



Red Hat Enterprise MRG 2

MRG Release Notes

Release Notes for the Red Hat Enterprise MRG 2

Tomáš Čapek
Cheryn Tan

Douglas Silas
Joshua Wulf

David Ryan

Red Hat Enterprise MRG 2 MRG Release Notes

Release Notes for the Red Hat Enterprise MRG 2

Tomáš Čapek
Red Hat Engineering Content Services
tcapek@redhat.com

Douglas Silas
Red Hat Engineering Content Services
dhensley@redhat.com

David Ryan
Red Hat Engineering Content Services
dryan@redhat.com

Cheryn Tan
Red Hat Engineering Content Services
cheryntan@redhat.com

Joshua Wulf
Red Hat Engineering Content Services
jwulf@redhat.com

Legal Notice

Copyright 2013 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, MetaMatrix, 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. All other trademarks are the property of their respective owners. 1801 Varsity Drive Raleigh, NC 27606-2072 USA Phone: +1 919 754 3700 Phone: 888 733 4281 Fax: +1 919 754 3701

Keywords

Abstract

These Release Notes contain important information available at the time of release of Red Hat Enterprise MRG 2 and subsequent minor version releases. Known issues and most important new features and bug fixes are described here. Read this document before beginning to use the Red Hat Enterprise MRG distributed computing platform.

Table of Contents

Chapter 1. System Requirements	5
1.1. Supported Hardware and Platforms	5
1.2. Installing and Configuring Red Hat Enterprise MRG	6
Chapter 2. MRG Messaging	7
2.1. Messaging 2.3 Release	7
2.1.1. Known Issues	7
2.1.2. Deprecation Notices	8
2.2. Messaging 2.2 Release	8
2.2.1. Known Issues	8
2.3. Messaging 2.1 Release	9
Changes introduced to Red Hat Enterprise MRG on April 30, 2012:	9
Changes introduced to Red Hat Enterprise MRG on January 23, 2012:	9
2.3.1. Known Issues	9
The following known issues were reported for Red Hat Enterprise MRG as of April 30, 2012:	9
2.4. Messaging 2.0 Release	10
2.4.1. Update Notes	11
Chapter 3. MRG Realtime	17
3.1. Realtime 2.3 Release	17
3.1.1. Technology Preview	17
3.2. Realtime 2.2 Release	18
3.2.1. Known Issues	18
3.3. Realtime 2.1 Release	18
3.3.1. Known Issues	19
3.4. Realtime 2.0 Release	19
3.4.1. Update Notes	19
Chapter 4. MRG Grid	21
4.1. Grid 2.3 Release	21
4.1.1. Technology Preview	21
4.1.2. Known Issues	22
4.2. Grid 2.2 Release	23
4.2.1. Known Issues	24
The following known issue was reported for Red Hat Enterprise MRG as of October 16, 2012:	24
4.3. Grid 2.1 Release	25
Changes introduced to Red Hat Enterprise MRG on February 6, 2012:	25
Changes introduced to Red Hat Enterprise MRG on January 23, 2012:	25
The following release notes were issued for Red Hat Enterprise MRG on February 6, 2012:	
4.3.1. Technology Previews	26
The following technology previews were introduced to Red Hat Enterprise MRG on January 23, 2012:	26
4.3.2. Known Issues	26
4.4. Grid 2.0 Release	27
4.4.1. Update Notes	28
Chapter 5. MRG Management Console	36
5.1. Management Console 2.2 Release	36
5.1.1. Known Issues	36
5.2. Management Console 2.1 Release	37
5.2.1. Technology Previews	38

Dependencies	40
Feedback	40
Full Support	40
Where to Find This Information	40
5.2.2. Known Issues	40
The following known issues were reported for Red Hat Enterprise MRG as of February 6, 2012:	40
The following known issues were reported for Red Hat Enterprise MRG as of January 23, 2012:	40
5.3. Management Console 2.0 Release	41
5.3.1. Update Notes	41
Chapter 6. Red Hat Enterprise MRG 2 Package Manifest	49
6.1. Red Hat Enterprise MRG 2.1	49
6.2. Red Hat Enterprise MRG 2.0	49
6.2.1. Messaging 2.0 Release	49
6.2.2. Realtime 2.0 Release	50
6.2.3. Grid 2.0 Release	51
6.2.4. Management Console 2.0 Release	53
Revision History	54

Chapter 1. System Requirements

This section contains information related to installing Red Hat Enterprise MRG, including hardware and platform requirements.

1.1. Supported Hardware and Platforms

Red Hat Enterprise MRG is highly optimized to run on Red Hat Enterprise Linux 6.2 and later due to its inclusion of MRG Realtime. The MRG Messaging and MRG Grid capabilities can also run on other platforms, but without the full benefits of running on Red Hat Enterprise Linux 5.7 and later.

Table 1.1. Supported Hardware and Platforms

	Red Hat Enterprise Linux 5.7 (32-bit and 64-bit)	Red Hat Enterprise Linux 6.2 (32-bit and 64-bit)	Windows XP SP3+ (32-bit)	Windows Server 2003+ (32-bit and 64-bit)	Windows Server 2008 (32-bit and 64-bit)	Windows Server 2008 R2 (64-bit)	Windows 7 (32-bit and 64-bit)
MRG Messaging Native Linux Broker	X	X					
MRG Messaging Client - Java/JMS [a]	X	X					
MRG Messaging Client - C++	X	X	X	X	X	X	X
MRG Messaging Client - Python	X	X	X	X	X	X	X
MRG Messaging Client - Ruby <i>preview</i>	X	X					
MRG Grid Scheduler	X	X					
MRG Grid Execute Node	X	X	X	X	X	X	X
MRG Realtime		X (64-bit only)					

[a] The Java and JMS MRG Messaging Clients are supported for use with Java 1.5 and Java 6 JVMs. For Sun JVMs, it is recommended to use Java 1.5.15 or later or 1.6.06 or later.

1.2. Installing and Configuring Red Hat Enterprise MRG

In order to download and install Red Hat Enterprise MRG 2.1 on your system, you need to subscribe to the appropriate channels on the Red Hat Network (RHN).

Table 1.2. Red Hat Network Channels

Channel Name	Platform	Architecture
MRG Grid	RHEL-5 Server	32-bit, 64-bit
MRG Grid	RHEL-6 Server	32-bit, 64-bit
MRG Grid	non-Linux	32-bit
MRG Grid Execute Node	RHEL-5 Server	32-bit, 64-bit
MRG Grid Execute Node	RHEL-6 Server, Compute Node	32-bit, 64-bit
MRG Management	RHEL-5 Server, RHEL-6 Server, Compute Node	32-bit, 64-bit
MRG Messaging	RHEL-5 Server	32-bit, 64-bit
MRG Messaging	non-Linux	32-bit
MRG Messaging Base	RHEL-5 Server	32-bit, 64-bit
MRG Realtime	RHEL-6 Server	64-bit

Chapter 2. MRG Messaging

MRG Messaging is a high-speed reliable messaging distribution for Linux based on AMQP (Advanced Message Queuing Protocol). This open protocol standard for enterprise messaging is designed to make mission-critical messaging widely available as a standard service and to make enterprise messaging interoperable across platforms, programming languages, and vendors. MRG Messaging includes an AMQP 0-10 messaging broker; AMQP 0-10 client libraries for C++, Java JMS, and Python; as well as persistence libraries and management tools.

2.1. Messaging 2.3 Release

These updated packages for Red Hat Enterprise Linux provide numerous enhancements and bug fixes for the Messaging component of MRG. Some of the most important enhancements include:

- ▶ The **--max-session-rate** option has been removed from the broker options. Queue-based flow control should be used instead.
- ▶ The number of queues per user may now be limited using the **--max-queues-per-user** broker option.
- ▶ ACL-based control of queues has been enhanced with the addition of parameters to control the upper and lower limits of file count and file size on queues.
- ▶ ACL rules now support a wider range of wildcards.
- ▶ ACL rules now support user and domain name substitution, to allow a single rule to apply to all users.
- ▶ Logging has been enhanced. A new logging category allows object lifecycle tracking for auditing and troubleshooting.
- ▶ Expired messages are now logged with the queue name and message properties at debug level.
- ▶ A new ACL Lookup Query Method enables testing of ACL rules.
- ▶ The JMS client can now send and receive messages encoded as AMQP 0-10 lists, which enables the use of QMF while using standard JMS interfaces.
- ▶ Broker QMF management events now supports additional parameters to allow ease of identifying when a remote client has connected or disconnected from a broker.
- ▶ The mechanism list in the file **/etc/sasl2/qpidd.conf** has been changed to "ANONYMOUS DIGEST-MD5 EXTERNAL PLAIN". GSSAPI is no longer included in this list.
- ▶ SSL/TLS support has been added for the messaging API, which allows clients using the messaging API to connect to the broker over a connection encrypted using SSL/TLS.
- ▶ The browse-only queue type has been introduced. This queue allows clients to only browse, but not consume, messages.
- ▶ The command-line tools **qpidd-stat** and **qpidd-config** have been updated. Some command-line options are changed. Refer to the Installation and Configuration Guide for details.
- ▶ Refer to the updated documentation for a list of new and changed sections.

2.1.1. Known Issues

qpidd-cpp, BZ#[909986](#)

Messages sent from the C++ client to a Header exchange are not routed by default. This is because headers are sent from the C++ client encoded in binary, which is not read by the broker. The solution is to specify utf-8 encoding for message headers which are used for binding when sending to a Headers exchange from the C++ client, like so:

```
message.getProperties()["header1"].setEncoding("utf8");
message.getProperties()["header2"].setEncoding("utf8");
```

qpid-cpp, BZ#[884844](#)

The Windows qpid client currently does not support the **DIGEST_MD5** authentication mechanism. The supported SASL mechanisms on Windows qpid clients are **ANONYMOUS** and **PLAIN**. Windows HTCondor nodes should use **PLAIN** authentication to communicate with the qpid broker.

2.1.2. Deprecation Notices

The following are features that may be removed in future releases. New implementations should not utilize these features, and existing implementations should think about alternatives.

qpid::clientAPI

The **qpid::client** API is ABI unstable and should not be used. The **qpid::messaging** API supersedes it, and should be used instead.

QMF v.1

QMF version 1 is superseded and should not be used. Use QMF v.2 instead. The QMF v.1 API namespaces are **qpid::management** and **qpid::console**.

qpid-perftest

The **qpid-perftest** tool will be removed in a future release. A replacement utility will be available at that time.

2.2. Messaging 2.2 Release

These updated packages for Red Hat Enterprise Linux provide numerous enhancements and bug fixes for the Messaging component of MRG. Some of the most important enhancements include:

- ▶ This update adds SSL support with client authentication (using PEM certificates) to all management tools in the *qpid-tools* for Red Hat Enterprise Linux 6. Now, connections using SSL work correctly both with and without client authentication.
- ▶ This update introduces the new **--max-negotiate-time** qpid broker option. This option prevents a possible denial of service security concern. It specifies the maximum amount of time a new connection to the broker has in order to do protocol negotiation and authenticate itself. If this procedure does not complete in the specified time, the connection is aborted by the broker.
- ▶ This update introduces new command-line parameters to specify connection limits. A new code to monitor connections enforces the limits. Now, individual users cannot consume all the broker's connection resources and deny service to other users.

