Red Hat Enterprise MRG 3
3.0 MRG Messaging Release Notes

Release Notes for Red Hat Enterprise MRG Messaging 3.0 and its patch releases

Joshua Wulf

Red Hat Enterprise MRG 3 3.0 MRG Messaging Release Notes

Release Notes for Red Hat Enterprise MRG Messaging 3.0 and its patch releases

Joshua Wulf
jwulf@redhat.com

Legal Notice

Abstract

These release notes contain important information related to Red Hat Enterprise MRG Messaging v3.0.0, v3.0.1, and v3.0.2. Read these Release Notes in their entirety before installing the product.

Table of Contents

Chapter 1. Introducing Messaging 3 . . . . . **2**
1.1. Red Hat Enterprise Messaging 3.0 2
1.2. The Top Six Differences between MRG Messaging 2 and 3 2
1.3. Documentation 2

Chapter 2. Changes in 3.0.0 . . . . . **3**
2.1. Known Issues 3
2.2. Enhancements 3
2.3. Bug Fixes 6
2.4. Deprecated Functionalities 10

Chapter 3. Changes in 3.0.1 . . . . . **11**
3.1. Errata 11
3.1.1. RHBA-2014:1952 – Red Hat Enterprise MRG Messaging 3.0 Release 11
3.2. Customer Reported Defects Since 3.0.0 11

Chapter 4. Changes in 3.0.2 . . . . . **12**
4.1. Errata 12
4.1.1. RHSA-2015:0707 – Moderate: qpid-cpp security and bug fix update 12
4.2. Customer Reported Defects Since 3.0.1 13

Revision History **14**

Chapter 1. Introducing Messaging 3

1.1. Red Hat Enterprise Messaging 3.0

Red Hat Enterprise Messaging version 3 is the next generation release of the Red Hat Enterprise Messaging product, based on the Apache Qpid project.

Version 3 introduces enhancements and replacement components, notably High Availability and support for the AMQP 1.0 standard.

Report a bug

1.2. The Top Six Differences between MRG Messaging 2 and 3

These are the most significant differences between MRG 2 and MRG 3:

1. The broker and the C++ messaging library (**`qpid::messaging`**) now offer amqp1.0 support via the Apache Proton library (note that transactions are not yet available over amqp1.0).
2. Clustering has been replaced with a new High Availability implementation.
3. Queue Threshold alerts are now edge-triggered, rather than level-triggered. This improves alert rate limiting.
4. The flow-to-disk implementation has been changed to *disk-paged queues* to more efficiently use memory.
5. The **`ring-strict`** limit policy has been dropped.
6. The messaging journal has been replaced with a new implementation - the dynamically-expanding *Linear Store*.

Report a bug

1.3. 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 3.0.0

2.1. Known Issues

qpid-cpp

1095809 - Unwanted and misleading SSL info messages on qpidd service startup

It was discovered that misleading SSL messages appeared in logs if the HA Module was loaded into the Broker and logging was turned up to INFO or higher. Messages similar to the following are displayed: "2014-04-29 10:17:34 [Security] info SSL connector not enabled, you must set QPID_SSL_CERT_DB to enable it." This message is harmless. The message is not referring to the regular SSL functionality of the broker: it is referring to the outgoing client functionality of the HA client. There is no workaround to this issue, except to disregard this message if it appears in the logs.

1129997 - qpid c++ client AMQP 1.0 throughput performance regression

It was discovered that if the capacity (property of qpid::messaging::Receiver class) and ack-frequency (setting on qpid-recv utility) were both at 100, there was a sudden drop in throughput. Reducing the ack-frequency or increasing the capacity, even by a very small amount, was found to make a considerable difference in throughput.

In general, a value higher than 100 is recommended to test throughput. A suggested value for the qpid::messaging::Receiver capacity parameter would be between 500-1000. A lower value is suitable for the qpid-recv utility ack-frequency parameter (which sets the frequency at which qpid::messaging::Session::acknowledge() is called). For example, acknowledging every 10 messages is unlikely to negatively impact the performance.

Report a bug

2.2. Enhancements

qpid-cpp

588504 - [RFE] [MRG] Messaging broker to listen only on SSL port

The MRG Messaging Broker is now able to require SSL connections. This feature is required to allow customers to comply with some security policies. The MRG Messaging Broker can now be prohibited from listening for regular TCP connections by specifying `--listen-disable tcp` on the broker command line, or by using the equivalent configuration file option.

