



Red Hat JBoss Data Grid 6.4

6.4.1 Release Notes

Known and resolved issues for Red Hat JBoss Data Grid 6.4.1

Red Hat JBoss Data Grid 6.4 6.4.1 Release Notes

Known and resolved issues for Red Hat JBoss Data Grid 6.4.1

Misha Husnain Ali

Red Hat Engineering Content Services

mhusnain@redhat.com

Legal Notice

Copyright © 2015 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

The Red Hat JBoss Data Grid 6.4.1 Release Notes list and provide descriptions for a series of bugzilla bugs. The bugs highlights issues that are known problems and resolved issues for the release.

Table of Contents

CHAPTER 1. INTRODUCTION TO RED HAT JBOSS DATA GRID 6.4.1	3
1.1. ABOUT RED HAT JBOSS DATA GRID	3
1.2. OVERVIEW	3
CHAPTER 2. SUPPORTED CONFIGURATIONS	4
CHAPTER 3. COMPONENT VERSIONS	5
CHAPTER 4. KNOWN AND RESOLVED ISSUES	6
4.1. KNOWN ISSUES	6
4.2. RESOLVED ISSUES	8
APPENDIX A. REVISION HISTORY	11

CHAPTER 1. INTRODUCTION TO RED HAT JBOSS DATA GRID

6.4.1

Welcome to the Red Hat JBoss Data Grid 6.4.1. As you become familiar with the newest version of JBoss Data Grid, these Release Notes provide you with information about new features, as well as known and resolved issues. Use this document in conjunction with the entire JBoss Data Grid documentation suite, available at the Red Hat Customer Service Portal's [JBoss Data Grid documentation page](#).

[Report a bug](#)

1.1. ABOUT RED HAT JBOSS DATA GRID

Red Hat's JBoss Data Grid is an open source, distributed, in-memory key/value data store built from the Infinispan open source software project. Whether deployed in client/server mode or embedded in a Java Virtual Machine, it is built to be elastic, high performance, highly available and to scale linearly.

JBoss Data Grid is accessible for both Java and Non-Java clients. Using JBoss Data Grid, data is distributed and replicated across a manageable cluster of nodes, optionally written to disk and easily accessible using the REST, Memcached and Hot Rod protocols, or directly in process through a traditional Java Map API.

[Report a bug](#)

1.2. OVERVIEW

This document contains information about the known and resolved issues of Red Hat JBoss Data Grid version 6.4.1. Customers are requested to read this documentation prior to installing this version.

[Report a bug](#)

CHAPTER 2. SUPPORTED CONFIGURATIONS

For supported configurations, see the Red Hat JBoss Data Grid Supported Configurations reference on the Customer Portal at <https://access.redhat.com/site/articles/115883>.

[Report a bug](#)

CHAPTER 3. COMPONENT VERSIONS

For component versions, see the Red Hat JBoss Data Grid Supported Configurations reference on the Customer Portal at <https://access.redhat.com/articles/488833>.

[Report a bug](#)

CHAPTER 4. KNOWN AND RESOLVED ISSUES

4.1. KNOWN ISSUES

BZ-1175272 - CDI fails when both remote and embedded uber-jar are present

In Red Hat JBoss Data Grid, when both *infinispan-remote* and *infinispan-embedded* dependencies are on the classpath, the Infinispan CDI extension does not work as expected. This is due to bundling the Infinispan CDI extension in both jar files. As a result, CDI fails with ambiguous dependencies exception.

To prevent this issue either EAP modules may be used, or separate jar files may be bundled instead of using the uber-jars.

BZ-1158559 - C++ HotRod Client, RemoteCache.clear() will throw out exception when data is more than 1M

In Red Hat JBoss Data Grid, when a cache contains a large number of entries, the clear() operation can take an unexpectedly long time and possibly result in communication timeouts. In this case, the exception is reported to the Hot Rod client.

This is a known issue and no workaround is currently available for this issue.

BZ-1158839 - Clustered cache with FileStore (shared=false) is inconsistent after restarting one node if entries are deleted during restart

In Red Hat JBoss Data Grid, when a node restarts, it does not automatically purge entries from its local cache store. As a result, the Administrator starting the node must change the node configuration manually to set the cache store to be purged when the node is starting. If the configuration is not changed, the cache may be inconsistent (removed entries can appear to be present).

This is a known issue and no workaround is currently available for this issue.

BZ-1024373 - Default optimistic locking configuration leads to inconsistency

In Red Hat JBoss Data Grid, transactional caches are configured with optimistic locking by default. Concurrent `replace()` calls can return true under contention and transactions might unexpectedly commit.