2.2.1. Known Issues**qpid-cpp, BZ#[771169](#)**

Under rare circumstances, use of the **qpid.alert_repeat_gap** or **x-qpid-minimum-alert-repeat-gap** queue parameters introduces a timer-like event into the broker that can cause a clustered broker to fail, if another broker in the cluster has **qpid-tool**, or some other management tool, attached. This low-frequency failure occurs when one of the brokers reports a threshold event to the management tool (for example, when a ring-queue becomes full) while

the other broker continues to handle new input. To work around this bug, use the **default-event-threshold-ratio=0** option on all clustered brokers.

2.3. Messaging 2.1 Release

These updated packages for Red Hat Enterprise Linux provide numerous enhancements and bug fixes for the Messaging component of MRG. Some of the most important enhancements include:

Changes introduced to Red Hat Enterprise MRG on April 30, 2012:

- ▶ Support for Microsoft Visual Studio 2010 has been added to the *qpidd-winsdk* package.
- ▶ Support for the IPv6 protocol has been added to the **qpidd** C++ libraries for MRG Messaging.
- ▶ Support for message grouping with strict sequence consumption across multiple consumers has been added to the *qpidd-cpp* package.
- ▶ This update adds support for DTX (distributed transactions) in clusters to the *qpidd-cpp* package.
- ▶ The JMS client now sets the **TCP_NODELAY** property to **true** by default as it shows an improvement in many general cases. If there is a configuration error, such as an error in connection URL, this property will still be set to **true**.



Note

Note that for high-throughput scenarios, packet overhead is reduced and congestion collapse is prevented if **TCP_NODELAY** is turned off. For further details, refer to the section [3.2.2. Connection URLs](#) in the [programming guide](#).

- ▶ The **SSL** module of the *qpidd-cpp* package has been improved to support a single port for both SSL and non-SSL traffic.
- ▶ The value for the **JMSDeLiveryMode** message header field can now be used as a string in the selector.
- ▶ The QMF Broker method **query** now returns extra detail for a message group queue's internal state.
- ▶ With this release, a JCA-compliant (Java EE Connector Architecture) resource adapter has been provided for JEE integration.
- ▶ In order to upgrade to MRG Messaging 2.1 on Red Hat Enterprise Linux 6.2, the **Red Hat Enterprise Linux Server EUS** channel needs to be enabled via the Red Hat Network. Some components of MRG Messaging 2.1 depend on Red Hat Enterprise Linux updates delivered through this channel.
- ▶ As of MRG Messaging 2.1, the QMF version 1 interfaces are deprecated. Users are advised to use the QMF version 2 interfaces instead. The version 1 interfaces will be removed in a future release.

Changes introduced to Red Hat Enterprise MRG on January 23, 2012:

- ▶ Sesame has moved from the **MRG Messaging** yum group to the **MRG Management** group.

2.3.1. Known Issues

The following known issues were reported for Red Hat Enterprise MRG as of April 30, 2012:
qpidd-cpp, BZ#[817283](#)

Connection options which are used to enable SSL fail to properly enable SSL. To work around

this issue, create SSL connections using the following syntax in your connection URL:

```
amqp:ssl:example.net:5671
```

qp_{id}-qmf, BZ#[786962](#)

The **PLAIN** authentication does not work for method calls if the *python-saslwrapper* package is not installed. The wrapper should not be required for this authentication to work. To work around this problem, a dependency for *python-saslwrapper* has been added to Red Hat Enterprise MRG. In older builds of Red Hat Enterprise MRG, which do not have this dependency, this issue can be resolved by running the **yum install python-saslwrapper** command.

qp_{id}-qmf, BZ#[798521](#)

If the *python-saslwrapper* package is installed and the default broker supports the **ANONYMOUS** and **PLAIN** authentication types, Cumin authenticates to the broker as **anonymous**. Consequently, method calls on the scheduler object fail. To work around this problem, the **sasl-mech-list** parameter in the **cumin.conf** file must be set to exclude **anonymous** users during authentication (for example, **sasl-mech-list: PLAIN**).

2.4. Messaging 2.0 Release

The 2.0 release of MRG Messaging contains several new features and enhancements, including:

- ▶ Thresholds in queues are now available to alert the user about an elongated queue.
- ▶ Message priority is now considered when messages are delivered.
- ▶ A delay is now added between the time a queue is unattached to any session and when it is automatically deleted.
- ▶ LVQ is enhanced with improved handling of incoming messages.
- ▶ Flow control is added to measure the amount of data in each queue.
- ▶ Statistics are enhanced to track the number of messages transferred instead of just the number of frames and bytes transferred over the connection.
- ▶ *AMPQ* now allows inspection of exclusive queues that were previously inaccessible.
- ▶ The level of details being logged can now be altered at runtime and applied without requiring a restart.
- ▶ The Python API now includes a **tcp_nodelay** option.
- ▶ Flow control mechanism is now on by default.
- ▶ Improved handling of the exception code **530**.
- ▶ Improved handling of erroneous cancellations of messages is handled with a **404** error.
- ▶ Invalid arguments now result in a rejected **queue-declare** where previously they were ignored.
- ▶ **QMFv2** event broadcast is now enabled by default. **QMFv1** is also enabled by default and is now independently toggled (previously either **QMFv1** or **QMFv2** had to be selected).
- ▶ The C++ client now recognizes connection names that are the same as Python. Old names are also still supported and an exception occurs if an unrecognizable option is encountered.
- ▶ Performance enhancements are added for synchronous transfers (the default for C++ and Python clients) with durable messages. The transfer call is blocked until the sent message is saved to the store. For messages less than **4** Kilobytes in size, this process can remain pending until the one-

second flush timer expiry.

Installation information for MRG Messaging is available in the *MRG Messaging Installation Guide*. For use and configuration details, see the *MRG Messaging User Guide*. For information on developing your own programs for MRG Messaging, start with *Programming in Apache Qpid*.

Red Hat Enterprise MRG documentation is available for download at the [Red Hat Documentation Website](#).

2.4.1. Update Notes

The following is a list of issues to be aware of when upgrading to the 2.0 release of MRG Messaging. This is not a complete list of bugs fixed in this release. For a complete list, please refer to the *Red Hat Enterprise MRG Technical Notes* for this release.

Table 2.1. MRG Messaging Update Notes

Description	Workaround	Reference
<p>Customers who are using the qpId-cpp-server-xml module must be subscribed to the following Red Hat Network channel in order to satisfy the XML dependencies for this release: RHEL Server Optional.</p> <p>Customers who are using the qpId-cpp-server-cluster module must be subscribed to the RHEL High Availability channel in order to satisfy all dependencies.</p> <p>If you encounter yum dependency problems, ensure that you are subscribed to all of the required channels as detailed above.</p>	No action required.	BZ# 720731
<p>Previously, using a java.util.UUID value as a Map message value or as part of a Map or List used within a Map message value displayed an exception. Additionally, the UUID from an incoming map message could not be read. A UUID is now correctly handled from an incoming Map message and the JMS client now allows a UUID to be set as a Map message value or as a Map or List within a Map message value.</p>	No action required	BZ# 632396
<p>Red Hat Enterprise MRG now includes the tcp_nodelay option for the Python API.</p>	No action required.	BZ# 667463
<p>Previously, the messaging-broker connected successfully from a client to a broker using SASL and SSL as the external SASL mechanism, but a similar connection between two brokers of a federation was not possible. Changes have been made to <i>qpId-config</i> and the</p>	No action required.	BZ# 500430

location of **SASL**-related code to allow one federation broker to act as an **SASL** server while the other acts as an **SASL** client. Federated links can now be connected with **SASL**, with the external mechanism of **SSL**. A test demonstrating this new connectivity is available at `cpp/src/tests/sasl_fed_ex`.

<p>A flow control mechanism is now added, allowing the broker to measure the current level of data in each queue using <i>high_watermark</i> and <i>low_watermark</i> flags. This flow control mechanism allows credit to be used to prevent a queue overflow event and provide information to a client about data levels in a queue.</p>	No action necessary.	BZ# 660291
<p>AMQP now allows the user to inspect exclusive queues that previously could not be browsed.</p>	No action required.	BZ# 624793
<p>Previously, queues marked for automatic deletion were deleted immediately after being released from a session and , in the case of failover, queues were permanently lost. A delay has now been introduced between the time a queue is eligible for automatic deletion and the time it is actually deleted. Additionally, if this delay period is longer than the failover time, then the queue survives the failover and then, if it is not required, is automatically deleted.</p>	No action required.	BZ# 585844
<p>Previously, the Messaging Broker did not consider signaled message priority during message delivery. The Messaging Broker can now be configured to recognize higher priority messages and adjust delivery accordingly.</p>	No action required	BZ# 453538
<p>Previously, applications were forced to use older APIs and</p>	No action required.	BZ# 547743

workarounds to dynamically create and delete broker entities as the messaging API was unable to perform these actions on entities such as queues, exchanges and their bindings.

QMF can now deal with the creation and deletion of broker entities and **QMFv2** can perform the same actions by sending a message of a defined format to a specified address.

When the Messaging component runs out of space, it must remove older messages to make space for new incoming messages. Messages to be deleted first are selected using an algorithm that assesses both the priority and the age of the message. This algorithm allows the oldest of the low priority messages to be considered expendable while the high priority messages are preserved.

No action required.

BZ#[606357](#)

Previously, browsing an LVQ prevented the messages browsed from being replaced, resulting in a queue that continues to grow as updates are added. This is now altered so that message equivalence is determined by the ***gpId.last_value_queue_key*** parameter, ensuring that LVQ receives the latest updates and new messages correctly replace their older versions.

No action required.

BZ#[632000](#)

Previously, the level of logging being done could not be altered at runtime without restarting the broker. A management method is now used to allow the user to change the level of logging while the program run without requiring a restart. This allows users to get detailed logs during troubleshooting and return to normal logging settings to prevent excessive logs.

No action required.

BZ#[657398](#)

Previously, messaging was

No action required.

BZ#[660289](#)

unable to monitor the growing queue depth without polling constantly or waiting until the maximum level was reached to issue a warning. The broker now allows **QMF** events to be generated when the queue depth reaches a previously configured threshold, providing an early warning for elongated messaging queues.

A new feature is added that tracks statistics about the number of messages transferred over the connection instead of tracking only the numbers of frames and bytes transferred across the connection.

No action required.

BZ#[667970](#)

Previously, the **RDMA** protocol transport for *Qpid* supported only Infiniband network interfaces. As a consequence of using *Qpid* **RDMA** with an iWarp network interface, the client process was unable to transmit more than 30-40 messages on a single connection due to lost flow control messages. *Qpid*'s use of **RDMA** now has changed to support iWarp network interfaces.

Current users of RDMA must upgrade any brokers before upgrading their clients if the upgrade is staged. This upgrade order is necessary as new brokers can detect both old and new protocols and switch automatically, but new clients will only use the new protocol.

BZ#[484691](#)

When there are multiple subscriptions on an **AMQP** queue, messages are distributed among subscribers. For example, two queue routes being fed from the same queue will each receive a load-balanced number of messages. If fanout behavior is required instead of load-balancing, use an exchange route.

No action required.

BZ#[656226](#)

A journal that is in use can quit after it displays a fatal error if its message store is full.

