



Red Hat Enterprise MRG 3

3.2 MRG Messaging Release Notes

Release Notes for Red Hat Enterprise MRG Messaging 3.2, and its patch releases

Red Hat Enterprise MRG 3 3.2 MRG Messaging Release Notes

Release Notes for Red Hat Enterprise MRG Messaging 3.2, and its patch releases

Red Hat Customer Content Services

Legal Notice

Copyright © 2018 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

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 important information related to Red Hat Enterprise MRG Messaging v3.2.0. Read these Release Notes in their entirety before installing the product.

Table of Contents

CHAPTER 1. INTRODUCING MESSAGING 3.2	3
1.1. RED HAT ENTERPRISE MESSAGING 3.2	3
1.2. DOCUMENTATION	3
CHAPTER 2. CHANGES IN 3.2.0	4
2.1. ENHANCEMENTS	4
2.2. BUG FIXES	6
2.3. KNOWN ISSUES	9
2.4. DEPRECATED FUNCTIONALITIES	10
CHAPTER 3. CHANGES IN 3.2.1	11
3.1. ERRATA	11
CHAPTER 4. CHANGES IN 3.2.2	12
4.1. ERRATA	12
CHAPTER 5. CHANGES IN 3.2.3	14
5.1. ERRATA	14
CHAPTER 6. CHANGES IN 3.2.4	15
6.1. ERRATA	15
CHAPTER 7. CHANGES IN 3.2.5	16
7.1. ERRATA	16
CHAPTER 8. CHANGES IN 3.2.6	17
8.1. ERRATA	17
CHAPTER 9. CHANGES IN 3.2.7	18
9.1. ERRATA	18
CHAPTER 10. CHANGES IN 3.2.8	19
10.1. ERRATA	19
CHAPTER 11. CHANGES IN 3.2.9	20
11.1. ERRATA	20
CHAPTER 12. CHANGES IN 3.2.10	21
12.1. ERRATA	21
CHAPTER 13. CHANGES IN 3.2.11	22
13.1. ERRATA	22
APPENDIX A. REVISION HISTORY	23

CHAPTER 1. INTRODUCING MESSAGING 3.2

1.1. RED HAT ENTERPRISE MESSAGING 3.2

Red Hat Enterprise Messaging (MRG-M) version 3.2 is the next generation release of the MRG-M product. Based on the Apache Qpid project, this release builds upon the foundation laid in the MRG-M 3.0 releases.

Version 3.2 introduces the enhancements detailed below, and numerous customer facing bug fixes that comprise the errata prepared for this release.

[Report a bug](#)

1.2. DOCUMENTATION

See the product documentation for further details on installing, configuring, and migrating existing installations and applications. All product documentation is available from the [Red Hat Enterprise MRG Documentation Home Page](#)

The [Messaging Installation and Configuration Guide](#) contains information on server installation, configuration, and migration.

The [Messaging Programming Reference](#) contains information on developing applications and migrating existing applications to run against the latest release of the server.

[Report a bug](#)

CHAPTER 2. CHANGES IN 3.2.0

2.1. ENHANCEMENTS

distribution

BZ#1215806 - Provide qpid JMS client that supports AMQP 1.0

MRG-M now offers a qpid JMS client at version 6.2, which supports AMQP 1.0

This support has been added as a result of requests from customers

Customers can now use AMQP 1.0 through the JMS client. Documentation for this feature is available from <http://qpid.apache.org/components/jms/index.html>.



NOTE

The new AMQP 1.0 qpid-jms client does not support the MRG-specific address strings used by the older AMQP 0.10 java client.

In the new client, the *queue name* and *topic name* strings given to the new client represent only the name of the Queue or Topic to consume from or publish to.

Other concepts from the old client syntax such as subjects or operations to create and delete entities are not supported.

See http://bugzilla.redhat.com/show_bug.cgi?id=1215806

qpid-cpp

BZ#1261180 - [Enhancement] Make the journal flush timeout configurable in the C++ broker

The qpidd persistent message store flush timeout can be configured using a command line option or using the qpidd configuration file. In previous versions of the product, this time was set at 500ms.

The appropriate command line option is:

```
--journal-flush-timeout SECONDS
```

The parameter is in seconds but can be a decimal, and will also accept suffixes to allow specification in seconds (**s**), milliseconds (**ms**), microseconds (**us**) and nanoseconds (**ns**). (this is consistent with the other parameters that can also be times).