771830 - Enable high resolution timestamps for qpidd logs using QMF and a qpidd option.

It is now possible to enable or disable high resolution log time stamps while the broker is running. If time stamps are set to low resolution then many issues cannot be diagnosed effectively. Restarting the broker to enable high resolution time stamps was often not an option for users.

572456 - [Feature request] Provide the possibility to choose the network interface to bind qpidd.

In MRG Messaging 3.0, it is now possible to select the network interface to bind qpidd to. This allows customers to restrict the networks to which qpidd is exposed. This may be used

as part of the security policy of the installation. Customers can now use the qpidd --interface command to bind qpidd to a network interface, but not a specific port. This command is documented in the "General Broker Options" section of the Messaging Installation and Configuration Guide.

768114 - Enable TCP_NODELAY by default

The qpid C++ clients and broker now deactivate the Nagle algorithm by default. The Nagle algorithm improves the connection bandwidth by sacrificing some latency. Under most circumstances, using qpid will achieve identical or improved performance with the Nagle algorithm deactivated. Where this is not the case, the algorithm can be enabled by specifying the **`--tcp-nodelay=no`** option on the command line for the broker, or **`tcp_nodelay=no`** in the configuration file. This enhancement should result in a performance improvement for most user configurations. This feature is documented in the "General Broker Options" section of the Messaging Installation and Configuration Guide.

949014 - Setting --ack value on Federation links forces a fixed finite credit limit.

This feature allows credit level configuration on Federation connections. By default, the credit level was fixed to unlimited, or forced to twice the --ack value (if specified). There was no other way to configure the credit level. The user can now better tune the performance of the Federation link by specifying custom credit levels. This feature is documented in the "qpid Route Options" section of the Messaging Installation and Configuration Guide.

797092 - Support log category exclusion

Logs are sometimes flooded by thousands of log messages that may not be considered important during issue investigation. It is now possible to specify a broker option that turns off log messages using precise filters. During problem investigation, users can control log file content to better understand a specific issue, rather than manually filtering out extraneous log events. The feature is supported in the C++ Broker, by specifying --log-disable.

808105 - ACL syntax does not allow specifying '' exchange

ACL PUBLISH EXCHANGE rules now have a simplified way to refer to the nameless default exchange. In situations where the default exchange requires ACL rules, it is now possible to name the unnamed exchange by specifying the keyword **`amq.default`** in the ACL rule syntax.

709325 - Possibility to create queue files dynamically and resizing queues without deleting them

A new store implementation named Linearstore replaces Legacystore in this release. Linearstore addresses limitations imposed by Legacystore, specifically its use of fixed circular file buffers for queue journals. Linearstore utilizes a pool of fixed-size empty journal files and appends the journal files to a queue journal in a linear manner. The file is returned to the pool once a journal file is cleared of messages, and has no preceding journal files with messages waiting. There are a number of changes between Legacystore and Linearstore users must be aware of. See Comment 33 in the original Bugzilla Enhancement for further usage caveats and migration limitations. This information will be migrated to formal user documentation shortly after the 3.0 Release.

874516 - [RFE] Per-user configurable limit on the number of active connections

It is now possible to configure the active connection limit for individual users. Having one setting for all users was identified as being insufficient: Administrators and Broker monitors require many connections while public clients must be limited to few connections. With the new scheme, individual users can now have connection limits set through the ACL file. Both

users, and groups of users can now be configured to receive many or few connections (including zero connections if this is applicable).

885207 - Move all qpid config files under the qpid config dir

The qpidd.conf file has changed locations in MRG Messaging 3.0. This change improves consistency by moving Qpid daemon and client configuration under one location. The config file is installed to **/etc/qpid/qpidd.conf**, and the broker now uses this new **qpidd.conf** file. If any changes were made to the old **qpidd.conf** file, it is preserved as **/etc/qpidd.conf.rpmsave** to allow any configuration to be migrated into the new file manually.

915821 - RFE: Support per user queue quotas in ACL file

It is now possible to specify queue quotas on a per-user basis in the ACL file. The normal approach of making a single command line switch available for setting queue quotas was insufficient: Administrators need to create many queues and normal users must be constrained to fewer or none. With the settings available in the ACL file, each user, group of users, or all otherwise unnamed users can be given a different quota. A quota value of zero prevents the user from creating any queues.