A full journal can be resized using a utility that can store and transfer all active records to a new, larger journal. This process will only occur when the broker is not running. The resize utility is located in `/usr/libexec/qpid/` and to use it the Python path must

BZ#[617488](#)

include this directory.

The **resize** command resizes the message store then transfers all outstanding records from the old message store to the new one. An error message displays if the records are unable to fit in the file, but the old store remains preserved in a subdirectory. The **store_chk** command analyzes a store, and shows the outstanding records and transactions.

Chapter 3. MRG Realtime

As MRG Realtime provides an updated Linux kernel, it is certified for use on a subset of the hardware systems certified for Red Hat Enterprise Linux. MRG Realtime is certified on x86_64 architectures only. Red Hat works with hardware vendors to certify systems for use with MRG Realtime based on customer demand. For an updated list of certified systems, see the [Red Hat Hardware Catalog](#).

MRG Realtime is not supported for use with any virtualization technology.

The MRG Realtime kernel may be rebased over the lifetime of a Red Hat Enterprise MRG release, however there are no guarantees of a stable kernel Application Binary Interface (kABI) over the life of Red Hat Enterprise MRG.

Installation information for MRG Realtime is available in the *MRG Realtime Installation Guide*. For information on tuning MRG Realtime, see the *MRG Realtime Tuning Guide* and the *MRG Realtime Reference Guide*.

Red Hat Enterprise MRG documentation is available for download at the [Red Hat Documentation Website](#).

3.1. Realtime 2.3 Release

These updated packages for Red Hat Enterprise Linux provide numerous enhancements and bug fixes for the Realtime component of MRG. Some of the most important updates include:

- ▶ MRG Realtime is now rebased to latest stable Realtime kernel (3.6), so it can take advantage of latest Realtime design changes.
- ▶ The MRG Hewlett Packard Smart Array (HPSA) driver has been rebased to the latest upstream version (3.7) and updated with selected Red Hat Enterprise Linux 6 fixes and enhancements. The current MRG HPSA driver version is 2.0.2-4-RH1+MRG-RT.
- ▶ Tuna now has the ability to change individual thread policy attributes, when before it was only possible to change attributes at the thread-group level. Users can now modify the priority or affinity of individual threads.
- ▶ Realtime scheduler throttling is now enabled, so the default behavior is to allow **SCHED_FIFO** and **SCHED_RR** threads 950000 microseconds of continuous runtime, then allow 50000 microseconds for **SCHED_OTHER** threads.
- ▶ The MRG Realtime kernel, which is newer than the Red Hat Enterprise Linux 6 kernel, has a "filesystem error nag" feature that gets cleared by newer user-space tools so that it is only printed once. This feature is unknown to Red Hat Enterprise Linux 6 filesystem or logging tools, so it never gets cleared. Consequently, the filesystem status is sent to the console and system log each day. The kernel function **print_daily_error_info()** has been patched, and now returns without printing daily error message.
- ▶ This update removes the **scripts/conmakehash** and **scripts/pnmtologo** executable files from the MRG Realtime * **-devel** packages. These files were not changed between kernel variants and generated the same hash value for all variants, which had previously caused debuginfo conflicts.
- ▶ An upstream patch which introduced the **rt_mutex** construct had exported previous inline functions as **EXPORT_SYMBOL_GPL**, which prevented third-party kernel modules from building. The **EXPORT_SYMBOL_GPL** functions are now changed to **EXPORT_SYMBOL**, so third party modules can build successfully.

3.1.1. Technology Preview



Important

Technology Preview features are not currently supported under Red Hat Enterprise Linux subscription services, may not be functionally complete, and are generally not suitable for production use. However, these features are included as a customer convenience and to provide the technologies with wider exposure. Customers may find these features useful in non-production environments, and can provide feedback and functionality suggestions prior to their transition to fully supported status.

PTP Kernel Support, BZ#[866600](#)

Support has been added for IEEE 1588 Precision Time Protocol (PTP) to the MRG 2.3 kernel. PTP provides greater time synchronization than is offered by Network Time Protocol (NTP).

3.2. Realtime 2.2 Release

These updated packages for Red Hat Enterprise Linux provide numerous enhancements and bug fixes for the Realtime component of MRG. Some of the most important enhancements include:

- ▶ The MRG Realtime kernel is now able to work as a diskless client.
- ▶ With this update, MRG Realtime kernel provides the *mrg-rt-release* package, which maintains the `/etc/mrg-realtime-release` file. This file contains the MRG release string, which indicates the major, minor, and errata releases of the MRG Realtime kernel.

3.2.1. Known Issues

kernel

Vsyscalls allow certain calls to work without changing from user-mode to kernel-mode. Kernel maps memory to user space with read-only data. This memory also includes some native code that emulates the system call. Since this native code was located at a fixed address, it could theoretically be used in security exploits.

This update emulates the vsyscalls and removes exploitable instructions from the vsyscall page, thus providing additional security. Vsyscalls are now emulated by being trapped in the kernel. This emulation occurs without breaking any APIs but can be potentially slower than the old native code. This kernel emulation of vsyscalls supports three modes described below that can be activated using the `vsyscall=` kernel parameter:

- ▶ `vsyscall=emulate` – as of this update, this is the new default option.
- ▶ `vsyscall=native` – use this parameter to have vsyscalls operate as they did prior to this update.
- ▶ `vsyscall=none` – this option provides the most security but can break existing binaries and critical libraries such as `glibc`, thus it is not recommended.

3.3. Realtime 2.1 Release

These updated packages for Red Hat Enterprise Linux provide numerous enhancements and bug fixes for the Realtime component of MRG. Some of the most important enhancements include:

- ▶ The *kernel-rt* package has been upgraded to upstream version 3.0, which provides a number of bug fixes and enhancements over the previous version.
- ▶ Some applications use flawed versioning logic that cannot recognize new Linux kernel versions in the format of **3.x.y**. A 2.6 personality patch has been added to the MRG Realtime kernel, which changes the data returned by the **uname(2)** system call. This update also adds the **uname26** executable needed by the personality patch.
- ▶ Prior to this update, the *kernel* and *kernel-rt* packages delivered the same set of kernel manual pages. Consequently, file conflicts occurred when both *kernel-doc* and *kernel-rt-doc* were being installed. This update adds the **rt** suffix to the files with *kernel-rt-doc* manual pages and the file conflicts no longer occur.
- ▶ Certain kernel static data areas and kernel modules have writable or executable memory areas. Prior to this update, malicious software could overwrite the data and potentially execute code in these areas. With this update, the RO (Read-Only) and NX (No eXecute) bits have been added to the memory areas to prevent such actions.
- ▶ The **recvmsg()** and **sendmsg()** system calls were missing from the code and were previously unavailable. This update restores the code with the system calls.
- ▶ The **%pK** printk format specifier was not added when printing the data from the **/proc/kallsyms** and **/proc/modules** interfaces. This could cause kernel address leaks. With this update, **%pK** is properly used when returning data from the interfaces.
- ▶ Previously, the **tuna** utility regularly polled **/proc/[PID]** directories to gather information about processes and threads, which could lead to unnecessary workload. To avoid this, the **tuna** utility has been enhanced to use the **perf** events infrastructure.

3.3.1. Known Issues

kernel

Build-ID hash conflicts exist between identical files in Red Hat Enterprise Linux 6 kernel tree and the Red Hat Enterprise MRG Realtime kernel tree. To avoid conflict messages and failed package installed, please remove all **kernel-* -debuginfo** files before upgrading to Red Hat Enterprise MRG 2.1 .

3.4. Realtime 2.0 Release

3.4.1. Update Notes

The following is a list of issues to be aware of when upgrading to the 2.0 release of MRG Realtime. This is not a complete list of bugs fixed in this release. For a complete list, please refer to the *Red Hat Enterprise MRG Technical Notes* for this release.

Table 3.1. MRG Realtime Update Notes

Description	Workaround	Reference
<p>The MRG Realtime Kernel uses one PCI bridge by default, despite modern hardware being commonly equipped with multiple PCI host bridges. Due to this default setting, some devices may be inaccessible to the kernel.</p> <p>The below line, if present in the kernel boot output (located at <code>/var/log/dmesg</code>), indicates an inconsistency in the number of PCI host bridges in the hardware and the MRG Realtime kernel:</p> <pre>pci_root PNP0A03:00: ignoring host bridge windows from ACPI; boot with "pci=use_crs" to use them</pre>	<p>If the aforementioned kernel boot output displays, edit the corresponding boot entry at <code>/etc/grub.conf</code> and add the following text to the kernel line:</p> <pre>pci=use_crs</pre> <p>This edit enables the use of the ACPI property Current Resource Settings to enumerate all available Host Bridges.</p>	None
<p>The system display adapter may not function correctly if an ATI Radeon display adapter is in use.</p>	<p>Add the below boot parameter string to the grub entry for the kernel:</p> <pre>radeon.hw_i2c=0</pre>	None
<p>Due to a path conflict, the <code>kernel-rt-doc</code> package conflicts with the <code>kernel-doc</code> package in Red Hat Enterprise Linux 6.</p>	<p>To prevent this conflict, only one of the two packages can be installed at a time.</p>	None

Chapter 4. MRG Grid

MRG Grid provides high-throughput computing and enables enterprises to achieve higher peak computing capacity as well as improved infrastructure utilization by leveraging their existing technology to build high performance grids. MRG Grid provides a job-queueing mechanism, scheduling policy, and a priority scheme, as well as resource monitoring and resource management. Users submit their jobs to MRG Grid, where they are placed into a queue. MRG Grid then chooses when and where to run the jobs based upon a policy, carefully monitors their progress, and ultimately informs the user upon completion.

4.1. Grid 2.3 Release

These updated packages for Red Hat Enterprise Linux provide numerous enhancements and bug fixes for the Grid component of MRG. Some of the most important updates include:

- ▶ This update introduces a discovery mechanism for Aviary endpoints through the Aviary locator service. Cumin is now fully integrated with the Aviary web servers, and is enabled by default. If this feature is disabled, Cumin will use QMF methods instead. This feature was not available in the Technology Preview in the previous release, where Cumin used QMF method for remote grid operations by default.
- ▶ This update improves custom kill signal use. As a result of a rebase to HTCondor 7.8, custom kill signals are no longer used during a fast shutdown. If a job is to use custom kill signals, it will be gracefully removed. Additionally, some instances where custom kill signals were sent more than once have been removed.

4.1.1. Technology Preview



Important

Technology Preview features are not currently supported under Red Hat Enterprise Linux subscription services, may not be functionally complete, and are generally not suitable for production use. However, these features are included as a customer convenience and to provide the technologies with wider exposure. Customers may find these features useful in non-production environments, and can provide feedback and functionality suggestions prior to their transition to fully supported status.

Reporting capabilities with Plumage data, BZ#[807838](#)

This feature allows Cumin to use data from the condor-plumage (ODS) database to generate long-duration visualizations of grid system behavior. This feature is intended to be fully supported in an upcoming release.

