Red Hat AMQ 7.6

Release Notes for Red Hat AMQ Broker 7.6

Release Notes for AMQ Broker

Last Updated: 2020-03-19

Red Hat AMQ 7.6 Release Notes for Red Hat AMQ Broker 7.6

Release Notes for AMQ Broker

Legal Notice

Abstract

These release notes contain the latest information about new features, enhancements, fixes, and issues contained in the AMQ Broker 7.6 release.

Table of Contents

CHAPTER 1. ENHANCEMENTS

- **ENTMQBR-875 - Add support for retroactive addresses**
  AMQ Broker 7.6 adds support for retroactive addresses. Configuring an address as *retroactive* enables you to preserve messages sent to that address, including when there are no queues yet bound to the address. For more information, see Configuring Retroactive Addresses.

- **ENTMQBR-2188 - Support wildcards in role-based access control configuration**
  In AMQ Broker 7.6, when configuring the **role-access** element in your **broker.xml** configuration file, you can now use a wildcard expression in the **key** attribute of the **match** element. For more information, see Configuring Authentication Based on Roles .

- **ENTMQBR-2519 - Certification for Oracle 19c**
  AMQ Broker 7.6 adds support for Oracle Database 19c. Oracle Database 19c enables you to configure Oracle Real Application Clusters (RAC). RAC is an option for the Oracle Database that provides high availability (HA) and scalability to the Oracle Database without requiring any application changes.

  For more information about the databases supported by different versions of AMQ Broker, see Red Hat AMQ 7 Supported Configurations . To learn more about Oracle RAC, consult the Oracle documentation.

- **ENTMQBR-2544 - Set LDAP connection timeout**
  In AMQ Broker 7.6, you can set timeouts for connection and read requests from the broker to an LDAP directory server. For more information, see Integrating with LDAP.

- **ENTMQBR-2755 - Support custom validation of TLS client certificates**
  In AMQ Broker 7.6, an administrator can specify an SSL **TrustManagerFactory** plugin in the broker configuration to associate with the SSL context. Specifically, a new configuration parameter, **trustManagerFactoryPlugin** can now be specified on an acceptor or connector. This option enables custom validation of TLS client certificates.

  For more information, see **Table A.3. Netty TLS/SSL Parameters** in Appendix A. Acceptor and Connector Configuration Parameters.

- **ENTMQBR-2852 - Configurable page sync tiemout for AIO**
  If you are using AIO for journal-based persistence (that is, **journal-type** is set to **ASYNCIO** in your **broker.xml** configuration file), you can now use the **page-sync-timeout** configuration parameter to specify the page synchronization timeout duration.

CHAPTER 2. TECHNOLOGY PREVIEW

This section details Technology Preview features in AMQ Broker 7.6.

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.

- **Federated queues and addresses**
  Federated queues and addresses enable 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 3. FIXED ISSUES

This section details the issues that have been fixed in AMQ Broker 7.6.

- ENTMQBR-1444 - **[amqp] when receiver client connects without source being set, broker prints NPE**
  In previous releases, if an AMQP receiver client connected to the broker without the receiver's source attribute (**:source**) being set, the broker displayed a "NullPointerException" warning. This issue is resolved. The broker no longer shows an exception if an AMQP receiver client does not have the **:source** attribute set when it connects to the broker.

- ENTMQBR-2707 - **[amqp] when receiver client connects without source being set, broker prints NPE**
  Previously, if AMQP messages in a pre-fetch buffer were returned to the queue, unacknowledged messages were returned in the wrong order. As a result of this issue, clients might later consume these messages in the wrong order, negatively affecting applications for which message ordering was important. This issue is resolved. Pre-fetched AMQP messages are always returned to the queue in the original order.

- ENTMQBR-2767 - **[AMQ7, purge message, OutOfMemoryException] with a large queue size, removeAllMessages() takes a long time and can eventually result in an OOM exception**
  Previously, for a queue with a large number (that is, in the order of millions) of messages, the JMX operation **removeAllMessages** could take up to several minutes to complete. In addition, the operation sometimes failed to complete, returning an out-of-memory (OOM) exception instead. This issue is now resolved.

- ENTMQBR-2878 - **AMQ broker does not clean the connection (MQTT) when the connection is broken**
  Previously, when an MQTT client dropped its connection to the broker without properly closing it, the broker did not close the underlying physical connection. This issue is now resolved.

- ENTMQBR-2889 - **Don't delete auto-created queues when FORCE is used for configuration changes**
  Previously, if you specified **config-delete-addresses=FORCE** in your **broker.xml** configuration file, auto-created addresses and their associated queues were also deleted when the configuration was reloaded. This issue is now resolved.

- ENTMQBR-2912 - **Messages greater than 50kb do not appear in the AMQ Broker management console**
  Previously, if a large message was compressed and the compressed size fell below the minimum large message size, the message was sent as a normal message and did not appear in the AMQ Broker management console. This issue is now resolved. All large messages can be browsed in the management console, regardless of whether they have been compressed or not.

- ENTMQBR-2932 - **Remote JMX server on slave shuts down during failback**
  Previously, when a slave broker in a master-slave pair failed back to the original master broker, the JMX sever on the slave broker did not restart and could no longer be used. As a result of this issue, you could not manage or control the slave broker via JMX. However, other management methods (for example, via Jolokia or HTTP) were unaffected. This issue is now resolved. When a slave broker fails back to the original master broker, the JMX server is restarted.