Two concurrent commands, `replace(key, A, B)` and `replace(key, A, C)` may both overwrite the entry. The command which is finalized later wins, overwriting an unexpected value with new value.

As a workaround, enable write skew check and the *REPEATABLE_READ* isolation level. This results in concurrent replace operations working as expected.

BZ-1114080 - HR client SASL MD5 against LDAP fails

In Red Hat JBoss Data Grid, the server does not support pass-through MD5 authentication against LDAP. As a result, the Hot Rod client is unable to authenticate to the JBoss Data Grid server via MD5 as the authentication is backed by the LDAP server.

As a workaround use the PLAIN authentication over end-to-end SSL encryption.

BZ-1092403 - JPA cachestore fails to guess dialect for Oracle12c and PostgresPlus 9

In Red Hat JBoss Data Grid, JPA Cache Store does not work with Oracle12c and Postgres Plus 9 as Hibernate, an internal dependency of JPA Cache Store, is not able to determine which dialect to use for communication with the database.

As a workaround for this issue, specify the Hibernate dialect directly by adding the following element in the `persistence.xml` file:

```
<property name="hibernate.dialect" value="${hibernate.dialect}" />
```

Set `${hibernate.dialect}` to `org.hibernate.dialect.Oracle10gDialect` or `org.hibernate.dialect.PostgresPlusDialect` for Oracle12c or Postgres Plus 9 respectively.

BZ-1101512 - [CLI] UPGRADE command fails when testing data stored via CLI with REST encoding

In Red Hat JBoss Data Grid, the CLI `upgrade` command fails to migrate data from the old cluster to the new cluster if the data being migrated was stored in the old cluster via CLI with REST encoding (e.g by issuing a command such as `put --codec=rest key1 val1`). This issue does not occur if data is stored via REST clients directly.

This issue only occurs when upgrading data that has been put in the cache using the CLI and the REST codec; any other method of placing data or any other codec will not encounter this issue.

BZ-881080 - Silence SuspectExceptions

In Red Hat JBoss Data Grid, `SuspectExceptions` are routinely raised when nodes shut down because they are unresponsive as they shut down. As a result, a `SuspectException` error is added to the logs. The `SuspectExceptions` do not affect data integrity.

This is a known issue and no workaround is currently available for this issue.

BZ-807674 - JDBC Cache Stores using a JTA Data Source do not participate in cache transactions

In Red Hat JBoss Data Grid's library mode, JDBC cache stores can be configured to use a JTA-aware datasource. However, operations performed on a cache backed by such a store during a JTA transaction will be persisted to the store outside of the transaction's scope. This issue is not applicable to JBoss Data Grid's Remote Client-Server mode because all cache operations are non-transactional.

This is a known issue and no workaround is currently available for this issue.

BZ-1163665 - Node can temporarily read removed data when another node joins the cluster, leaves or crashes

In Red Hat JBoss Data Grid, the distribution of entries in the cluster changes when a node joins, leaves or crashes. During this brief period, a read on the previous node owner can return stale data. When the rebalance process is completed, further reads return up-to-date data.

This is a known issue and no workaround is currently available for this issue.

BZ-1178965 - Fail fast when using two phase commit with ASYNC backup strategy

In Red Hat JBoss Data Grid, using the two phase commit with the ASYNC backup strategy results in one phase commit unexpectedly being used instead.

This is a known issue and no workaround is currently available for this issue.

BZ-1012036 - RELAY2 logs error when site unreachable

In Red Hat JBoss Data Grid, when a site is unreachable, JGroups's **RELAY2** logs an error for each dropped message. Infinispan has configurable fail policies (**ignore/warn/abort**), but the log is filled with errors despite the **ignore** policy.

This is a known issue and no workaround is currently available for this issue.

BZ-1107613 - SASL GSSAPI auth doesn't use principal configured login_module

In Red Hat JBoss Data Grid, the server principal is always constructed as **jgroups/server_name** and is not loaded from the Kerberos login module. Using a different principal results in an authentication failure.

As a workaround for this issue use **jgroups/server_name** as the server principal.

BZ-1088073 - Move rebalancing settings in JON from cache level to global cluster level

In Red Hat JBoss Data Grid, it is possible to change rebalancing settings using the JBoss Operations Network UI when navigating to the JBoss Data Grid Server/Cache Container/Cache/Configuration (current)/Distributed Cache Attributes/Rebalancing.