955674 - ACL delete action should not ignore object's properties other than name

Users can now further restrict ACL rules with respect to the queues and exchanges they are allowed to delete. It is now possible to allow or deny access to deleting queues and exchanges based on durability and other settings. Restricting queue and exchange deletion prevents users from interfering with each other in a broker.

691411 - [RFE] mechanism to detect when messages are overwritten in ring-type queues

Because users need to be able to determine when a ring queue was full and further writes would overwrite older messages, it is now possible to store and retrieve a queue level sequence number in message properties. Enabling queue sequence numbers allows gap detection in the sequence numbers, which indicate that messages were overwritten. Sequence numbers are retrieved from the message properties using the same user defined key. The feature is enabled by declaring a queue with argument **qpid.queue_msg_sequence** set to the user-defined key. There is no persistence support in this implementation: sequence numbers are reset after a broker restart.

453539 - Support message selectors

MRG Messaging C **components now include support for Message Selectors. Other languages had support for selectors, but C** was missing this. Customers required this functionality in the product. Selector support is now available in the 0-10 and 1-0 protocols for the C++ messaging client and broker code, and implements the selector language with some limitations. This feature is documented in the "Select Messages Using a Filter" section of the Messaging Programming Reference Guide.

824988 - [RFE] Federated link heartbeat interval is hard coded to 120 seconds

This Enhancement introduces a configurable link heartbeat interval for the qpidd broker. In a worst-case scenario, the previous heartbeat default of 120 seconds would result in a system recovery under 240 seconds. For High Availability environments, this amount of time was considered to be too long, and a user-configurable time in seconds was required. The qpid broker now has an option **link-heartbeat-interval**, which allows a custom heartbeat interval (in seconds) to be configured. This feature is documented in the "Broker HA Options" section in the Messaging Installation and Configuration Guide.

862899 - [RFE] Extend the qpidd --wait option to work at shutdown

It is now possible to use the --wait option available in C++ Brokers to control both startup and shutdown wait times. Brokers with large databases and process spaces may take longer than the hard-coded 10 seconds to shut down successfully. Enabling the --wait option to use it's value in the shutdown timer allows users to specify the shutdown wait time beyond 10 seconds. This allows users to successfully shut down large-scale brokers without error.

qpid-tools

710429 - qpid-cluster, qpid-tool and qmf-tool do not allow SASL mechanism to be chosen

The qpid-tool did not allow the SASL mechanism to be chosen on the command line. It was not possible to override the default logic of choosing the most secure method available. The enhancement adds the --sasl-mechanism and --ssl-certificate command line options. The SASL mechanism and certificate file can now be specified on the command line to override the default.

915774 - qpid-tool to have -b option as other qpid tools

To align qpid-tool with functionality provided by other qpid tools, a "-b" command line option is now available to specify a broker address. The parameter takes the same form as the other qpid tools.

Report a bug

2.3. Bug Fixes

python-qpid

950501 - python clients throw "[Errno 1] _ssl.c:1217: error:1409F07F:SSL routines:SSL3_WRITE_PENDING:bad write retry"

In circumstances when the python client code tried to re-write queued data to a previously blocked socket, it was discovered that the python client code was changing the pointer used prior to the socket blocking. The underlying OpenSSL library detected that the pointer had changed since the last call to write, and threw an exception. The python client is now modified to keep the original pointer constant when new data is added. When the socket becomes writable again, the address of the pointer passed to the socket is identical to the address passed prior to the socket blocking, as required by OpenSSL.

908224 - reconnect_timeout ignored in qpid.messaging.Connection()

It was discovered that the **`reconnect_timeout`** argument was ignored when creating a connection. Passing a **`reconnect_timeout`** when creating a connection did not raise an exception after the specified number of seconds. The fix passes the **`reconnect_timeout`** to the **`open()`** method, which allows an attempt to create a connection to timeout correctly.

719589 - Python example "server" fails

It was discovered that the server example application referenced a non-existent **`ReceiveError`**, which caused the server example to crash upon start-up. The fix corrects the ReceiveError to ReceiverError, which allows the server example to operate as expected.

772028 - malfunctioning unsettled() receiver method

It was discovered that the python messaging method **`receiver.unsettled()`** referenced a non-existent "acked" variable. Calling **`receiver.unsettled()`** raised an exception. The fix now uses **`session.acked`** instead of **`receiver.acked`**, which ensures the correct number of unsettled messages are returned.

qpid-cpp

1023322 - qpid c++ client Address string compatibility regression in parsing (long) int value postfixed by 'L'

It was discovered that numeric queue configuration options specified as strings were not checked when parsed, to ensure there were no trailing non-digit characters after the number. Values such as 40000L were previously accepted as was 40000xyz, with the trailing characters simply being ignored. This may have given the incorrect impression in some cases that values such as 40000L were recognized as being a long integer. The parsing logic when converting from string to numeric option now ensures there are no illegal trailing characters. Values such as 40000L are now rejected as invalid.