Setting this option in the qpidd configuration file follows the syntax shown below:

```
journal-flush-timeout=1ms
```

See http://bugzilla.redhat.com/show_bug.cgi?id=1261180

BZ#1151446 - Heartbeats for AMQP 1.0 Qpid Messaging API

When an AMQP 1.0 client advertises a connection idle timeout value, twice that value is used by the broker as its idle timeout threshold for that connection. This is consistent with the existing 0-10 behavior.

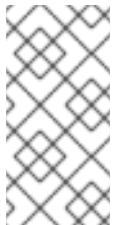
This causes the broker to terminate any client that has been idle for twice the value of the client's idle timeout.

It is important to note that in this context, *advertized* means the value sent as the idle timeout in the open frame.

Most clients advertize half their configured timeout, but a client may advertise a full configured timeout.

This interval is set by passing the 'heartbeat' option to the `qpidd::messaging::Connection` constructor. The value is expressed in seconds. You can pass these via the `--connection-options` command line argument to `qpidd-receive`:

```
qpidd-receive --connection-options "{protocol: amqp1.0, heartbeat: 10}"
....
```



NOTE

The broker does not support the configuration of a default idle-timeout on the broker side. The idle timeout value for the connection is entirely determined by the configuration of the connection made by the client. There is no way to override this value on the broker side via management.

See http://bugzilla.redhat.com/show_bug.cgi?id=1151446

BZ#1015448 - AMQP 1.0 qpidd::messaging c++ local transactional support to be enabled (Transactions not yet supported)

Local transactional support is now available for the AMQP 1.0 `qpidd::messaging` c++ client. This feature was available with the AMQP 0-10 protocol, but was missing as a feature of the AMQP 1.0 protocol. Any customer who attempted to use the feature with the old protocol, and tried to upgrade, would have encountered a **Transactions not yet supported** error.

Transactional support in AMQP 1.0 works as it did in AMQP 0-10, and customers can now continue to use the same approach in the latest version of AMQP.

See http://bugzilla.redhat.com/show_bug.cgi?id=1015448

BZ#1160367 - [linearstore] Add ability to use disk partitions and select per-queue EFPs

The store now supports partitions for the empty file pools and store journal files.

To select a default broker partition use the `--efp-partition` option which can be used together with the `--efp-file-size` option if needed. If these options are not used, the default partition is 1 (`p001` in the store directory) and the size is 2048k.

See the **Messaging Installation and Configuration** Guide's **Linearstore Custom Broker EFP Partitions** section for more information, and links to related section describing the new parameters in more detail.

See http://bugzilla.redhat.com/show_bug.cgi?id=1160367

qpidd-sdk

BZ#1152686 - qpidd-winsdk should support Microsoft Visual Studio 2012

Feature: qpidd-winsdk now supports Microsoft Visual Studio (VS) 2012, which now means Visual Studio 2008, 2010, 2012 are supported for qpidd-winsdk. Information about the correct files to consume from qpidd-winsdk can be found in the *Messaging Programming Reference Guide's C++ on Windows* section. This section has been improved to describe the correct directory structure of the download binaries, and provides steps to correct consume the contents of the binaries for environments with an established (VS) environment.

See http://bugzilla.redhat.com/show_bug.cgi?id=1152686

[Report a bug](#)

2.2. BUG FIXES

qpidd-cpp

BZ#1002605 - Trace queue deletion statistics show msgDepth:0 every time

It was discovered that the event and log about deleting some provisionable objects (for example, the queue) have zero counters in many statistics, regardless of the truth values. This caused the information in the event and the log to be incorrect. An update in deletion procedures of the objects now ensures the statistic counters provide proper values both in the event and the log.

See http://bugzilla.redhat.com/show_bug.cgi?id=1002605

BZ#1224300 - [MRG 3.2] linearstore raising JERR_LFCR_SEQNUMNOTFOUND after sending many DTX transactions

In previous releases of the product, customers could encounter a **JERR_LFCR_SEQNUMNOTFOUND** error. This is because after sufficient journal files have been used, the journal sequence number overflows to 0 which is a reserved and non-existent file number. This is an artifact of the older legacystore, where file numbers were limited in a circular buffer to 64 files.

In this release, all use of the journal file sequence number has been updated to use 64-bit unsigned integers, preventing the error from occurring in any practical usage situation.