This operation is currently misrepresented as a cache-level operation when the changed rebalancing settings automatically apply for all the parent cache manager's caches and for all nodes in a particular cluster.

This is a known issue and no workaround is currently available for this issue.

BZ-881791 - Special characters in file path to JDG server are causing problems

In Red Hat JBoss Data Grid, when special characters are used in the directory path, the JBoss Data Grid server either fails to start or a configuration file used for logging cannot be loaded properly. Special characters that cause problems include spaces, # (hash sign), ! (exclamation mark), % (percentage sign), and \$ (dollar sign).

As a workaround for this issue avoid using special characters in the directory path.

[Report a bug](#)

4.2. RESOLVED ISSUES

BZ-1168043 - Better documentation of state-transfer configuration and behaviour

In previous versions of Red Hat JBoss Data Grid, for a cache store where *sharing* is set to **false**, the *fetchInMemory* and *fetchPersistence* parameters must also be set to **true** or different nodes may contain different copies of the same data.

For a cache store where the *shared* parameter is set to `true`, set *fetchPersistence* must be set to `false` because the persistence is shared and enabling it results in unnecessary state transfers.

However, the *fetchInMemory* parameter can be set to either `true` or `false`. Setting this to `true` loads the in-memory state via the network and resulted in a faster start up. Setting the value to `false` loads the data from persistence without transferring the data remotely from other nodes.

The *Administration and Configuration Guide* has been updated to include additional information regarding state-transfer configuration and behaviour.

BZ-1184373 - During state transfer, previously removed entry was revived

In previous versions of Red Hat JBoss Data Grid, an entry removed or updated during ongoing state-transfer after previous coordinator leave can be overwritten on some nodes. This results in stale entry with old value (before the removal or update).

The handling of JGroups views was improved and cluster topology that sent from old coordinators is now ignored. Additionally, the response returned from a transaction that was rolled back has been fixed.

BZ-1185779 - Transaction cannot be recommitted in new topology

In previous versions of Red Hat JBoss Data Grid, each transaction can be executed on each node only once. When a transaction affects multiple entries, one of them owned and another not owned, only the owned entry is modified during execution. However, if there is a topology change (change of entry ownership) during the transaction, the node may become owner of some entry while receiving old (pre-transaction) value via the state-transfer. The transaction should execute again on the newly owned entries, but it does not so since it is already marked as completed.

This issue was addressed by reworking how transactions are marked as completed.

BZ-1200360 - Hot Rod client does not full recover after a complete server shutdown

In previous versions of Red Hat JBoss Data Grid, after a complete server shutdown the Hot Rod client was only able to reconnect to the last known server; in addition, the cluster view was no longer updated.

The Hot Rod client will now check to see if the cluster is available, and if not will try to restore the initial host configuration and see if the cluster has been restarted. If the Hot Rod client is not able to connect to the cluster it will fail and log a message indicating that the cluster is not available.

BZ-1192324 - Inconsistent logging for starting and finished rebalance

In previous versions of Red Hat JBoss Data Grid the finished cluster-wide rebalance event was previously logged at DEBUG level which was not consistent with other events related to re-balancing.

This issue is now fixed and the event is logged on INFO level.

BZ-1198452 - LIRS eviction strategy fixes

In previous versions of Red Hat JBoss Data Grid, LIRS eviction can cause some elements to be evicted prematurely, resulting in data not being passivated to the cache store.

The eviction policies have been updated, and the data container now ensures atomicity when passivating and activating entries to address this issue.

BZ-1200514 - Persistent Store containing a Map written by JDG <=6.3 can not read with JDG 6.4.0

After upgrading to Red Hat JBoss Data Grid 6.4.0 from a prior version users were unable to load data where a Map was used as a key or value.

This issue has been addressed by updating Map serialization to support both the 6.3 and the 6.4.0 method, allowing maps from either version to be read successfully.

BZ-1190001 - Avoid invalid topology

In previous versions of Red Hat JBoss Data Grid, when clients make requests to the server while there is a topology view change ongoing the server may send back partial topology updates; this behavior results in clients talking to the server with an incorrect view which may result in segments having no owners.

Now if a segment is found to have no owners, then the first node of the topology is sent as a segment owner. Once the cluster has stabilized, then the fully formed topology will be received by the clients.

[Report a bug](#)

APPENDIX A. REVISION HISTORY

Revision 6.4.1-1 Adjusted version.	Wed 1 Apr 2015	Christian Huffman
Revision 6.4.1-0 Created initial draft. Updated list of Known and Resolved Issues.	Tue 10 Feb 2015	Misha Husnain Ali