1012852 - [AMQP 1.0] Message size resolution differs for AMQP 0.10 and 1.0

It was discovered that qpid-cpp was only reporting broker statistics or enforcing queue depth limits based on the size of the AMPQ 0-10 message body, instead of additionally including the header size. This caused statistics to incorrectly reflect the size of headers in the message. For example, a message with no body but many headers would not contribute to the aggregate size in bytes of a queue onto which it was enqueued. Because AMQP 1.0 messages included headers and other meta-data with the message, this issue had the potential to introduce unnecessary confusion. The size of AMQP 0-10 messages is now fixed to take the sum of the content and headers segments into account when calculating the logical size of the message. Statistics and queue limits reported or controlled in bytes now more accurately reflect the logical size of the message, and there is less difference between the 1.0 and 0-10 encodings in terms of their overall size. It is expected that queue limits will be reached earlier due to the correct logical size now being reported and enforced.

913448 - invoking Receiver::fetch() in a loop for slow producer causes only first <prefetch> messages received

It was discovered that if a fetch timed-out, information about the current credit window quota was cleared from the broker. This prevented the credit window reported by the client to be moved forward by the broker, and prevent a receiver from receiving messages even though there were messages available on the queue. The broker is now notified which messages have been received whenever a fetch times out. The credit window is now always moved forward correctly, and messages continue to be received when available.

919017 - XML exchange does not update statistics when no binding key matches routing key

It was discovered that doRoute(msg, b) was not called when the routing key of a message did not match the binding key of an XML exchange. This caused the QMF statistics msgIn, msgDrop, byteIn and byteDrop to incorrectly update. The fix calls doRoute(...) even if the routing key and binding key do not match. The QMF statistics now update as expected.

865689 - "Recovery found" logs of wrong severity

An issue with journal recovery messages reported at the WARN level caused customers to assume that action on their part was required to fix the issue when this was not the case. The fix lowers the priority of the recovery messages to INFO level, which sets the appropriate log level for this normal provisioning setting.

955578 - Multi-threaded fetching messages from different queues causes fetch delay

It was discovered that if there was more than one thread concurrently processing the session queue, a message of interest to one thread may be processed by another thread preventing the *correct* thread from seeing it. The fix ensures only one thread at a time can process the incoming queue. Concurrent fetch() calls now work as expected without needing to timeout to see a message as it arrives.

912519 - broker ignores if_empty and if_unused flags when deleting queues

It was discovered that the C++ broker ignored the **`if_empty`** and **`if_unused`** flags when a queue was deleted. If the queue was in use or non-empty, and a call was made to delete the queue only if it was unused or empty, the queue would incorrectly be deleted anyway. The fix inspects the **`if_empty`** and **`if_unused`** flags, and will not delete the queue if it is in use or non-empty when appropriate. Queues are no longer incorrectly deleted when the **`if_empty`** or **`if_unused`** flags are used.

740152 - Invalid handling of qpid.max_size

It was discovered that the broker retrieved the maximum queue size option as a uint32, even though it should be a uint64. The maximum value expressible for the aggregate queue size in bytes was needlessly limited, which prevented large queue sizes from being configured. The broker now correctly retrieves the size as a uint64. Any value expressible as a uint64 is now valid as the maximum queue size (in bytes).

qpid-java

784270 - AMQDestination equals() validation is broken for addresses

It was discovered that the AMQDestination equals method did not take into account the differences between the ADDR and BURL syntax. The legacy fields were not populated until the address was validated resulting in a null pointer exception if equals was called beforehand. The fix now handles equals and hashcode separately for ADDR and BURL, which allow the equals and hashcode methods to behave as expected.

828442 - When sending a message, Java client gets unauthorized-access error due to using "ILADDRESS=<emailaddr>" as the user-id

The Java client code was excepting the Certificate Name (CN) to be the first entry in the name string. If the CN was not the first entry, the extracted value was not the correct user-id, which could have resulted in issues when message authentication was used. The fix now ensures the parsing logic finds the CN string first before extracting the user-id, which results in the correct user-id not being extracted from the name string.

