



Red Hat JBoss AMQ 7.0

AMQ Clients 1.1 Release Notes

Release Notes for Red Hat JBoss AMQ Clients

Red Hat JBoss AMQ 7.0 AMQ Clients 1.1 Release Notes

Release Notes for Red Hat JBoss AMQ Clients

Legal Notice

Copyright © 2017 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

These release notes contain the latest information about new features, enhancements, fixes, and issues contained in the AMQ Clients 1.1 release.

Table of Contents

CHAPTER 1. FEATURES	3
CHAPTER 2. ENHANCEMENTS	4
2.1. AMQ C++, AMQ JAVASCRIPT, AND AMQ PYTHON	4
2.2. AMQ JMS	4
CHAPTER 3. RESOLVED ISSUES	5
3.1. AMQ C++ AND AMQ PYTHON	5
3.2. AMQ C++	5
3.3. AMQ PYTHON	5
3.4. AMQ .NET	5
CHAPTER 4. IMPORTANT NOTES	6
4.1. NMS AND CMS	6
4.2. AMQ JMS	6
4.3. AMQ C++	6

CHAPTER 1. FEATURES

- AMQ JMS now supports the JMS 2.0 API over AMQP.
- AMQ C++ is a new event-driven C++ messaging API.
- AMQ JavaScript is a new event-driven Node.js messaging API.

CHAPTER 2. ENHANCEMENTS

2.1. AMQ C++, AMQ JAVASCRIPT, AND AMQ PYTHON

- **Reconnect after `amqp:connection:forced` error**

In earlier releases of the product, the reconnect mechanism of these clients did not treat the **`amqp:connection:forced`** close condition in the same way as an unexpected network disconnect. Servers may use this AMQP condition to indicate temporary disruption of service.

In this release, the client treats an **`amqp:connection:forced`** close condition the same as an unexpected network disconnect, allowing the client to reconnect as configured by the user.

2.2. AMQ JMS

- **Implement the JMS 2.0 API**

AMQ JMS now implements the JMS 2.0 API. This introduces support for new functionality such as shared topic subscriptions, an asynchronous send API, and delayed message delivery, along with various API improvements.

- **[ENTMQCL-283](#) - OSGi enable the AMQ JMS client jar**

The **`qpidd-jms-client`** jar now includes OSGi metadata in its manifest to enable use as an OSGi bundle. This allows using the client in an OSGi container without requiring it be wrapped to add such metadata.

- **[ENTMQCL-426](#) - AMQP-based server-provided failover details**

AMQ JMS can now use its initial connection to discover alternate connection endpoints for use if the current connection fails. With AMQ Broker, clients can use this to discover the network address of the backup server in an HA pair. Use the **`failover.amqpOpenServerListAction`** connection URI option to control its behavior in detail.

CHAPTER 3. RESOLVED ISSUES

3.1. AMQ C++ AND AMQ PYTHON

- **ENTMQCL-459 - Cyrus SASL accesses strings that have been freed**

In earlier releases of the product, the Proton C library could cause the Cyrus SASL library to access freed memory.

In this release, the issue has been resolved.

3.2. AMQ C++

- **ENTMQCL-430 - C++03 version of the schedule API crashes on RHEL 7**

In earlier releases of the product, if you compiled a C program with C03 and linked it to **libqpidd-proton-cpp** (which is compiled with C++11) and use the **proton::container::schedule** API, then the application would crash.

In this release, the issue has been resolved.

3.3. AMQ PYTHON

- **ENTMQCL-428 - "Buffer" type in message body should map to AMQP binary encoding**

In earlier releases of the product, the Python **buffer** type was not allowed as message data.

In this release, the **buffer** type is automatically converted to the AMQP binary type.

- **ENTMQCL-457 - BlockingConnection leaks Proton C memory**

In earlier releases of the product, the AMQ Python **BlockingConnection** did not release all underlying Proton C data structures on close, causing a memory leak and sometimes a leak of several file descriptors.

In this release, the issue has been resolved.

3.4. AMQ .NET

- **ENTMQCL-418 - Support intercepting rejection of sent messages**

In earlier releases of the product, the error indication status of rejected messages was not being passed between the broker and the client program. This could create a situation wherein a client was not aware that the broker had rejected a message and would subsequently take no action to recover from the error.

In this release, the broker signals the error through AMQP mechanisms as expected, and clients can take appropriate action to recover.

CHAPTER 4. IMPORTANT NOTES

4.1. NMS AND CMS

- **Deprecation of the CMS and NMS APIs**

The ActiveMQ CMS and NMS messaging APIs are deprecated in AMQ 7. It is recommended that users of the CMS API migrate to AMQ C++, and users of the NMS API migrate to AMQ .NET. The CMS and NMS APIs may have reduced functionality in AMQ 7.

4.2. AMQ JMS

- **RPMs for AMQ JMS are no longer distributed**

Red Hat is no longer distributing AMQ JMS as an RPM. AMQ JMS will continue to be available as a zip file on the Customer Portal and as a Maven artifact in the JBoss Maven repository.

4.3. AMQ C++

- **Experimental interfaces in the AMQ C++ API**

The AMQ C++ messaging API includes APIs that are not yet proven and may change in future releases. Be aware that use of these APIs may require changes to your application code in the future.

These APIs are marked **Experimental** in the API reference. They include the interfaces in the `proton::codec` and `proton::io` namespaces and the following interfaces in the `proton` namespace.

- `annotation_map`
- `event_loop`
- `filter_map`
- `listen_handler`
- `property_map`
- `ssl_certificate`
- `ssl_client_options`
- `ssl_server_options`
- `thread_safe`
- The `stop` methods on `container`
- The `on_sender_drain_start` and `on_sender_drain_finish` methods on `messaging_handler`
- The `draining` and `return_credit` methods on `sender`
- The `draining` and `drain` methods on `receiver`

Any APIs present in header files but not yet documented are also considered experimental and subject to change.

Revised on 2017-09-01 11:05:15 EDT