In order for `cumin-report` to be able to access the condor-plumage ODS, the Grid installation must both include and enable the condor-plumage package. The only configuration necessary to enable CuminReporting is in `/etc/cumin/cumin.conf`. This can be achieved by removing the comment from the `# reports: report` line in the following example.

```
# Reporting is off by default.  
# To enable reporting features, uncomment the following line.  
# reports: report
```

Cumin expires data samples in the database after 90 days by default. In order to allow the reporting feature to store data for a longer period of time, the following can be added to the `[report]` section of `cumin.conf`.


```
expire-threshold: 1y #for 1 year of data retention, 30d would give 30
days, and so on.
```

Data is pulled from the ODS (mongoDB) by *cumin-report* as a background operation. A full data load may take a considerable length of time due to a potential high count of records that can be in the many millions. Any data that has been loaded will immediately show up in the charts in the cumin UI. There are threads that maintain an archive load (starting with current records and moving back in time) and another thread that loads current records every 5 minutes.

The CuminReporting feature has a dependency on Red Hat Enterprise Linux 6 or newer, and PyMongo version 1.9-8 or newer. To install PyMongo you can run the following:

```
# yum install pymongo
```

Content similar to this Release Note may be found in the file `/usr/share/doc/cumin-*/REPORTING-README` after the software is installed. However, the Release Note should be considered more up to date and where there are any discrepancies the Release Note supersedes the readme file.

4.1.2. Known Issues

MRG Management Console BZ#[911142](#)

The MRG Management Console, also known as Cumin, in MRG 2.3 is not fully backwards compatible with the *condor-aviary* package from MRG 2.2, which is based on *condor-aviary-7.6.5-0.22*.

Consequently, in a mixed environment where the MRG 2.3 Cumin application is installed along with MRG 2.2 aviary components, the environment on the Cumin host determines whether Cumin can communicate with MRG 2.2 or MRG 2.3 aviary servers.

- ▶ If Cumin is installed on a host where there is no *condor-aviary* package installed, it will be able to communicate exclusively with MRG 2.3 aviary servers.
- ▶ If Cumin is installed on a host where the *condor-aviary* package from MRG 2.2 or MRG 2.3 is installed, it will be able to communicate exclusively with MRG 2.2 or MRG 2.3 aviary servers respectively.

ViewServer needs directory created by Plumage BZ#[896582](#)

Configuring a ViewServer feature through Remote Configuration without also configuring Plumage will prevent the ViewServer from starting. To workaroud this issue, install the **condor-plumage** package, create the directory `/var/lib/condor/ViewHist`. Alternatively set **POOL_HISTORY_DIR** to the appropriate directory. The directory should be owned by condor.condor and be writeable by the condor user. Once one of the workarounds is applied, the ViewServer will start.

Small hard memory limit support BZ#[892742](#)

A known issue with cgroups exists where small hard memory limits are not supported. When the hard memory limit is set below 200 MB, the job will be OOM killed when it breaches the limit. The expected behavior is the total amount of physical memory used by the sum of all processes in this job will not be allowed to exceed the limit.

If the processes try to allocate more memory, the allocation will succeed, and virtual memory will

be allocated, but no additional physical memory will be allocated. The system will keep the amount of physical memory constant by swapping some page from that job out of memory.

Virtual Machine universe jobs on Red Hat Enterprise Linux 5 and Xen BZ#[877428](#)

A known issue exists that affects the launch of virtual machine images in HTCondor's virtual machine universe on a 32-bit Red Hat Enterprise Linux machine using the Xen hypervisor. In this instance, a job will fail to run where the virtual machine image which is greater than 2 GiB.

Recommended workarounds include the use of the 64-bit version of Red Hat Enterprise Linux 5, the creation and use of disk images smaller than 2 GiB, the use KVM instead of Xen, or the use of Red Hat Enterprise Linux 6. Using these workarounds will allow virtual machine jobs of any size to be run.

Multiple entity actions with `condor_configure_store` BZ#[912357](#)

A known issue affects actions upon multiple entities with `condor_configure_store`. It can occur where at least one of the entity names would produce an error because the entity does or does not exist as required by the action. When this occurs, the `condor_configure_store` tool will print an error and not act on any of the entities.

A recommended workaround is to re-run the `condor_configure_store` command with a list of entities that satisfy the requirements of the action.

Activation and saving with `condor_configure_pool` BZ#[912391](#)

A known issue exists when using using `condor_configure_pool` to apply changes to pool configuration in the configuration store. In this case, the `condor_configure_pool` tool will create a snapshot if attempts are made to activate the changes, even if the changes are not valid.

In previous versions of `condor_configure_pool` a snapshot would be created only if the configuration is successfully activated. In this release the tool will create a snapshot every time a configuration is attempted to be activated.

4.2. Grid 2.2 Release

These updated packages for Red Hat Enterprise Linux provide numerous enhancements and bug fixes for the Grid component of MRG. Some of the most important enhancements include:

- ▶ This update adds Deltacloud support to the Condor's Grid Universe so that users can now submit their jobs to Condor, which run against the Deltacloud API.
- ▶ With this update, the ClassAd log file system has been enhanced to provide record numbers that cause parse errors. This allows parse errors to be easily located in ClassAd transaction log files.
- ▶ With this update, the `startd` daemon has been enhanced to allow additional local machine resources of arbitrary nature to be specified and managed.
- ▶ This update adds named groups for scoping multiple default concurrency limits based on a limit name prefix to the Negotiator. This allows Concurrency limits to be defined with multiple possible default values without invoking frequent Negotiator re-configurations.
- ▶ This update introduces a new set of tools that enables management of multiple HA Scheduler configurations on a node by Red Hat High Availability.
- ▶ This update adds new attributes, `TotalSlotCpus`, `TotalSlotMemory`, and `TotalSlotDisk` to

the slot ClassAd. The CPU accounting now works better on configurations with multiple partitionable slots on a single machine and users can now run simple queries to determine partitionable slot utilization.

4.2.1. Known Issues

Aviary

The Axis2/C engine retains a stream buffer for responses. It does not reallocate buffers based on the size of each new response but instead just checks to see if the old buffer fits the new data to be written. If not, a new buffer is allocated and the old one freed. Thus, over time, the buffer remains at a high-water mark of the last largest response size. In the case of the **aviary_query_server** utility and depending on the operation invoked, the resident size of the process in the memory (RES) grows because of the heap still in use.

For example, a request for one job status after a request for 500 job details will not grow the RES of the process; other non-RPC memory allocations notwithstanding. The user can simply restart the **aviary_query_server** daemon if they are concerned about the apparent RES size. The **aviary_query_server** daemon then reconstructs its internal job collections and the Axis2/C response buffer is then reset.

cyrus-sasl, BZ#[761041](#)

Adding a new user to the SASL database using the **saslpasswd2 -c username** command causes the following error messages to be written to the **/var/log/messages** log file:

```
error deleting entry from sasldb
```

These error messages are misleading and should not be returned as the command itself finishes its operation successfully and the user exists.

grid-condor-vm-gahp, BZ#[839961](#)

When multiple VM universe jobs refer to the same VM image, they are not prevented from using the image at the same time in KVM. Multiple jobs, represented by **qemu-kvm** processes, accessing the same VM image can cause data corruption. To work around this bug, do not submit multiple jobs referring to the same VM image.

The following known issue was reported for Red Hat Enterprise MRG as of October 16, 2012:

Wallaby

When running the new version of the Wallaby configuration service (0.16.0-9), Wallaby sometimes fails to operate properly and returns error messages due to invalid or missing object schemas. To work around this problem, restart your Qpid broker to remove cached object schemas, which were used with the previous version of Wallaby.



Note

Note that even if you experience no problems with Wallaby, it is always advisable to restart the Qpid broker before running a new version of Wallaby to prevent possible conflicts with incompatible or missing entries in object schemas.

4.3. Grid 2.1 Release

These updated packages for Red Hat Enterprise Linux provide numerous enhancements and bug fixes for the Grid component of MRG. Some of the most important enhancements include:

Changes introduced to Red Hat Enterprise MRG on February 6, 2012:

- ▶ Addition of **-sort** option to **condor_status**
- ▶ Customized output from **condor_q -run** for EC2 jobs
- ▶ Enhanced the summary line provided by **condor_q**
- ▶ Improved Collector performance around blocking network calls
- ▶ Fixed a memory leak associated with *python-psycopg2* hit by **cumin-data**

Changes introduced to Red Hat Enterprise MRG on January 23, 2012:

- ▶ Inventory – integration of configuration management (wallaby) inventory into the management console (cumin)
- ▶ Node grouping and feature application – integration of configuration management tagging/feature application into the management console
- ▶ Performance and utilization – speedup in configuration management for saving and activating configuration; increased utilization through improved re-use of scheduler resources, signal escalation management and claim re-use; elimination of expensive disk operations for parallel jobs
- ▶ Suspend and continue for jobs – ability to stop in place and later resume jobs
- ▶ SSL security mechanism for the Aviary API
- ▶ Simplified configuration management database upgrades, via a database patching tool
- ▶ Availability of Cluster Suite managed HA of the scheduler
- ▶ Configuration management command-line tools from the wallaby shell, as Technology Preview
- ▶ Operational Data Store command-line tools, as Technology Preview
- ▶ Cumin and Aviary integration, as Technology Preview
- ▶ The remote configuration database needs to be upgraded to take advantage of the latest features. To upgrade the database, make sure the *condor-wallaby-base-db* and *wallaby-utils* RPMs are updated to the latest package in the release and execute:

```
wallaby upgrade-db
```

The following release notes were issued for Red Hat Enterprise MRG on February 6, 2012:

Condor, BZ#[741716](#)

This update allows users to actually remove jobs marked for removal, even if a later **condor_hold** request comes in. The **-constraint** configurable option has been introduced to avoid the regular condition for held jobs.

Condor, BZ#[745348](#)

Previously, the **condor_status** only allowed output sorting by simple attribute names. Users could not sort their classads by generalized expressions. Construction of the internal sorting expression has been modified so that it could refer to a generalized classad expression. Users can now provide general classad expressions to the **-sort** argument to sort the classads.

4.3.1. Technology Previews



Technology Preview Policy

Technology Preview features are not currently supported under Red Hat Enterprise Linux subscription services, may not be functionally complete, and are generally not suitable for production use. However, these features are included as a customer convenience and to provide the technologies with wider exposure.

Customers may find these features useful in non-production environments, and can provide feedback and functionality suggestions prior to their transition to fully supported status. Erratas will be provided for high-priority security issues.

During its development additional components of a Technology Preview feature may become available to the public for testing. It is the intention of Red Hat to fully support Technology Preview features in a future release.

The following technology previews were introduced to Red Hat Enterprise MRG on January 23, 2012:

Grid NoSQL Plug-in Technology Preview, BZ#[733507](#)

Previously, existing statistic collection facilities (ViewCollector) used flat files that were only usable through the `condor_stats` tool. With this update, a new view collector plug-in has been developed to provide a new operational data store capability for Grid using a NoSQL database. The plug-in writes classad data (Machine, Submitter) to a mongodb NoSQL database. Grid machine and submitter statistics are now generally available to a variety of mongodb programming language drivers for C++, Python, Ruby, and other languages.

Grid ODS Query Tool Technology Preview, BZ#[733511](#)

Previously, existing statistic collection facilities (ViewCollector) used flat files that were only usable through the `condor_stats` tool. With this update, a new general query tool to access data generated from new Grid ODS facilities and examine statistics from the ODS database store has been developed in Python. Now, users have the means to start working with Grid statistics data in a dynamic language tool.