845999 - second invocation of createConsumer fails for queue in JNDI properties file

If the address had a delete directive, the queue or exchange would be deleted when the consumer or producer closed (as specified in the delete directive). Because the destination was marked as resolved, no validation (or creation) of the queue or exchange was performed when it was used again. A not-found-exception occurred if the same destination was reused to create another producer or consumer after the queue or exchange was deleted. The fix now resets the address resolution flag when the queue or exchange is deleted. If the same exchange is re-used, it will go through address resolution again. If there is a directive to create the queue or exchange, it will be created. An error is thrown otherwise.

890532 - NullPointerException raised in AMQDestination.equals method

It was discovered that the AMQDestination.equals method did not take into account the differences between the ADDR and BURL syntax. The legacy fields were not populated until the address was validated, which resulted in a null pointer exception if equals was called beforehand.The fix now handles equals and hashcode separately for ADDR and BURL, which now allow the equals and and hashcode methods to behave as expected.

896570 - retries parameter ignored for the very first broker

It was discovered that when the current broker details were not set, the _serverRetries parameter was set to zero. This caused the Java client failover to not attempt to retry after the very first connection loss to the broker it was already connected to. The fix sets the current broker details in the constructor, which ensures the retry parameter is honored on the first broker when set.

728509 - Specifying username/password in JMS clients should not be mandatory

It was discovered that the URLParser threw an exception if the username and password was missing from a connection URL. Due to this behavior, it was mandatory to specify a username and password, even if the SASL mechanism chosen did not require it. The URL parser no longer throws an exception if the username or password is missing. Instead it checks if the chosen SASL mechanism (selected during connection negotiation) requires it and then throws an exception at that point.

755651 - getJMSReplyTo does not return null for null ReplyTo property

It was discovered that the replyTo structure in getJMSReplyTo was checked for null values, however it's attributes were not checked. The replyTo struct could be set with null for the exchange and routing key, which resulted in a Destination being created with the exchange and routing key set to null on the receivers side. The fix checks for a null exchange and routing key, and returns null. This allows getJMSReplyTo to return null as designed.

705015 - Message.setJMSMessageID() not compatible with ActiveMQ MessageID

It was discovered that the message implementation did not allow a non-UUID string to be set as the message ID. When a Qpid JMS message was sent through a third-party message provider, setting the message-id resulted in an exception if the message-id was not a valid UUID. The fix now allows any string prefixed with an "ID:", to be set as the message ID. When sending the message through the Qpid JMS producer, it is set as a UUID (as per the JMS spec and AMQP spec).

qpid-sdk

715284 - Examples C++ "hello_world" and "hello_xml" are not able to be compilled from solution file

It was discovered that the hello_world and hello_xml projects were not produced by default from the distribution kit. Users requiring these projects needed to add the projects manually to the `CMakeLists.txt` file before running cmake. The fix adds the files to CMakeLists.txt, which results in hello_world and hello_xml being produced by default.

qpid-tools

872111 - qpid-config traceback in case of ACL denial

It was discovered that if an operation was performed that caused an ACL denied exception, and that connection was closed, the entire backtrace of the ACL exception was displayed instead of only the ACL denied error message. The fix now detects when an ACL exception is raised during connection close, and does not display the backtrace. Instead, only the ACL error text is displayed when the connection is closed.

895515 - '--ssl-key' option missing in several management tools

It was discovered that some of the QPID command-line tools did not provide a way for the user to supply a private key when a certificate was used to identify the user of the command to the broker. This caused the command to fail because it was not able to use the certificate without the key. The fix ensures all QPID command line tools that allow user identification through a self-identifying certificate now allow the private key to be supplied via the **--ssl-key** option. This option takes a path to a file that contains the certificate's private key in PEM format. The command line tool now presents the certificate to the broker for authorization, and the command is executed successfully. This feature is documented in the "Enable SSL in Python Clients" section of the Messaging Installation and Configuration Guide and the "Connection Options Reference" of the Messaging Programming Reference Guide.

Report a bug

2.4. Deprecated Functionalities

qpid-cpp

995039 - Remove the old client API

The Old Qpid Client C++ API (in namespace qpid::client) has been removed from the developer packages. This API is now old, not actively maintained, brings in some undesirable dependencies, and is therefore not recommended for new code. The Qpid Messenging API (in namespace qpid::messaging) has equivalent functionality and can be used instead. All new C++ messagaing code should use the Qpid Messaging API. Previous code written using the Qpid Client code may require modification to use the Qpid Messaging API.