See http://bugzilla.redhat.com/show_bug.cgi?id=1224300

BZ#1192924 - A broker configured with disabled authentication requires authentication from a client connected from Windows

In previous versions of the **qpidd::messaging** client on **Windows** (*c#* or *c++*), if a username was not specified when using the default (+PLAIN) authentication mechanism, the empty string was included as the identity causing authentication to fail. This would happen even if authentication was disabled on the broker.

The broker has been modified to accept an empty string as the identity, but only when authentication is disabled. If authentication is enabled the empty string will still fail.

As a result, a `qpidd:messaging` client in **Windows** can connect to a broker without needing to specify any extra connection options with authentication disabled.

See http://bugzilla.redhat.com/show_bug.cgi?id=1192924

BZ#1158903 - [linearstore] segfault when 2 journals request new journal file from empty EFP

It was discovered that when two journals concurrently requested a new journal file from an empty EFP, the Broker could segfault. A fix to `popEmptyFile` now performs an *atomic* test-and-create-and-grab for the EFP file, which prevents the segfault from occurring.

See http://bugzilla.redhat.com/show_bug.cgi?id=1158903

BZ#1240552 - No error info logged for io errors with ssl

There was a regression in the error messages of the MRG-3.0 release as compared to previous releases. This relates to errors occurring with SSL/TLS connections.

In the 3.0 release, errors related to SSL/TLS were not informative and even appeared not to be errors:

```
Error reading socket: Success(0) `
```

The regression has been fixed and error messages relating to SSL/TLS now have the same information as previous releases.

See http://bugzilla.redhat.com/show_bug.cgi?id=1240552

BZ#1234423 - Outdated "README-winsdk.txt" file

With the addition of Microsoft Visual Studio 2012 to the list of supported versions, it was discovered that information about how to build the C examples in Visual Studio 2012, along with the required CMake version required to build the C projects were missing from the README. The README has been updated to add the required steps, which will help customers build the examples correctly.

See http://bugzilla.redhat.com/show_bug.cgi?id=1234423

BZ#1167911 - [linearstore] Symlink creation fails if store dir path is not absolute

It was discovered that symlink creation failed if the specified linearstore `store-dir` path was not absolute. The `store-dir` parameter should never have been specified as a relative path, however the issue was benign for some time in both linearstore and legacystore due to the lack of symlink support in journal directories (introduced as part of [BZ#1160367](#)). The fix for this bug now adds a prefixed `/` to the `store-dir` parameter if the path is not specified as absolute.

See http://bugzilla.redhat.com/show_bug.cgi?id=1167911

BZ#1152032 - purging TTL expired messages via purge task should not increase acquires counters

It was discovered that when time-to-live (TTL) messages were removed using a purge task, the `acquires` counter was increased for each broker and queue. Purging the same message by removing it when finding what message to send/acquire to a consumer did not increase the counter. Because TTL purged messages are not technically acquired, this behavior was misleading. The fix removes code from `Queue.cpp` that counts `acquires` on `mgmtObject` and `brokerMgmtObject`.

See http://bugzilla.redhat.com/show_bug.cgi?id=1152032

BZ#1150127 - AMQP 1.0 consumer should be able to get messages from browse-only queue

Qpid broker allowed 0-10 consumers on browse-only queues, but did not allow 1.0 consumers to attach to these queues. This caused read-only queues not to work as expected for customers. Improvements to `Queue.h` and `amqp/Session.cpp` now allow 1.0 consumers to attach to browse-only queues. The consumers are marked internally as browsers, which is consistent in behavior to 0-10 consumers.

See http://bugzilla.redhat.com/show_bug.cgi?id=1150127

BZ#1148807 - [linearstore] Restarting broker with empty journal raises confusing warning

Provisioning a durable message (backed up by linearstore journal) and restarting the broker without enqueueing a message into it raised a warning message **Journal .. is uninitialized or corrupted**.

This message (on this verbosity level) was confusing to users, as it incorrectly suggested a potential problem with the journal.

The fix to Linearstore now uses the uninitialized file as the first journal file in the queue.

See http://bugzilla.redhat.com/show_bug.cgi?id=1148807

BZ#1136418 - QMF Session name to contain user ID

It was discovered that QMF could not handle sessions of the same name attached in parallel by different users because it stored session objects by their name: unique per user only.

