



Red Hat JBoss Data Grid 6.3

6.3.1 Release Notes

Known and resolved issues for Red Hat JBoss Data Grid 6.3.1

Red Hat JBoss Data Grid 6.3 6.3.1 Release Notes

Known and resolved issues for Red Hat JBoss Data Grid 6.3.1

Misha Husnain Ali

Red Hat Engineering Content Services

mhusnain@redhat.com

Legal Notice

Copyright © 2014 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 6.3.1 Release Notes list and provide descriptions for a series of bugzilla bugs. The bugs highlights issues that are known problems for the relevant release.

Table of Contents

CHAPTER 1. INTRODUCTION TO RED HAT JBOSS DATA GRID	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 ISSUES	6
CHAPTER 5. RESOLVED ISSUES	8
APPENDIX A. REVISION HISTORY	11

CHAPTER 1. INTRODUCTION TO RED HAT JBOSS DATA GRID

Welcome to the Red Hat JBoss Data Grid 6.3.1. As you become familiar with the newest version of JBoss Data Grid, these Release Notes provide you with information about known and resolved issues. Use this document in conjunction with the entire JBoss Data Grid 6.3.x 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 known and resolved issues for Red Hat JBoss Data Grid version 6.3.1. Customers are requested to read this document prior to installing this version.

[Report a bug](#)

CHAPTER 2. SUPPORTED CONFIGURATIONS

For supported hardware and software 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

The full list of component versions used in Red Hat JBoss Data Grid is available at the Customer Portal at <https://access.redhat.com/site/articles/488833>.

[Report a bug](#)

CHAPTER 4. KNOWN ISSUES

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 in JBoss Data Grid 6.3.1. No workaround is currently available for this issue.

BZ#881080 - Silence SuspectExceptions

In Red Hat JBoss Data Grid, `SuspectExceptions` are routinely raised when nodes are shutting 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 in JBoss Data Grid 6.3.1. No workaround is currently available for this issue.

BZ#1101512 - CLI UPGRADE command fails when testing data stored via CLI

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 (for example, by issuing a command such as `put --codec=hotrod key1 val1`). This issue does not occur if data is stored via the Hot Rod or REST clients directly.

This is a known issue in JBoss Data Grid 6.3.1. There is currently no workaround for this solution.

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.

This is an open issue in JBoss Data Grid 6.3.1. As a workaround for around 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#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)`, `replace(key, A, C)` may both overwrite the entry. The command which is finalized later wins, overwriting an unexpected value with new value.

This is a known issue in JBoss Data Grid 6.3.1. As a workaround, enable write skew check and the *REPEATABLE_READ* isolation level. This results in concurrent replace operations working as expected.

BZ#1118204 - Infinispan Query - Concurrency problem with WeakIdentityHashMap in FullTextIndexEventListener

In Red Hat JBoss Data Grid, when using put/remove operations via Hot Rod, some objects are not be indexed if the system is heavily loaded. These object are not returned by Remote Query or based on their previous state. This also affects embedded querying when the cache is not transactional.

This bug may also cause loss of update and delete operations, so while it's true that subsequent queries might miss results of newly inserted objects, they might also contain results which aren't supposed to be returned. The problem on missed delete operations only affects usage of projection, as when the user asks to return the object from the grid we remove the missing ones from the results.

This is a known issue in JBoss Data Grid 6.3.1. The workaround is to mark the cache transactional, regardless of the cache being accessed over Hot Rod (note that then every single operation is enclosed in a transaction) or locally as an embedded cache.

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 for JBoss Data Grid 6.3.1 and no workaround is currently available.

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).

This is a known issue in JBoss Data Grid 6.3.1. A workaround for this issue is to avoid using special characters in the directory path.

BZ#1076084 - RHQ server plugin: remote store cache child creation fails

Currently, it is not possible to create a remote cache store child resource for a cache using the JBoss Data Grid Remote Client-Server plug-in for JBoss Operation Network. The operation subsequently fails and the JBoss Operations Network Agent records the failure in its log file. As a result, a remote cache store cannot be configured using the JBoss Operations Network user interface.

This is a known issue in JBoss Data Grid 6.3.1. As a workaround for this issue, modify the JBoss Data Grid server configuration file to manually configure the remote cache store.