4.3.2. Known Issues

Condor, BZ#[728285](#)

When MRG Grid ran on a node with multiple network interfaces, it tried to estimate the correct interface for its communications with the remaining MRG Grid nodes. As a consequence, the node could have failed to communicate with other parts of MRG Grid correctly if the wrong interface had been chosen. As a workaround to this issue, MRG Grid can be forced to use a specific network interface by setting the **NETWORK_INTERFACE** parameter to the IP address of that interface. To determine which interface was used by MRG Grid when it fails to communicate with other parts of the grid, include the **D_HOSTNAME** variable in the logging configuration of the corresponding daemon.

Condor, BZ#[751834](#)

The **EC2 Enhanced** feature does not support specifying a region or provider for AWS (Amazon Web Services) AMIs (Amazon Machine Images). As a consequence, **EC2 Enhanced**

can only be used with the default availability zone.

Aviary, BZ#[733495](#)

The Aviary (SOAP-based) Grid API provides users the ability to query for submission information. Since a submission is a shared attribute value of a related group of individual jobs, it is possible that jobs with duplicate **cluster-proc** identifiers can hide submissions depending on the order that the jobs are processed within the API implementation. These duplicate **cluster-proc** identifiers can occur if the maximum cluster value has been reached (i.e., the **SCHEDD_CLUSTER_MAXIMUM_VALUE** variable has been set to an arbitrarily lower value than the default value of **INT_MAX**) or if the schedd job queue files have been deleted while history job files remain.

Condor, BZ#[759403](#)

When an OpenMPI (Message Passing Interface) job is submitted to the parallel universe environment, if SELinux is set to the **Enforcing** mode, the job fails when an attempt to generate SSH keys is made through the `/usr/libexec/condor/sshd.sh` utility.

4.4. Grid 2.0 Release

The 2.0 release of MRG Grid contains several new features and enhancements, including:

- ▶ Elastic binding for *Condor* EC2 jobs are now supported.
- ▶ A customizable Power Management feature is now added.
- ▶ *Condor* now imports validation for power management features to support IDLE machine hibernation.
- ▶ *Condor*'s wallaby database now include the AviaryPlugin, QueryServer and Axis2Home features.
- ▶ *Condor* is now able to appropriately handle SIGHUP.
- ▶ A simpler web service interface called **Aviary** is now included.
- ▶ The Aviary API now replaces the SOAP API.
- ▶ *Condor*'s **condor_dagman** now recognizes its schedd and caches its own *Condor* schedd address file.
- ▶ The administrator can now control the ads forwarded to a **condor_view_host**.
- ▶ Configurations are now validated for syntax to prevent crashes due to invalid configuration parameters.
- ▶ The PreJobPrio1, PreJobPrio2, PostJobPrio1, PostJobPrio2 job ad attributes are now included.
- ▶ The scheduler now gathers additional statistics for detailed statistics presentation.
- ▶ *Condor* now supports multiplexing among multiple view servers.
- ▶ In a group quote scenario, the negotiator now includes submitter names for enhanced monitoring of group quota limits.

MRG Grid includes the ability to schedule workloads to Amazon EC2. For Red Hat Enterprise MRG 2.0, Red Hat is transitioning the ability to purchase this capability at Amazon from Amazon's Dev Pay system to Red Hat's Cloud Access program. For more information refer to the Cloud Access website <http://www.redhat.com/solutions/cloud/access/>. Purchasing via Cloud Access will be enabled shortly after the Red Hat Enterprise MRG 2.0 release date.

Installation and configuration information for MRG Grid is available in the *MRG Grid Installation Guide*. For user and configuration details, see the *MRG Grid User Guide*.

Red Hat Enterprise MRG documentation is available for download at the [Red Hat Documentation Website](#).

4.4.1. Update Notes

The following is a list of issues to be aware of when upgrading to the 2.0 release of MRG Grid. This is not a complete list of bugs fixed in this release. For a complete list, please refer to the *Red Hat Enterprise MRG Technical Notes* for this release.

Table 4.1. MRG Grid Update Notes

Description	Workaround	Reference
Previously, running the wallaby feature-import command without supplying a file name resulted in an unclear error message. The wallaby feature-import command can now trap the error to display a clearer error message when run without a file name .	No action required.	BZ# 673502
Certain <i>wallaby</i> utility subcommands caused <i>wallaby</i> to exit with an unintuitive exit code. For example, 0 (zero) which indicates success, instead of a non-zero exit code (indicating failure). With this update, <i>wallaby</i> exits with more intuitive exit codes that reflect the success or failure of the underlying operation.	No action required.	BZ# 673520
Validation for <i>Condor</i> 's power management features are now imported to allow support for hibernation of IDLE machines.	No action required.	BZ# 674161
Previously, <i>Condor</i> EC2 jobs could not be bound to an elastic IP, forcing a dynamic IP to be created for each instance. Now the ec2_elastic_ip parameter supports elastic IP binding for <i>Condor</i> EC2 jobs.	No action required.	BZ# 621899
Previously, <i>Condor</i> did not support the EC2 RESTful API, that enables support for providers other than Amazon. <i>Condor</i> now provides validation and bug fixes for ec2_gahp and has replaced amazon_gahp , enabling all cloud providers support for the EC2 RESTful API.	No action required.	BZ# 679553
Red Hat Enterprise MRG 2.0 includes a Power Management feature, configurable manually via the <i>wallaby</i> component. Power Management is configured through the remote	No action required.	BZ# 678394

configuration feature.		
Condor's <i>wallaby</i> database now includes the AviaryPlugin , QueryServer and Axis2Home features with their respective parameters, as well as the subsystem <code>query_server</code> .	No action required.	BZ# 692801
Previously, when condor_configure_pool was used to add a feature containing the <i>must_change</i> parameter without a value, the condor_configure_pool did not prompt the user to supply the missing value. The condor_configure_pool now uses a new API call to detect <i>must_change</i> parameters and prompt the user if a <i>must_change</i> parameter value is missing.	No action required.	BZ# 627957
Previously, when a reconfigure signal from Red Hat Enterprise MRG Grid or SIGHUP was sent to condor_configd , condor_configd would unexpectedly fail and then quit. Condor_configd is now able to handle SIGHUP on Linux, UNIX, and similar operating systems and then exit gracefully.	No action required.	BZ# 680518
Previously, _triggerd 's C++ Console interface in <i>Condor</i> could not detect and report absent nodes as ENABLE_ABSENT_NODES_DETECTION was set to FALSE as a default. The ENABLE_ABSENT_NODES_DETECTION is now set to TRUE as the default in <i>Condor</i> , allowing _triggerd to raise an event for each node in <i>wallaby</i> without a corresponding master qmf object.	No action required.	BZ# 705325 and BZ# 602766
Red Hat Enterprise MRG Grid 2.0 offers a simpler web interface called Aviary , created with the use of Axis2/C and WSO2.	No action required.	BZ# 674349

<p>Previously, SOAP API was provided by the gSOAP library. This update replaces gSOAP with the WSO2-axis framework, which provides the Aviary API for SOAP.</p>	<p>No action required.</p>	<p>BZ#674384</p>
<p>The -schedd-daemon-ad-file and -schedd-address-file flags are now added to condor_submit_dag to allow targeting a dag to a specific Shedd and binding all its operations to the Schedd. This operation was possible previously using -remote, with an impact on the performance due to collector queries.</p>	<p>No action required.</p>	<p>BZ#584562</p>
<p>The newly added CONDOR_VIEW_CLASSAD_TYPES configuration parameter allows an administrator to control the ads that are forwarded to a CONDOR_VIEW_HOST. The CONDOR_VIEW_CLASSAD_TYPES parameter can be changed with a reconfiguration.</p>	<p>No action required.</p>	<p>BZ#610258</p>
<p>Previously, when runtime reconfiguration was enabled, an authorized user could cause a daemon to crash by providing a faulty configuration. This configuration was accepted because neither condor_config_val -set/rset or the daemon being reconfigured would validate the input. This is changed to ensure both condor_config_val -set/rset and the target daemon validate the configuration provided, preventing crashes during runtime.</p>	<p>No action required.</p>	<p>BZ#668038</p>
<p><i>Condor</i> now includes the PreJobPrio1, PreJobPrio2, PostJobPrio1, PostJobPrio2 job ad attributes, which allow jobs to be ordered outside the previously present JobPrio attribute.</p>	<p>No action required.</p>	<p>BZ#674659</p>
<p>Previously, the negotiator ran out of file descriptors and</p>	<p>The NEGOTIATOR.MAX_FILE_DESCR</p>	<p>BZ#603663</p>

crashed when assigned a large number of jobs.

IPTORS value can be user edited to a number larger than the expected number of jobs for the negotiation cycle. The recommended value for ***NEGOTIATOR.MAX_FILE_DESCR IPTORS*** is double the number of jobs per negotiation cycle.

The **scheduler** was updated to collect the following additional statistics:

No action required.

BZ#[678025](#)

- ▶ ***WindowedStatWidth*** : value of configuration parameter ***WINDOWED_STAT_WIDTH*** at the time the target ad was published.
- ▶ ***UpdateInterval*** : number of seconds between current schedd ad publish time and previous ad.
- ▶ ***JobsSubmitted*** : number of jobs submitted over the most recent sampling window.
- ▶ ***JobSubmissionRate*** : rate of job submissions (jobs/sec) over the sampling window.
- ▶ ***JobsStartedCum*** : number of jobs initiated over the schedd's lifetime.
- ▶ ***JobsStarted*** : number of jobs started in the stat window (***WINDOWED_STAT_WIDTH***).
- ▶ ***JobStartRate*** : rate (jobs/sec) of jobs starting in the stat window.
- ▶ ***JobsCompleted*** : number of jobs successfully completed in the sampling window.
- ▶ ***JobCompletionRate*** : rate of successful job completions in the sampling window.
- ▶ ***JobsExited*** : number of jobs that exited (successful or otherwise) over in the sampling window.
- ▶ ***ShadowExceptions*** : number of shadow exceptions in the sampling window.
- ▶ ***ExitCodeXXX*** : number of jobs exited with code ***XXX***

(100, 115, etc.) in the sampling window.

- ▶ **JobsSubmittedCum**: number of jobs submitted over the schedd's lifetime.
- ▶ **JobsCompletedCum**: number of jobs successfully completed over the schedd's lifetime.
- ▶ **JobsExitedCum**: number of jobs exited (successfully or otherwise) over the schedd's lifetime.
- ▶ **ShadowExceptionsCum**: number of shadow exceptions over the schedd's lifetime.
- ▶ **ExitCodeCumXXX** : number of jobs exited with code **XXX** over the schedd's lifetime.
- ▶ **MeanTimeToStartCum**: mean time a job waits in the schedd until first started, in the schedd's lifetime.
- ▶ **MeanRunningTimeCum**: mean running time for jobs in the schedd (wall-clock time), over the schedd's lifetime.
- ▶ **SumTimeToStartCum**: sum of job wait times to first start, over the schedd's lifetime (intended for consumption by software like **cumin**).
- ▶ **SumRunningTimeCum**: sum of job running times over the schedd's lifetime.
- ▶ **MeanTimeToStart**: mean time a job waits in schedd until first start, over stat window.
- ▶ **MeanRunningTime**: mean running (wall-clock) time of jobs in schedd, over stat window.

The follow are now published to all daemon ads:

- ▶ **DetectedMemory**: detected machine RAM.
- ▶ **DetectedCpus**: detected machine CPUs/cores.