The fix implements a session ID for QMF in `SessionState.cpp`, by requesting `std::string name(getId().str());` instead of `std::string name(getId().getName());`.

See http://bugzilla.redhat.com/show_bug.cgi?id=1136418

BZ#1152012 - [linearstore] missing journal id in "trace Mgmt create journal." log

It was discovered that the broker log did not initially log the journal ID in the log entry. While the journal ID in the relevant QMF object was confirmed to be set properly after the fact, the log was not indicating this was the case. A fix to how the log entry is created now correctly inserts the journal name.

See http://bugzilla.redhat.com/show_bug.cgi?id=1152012

qpid-java

BZ#1207368 - NullPointerException from Hello example (hello.properties file missing)

Previously the `hello.properties` file was located in `org/apache/qpid/example` (together with the example sources) and the binaries were compiled to the same location. Starting with MRG-3.1, the source and `hello.properties` location was changed to `src/main/java/org/apache/qpid/example/` while the compiled binaries were still saved to the old location. This caused the Hello example to not be able to read the `hello.properties` file and resulted in a `NullPointerException`. The bug fix

permanently moves the java source for examples into `/examples/org` instead of `examples/src/main/java/org` which puts the content back into the correct classpath context for the example start script. This corrects the workaround present in [BZ#1206503](#).

See http://bugzilla.redhat.com/show_bug.cgi?id=1207368

qpidd-tools

BZ#1155553 - qpidd-tool to halt when initial connection to broker failed

When the qpidd-tool was run with invalid credentials or an incorrect host name, the tool did not raise an exception or warning. This lack of context left the user with the impression that the tool connected successfully, when this was not the case. qpidd-tool now prints a warning to stdout: **qpidd: Failed to connect: Exception during connection setup: error - [Errno 111]** if incorrect parameters are passed while the broker is not running.

See http://bugzilla.redhat.com/show_bug.cgi?id=1155553

BZ#1136191 - qpidd-ha -h or --help should exit with returncode of 0

The `-h` and `--help` options were not recognized by qpidd-ha, and were treated the same way as running qpidd-ha (without any argument). All returned the exit code `1` instead of `0`, which displayed **Exception: No valid command**.

Fix: The `-h` and `--help` options are now implemented to return the result from `--help-all`, which displays command usage for qpidd-ha regardless of the commonly-passed method chosen by the user.

See http://bugzilla.redhat.com/show_bug.cgi?id=1136191

[Report a bug](#)

2.3. KNOWN ISSUES

qpidd-cpp

BZ#1242954 - SELinux prevents qpidd from starting after update MRGM 3.1 to MRGM 3.2



IMPORTANT

Customers should be aware that they need to upgrade to the latest version of Red Hat Enterprise Linux 6 (6.7.2) and the SELinux policy files shipped with it to get started with MRG-M 3.2. Any other version of RHEL/SELinux can produce **Permission denied** errors.

See http://bugzilla.redhat.com/show_bug.cgi?id=1242954

qpidd-java

BZ#1267276 - Inappropriate message logged by the SSL client

When the java client (qpid-java package) is verifying the hostname from a valid server certificate, an error message is logged even though the verification succeeds:

```
IoReceiver - <hostname>/<ip>:<port> 2015-09-21 14:52:46,392 WARN  
[network.security.ssl.SSLUtil] Exception received while trying to verify  
hostname
```

The message can be ignored.

See http://bugzilla.redhat.com/show_bug.cgi?id=1267276

BZ#1267275 - The hostname from the server certificate is parsed incorrectly

When the java client (qpid-java) parses the server certificate, it expects the 'subject' to begin with **CN=<hostname>**.

If it does not, the certificate fails to parse correctly and the SSL hostname verification will fail. This will fail even if the **CN** appears elsewhere in the subject.

The workaround is to make sure the subject starts with **CN**.

See http://bugzilla.redhat.com/show_bug.cgi?id=1267275

Release_Notes

BZ#1203298 - User options "qpid.file_size" and "qpid.file_count" are always blocked by ACL rules

ACL limit properties "filemaxsizelowerlimit" and "filemaxsizeupperlimit" were not honored for "qpid.file_size". Similarly, "filemaxcountlowerlimit" and "filemaxcountupperlimit" were not honored for the "qpid.file_count" option. Both options are no longer supported with LinearStore, and should not be used.