[Report a bug](#)

CHAPTER 5. RESOLVED ISSUES

BZ#1122162 - Infinispan Core module for EAP not importing Services from the Query module

Previously in Red Hat JBoss Data Grid, the module definition for the productized Infinispan Core in `$EAP_MODULES_LIBRARY/modules/org/infinispan/jdg-6.3` depended on the Query module but failed to import its services. Not loading the services allowed the Indexing feature to be enabled, although several operations were failing at run time. This issue is now fixed in JBoss Data Grid 6.3.1.

BZ#1134184 - Hot Rod client receives `ArrayIndexOutOfBoundsException` and `InvalidResponseException` when topology changes

Previously in JBoss Data Grid, the new segment-based topology information sent to Hot Rod clients would incorrectly compute the segments. As a result, segment entries were incorrectly added for nodes that were not present. This presented a runtime error when the distributed cache view changes were processed by the Hot Rod client. This is fixed in JBoss Data Grid 6.3.1 so that servers that are not part of the topology are filtered out from the segment information sent to Hot Rod clients. As a result, the Hot Rod clients correctly process topology views.

BZ#1136109 - Improve `SyncConsistentHashFactory` key distribution

Previously in Red Hat JBoss Data Grid, data distribution was unexpectedly uneven when using `SyncConsistentHashFactory` and `TopologyAwareSyncConsistentHashFactory` due to a problem that prevented these two classes from distributing the segments randomly between nodes as expected. This is now fixed in JBoss Data Grid 6.3.1. `SyncConsistentHashFactory` and `TopologyAwareSyncConsistentHashFactory` can now distribute data correctly so that each node is only given a certain percentage of data compared to other nodes. As a result, data is now distributed evenly between nodes.

BZ#1136064 - Externalizers not being used in the Lucene Directory

Red Hat JBoss Data Grid includes a Wildfly module responsible for the query feature. This module included declared Externalizers that were not being loaded. As a result, a plain Java serialization was used to transfer index files instead. This affected indexed queries that used the Infinispan Lucene directory for storage. This is now fixed in JBoss Data Grid 6.3.1 and querying now works as expected.

BZ#1135924 - Race condition during unmarshalling in the Infinispan Lucene Directory

Previously in Red Hat JBoss Data Grid, if Hibernate Search was configured to use the Infinispan directory provider, deploying an application using Hibernate Search resulted in a `java.lang.ClassCastException`. This issue is fixed in JBoss Data Grid 6.3.1 and Hibernate Search works as expected when using Infinispan directory provider.

BZ#1135580 - `LuceneCacheLoader` doing unnecessary IO

Previously in JBoss Data Grid, after a read-only Lucene index was preloaded via the `cacheloader`, the backing index was still accessed for small segments. This is now fixed in JBoss Data Grid 6.3.1 so that preloading a read-only Lucene index works as expected.

BZ#1135553 - `QueryInterceptor` & `ClusterRegistry` racing conditions fixes

Previously in Red Hat JBoss Data Grid, race conditions caused intermittent timeouts in indexed queries and missing `@Indexed` annotation errors. This is now fixed in JBoss Data Grid 6.3.1 and

these race conditions work as expected and do not cause unexpected timeouts and annotation errors.

BZ#1136438 - Caching of parsed HQL query objects

Previously in Red Hat JBoss Data Grid, the same queries were periodically parsed multiple times, which results in a minor performance issue. This is fixed in JBoss Data Grid 6.3.1 so that a local cache is introduced to prevent the same query being unnecessarily parsed more than once.

BZ#1131117 - Log rebalancing messages to specific category

Previously in Red Hat JBoss Data Grid, users had to enable the DEBUG logging level to view re-balancing events. This is now fixed in JBoss Data Grid 6.3.1 so that logging major re-balancing events such as re-balancing started/enabled/suspended are now logged under a specific category (`org.infinispan.CLUSTER`) and the INFO level.

BZ#1113585 - LevelDBStore.stop() crashes JVM in native code

Previously in Red Hat JBoss Data Grid, when a cache using LevelDB cache store was stopped (for example, as a consequence of stopping the cache manager), the LevelDB native implementation caused a segmentation fault in the JVM process. As a result of this segmentation fault, the process crashed. This issue is now fixed in JBoss Data Grid 6.3.1 so that using the LevelDB cache store native implementation works as expected.

BZ#1130493 - Inserting into cache with indexing fails for XA transactions

Previously in Red Hat JBoss Data Grid, when a transactional cache was configured as an XA resource (`useSynchronization="false"`) and indexing is enabled on this cache, inserting an entry into this cache may fail due to deadlock between this cache and internal caches. This deadlock occurs under race conditions and only once per a type (class) of value. This is fixed in JBoss Data Grid 6.3.1 and the transactional cache works as expected when configured as an XA resource and with indexing enabled.