<p>Previously, <i>Condor</i> allowed declaration of only a single view server, this did not allow multiplexing among multiple view servers. The <i>Condor collector</i> now supports a list of multiple view servers declared using <i>CONDOR_VIEW_HOST</i> for improved scalability.</p>	<p>No action required.</p>	<p>BZ#610251</p>
<p>Previously, the <i>LastNegotiationCycleSubmittersShareLimitNegotiator classad stat</i> attribute did not account for a submitter reaching the share limits in a <i>group-quote</i> scenario. The negotiator now includes submitter names in the attribute when any submitter reaches the submitter limit, including group quota limits.</p>	<p>No action required.</p>	<p>BZ#674669</p>
<p>Due to case folding of submitted names in <i>Accountant</i>, multiple submitter name entries would be created in <i>condor_userprio</i> (one entirely in lower case and one with the correct mix of upper and lower case characters) if the submitter name entry contained upper case letters. Explicit case folding is now removed from the <i>Accountant</i> and data maps are updated with a case-insensitive sorting function. As a result, submitter names with upper case letters no longer appear as multiple entries and <i>Accounting</i> group entries now match updated entries by case and full submitted entries are case sensitive.</p>	<p>No action required.</p>	<p>BZ#675703</p>
<p>The top-level package <i>condor-classads</i> replaces the deprecated package <i>classads</i> in <i>Condor</i> version 7.6.1-0.1 and later. The old and new packages share <i>debuginfo</i> packages, which requires the user to manually remove</p>	<p>If old and new packages share <i>debuginfo</i> packages, manually remove <i>classads-debuginfo</i> packages.</p>	<p>BZ#696324</p>

classads-debuginfo packages.

The remote configuration database requires an update as loading a database snapshot destroys pool configuration.

Previously, the **upgrade-wallaby-db** tool was required to upgrade an existing deployment's database. That is no longer necessary.

BZ#[688344](#)

Chapter 5. MRG Management Console

Installation and configuration information for the MRG Management Console is available in the *MRG Management Console Installation Guide*.

5.1. Management Console 2.2 Release

These updated packages for Red Hat Enterprise Linux provide several enhancements for the Management Console component of MRG. The most important enhancements are described below:

- ▶ With this update, the Cumin web server supports secure communication between a web browser and the web server using SSL.
- ▶ Now, Cumin has been enhanced to allow the use of LDAP servers for authentication. If a user is not found in the Cumin database, Cumin attempts to authenticate a user against a specified list of LDAP directories.

5.1.1. Known Issues

Cumin, BZ#[852378](#)

Beginning with the release of **cumin-0.1.5098-2**, a facility has been included to modify the database schema as needed when upgrading the *cumin* package. However, the upgrade mechanism only works correctly when upgrading from **cumin-0.1.4916-1** or newer.

When upgrading from a package older than **cumin-0.1.4916-1** to **cumin-0.1.5098-2** or newer, the schema upgrade mechanism should not be used. Instead, the database schema should be dropped and recreated.



Note

This issue also applies to past actions – if you previously upgraded from a package older than **cumin-0.1.4916-1** directly to **cumin-0.1.5098-2** or newer using the schema upgrade mechanism, the database schema may be incorrect.

The following procedure drops and recreates the schema while preserving user accounts. The sample data, which is used to populate graphs, will be lost but will be repopulated as the service runs. Although the **cumin-admin upgrade-schema** command is not being relied on in this procedure, the command itself must still be issued to ensure that subsequent commands work as expected:

```
# service cumin stop
# cumin-admin upgrade-schema
# cumin-admin export-users users
# cumin-admin drop-schema
# cumin-admin create-schema
# cumin-admin import-users users
# rm users
# service cumin start
```

cumin-admin

Previously, the **cumin-admin** application was run as **root** and file permissions on exported user files were not an issue. This update modifies **cumin-admin** to run commands as the **cumin** user. As a consequence, the user must ensure that the **cumin** user is able to write or

read files specified for the **export-users** or **import-users** commands respectively. For **export-users**, the directory specified in the path must allow the **cumin** user to create files. Alternatively, an existing file that is writable by the **cumin** user may be specified. For **import-users**, an exported-users file passed to the command must be readable by the **cumin** user.

Cumin, BZ#[850759](#)

Web forms left in a browser window after a session has expired are no longer valid and the Cumin server rejects them if submitted. This includes the login form itself. This may result in a CSRF (Cross Site Request Forgery) exception trace if an expired form is submitted. At this time, such an exception is expected behavior in the described scenario and should cause no concern. To work around this issue, simply navigate to the Cumin main page and log in again. To prevent the exception from occurring, log out and close the Cumin window when you finish your work in Cumin.

Cumin, BZ#[838619](#)

When a job is submitted from Cumin, the **description** value is used to look up submission objects in MRG Grid. If no submission object exists with that submission description, a new submission object is created that is owned by the submitting user. If a submission object already exists with that description, the new job becomes part of the existing submission. However, the existing submission object may be owned by a different user. Submissions in Cumin are sorted by description and filtered by owner in the **Grid User** view. The submission list under **Grid User ->Submissions** only shows submission objects owned by the logged-in user. If the user submits a job with a description that matches an existing submission object owned by another user, the user will not see their new submission in the **Grid User** view in Cumin.

To work around this problem, we advise to adopt a naming convention to help avoid submission description duplicates. For example, use the **<username>** suffix with each of your descriptions to create unique identifiers. Alternatively, the **Administrator** view in Cumin can be used since all submissions are visible in it, regardless of their owner. The latter workaround is only practical if role enforcement is disabled or the **admin** role is granted to a large number of users, effectively disabling the role enforcement.

Cumin, BZ#[848344](#)

The CuminAviary Technology Preview feature requires command arguments for jobs submitted in Cumin via Aviary. Submissions that specify commands without arguments are rejected by Cumin with an error. This applies to regular submissions and VM submissions. To work around this problem, expand the submission form using the **Show more** button and add the **Args =** string to the **Extra attributes** field. This satisfies the restriction in Cumin without defining any actual arguments and the job will run in Condor without any arguments being passed to the command.

5.2. Management Console 2.1 Release

These updated packages for Red Hat Enterprise Linux provide several enhancements for the Management Console component of MRG. The most important enhancement is described below:

Cumin integration with Wallaby

Cumin is now integrated with Wallaby, exposing Wallaby data in the Inventory tab as well as

Configuration tab under the Grid tab (Grid::Configuration). This list contains information that comes from both Sesame and Wallaby; entries in **Tags** and **Last check-in** columns come from Wallaby, entries in the other columns come from Sesame.

If you don't see any information in columns populated by Wallaby, it indicates that either, Wallaby is not running properly, or that the given node is not provisioned via Wallaby. Both situations should be handled in Cumin.

If you don't see any information in columns populated by Sesame (Kernel, Architecture, Free Memory, Load average), it indicates that either, Sesame is not running on that host, or is otherwise not working properly. In either case, when you click on the node name in the **host** column, the Overview tab will show the **there is no data** message. Likewise, the Details tab will also indicate the lack of data.

When clicking on a host name from the Inventory table, the new Configuration tab will open. That tab shows information about tags for that node. It is *not* currently possible to edit the tags data from this screen. Tags and features can be changed in the **Grid::Configuration** tab.

The **Grid::Configuration** tab lists tags that have been created so far. A tag can have zero or more features and/or hosts attached to them. New entries of this type can be attached to a tag by clicking the tag name in the table; a new page will open that lists the features and hosts currently associated with the tag. The user can then use the **Edit features** or **Edit hosts** links to add a new entry.

Some features, such as **NodeAccess**, have mandatory parameters that must be entered when activating a Wallaby configuration. With this release, it is *not* possible to enter the required parameter information for the activation via the Cumin user interface. Wallaby command-line tools need to be used instead. This missing feature might be a part of a future Cumin update.

5.2.1. Technology Previews



Technology Preview Policy

Technology Preview features are not currently supported under Red Hat Enterprise Linux subscription services, may not be functionally complete, and are generally not suitable for production use. However, these features are included as a customer convenience and to provide the technologies with wider exposure.

Customers may find these features useful in non-production environments, and can provide feedback and functionality suggestions prior to their transition to fully supported status. Erratas will be provided for high-priority security issues.

During its development additional components of a Technology Preview feature may become available to the public for testing. It is the intention of Red Hat to fully support Technology Preview features in a future release.

CuminAviary Technology Preview, BZ#[733677](#)

This feature allows Cumin to use the Aviary web services provided in the *condor-aviary* package for certain functions in the user interface. If the CuminAviary feature is enabled, Cumin will use Aviary services rather than QMF method calls where possible.

The CuminAviary feature is controlled through the cumin configuration file. Relevant configuration parameters with descriptive comments can be found in the default `/etc/cumin/cumin.conf` file by searching for a line containing **Aviary interface to**

condor.

Aviary provides a job service and a query service; Cumin may use either, both or neither. By default, Cumin will use QMF methods rather than Aviary services.

To enable use of the Aviary job service, the ***aviary-job-servers*** parameter must be uncommented and set (see the comments in the configuration file for further details). Setting this parameter will cause Cumin to use the Aviary job service for job submission, for the hold, release, and remove job control functions, and for editing of job and attributes.

To enable use of the Aviary query service, the ***aviary-query-servers*** parameter must be uncommented and set (see the comments in the configuration file for further details). Setting this parameter will cause Cumin to use the Aviary query service for retrieving job output files, retrieving job ad details, and retrieving the list of jobs in a submission.

Cumin will make **INFO** level entries in the log file for cumin-web that indicate whether the use of the job and/or query services has been enabled and what type of certificate validation will be used for servers configured for SSL (see below). These log entries will begin with **AviaryOperations:** or contain the string **Aviary** somewhere in the message. If an Aviary operation fails, the yellow task banner associated with the operation will contain error information.

By default, the Aviary services in Condor will not use SSL (Secure Socket Layer) for communication and no other configuration parameters need to be set for this feature. However, if the Aviary services in Condor have been configured to use SSL, then additional configuration parameters must be set.

First, note that the scheme for Aviary servers will change from **http** to **https** for any server using SSL. Failure to specify schemes correctly in the ***aviary-job-servers*** or ***aviary-query-servers*** parameters will prevent the CuminAviary feature from functioning. An incorrect server address may result in a default 90 second timeout when Cumin attempts to perform an operation using that server.

Second, the ***aviary-key*** and ***aviary-cert*** parameters must be set. These parameters give the full paths to a PEM formatted private key file and PEM formatted certificate file that Cumin will use as a client to access the Aviary services. The Aviary servers will validate Cumin's client certificate and allow access if validation succeeds.

Optionally, the ***aviary-root-cert*** parameter may be set. This is the full path to a PEM formatted file containing CA (certificate authority) certificates that Cumin will use to validate the server certificate. If this parameter is unset Cumin will NOT validate server certificates.

Here is a note relating to the ordering of certificate chains within a file from the OpenSSL documentation:

```
"SSL_CTX_use_certificate_chain_file() loads a certificate chain from file into ctx. The certificates must be in PEM format and must be sorted starting with the subject's certificate (actual client or server certificate), followed by intermediate CA certificates if applicable, and ending at the highest level (root) CA. There is no corresponding function working on a single SSL object."
```

Lastly, the ***aviary-domain-verify*** parameter controls whether or not Cumin checks the hostname of the server against the server certificate during validation. This parameter has no effect unless ***aviary-root-cert*** is set. The default value is **True**; it may be useful to set this

parameter to **False** if the server is using a self-signed certificate with a non-matching hostname.