Report a bug

Chapter 3. Changes in 3.0.1

3.1. Errata

3.1.1. RHBA-2014:1952 – Red Hat Enterprise MRG Messaging 3.0 Release

RHBA-2014:1952 – Red Hat Enterprise MRG Messaging 3.0 Release

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

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 BZ#1150397 for more information.

A vulnerability was discovered in the SSLv2 and SSLv3 protocols, which is commonly referred to as POODLE. POODLE stands for Padding Oracle On Downgraded Legacy Encryption. This vulnerability allows a man-in-the-middle attacker to decrypt ciphertext using a padding oracle side-channel attack. POODLE affects older standards of encryption, specifically Secure Socket Layer (SSL) version 2 and 3. It does not affect the newer encryption mechanism known as Transport Layer Security (TLS). As such, these updated packages implement the recommended action to disable SSLv2 and SSLv3 in the C++ broker (qpid-cpp) (BZ#1153763), Windows C++ client (qpid-qmf) (BZ#1153775), and Java client (qpid-java) (BZ#1153779).

Users of the Messaging capabilities of Red Hat Enterprise MRG 3.0, which is layered on Red Hat Enterprise Linux 6, are advised to upgrade to these updated packages, which provide numerous bug fixes and enhancements.

Report a bug

3.2. Customer Reported Defects Since 3.0.0

qpid-cpp

BZ#1150397 - [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.

Report a bug

Chapter 4. Changes in 3.0.2

4.1. Errata

4.1.1. RHSA-2015:0707 – Moderate: qpid-cpp security and bug fix update

RHSA-2015:0707 – Moderate: qpid-cpp security and bug fix update

Updated qpid-cpp packages that fix two security issues and one bug are now available for Red Hat Enterprise MRG Messaging 3 for Red Hat Enterprise Linux 6.

Red Hat Product Security has rated this update as having Moderate security impact. Common Vulnerability Scoring System (CVSS) base scores, which give detailed severity ratings, are available for each vulnerability from the CVE links in the References section.

Security Fixes

CVE-2015-0223

It was discovered that the Qpid daemon (qpidd) did not restrict access to anonymous users when the ANONYMOUS mechanism was disallowed.

CVE-2015-0224

A flaw was found in the way the Qpid daemon (qpidd) processed certain protocol sequences. An unauthenticated attacker able to send a specially crafted protocol sequence set could use this flaw to crash qpidd.

Bug Fixes

BZ#1175872

Previously, the neutron messaging client rewrote (by method of "monkey-patching") the python selector module to support eventlet threading. The rewritten client did not update select.poll() during this process, which is used by qpid-python to manage I/O. This resulted in poll() deadlocks and neutron server hangs. The fix introduces updates to the python-qpid library that avoid calling poll() if eventlet threading is detected. Instead, the eventlet-aware select() is called, which prevents deadlocks from occurring and corrects the originally reported issue.

BZ#1186694

It was discovered that the QPID Broker aborted with an uncaught UnknownExchangeTypeException when the client attempted to request an unsupported exchange type. The code for the Exchange Registry and Node Policy has been improved to prevent this issue from happening again.

BZ#1193425

Because of changes in the included security fixes, starting with this release Windows client programs that use the default PLAIN authentication may now fail if authentication is turned off on the broker, where before they erroneously succeeded. To work around this, explicitly use ANONYMOUS by using **`--connection-options '{sasl_mechanisms:ANONYMOUS}'`** or you can use a valid username (to work with the default PLAIN mechanism if you have it set up). For example, with valid user 'jsmith', it would be **`--connection-options '{username:jsmith}'`**.

Report a bug

4.2. Customer Reported Defects Since 3.0.1

There are no customer reported defects included in this micro release, apart from the fixes the errata provide.

Report a bug

Revision History

Revision 3.0.2-2	**Fri Mar 20 2015**	**Jared Morgan**
Prepared for MRG-M 3.0.2 errata.		
Revision 3.0.2-1	**Tue Mar 10 2015**	**Jared Morgan**
Prepared for MRG-M 3.0.2 GA		
Revision 3.0.1-2	**Fri Dec 5 2014**	**Jared Morgan**
Prepared for MRG-M 3.0.1 GA Minor correction to URL.		
Revision 3.0.0-4	**Wed Sep 24 2014**	**Jared Morgan**
Prepared for MRG-M 3.0 GA		