See http://bugzilla.redhat.com/show_bug.cgi?id=1203298

BZ#1171817 - Hello example incompatible with java-1.6

An issue with the Hello example not being compatible with java-1.6 because of "try-with-resources" (introduced in java-1.7) caused compilation errors when building with java-1.6. Using Java qpid client applications and examples require java-1.7 or higher, therefore it is mandatory to upgrade your java environment to prevent compilation and execution problems.

See http://bugzilla.redhat.com/show_bug.cgi?id=1171817

[Report a bug](#)

2.4. DEPRECATED FUNCTIONALITIES

There are no deprecated functionalities introduced for this release.

[Report a bug](#)

CHAPTER 3. CHANGES IN 3.2.1

3.1. ERRATA

[RHBA-2016:0443-1 – Red Hat Enterprise MRG Messaging 3.2.1 Release](#)

These updated packages for Red Hat Enterprise Linux 6 include a bug fix for the Messaging component of MRG.

It was found that any message property declared as an unsigned 16 bit integer would get converted to an unsigned 8 bit value in the broker. If the value was greater than 255, an error would occur in the conversion.

The code has been updated to ensure the conversion performs as expected and unsigned 16 bit integer values in the message header are now correctly maintained in the broker.(BZ#11301359)

[Report a bug](#)

CHAPTER 4. CHANGES IN 3.2.2

4.1. ERRATA

[RHBA-2016:2049 – Red Hat Enterprise MRG Messaging 3.2.2 Release](#)

BZ#1146849 - Re-subscribing to auto-delete queue fails when the queue is just being deleted

The creation of a receiver in the messaging APIs involves two main network interactions when using AMQP 0-10: declaring the queue, then subscribing to it. The declare will create the queue if necessary, however it did not reset any auto-deletion timer. Consequently it was possible for the queue to expire after the declare but before the subscribe, meaning the subscribe would fail. To resolve this issue, the declare semantics reset any auto-deletion timer associated with the queue. Now, the queue is unlikely to expire between the declare and the subscribe and the subscribe call will not fail.

BZ#1316507 - [C++ client] Message::setContent("") does not work

The message implementation can hold two views of message content: the raw bytes or a decode object form. These operate independently (changes to the raw bytes do not alter the object content). When a message was decoded into object form on receipt, then updated by setting the raw bytes in order to resend, the resent message used the original object content. Now, any previously held object content view is reset. Additionally, a setting was added to allow the automatic decoding of messages into object form to be disabled by environment variable or configuration file. The message content now reflects the application's actions.

BZ#1318180 - [MRG2] memory leak in primary broker when overwriting messages in a ring queue

Previously, the primary HA broker was accumulating message IDs and Poller handles, leading to a memory leak. The HA broker now frees message IDs and Poller handles when no longer needed, removing the memory leak.

BZ#1324845 - Federation does not interoperate between MRG-M 2 and 3

The management method `link.bridge` got extra 'credit' argument between MRG 2 and 3. This prevented MRG 2 tools from adding links to the MRG 3 broker. The 'link.bridge' method is now deprecated in favor of 'broker.create' which supports the credit argument. Both MRG 2 and MRG 3 tools can now be used with an MRG 3 broker.

BZ#1329582 - consuming messages from priority ring queue on backup broker can trigger broker segfault

A priority queue can result in messages being deleted from the queue even if they are acquired by a consumer. When this happens, the state of the message is changed to indicate that the message may no longer be valid. However the code that releases acquired messages that have not been acknowledged when the consumer session ends was not checking this state to ensure the messages to be released were still acquired. This meant that an attempt was made to release a dequeued message, causing a segmentation fault. Now, the state is checked before release, preventing the segmentation fault, even if the message has been dequeued.

BZ#1330539 - client does not update credit for expired messages

Messages that expire on the client side are not handled by the usual code path which includes the moving of the credit window. Therefore, if there are a large number of messages that expire on the client side, the flow of messages may stall as the window is never moved forward. This has been

corrected by ensuring the credit window is moved even for expired messages.

BZ#1333437 - segfault when rerouting and deleting a queue in parallel

Missing locks around queue management object meant concurrent queue delete and management operations could occasionally cause a broker crash. This release adds the appropriate locks, preventing the broker crashes.

