMQIllegalStateException errorType=ILLEGAL\_STATE message=AMQ119026: Backup Server was not yet in sync with live**

Removing the paging disk of a master broker while a backup broker is trying to sync with the master broker causes the master to fail. In addition, the backup broker cannot become live because it continues trying to sync with the master.
- **ENTMQBR-1942 - [AMQ 7.2, shared store, scale down] NullPointerException exception when slave activates and tries to scale down**

If you configure a slave broker to use a shared store and scaledown in a cluster, you can encounter an exception on the slave if the master broker goes offline.
- **ENTMQBR-2068 - some messages received but not delivered during HA fail-over, fail-back scenario**

Currently, if a broker fails over to its slave while an OpenWire client is sending messages, messages being delivered to the broker when failover occurs could be lost. To work around this issue, ensure that the broker persists the messages before acknowledging them.
- **ENTMQBR-2452 - Upgraded broker AMQ 7.3.0 from AMQ 7.2.4 on Windows cannot log**

If you intend to upgrade a broker instance from 7.2.4 to 7.3.0 on Windows, logging will not work unless you specify the correct log manager version during your upgrade process. For more information, see [Upgrading from 7.2.x to 7.3.0 on Windows](#) .
- **ENTMQBR-2470 - [AMQ7, openwire,redelivery] redelivery counter for message increasing, if consumer is closed without consuming any messages**

If a broker sends a message to an Openwire consumer, but the consumer is closed before consuming the message, the broker wrongly increments the redelivery count for the pending message. If the number of occurrences of this behavior exceeds the value of the **max-delivery-attempts** configuration parameter, the broker sends the message to the dead letter queue (DLQ) or drops the message, based on your configuration. This issue does not affect other protocols, such as the Core protocol.
- **ENTMQBR-2593 - broker does not set message ID header on cross protocol consumption**

A Qpid JMS client successfully retrieves a message ID only if the message was produced by another Qpid JMS client. If the message was produced by a Core JMS or OpenWire client, the Qpid JMS client cannot read the message ID.
- **ENTMQBR-2678 - After isolated master is live again it is unable to connect to the cluster**

In a cluster of three or more live-backup groups that is using the replication high availability (HA) policy, the live broker shuts down when its replication connection fails. However, when the replication connection is restored and the original live broker is restarted, the broker is



sometimes unable to rejoin the broker cluster. To enable the original live broker to rejoin the cluster, first stop the new live (original backup) broker, restart the original live broker, and then restart the original backup broker.

- **ENTMQBR-2711 - ServerSessionImpl cache does not clear names of deleted temporary destinations**

When you create and delete temporary queues, the broker does not remove these queue names from the internal **ServerSessionImpl** cache. Over time, this data accumulates, causing a memory leak.

- **ENTMQBR-2767 - [AMQ7, purge message, OutOfMemoryException] with a large queue size, removeAllMessages() takes a long time and can eventually result in an OOM exception**

For a queue with a large number (that is, in the order of millions) of messages, the JMX operation **removeAllMessages** can take up to several minutes to complete. In addition, the operation sometimes fails to complete, returning an out-of-memory (OOM) exception instead.

- **ENTMQBR-2890 - Infrequently, creating a CR instance with size  $n > 1$  causes the the  $n$ th broker pod to start and immediately restart once**

If you use a Custom Resource (CR) instance to deploy a broker cluster via the AMQ Broker Operator, the final broker Pod in the deployment (as determined by the **size** attribute of your CR) starts and then immediately restarts one time before it is usable. To work around this issue, incrementally increase the **size** attribute of your CR to reach the desired cluster size. First, set **size** to **1** and wait for the broker Pod to start. Then, set **size** to **2**, wait for the Pod to start, and so on. In this case, none of the broker Pods undergo a restart before they are usable.

- **ENTMQBR-2942 - Pod #0 tries to contact non-existent Pods**

If you change the **size** attribute of your Custom Resource (CR) instance to scale down a broker deployment, the first broker Pod in the cluster can make repeated attempts to connect to the drainer Pods that started up to migrate messages from the brokers that shut down, before they shut down themselves. To work around this issue, follow these steps:

- 1) Scale your deployment to a single broker Pod.

- 2) Wait for all drainer Pods to start, complete message migration, and then shut down.

- 3) If the single remaining broker Pod has log entries for an "unknown host exception", scale the deployment down to zero broker Pods, and then back to one.

- 4) When you have verified that the single remaining broker Pod is not recording exception-based log entries, scale your deployment back to its original size.

## CHAPTER 7. IMPORTANT LINKS

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

*Revised on 2020-06-18 16:02:31 UTC*