BZ#1128791 - Remove timeout from Map/Reduce jobs

Previously in Red Hat JBoss Data Grid, users had to extend the timeout value of map/reduce jobs because the default timeout value was the RPC timeout value. In JBoss Data Grid 6.3.1, this is fixed so that the timeout value for map/reduce jobs has been removed and the job will wait for an infinite period before timing out. If required, users can lower the timeout. As a result, users do not need to set the timeout as often as before.

BZ#1122269 - Replace fails with cache loader

Previously in Red Hat JBoss Data Grid, the `cache.replace(key, oldValue, newValue)` operation was comparing the requested new value with previous value, and if they differed it turned into a no-operation. However, `CacheLoaderInterceptor` did not load entries for a `ReplaceCommand`. If the entry only existed in the loader and not in memory, this caused the replace operation to fail. This issue is now fixed in JBoss Data Grid 6.3.1 and the replace operation works correctly even if the old value is only in the cache store and not in memory.

BZ#1131645 - RemoveCommand does not activate the key in Passivation

Previously in Red Hat JBoss Data Grid, when a persistent cache store was used with Passivation, the `remove` command did not activate the entry in cache store, which resulted in an inconsistency because the entry was never removed. It was removed from internal `DataContainer` but not from

the cache store and during any operation, the old value could be loaded from the cache store back again. This is fixed in JBoss Data Grid 6.3.1 and the remove operation now works correctly.

BZ#1080359 - ConfigurationTest.testTableProperties fails constantly on all environments with JDK6

Previously in Red Hat JBoss Data Grid, users were unable to use the `.withProperties()` method to configure JDBC cache store when running JDK 1.6. This was caused by the fact that the JDK could not find the correct property editor for the `DatabaseType` class. This problem does not occur with JDK 1.7 because `com.sun.beans.editors.EnumEditor` is used by default. However, this class is not present in JDK 1.6 so a specific property editor was added for deployments on JDK 1.6. This issue is now fixed in JBoss Data Grid 6.3.1.

BZ#1119780 - CDI annotations

Previously in Red Hat JBoss Data Grid, certain user code failed to compile because some Infinispan CDI classes had been refactored to different packages located in other modules. In JBoss Data Grid 6.3.1, the Infinispan CDI classes have been moved back to the original package and module. As a result, the user code no longer fails to compile because the CDI classes are not available where expected.

BZ#1132912 - The JGroups configuration files included with JDG reference the unsupported tom.TOA protocol

Previously in Red Hat JBoss Data Grid, the UDP, TCP, and EC2 JGroups configuration files included the `<tom.TOA/>` element. However, the TOA protocol is not supported in JBoss Data Grid. This issue is now resolved in JBoss Data Grid 6.3.1 and the TOA protocol is now removed.

BZ#1134085 - Loading LDAP roles fails when some principal doesn't have an LDAP record

Previously in Red Hat JBoss Data Grid, user authentication failed with a `NamingException` because the principal containing the remote client's network address could not be located in the LDAP directory. As a result, when resolving the roles associated with a Hot Rod authenticated user against an LDAP directory, the list of principals included an `InetAddressPrincipal` which contained the network address of the remote client. This is fixed in JBoss Data Grid 6.3.1. The role resolution logic has been modified so that the network address principal (`InetAddressPrincipal`) is not included in the list of principals that are verified against the LDAP directory. As a result, the role resolution of users authentication over Hot Rod works as expected.

BZ#1124743 - Include EAP 6.2.x schemas for the server config and update example configs accordingly

Previously, Red Hat JBoss Data Grid did not include the Red Hat JBoss Enterprise Application Platform 6.3.x schemas. The `jboss-as-config_1_5.xsd` schema in particular is useful because it documents support for LDAP in the security realm authorization element. This is fixed in JBoss Data Grid 6.3.1 with the schema file included in the `docs/schema` directory in the server distribution.

[Report a bug](#)

APPENDIX A. REVISION HISTORY

Revision 6.3.1-4 Fixed minor typo.	Mon Sep 22 2014	Misha Husnain Ali
Revision 6.3.1-3 Included one additional bug.	Mon Sep 22 2014	Misha Husnain Ali
Revision 6.3.1-1 Implemented minor changes based on QE feedback.	Fri Sep 19 2014	Misha Husnain Ali
Revision 6.3.1-0 Added first draft of content.	Tue Sep 16 2014	Misha Husnain Ali