BZ#1333767 - Occasional segfault when creating & deleting many queues just before primary failover

In previous releases, an interrupted backup recovery left invalid data that could cause memory errors. This release has improved clean up if recovery is interrupted and improved memory safety.

BZ#1339449 - [RFE] Improved broker logging for expired messages

This release contains improved broker logging for expired messages. This feature was added because messages that were expired by the queue cleaner were not logged. Expired messages now receive the same logging treatment as other messages, with the queue name and message properties included in debug-level logs.

BZ#1350023 - Memory leak in primary broker when creating & unsubscribing from an autoDel queue in a loop

When creating then unsubscribing from an autodelete queue, unnecessary information was being stored in the primary cluster node and never released. Thus memory use would increase over time. Now, the code that stored the unnecessary information has been removed and memory no longer accumulates.

BZ#1359506 - Small memory leak when trying to over-subscribe queue with exclusive subscription

Previously, if any worker thread was idle for long periods, routine memory flushing may not have occurred, causing the appearance of a memory leak. This release ensures that all worker threads are made active at least once per minute, meaning memory is released regularly and does not accumulate.

[Report a bug](#)

CHAPTER 5. CHANGES IN 3.2.3

5.1. ERRATA

[RHBA-2016:2921 – Red Hat Enterprise MRG Messaging 3.2.3 Release](#)

BZ#1396281 - source broker of dynamic route doesnt delete bindings after two consumers of same queue bound to dest.broker are gone

It was found that if a dynamic route is created between two brokers, and there are two consumers, both subscribed to the same queue via x-bindings such that the queue will be deleted upon exit, and both consumers exit, the first broker does not delete the binding.

This can cause unexpected problems if a consumer then tries to send messages to the still-existing binding, as the binding is no longer connected to the expected queue.

The code has been fixed so that the bindings are deleted as expected on both brokers when two consumers create then delete the same queue.

[Report a bug](#)

CHAPTER 6. CHANGES IN 3.2.4

6.1. ERRATA

[RHBA-2017:0496 – Red Hat Enterprise MRG Messaging 3.2.4 Release](#)

BZ#1425576 - Broker memory leak in DirectExchange

Previously, brokers could experience slow, consistent memory growth.

This problem occurred because the Exchange::unbind call for DirectExchange was coded to assume that the binding actually existed. When the binding did not exist, Bindingkey entries continued to be added in the BindingKey map until the broker was stopped.

The code has been updated to fix this memory leak.

[Report a bug](#)

CHAPTER 7. CHANGES IN 3.2.5

7.1. ERRATA

[RHBA-2017:27935-02 – Red Hat Enterprise MRG Messaging 3.2.5 Release](#)

BZ#1294824 - Allow qpid C++ client and broker to use TLS 1.1 and TLS 1.2

Previously, only Transport Layer Security (TLS) version 1.0 was supported for use by the broker and C++ client.

As a result, the code did not accept TLS 1.1 and TLS 1.2 as valid protocol versions.

The code has been updated to now support both TLS 1.1 and TLS 1.2 as valid protocol versions.

BZ#1315485 - [AMQP1.0] Setting idle-timeout to ffffffff, broker fires infinite loop of ConnectionTicker with zero timeout

Setting idle-timeout to 0xffffffff on an incoming AMQP 1.0 connection caused broker to go into a loop.

As a result, messages were sent to the logfile and CPU usage increased.

The code generating timer events was corrected to handle this case.

BZ#1332143 - Broker fails to restart after sending too-big message into paged queue

If you had a durable paged queue configured and sent a durable message that was too large to fit into a single page, the broker rejected it, as expected. However, the rejected message was still written to disk, which was not expected.

If you then stopped and attempted to restart the broker, it failed with an error message.

The broker no longer writes a durable message to disk if it is too large to fit in a single page and the broker no longer fails to restart.

BZ#1359372 - [linearstore] overwrite-before-return causing latencies for broker's clients

When the overwrite-before-return option was used on journal files, overwriting the files took a significant amount of time.

Because of this delay, clients were blocked for seconds at a time.

The method used for overwriting journal files has been refactored. Customers should no longer notice a delay or client unavailability when using the overwrite-before-return option.