- ENTMQBR-3012 - **[AMQ7, large messages] LargeMessage doesn't make a full copy of its props**

Previously, a copy of a large message still referenced some properties of the original message. This caused unexpected behaviour when a thread modified the properties. This issue is resolved. When the broker copies a large message, all properties are included in the copy.

- ENTMQBR-3069 - **AMQ JMS client doesn't recover after a complete outage**
  Previously, if a master-slave broker pair failed back to the original master broker following an outage, an AMQ JMS client could not reconnect to the master. This failed connection was caused by failure of the broker to remove its existing client IDs following failback. This issue is now resolved.

- ENTMQBR-3128 - **Failback does not work for master/slave group using NFS shared store**
  In AMQ Broker 7.5, if you configured a live-backup broker group to use the shared store high availability (HA) policy with a Network File System (NFS), the backup broker did not fail back to the original live broker when the live broker became available again after a previous failure. This issue occurred even if you set **allow-failback** to **true** in the configuration of your backup broker. This issue is now resolved.

- ENTMQBR-3136 - **Improper Quoting in Generated artemis.profile File - Causing Start Failures in Some Environments**
  In previous releases, the **JAVA_ARGS** line of the generated **artemis.profile** file incorrectly included quotation marks around the default value of the **hawtio.offline** parameter (that is, **Dhawtio.offline="true"**). This formatting error could cause the broker to fail to start in some environments. This issue is now resolved.

- ENTMQBR-3163 - **Memory Leak when Opening and Closing AMQP Consumers in the Same Session / Context**
  Previously, if you consecutively opened and then closed a series of AMQP clients within the same **JMSContext** instance, this caused a memory leak on the broker. This issue is now resolved.

- ENTMQBR-3215 - **Broker rejects client re-connection after broker stop/start**
  Previously, if a broker was stopped and restarted, an AMQP client (for example, an AMQ JMS client) with reconnect logic could not reconnect to the broker. Specifically, the broker rejected the connection, indicating that the client ID already existed. This issue is now resolved.

CHAPTER 4. KNOWN ISSUES

This section details known issues in AMQ Broker 7.6.

- **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, causingJavaNullPointerException, 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 knownamq-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.JMSException: 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] NullPointer 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-2890 - **Infrequently, creating a CR instance with size n > 1 causes the the nth 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-2928 - **Broker Operator unable to recover from CR changes causing erroneous state**
  If the AMQ Broker Operator encounters an error when applying a Custom Resource (CR) update, the Operator does not recover. Specifically, the Operator stops responding as expected to further updates to your CRs.

  For example, say that a misspelling in the value of the **image** attribute in your main broker CR causes broker Pods to fail to deploy, with an associated error message of **ImagePullBackOff**. If you then fix the misspelling and apply the CR changes, the Operator does not deploy the specified number of broker Pods. In addition, the Operator does not respond to any further CR changes.

  To work around this issue, you must delete the CRs that you originally deployed, before redeploying them. To delete an existing CR, use a command such as **oc delete -f <CR name>**.

- 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.

- ENTMQBR-3103 - **[AMQ7, AMQP, Openwire] Issue consuming AMQP message using OpenWire consumer**

When a message is converted from OpenWire to AMQP and then back to OpenWire, the message can no longer be consumed by an OpenWire client. For example, this situation might occur when:

1) An OpenWire client sends a message to the broker

2) An AMQP client consumes the message

3) The AMQP client returns the message to the broker

4) An OpenWire consumer tries to consume the message

- ENTMQBR-3131 - **Topology Fails to Update correctly for Backup Brokers when Master is Killed**
  When a live broker fails in a cluster with more than four live-backup pairs, the live brokers, including the newly-elected live broker, all correctly report the updated topology. However, the remaining backup brokers might show the wrong topology in the following ways:

  - If a backup broker has failed over in place of the failed live broker, the remaining backup brokers show this backup broker twice in the topology.

  - If a backup broker has not yet failed over in place of the failed live broker, the remaining backup brokers still show the failed live broker in the topology.

  To work around this issue, ensure that the first **connector-ref** element in the **cluster-connection** > **static-connectors** configuration of each backup broker specifies the expected live broker.

- ENTMQBR-3228 - **Operator Pod not authenticating correctly when processing address CR**
  If you use the AMQ Broker Operator to create a broker Pod that requires login (that is, by setting **requireLogin** in the Custom Resource (CR) used for the deployment), any addresses you try to create on the broker via a CR are not created. To work around this issue, use the AMQ Broker management console to create addresses on your broker. For more information on accessing the console, see Connecting a broker to the AMQ Broker management console .

- ENTMQBR-3365 - **Paging is Broken for AMQP**
  By default, when you first get a message after the page cache has been evicted, the broker reads the entire page back into the cache. This behavior is specified by the **read-whole-page** configuration parameter, which has a default value of **true**. If you set the value of **read-whole-page** to **false**, the broker reads messages back into the page cache on a message-by-message basis instead.

  Reading messages back into the cache on a message-by-message basis **does not work** if you are using the AMQP protocol. If you are using AMQP, ensure that the value of **read-whole-page** is set to **true**.

CHAPTER 5. IMPORTANT LINKS

- Red Hat AMQ Broker 7.1 to 7.5 Release Notes (aggregated)
- Red Hat AMQ 7 Supported Configurations
- Red Hat AMQ 7 Component Details

Revised on 2020-03-19 13:56:56 UTC