Cumin will provide server certificate validation using the Python **ssl** standard language module if available or M2Crypto otherwise. If neither of these components are available, server certificate validation will be disabled.

Dependencies

The CuminAviary feature has a dependency on *python-suds-0.4.1* or newer. This package has been added as a dependency in the cumin RPM package.

Feedback

Bug reports or requests for enhancement can be made through <http://bugzilla.redhat.com>. General questions about this feature can be handled through the email list cumin-users@lists.fedorahosted.org

Full Support

This feature is intended to be fully supported in an upcoming minor release.

Where to Find This Information

Content similar to this Release Note may be found in the file `/usr/share/doc/cumin-*/AVIARY-README` after the software is installed. However, the Release Note should be considered more up to date and where there are any discrepancies the Release Note supersedes the **README** file.

5.2.2. Known Issues

The following known issues were reported for Red Hat Enterprise MRG as of February 6, 2012:

Cumin, BZ#[759200](#)

A bug in the *python-psycopg2-2.0.14* package caused a reference leak when the **cumin-data** utility updated objects in the database. Specifically, the bug manifested when the **cursor.mogrify(operations, params)** or **cursor.execute(operations, params)** functions were called and the **operations** string referenced the same value from the **params** list more than once. Consequently, long-running instances of **cumin** could leak significant amounts of memory. For Red Hat Enterprise Linux 5, the **_mogrify()** routine has been fixed and the reference leak no longer occurs when **cumin-data** updates database objects. However, for Red Hat Enterprise Linux 6, this bug could not be fixed the same way and will be resolved in a future update to Red Hat Enterprise Linux.

Cumin, BZ#[765894](#)

The default virtualization technology supported in **Cumin** is KVM. Submitting a VM job from **Cumin** using the Xen virtualization technology appears to succeed in **Cumin**, but the actual virtual machine fails to start.

The following known issues were reported for Red Hat Enterprise MRG as of January 23, 2012:

Cumin, BZ#[733365](#)

The node tagging functionality can be found under the **Grid::Configuration** tab. The table there contains all tags that are currently stored in the Wallaby configure store. New tags can be created via the **Create tags** link. Tags can be deleted by selecting them and clicking the **Delete tags** button located above the tags list table.

When performing any action with tagging (create, remove, apply to nodes, apply features), the status of the task will appear in the yellow invocation banner at the top of the Cumin user interface. Even after an operation gets the **OK** status, it may take a short time for the changes to be reflected in the display. This is due to the way Wallaby caches information. Each action triggers the cache to be reloaded and in most cases, the operation takes several seconds to finish. Delays that take considerably more time than that should be brought to the attention of the support team.

5.3. Management Console 2.0 Release

The 2.0 release of MRG Management Console contains several new features and enhancements, including:

- ▶ QMFv2 C++ library and the QMFv2 Ruby binding is added.
- ▶ QMFv2 Python is now available.
- ▶ A **Persona** feature allows customizable views.
- ▶ Version information can now be viewed in the User Interface.
- ▶ Qpid-tool's python clients can now select their authentication mechanism.
- ▶ The User Interface displays elements to assist in selecting and sorting columns.
- ▶ Tables that receive data from brokers now provide a search feature.
- ▶ The console now informs the user when an action requesting data from a broker is pending.
- ▶ Tables can be exported to a comma separated value format file.
- ▶ Statistics involving the overall health of the grid are now available.
- ▶ Slot icons are now simply grouped as either **Busy**, **Transitioning**, **Owner** or **Unclaimed**.

Red Hat Enterprise MRG documentation is available for download at the [Red Hat Documentation Website](#).

5.3.1. Update Notes

The following is a list of issues to be aware of when upgrading to the 2.0 release of the MRG Management Console. This is not a complete list of bugs fixed in this release. For a complete list, please refer to the *Red Hat Enterprise MRG Technical Notes* for this release.

Table 5.1. MRG Management Console Update Notes

Description	Workaround	Reference
The QMFv2 C++ library and the QMFv2 Ruby binding have been added to the qpidd-cpp component.	No action required.	BZ# 659093
QMFv2 Python has been added to the python-qmf component.	No action required.	BZ# 659095
Previously, users could not customize the content displayed in the GUI for a grid-only view or messaging-only view based on the deployment requirements. A persona feature is now added to the web section of the <i>cumin</i> configuration file, that allows the user to select grid-only or messaging-only views. The default view displays a mixture of both grid and messaging views.	No action required.	BZ# 678029
Previously, the value for Max Allowance in <i>Cumin</i> was incorrectly displayed as an integer despite being a float value and unlimited was assigned as the default value instead of assigned to a specific limit. The Max Allowance value is correctly displayed as a float value and the unlimited value is now only assigned to values over 1,000,000 .	No action required.	BZ# 635197
Previously, the version information for <i>Cumin</i> could not be viewed from the web user interface, requiring users to log into the server host and use rpm commands to view the installed package information. The <i>Cumin</i> UI now has an About the console tab under the Your Account page where version information stored in \$CUMIN_HOME/version displays.	No action required.	BZ# 630544
File conflicts as a result of package reorganization (the QMF code was moved from the	The workaround for this problem is to manually uninstall the previous version of the	BZ# 684182

qpidd-mrg set of packages to the *qpidd-qmf* set of packages) caused direct upgrades of *debuginfo* from version 1.3.2 to a later version to fail if debug symbols were automatically upgraded.

debuginfo package before an installation or upgrade of the newer messaging and qmf packages is done, such as:

```
$ rpm -ev qpidd-cpp-
mrg-debuginfo
```

This workaround does not introduce any limitations and is simple to execute for *debuginfo* packages users.

Qpidd-tool's python clients (such as **qpidd-config**, **qpidd-queue-stats**, **qpidd-route**, **qpidd-stat** and **qpidd-printevents**) are now able to select the mechanism used by them for authentication.

No action required.

BZ#[604149](#)

The MRG Management Console sorts data in ascending or descending order when the user clicks on the relevant column header. Previously in the GUI, there was no indication of the column used to sort rows, or if the sort order was ascending or descending. Now an arrow appears on the relevant column header, this indicates the sort order and a pop-up tooltip appears when the mouse cursor hovers over the column header describing how the column will be resorted if selected.

No action required.

BZ#[673178](#)

The database schema used by the MRG Management Console has changed in version 2.0.

It is necessary to rebuild the database after MRG Management Console is installed. This procedure will delete all database data except for user and password information. As the data in the console stores is largely dynamic, this should not present a problem. After the MRG Management Console is restarted it will gather information from the Qpidd messaging broker about systems in the MRG deployment and resume calculating statistics as part of its normal

BZ#[683975](#)

operation. The first step is to preserve the console user information. This procedure exports a list of users, roles, and encrypted passwords to a text file. After the database is rebuilt this file can be imported by **cumin-admin** to recreate the user data.

Become the root user. Stop the cumin service if it is running.

```
# /sbin/service cumin
stop
```

Export the user list to a file (here the file chosen is **users.bak**)

```
# cumin-admin export-
users users.bak
```

Remove the existing cumin database. Enter **yes** when prompted.

```
# cumin-database drop
```

Recreate the database

```
# cumin-database
create
```

Recreate the user list

```
# cumin-admin import-
users users.bak
```

Restart the cumin service

```
# /sbin/service cumin
start
```

Previously, the MRG Management Console displayed tables that receive data directly from the broker but were unable to search for the desired records. Tables that receive their data directly from the broker now have the ability to search for specific records.

No action required.

BZ#[673180](#)

Certain actions in the MRG Management Console, such as

No action required.

BZ#[673183](#)

displaying the job summary info and displaying the group quotas, get data directly from the **broker**, this process takes a few seconds to complete. While data retrieval occurs, no feedback is displayed to inform the user that the action is pending. The MRG Management Console now includes a mechanism to inform the user that an action is pending when data is being requested from the **broker**.

Cumin displays data in tables that display **100** records per page. When more than **100** records are present in a table, no easy method was available to save all the records to a file. *Cumin* now allows a user to save all records in a table to a comma separated value file.

No action required

BZ#[673187](#)

Previously, the Red Hat Enterprise MRG *Cumin* console presented a list of pools under the **Grid** tab. Generally, only one pool would display under the **Grid** tab, rendering the use of a dedicated page to display a list containing one entry unnecessary. The Red Hat Enterprise MRG *Cumin* console now does not display the list of pools and if more than one broker is listed in the **brokers=** line of the *Cumin* configuration file, the first broker is used as a default.

No action required

BZ#[673189](#)

An **overview** page was added to show the overall health of the grid and provide access to various grid statistics at a glance.

No action required

BZ#[673194](#)

The MRG Management Console now pulls data displayed for **Limits, Quotas** and **Job** summaries directly from the broker instead of the internal database. Tables that display these fields can now be exported whole into comma

No action required

BZ#[642405](#)

separated value format files.

The Red Hat Enterprise MRG *Cumin* console displays slot icons grouped by the slot's state and activity. The slot state is indicated by the icon color and the activity is indicated by the icon shape. To prevent confusion, slots are now displayed in four groups: **Busy**, **Transitioning**, **Owner**, and **Unclaimed**.

No action required.

BZ#[647500](#)

The *max_fsm_pages* parameter in `/var/lib/pgsql/data/postgresql.conf` affects PostgreSQL's ability to reclaim free space. Free space will be reclaimed when the MRG Management Console runs the **VACUUM** command on the database (the vacuum interval can be set in `/etc/cumin/cumin.conf`). The default value for *max_fsm_pages* is 20,000. In medium scale deployments, it is recommended that *max_fsm_pages* be set to at least 64K. PostgreSQL must be restarted in order for changes to *max_fsm_pages* to take effect. Cumin should be restarted when PostgreSQL is restarted.

Verify that *max_fsm_pages* is adequate using this procedure: Start an interactive PostgreSQL shell.

BZ#[699859](#)

```
$ psql -d cumin -U
cumin -h localhost
```

Run the following command from the PostgreSQL prompt. This will produce a large amount of output and may take quite a while to complete.

```
cumin=# VACUUM ANALYZE
VERBOSE;
```

Set *max_fsm_pages* to at least the indicated value in `/var/lib/pgsql/data/postgresql.conf`. Restart the PostgreSQL service and perform this process again, repeating until PostgreSQL indicates that free space tracking is adequate:

```
DETAIL: A total of
25712 page slots are in
use (including
overhead).
25712 page slots are
required to track all
free space.
Current limits are:
32000 page slots, 1000
relations, using 292 KB.
VACUUM
```

Depending on the configuration and number of *cumin* users, the default *max_connection*

The *max_connections* parameter in `/var/lib/pgsql/data/post`

BZ#[702482](#)

configuration value in the PostgreSQL database can be changed to accommodate a large number of users.

gresql.conf specifies the maximum number of concurrent connections allowed by the PostgreSQL server. This value is set to **100** by default.

The maximum number of concurrent connections needed by **cumin** can be estimated with the following formula:

$$(\text{cumin-web instances} * 36) + (\text{cumin-data instances}) + 2$$

For a default **cumin** configuration this number will be **43** but with multiple *cumin-web* instances, the number increases significantly. Ensure that the value of the **max_connections** parameter is set at a value that can accommodate the number of connections required by **cumin** and other applications that connect to the PostgreSQL server.