[Report a bug](#)

CHAPTER 8. CHANGES IN 3.2.6

8.1. ERRATA

[RHBA-2017:2892 – Red Hat Enterprise MRG Messaging 3.2.6 Release](#)

BZ#1487410 - Sending message via exchange when queue is full with reject policy causes closed connection

Previously, if a queue was set up with a reject-when-full policy, and messages were sent to a full queue via an exchange (for example, by using a topic) and AMQP 1.0 protocol was used, the connection to the client was incorrectly closed.

The code has been updated to handle the error correctly so that the connection is not closed. Now a "queue limit exceeded" error is returned.

BZ#1487729 - Improve performance of linearstore when the flush timeout is set to a low value

If the broker journal-flush-timeout config option was set to something below 1 ms, the broker continuously used a noticeable amount of CPU, even at idle.

The flush code has been reworked to minimize CPU usage if there are no messages pending. Now there is almost no CPU usage when the broker is idle, even if the journal-flush-timeout option is set to a very low value.

BZ#1487886 - The journal-flush-timeout config option parsing changed in an incompatible way

In MRG 3.2.5, the journal-flush-timeout config setting parsing was incorrectly changed so that the input value was assumed to be milliseconds.

This caused existing config files to not work the same way they did previously.

The parsing has been restored to its previous functionality, so that the setting is interpreted as seconds unless it has a suffix of "ns" (signifying nanoseconds) or "ms" (signifying milliseconds). Now the journal-flush-timeout setting is interpreted the way it was prior to MRG 3.2.5.

BZ#1487887 - Add tick method to Proton C++ connection_driver API so heartbeats can work with that API

A new method ("tick") was added to the Proton C++ connection_driver API.

Customers creating a server process using proton can now use this method to help manage heartbeats from clients.

[Report a bug](#)

CHAPTER 9. CHANGES IN 3.2.7

9.1. ERRATA

[RHBA-2017:3253 – Red Hat Enterprise MRG Messaging 3.2.7 Release](#)

BZ#1502855 - sending durable messages concurrently from 0-10 and 1.0 producer causes high latency on ACKing the 1.0 ones

Previously, when durable messages were sent to the same queue using both AMQP 1.0 and 0-10 concurrently, a condition could have been triggered where the store no longer flushed the 1.0 messages. When this condition was triggered, the 1.0 messages were only flushed by the 0-10 activity on the queue, or if there was none, then the messages were never flushed.

The code was changed so that any combination of 0-10 and 1.0 messages are successfully processed without causing messages to be delayed.

BZ#1507610 - HA cluster infinite ping-pong of session.detach "invalid session name" errors

In certain error situations, it was possible to create a cycle where each end of a connection replied to a "detach" frame with another "detach" frame in an endless loop. When this occurred, the session could not successfully detach and close, so the cluster could not recover.

The code was changed so that it does not send a detach frame when there is an exception on a session that is already detached. This prevents the endless loop and allows the session to detach and close, and subsequently, the cluster can recover.

BZ#1509058 - proton-c mixing up two links if one has a name that's a substring of the other

If a user created a receiver using a queue name that was a prefix of a previously created receiver's name (on the same connection/session), the broker could return access to the wrong one.

The code is fixed so that access to the correct queue is ensured.

[Report a bug](#)

CHAPTER 10. CHANGES IN 3.2.8

10.1. ERRATA

[RHBA-2018:0250 – Red Hat Enterprise MRG Messaging 3.2.8 Release](#)

BZ#1529929 - federation link heartbeat timer might not be fired on unreliable TCP connections

In previous releases, if TCP/IP connections were unreliable and packets were dropped or delayed, a TCP connection for a federation link could get stuck before the heartbeat timer started. As a result, the TCP connection could hang forever, because there was no way for brokers to know that the connection was unusable.

The heartbeat is now started when the federation link is created, thus allowing a damaged connection to be cleaned up when both sides detect that the heartbeat has timed out.

[Report a bug](#)

CHAPTER 11. CHANGES IN 3.2.9

11.1. ERRATA

[RHBA-2018:0524 – Red Hat Enterprise MRG Messaging 3.2.9 Release](#)

BZ#1554334 - Proton 0.16 C++ schedule API not thread-safe

In previous releases, Qpid Proton C++ version 0.16.0 schedule API was not thread-safe. As a result, clients could experience intermittent issues with missing events and could possibly stop processing incoming messages.

This issue was addressed by updating proton-c and C++ libraries to make the schedule() implementation thread-safe. Now when using the updated libraries, the client is properly notified of all new proton events.

