



Red Hat AMQ 2021.Q1

AMQ Clients 2.9 Release Notes

Release Notes for Red Hat AMQ Clients

Red Hat AMQ 2021.Q1 AMQ Clients 2.9 Release Notes

Release Notes for Red Hat AMQ Clients

Legal Notice

Copyright © 2021 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[®] 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 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 2.9 release.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE	3
CHAPTER 1. FIXED ISSUES	4
1.1. AMQ JMS	4
1.2. AMQ .NET	4
1.3. AMQ C++	4
CHAPTER 2. KNOWN ISSUES	5
2.1. AMQ C++	5
CHAPTER 3. IMPORTANT NOTES	6
3.1. END OF RED HAT ENTERPRISE LINUX 6 SUPPORT	6
3.2. LONG TERM SUPPORT	6
3.3. AMQ C++	6
3.4. PREFERRED CLIENTS	7
3.5. LEGACY CLIENTS	7
3.6. UPSTREAM VERSIONS	7
CHAPTER 4. IMPORTANT LINKS	9

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).

CHAPTER 1. FIXED ISSUES

1.1. AMQ JMS

- [ENTMQCL-2337](#) - Failure decoding large messages with special characters

In earlier releases of the product, the client failed to decode messages containing multi-byte characters if the characters spanned transfer frames.

In this release, large messages with multi-byte characters are correctly decoded.

- [ENTMQCL-2339](#) - The client can ignore transactions right after failover

In earlier releases of the product, the client sometimes failed to mark a message transfer as belonging to a transaction directly after connection failover.

In this release, transactions are correctly recovered after failover.

1.2. AMQ .NET

- [ENTMQCL-2241](#) - Client example sends empty begin frames

In earlier releases of the product, the **ReconnectSender** example program sent an AMQP **begin** frame lacking required fields.

In this release, the example correctly provides the mandatory fields.

1.3. AMQ C++

- [ENTMQCL-1863](#) - On receiving a forced close, send a corresponding protocol close and close the socket

In earlier releases of the product, when the client received a forced AMQP connection **close** from the remote peer, it then closed its local TCP socket without sending a matching AMQP **close** in return.

In this release, the client sends an AMQP **close** before closing the socket.

CHAPTER 2. KNOWN ISSUES

2.1. AMQ C++

- [ENTMQCL-2583](#) - Example build fails using CMake 2.8

The build metadata for the examples uses a CMake feature that is not available in CMake 2.8, causing the build to fail.

Workaround: Use a developer environment where CMake 3 is available.

CHAPTER 3. IMPORTANT NOTES

3.1. END OF RED HAT ENTERPRISE LINUX 6 SUPPORT

As of version 2.9, AMQ Clients no longer supports Red Hat Enterprise Linux 6. Red Hat Enterprise Linux 6 ended maintenance support on November 30th, 2020. For more information, see [Red Hat Enterprise Linux Life Cycle](#).

3.2. LONG TERM SUPPORT

AMQ Clients 2.9 has been designated as a long term support (LTS) release version. Bug fixes and security advisories will be made available for AMQ Clients 2.9 in a series of micro releases (2.9.1, 2.9.2, 2.9.3, and so on) for a period of at least 12 months.

This means that you will be able to get recent bug fixes and security advisories for AMQ Clients *without* having to upgrade to a new minor release.

Note the following important points about the LTS release stream:

- The LTS release stream provides only bug fixes. No new enhancements will be added to this stream.
- To remain in a supported configuration, you must upgrade to the latest micro release in the LTS release stream.
- The LTS version will be supported for at least 12 months from the time of the AMQ Clients 2.9.0 GA.

3.3. AMQ C++

- **Unsettled interfaces**

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

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

- **listen_handler**
- 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**

API elements present in header files but not yet documented are considered unsettled and are subject to change.

- **Deprecated interfaces**

Interfaces marked **Deprecated** in the API reference are scheduled for removal in a future release.

This release deprecates the following interfaces in the **proton** namespace.

- **void_function0** - Use the **work** class or C++11 lambdas instead.
- **default_container** - Use the **container** class instead.
- **url** and **url_error** - Use a third-party URL library instead.

3.4. PREFERRED CLIENTS

In general, AMQ clients that support the AMQP 1.0 standard are preferred for new application development. However, the following exceptions apply:

- If your implementation requires distributed transactions, use the AMQ Core Protocol JMS client.
- If you require MQTT or STOMP in your domain (for IoT applications, for instance), use community-supported MQTT or STOMP clients.

The considerations above do not necessarily apply if you are already using:

- The AMQ OpenWire JMS client (the JMS implementation previously provided in A-MQ 6)
- The AMQ Core Protocol JMS client (the JMS implementation previously provided with HornetQ)

3.5. LEGACY CLIENTS

- **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 might have reduced functionality in AMQ 7.
- **Deprecation of the legacy AMQ C++ client**
The legacy AMQ C++ client (the C++ client previously provided in MRG Messaging) is deprecated in AMQ 7. It is recommended that users of this API migrate to AMQ C++.
- **The Core API is unsupported**
The Artemis Core API client is not supported. This client is distinct from the AMQ Core Protocol JMS client, which is supported.

3.6. UPSTREAM VERSIONS

- AMQ C++, AMQ Python, and AMQ Ruby are now based on [Qpid Proton 0.33.0](#).
- AMQ JavaScript is now based on [Rhea 1.0.24](#).
- AMQ .NET is now based on [AMQP.Net Lite 2.4.0](#).
- AMQ JMS is now based on [Qpid JMS 0.55.0](#).
- AMQ Core Protocol JMS is now based on [ActiveMQ Artemis 2.16.0](#).
- AMQ OpenWire JMS is now based on [ActiveMQ 5.11.0](#).

- AMQ JMS Pool is now based on [Pooled JMS 1.2.1](#).
- AMQ Resource Adapter is now based on [AMQP 1.0 Resource Adapter 1.0.2](#).
- AMQ Spring Boot Starter is now based on [AMQP 1.0 JMS Spring Boot 2.3.6](#).
- AMQ Netty OpenSSL is now based on [netty-tcnative 2.0.34.Final](#).

CHAPTER 4. IMPORTANT LINKS

- [AMQ Clients 2.9.x Resolved Issues](#)
- [Red Hat AMQ 7 Supported Configurations](#)
- [Red Hat AMQ 7 Component Details](#)
- [AMQ Clients 2.8 Release Notes](#)
- [AMQ Clients 2.7 Release Notes](#)
- [AMQ Clients 2.6 Release Notes](#)
- [AMQ Clients 2.5 Release Notes](#)
- [AMQ Clients 2.4 Release Notes](#)
- [AMQ Clients 2.3 Release Notes](#)
- [AMQ Clients 2.2 Release Notes](#)
- [AMQ Clients 2.1 Release Notes](#)
- [AMQ Clients 2.0 Release Notes](#)
- [AMQ Clients 1.2 Release Notes](#)
- [AMQ Clients 1.1 Release Notes](#)

Revised on 2021-05-19 08:48:21 UTC