The text **OperationalError: FATAL: sorry, too many clients already** displayed in the UI, or contained in a Cumin log file, indicates that available database connections were exhausted and a Cumin operation failed.

The **args** attribute of a job can accommodate numeric values as a default. Non numeric values require special formatting in the attributes form.

To set the **args** attribute of a job to a non numeric value, encapsulate the value and non numeric characters within quotation marks.

For example, to change the **args** attribute from **60s** to **60m**, encapsulate **60m** within quotation marks ("**60m**"). If a non numeric value is provided without quotation marks, no error displays but the value remains unchanged.

[BZ#705819](#)

Chapter 6. Red Hat Enterprise MRG 2 Package Manifest

This chapter contains a complete list of packages released in Red Hat Enterprise MRG 2.0. With each subsequent release, a list of packages that have been added or removed from MRG is provided.

Core packages are provided with higher levels of support than non-core packages. To view details about the differences in support levels, refer to this [page](#).

6.1. Red Hat Enterprise MRG 2.1

The following packages have been added to Red Hat Enterprise MRG 2.1:

- ▶ MRG Grid
 - *python-wallaby* – used for Cumin integration with Wallaby
 - *condor-plumage* – the Grid ODS feature
 - *mongodb*, *pymongo*, *js* – used with *condor-plumage*
- ▶ MRG Realtime
 - *uname26* – kernel 2.6 personality patch

The following packages have been dropped from Red Hat Enterprise MRG 2.1:

- ▶ MRG Grid
 - *wso2-rampart*
 - *wso2-rampart-devel*

6.2. Red Hat Enterprise MRG 2.0

This section contains a complete list of packages released in Red Hat Enterprise MRG 2.0.

6.2.1. Messaging 2.0 Release

The following is a list of packages and comments for each package of MRG Messaging.

Table 6.1. Red Hat Enterprise MRG Messaging Package Manifest

Package Name	Core Package	Comments
mrg-release	Yes	
python-qpidd	Yes	
python-saslwrapper	No	
qpidd-cpp-server-ssl	Yes	
qpidd-cpp-server-store	Yes	
qpidd-cpp-server	Yes	
qpidd-tools	Yes	
saslwrapper	No	
qpidd-cpp-client-devel-docs	Yes	
qpidd-cpp-client-devel	Yes	
qpidd-cpp-client-ssl	Yes	
qpidd-cpp-client	Yes	
qpidd-cpp-server-cluster	Yes	
qpidd-cpp-server-devel	Yes	
qpidd-cpp-server-xml	Yes	
qpidd-java-client	Yes	
qpidd-java-common	Yes	
qpidd-java-example	Yes	
rhm-docs	Yes	
ruby-saslwrapper	No	
saslwrapper-devel	No	
sesame	Yes	
sesame-debuginfo	Yes	
xerces-c-devel	No	
xerces-c-doc	No	
xerces-c	No	
xqilla-devel	No	
xqilla	No	
qpidd-cpp-client-rdma	Yes	
qpidd-cpp-server-rdma	Yes	
qpidd-tests	Yes	
rh-qpidd-cpp-tests	No	
ruby-qpidd	No	

6.2.2. Realtime 2.0 Release

The following is a list of packages and comments for each package of MRG Realtime.

Table 6.2. Red Hat Enterprise MRG Realtime Package Manifest

Package Name	Core Package	Comments
kernel-rt	Yes	
mrg-realtime-docs	Yes	
rt-setup	Yes	
rtcheck	Yes	
rtctl	Yes	
tuna	Yes	
ibm-prtm	Yes	
kernel-rt-debug-devel	Yes	
kernel-rt-debug	Yes	
kernel-rt-devel	Yes	
kernel-rt-doc	Yes	
kernel-rt-firmware	Yes	
kernel-rt-trace-devel	Yes	
kernel-rt-trace	Yes	
kernel-rt-vanilla-devel	Yes	
kernel-rt-vanilla	Yes	
oscilloscope	No	Sub-package of tuna
python-numeric	No	Required by Tuna
python-linux-procfs	No	Required by tuna
python-schedutils	No	Required by tuna
rt-tests	Yes	
rteval-loads	Yes	
rteval	Yes	

6.2.3. Grid 2.0 Release

The following is a list of packages and comments for each package of MRG Grid.

Table 6.3. Red Hat Enterprise MRG Grid Package Manifest

Package Name	Core Package	Comments
condor	Yes	
condor-debuginfo	Yes	
condor-qmf	Yes	
mrg-release	Yes	
condor-aviary	Yes	
condor-classads	Yes	
condor-ec2-enhanced-hooks	Yes	
condor-ec2-enhanced	Yes	
condor-job-hooks	Yes	
condor-kbddd	Yes	
condor-low-latency	Yes	
condor-vm-gahp	Yes	
condor-wallaby-base-db	Yes	
condor-wallaby-client	Yes	
condor-wallaby-tools	Yes	
libyaml	No	Produced by libyaml spec and used by PyYAML
libyaml-devel	No	Produced by libyaml spec and used by PyYAML
libyaml-debuginfo	No	Produced by libyaml spec and used by PyYAML
PyYAML	No	Produced by the PyYAML spec and used by python-wallabyclient
PyYAML-debuginfo	No	Produced by the PyYAML spec and used by python-wallabyclient
python-boto	No	Produced by python-boto spec and used by condor-ec2-enhanced and condor-ec2-enhanced-hooks
python-condorec2e	No	Produced from condor-ec2-enhanced-hooks spec and used by condor-ec2-enhanced and condor-ec2-enhanced-hooks
python-condorutils	No	Produced by condor-job-hooks and used by condor-job-hooks, condor-low-latency, condor-ec2-enhanced and condor-ec2-enhanced-hooks
python-wallabyclient	No	Produced by condor-wallaby spec and used by condor-wallaby-client and condor-wallaby-tools
ruby-rhubarb	No	Produced by ruby-rhubarb spec

		and used by wallaby
ruby-spqr	No	Produced by ruby-spqr spec and used by wallaby
ruby-sqlite3	No	Produced by ruby-sqlite3 spec and used by ruby-rhubarb
ruby-sqlite3-debuginfo	No	Produced by ruby-sqlite3 spec and used by ruby-rhubarb
ruby-wallaby	No	Produced from wallaby spec and used by wallaby and wallaby-utils
spqr-gen	No	Produced by ruby-spqr spec and used by wallaby
wallaby-utils	Yes	
wallaby	Yes	
wso2-rampart	No	Produced by wso2-wsf-cpp and used by condor-aviary
wso2-rampart-devel	No	Produced by wso2-wsf-cpp and used by condor-aviary
wso2-axis2	No	Produced by wso2-wsf-cpp and used by condor-aviary
wso2-axis2-devel	No	Produced by wso2-wsf-cpp and used by condor-aviary
wso2-wsf-cpp	No	Produced by wso2-wsf-cpp and used by condor-aviary
wso2-wsf-cpp-devel	No	Produced by wso2-wsf-cpp and used by condor-aviary
wso2-wsf-cpp-debuginfo	No	Produced by wso2-wsf-cpp and used by condor-aviary

6.2.4. Management Console 2.0 Release

The following is a list of packages and comments for each package of MRG Management Console.

Table 6.4. Red Hat Enterprise MRG Management Package Manifest

Package Name	Core Package	Comments
cumin	Yes	
python-psycopg2	No	
python-psycopg2-doc	No	

Revision History

Revision 1-15	Sat Mar 2 2013	Cheryn Tan
Prepared for publishing (MRG 2.3).		
Revision 1-14	Fri Mar 1 2013	Cheryn Tan
Implemented fixes from SME review to Grid release notes.		
Revision 1-13	Wed Feb 27 2013	David Ryan
Added new release notes for MRG 2.3 - Grid.		
Revision 1-12	Fri Feb 22 2013	Joshua Wulf
Added new release notes for MRG 2.3 - Messaging.		
Revision 1-11	Mon Feb 18 2013	Cheryn Tan
Added new release notes for MRG 2.3 - Realtime.		
Revision 1-10	Mon Oct 15 2012	Tomáš Čapek
Added a new release note for Grid.		
Revision 1-9	Mon Sep 17 2012	Tomáš Čapek
Content for the MRG 2.2 release added.		
Revision 1-8	Thu Mar 22 2012	Tomáš Čapek
Added release notes and known issues for MRG Messaging asynchronous update		
Revision 1-7	Tue Feb 28 2012	Tim Hildred
Updated configuration file for new publication tool.		
Revision 1-6	Mon Feb 6 2012	Tomáš Čapek
Added additional release notes and known issues for Grid and Console as part of an MRG asynchronous update		
Revision 1-5	Mon Jan 23 2012	Tomáš Čapek
Added release notes and known issues for Grid, RT, and Console as part of MRG 2.1 release		
Revision 1-4	Thu Dec 4 2011	Tomáš Čapek
Added RT release notes, collapsed pkg manifest, many minor edits		
Revision 1-3	Thu Dec 1 2011	Tomáš Čapek
Many edits in preparation for the MRG 2.1 release publishing		
Revision 1-2	Wed Nov 23 2011	Tomáš Čapek
Publishing draft version of the book for MRG 2.1 release		
Revision 1-1	Thu Sep 22 2011	Alison Young
Version numbering change		
Revision 1-0	Thu Jun 23 2011	Alison Young

Prepared for publishing

Revision 0.1-17 BZ#674351 - minor update	Thur Jun 23 2011	Alison Young
Revision 0.1-16 updated the book	Wed Jun 22 2011	Alison Young
Revision 0.1-15 BZ#696324 - minor fix	Wed Jun 22 2011	Alison Young
Revision 0.1-14 BZ#674351 - platform support update	Fri Jun 17 2011	Alison Young
Revision 0.1-13 BZ#688344 - remote configuration database procedure update	Thu Jun 16 2011	Alison Young
Revision 0.1-12 BZ#712092 - add a note about Amazon EC2 Support	Tue Jun 14 2011	Alison Young
Revision 0.1-11 BZ#702482 - setting max_connections configuration for cumin postgresql database BZ#705819 - minor XML fix BZ#712092 - add a note about Amazon EC2 Support	Tue Jun 14 2011	Alison Young
Revision 0.1-10 Updated RHN Channels BZ#688344 - remote configuration database procedure	Thu Jun 09 2011	Alison Young
Revision 0.1-09 Minor Updates	Wed Jun 08 2011	Misha Husnain Ali
Revision 0.1-08 Minor Updates	Wed Jun 08 2011	Misha Husnain Ali
Revision 0.1-07 Minor Updates	Wed Jun 08 2011	Misha Husnain Ali
Revision 0.1-06 Updated RHN Channels	Wed Jun 08 2011	Alison Young
Revision 0.1-05 Minor edits and additions.	Tue Jun 07 2011	Misha Husnain Ali
Revision 0.1-04 Minor edits and additions.	Tue Jun 07 2011	Misha Husnain Ali
Revision 0.1-03 Minor edits and supported platform updates.	Mon Jun 06 2011	Misha Husnain Ali

Revision 0.1-02**Fri Jun 03 2011****Misha Husnain Ali**

First draft.

Revision 0.1-01**Tue Feb 22 2011****Alison Young**

Fork from 1.3