[Report a bug](#)

CHAPTER 12. CHANGES IN 3.2.10

12.1. ERRATA

[RHBA-2018:1334 – Red Hat Enterprise MRG Messaging 3.2.10 Release](#)

BZ#1564132 - invalid bridges can be detached in a stress test, resulting into infinite ping-pong of session.detach

When routes between brokers were deleted, a software defect sometimes allowed other open routes to be closed by mistake. This caused unexpected behavior, including possibly crashing the broker.

This issue was addressed by fixing the code to track which routes to close.

[Report a bug](#)

CHAPTER 13. CHANGES IN 3.2.11

13.1. ERRATA

[RHBA-2018:2680– Red Hat Enterprise MRG Messaging 3.2.11 Release](#)

BZ#1561819 - qpidd segfault at 10 ip 00007f27fbbabe19 sp 00007ffe478c05a0 error 4 in linearstore.so

Due to a coding error related to journal file management, the Qpid broker can sometimes crash at random times when durable queues are present.

The coding error was fixed and the random crashes should no longer occur.

BZ#1588015 - Qpidd deadlocks due to linearstore inactivity timer

A Qpid broker containing durable queues, and having a low value set for the 'journal-flush-timeout' parameter can experience a deadlock, causing the broker to hang. A timing flaw was identified with the underlying state machine that processes the asynchronous events related to the data store.

With the fix, Qpid no longer hangs due to this deadlock condition, and durable messages are processed normally.

BZ#1609230 - memory leak on 1.36 on randomly overwriting priority messages in ring prio queue

When a priority ring queue is filled to overflowing with messages of various priorities, and has no consumer, but only a browsing client, an internal data structure accumulates as the messages are delivered to the client, causing a slow memory leak.

The internal data is now released when the message is delivered to the browsing client and the memory leak is no longer present.

BZ#1616383 - proton [reactor] can fail to close the connection after detecting a local timeout

On idle timeout, the reactor queued a write of the close frame but waited for the OS to indicate that it had buffer space for the write before actually writing the frame and shutting down the socket. This usually happened immediately. But if the kernel output buffer was full at the time of the timeout, and the network was losing packet transfers with the peer such that it couldn't shrink the output buffer, the cleanup of the socket took arbitrarily long.

Now, a best-effort attempt is made to write the close frame when the idle timeout is detected and to close the socket immediately.

BZ#1618908 - Qpid client leaks memory when it connects to an unreachable broker IP address

If a Qpid client attempts to connect to an unreachable broker IP address repeatedly, each iteration causes a small memory leak. This can accumulate over time and cause system memory issues.

Now, after a failed connection, the temporary data is freed correctly and repeated attempts to connect no longer cause memory to accumulate.

[Report a bug](#)

APPENDIX A. REVISION HISTORY

Revision 3.2.0-21 Updated with 3.2.11 Errata.	Mon Sep 10 2018	Susan Jay
Revision 3.2.0-20 Updated with 3.2.10 Errata.	Wed May 16 2018	Susan Jay
Revision 3.2.0-19 Updated with 3.2.9 Errata.	Wed Mar 28 2018	Susan Jay
Revision 3.2.0-18 Updated with 3.2.8 Errata.	Mon Jan 29 2018	Susan Jay
Revision 3.2.0-17 Updated with 3.2.7 Errata.	Fri Nov 17 2017	Susan Jay
Revision 3.2.0-16 Updated with 3.2.6 Errata.	Fri Oct 13 2017	Susan Jay
Revision 3.2.0-15 Updated with 3.2.4 and 3.2.5 Errata.	Wed Jun 7 2017	Susan Jay
Revision 3.2.0-14 Updated with 3.2.3 Errata.	Wed Dec 21 2016	Scott Mumford
Revision 3.2.0-13 Updated with 3.2.2 Errata.	Thu Oct 13 2016	Scott Mumford
Revision 3.2.0-9 Updated with 3.2.1 Errata.	Mon Mar 21 2016	Scott Mumford
Revision 3.2.0-7 MRG-M 3.2 GA	Thu Oct 8 2015	Scott Mumford
Revision 3.2.0-6 Bugzilla sync Oct 7, 2015 12:29:28 PM AEST	Wed Oct 7 2015	Scott Mumford
Revision 3.2.0-1 Prepared for MRG-M 3.2 GA	Tue Jul 14 2015	Jared Morgan