



Red Hat JBoss Data Grid 6.6

6.6.0 Release Notes

Known and resolved issues for Red Hat JBoss Data Grid 6.6.0

Red Hat JBoss Data Grid 6.6 6.6.0 Release Notes

Known and resolved issues for Red Hat JBoss Data Grid 6.6.0

Christian Huffman

Red Hat Engineering Content Services

chuffman@redhat.com

Legal Notice

Copyright © 2017 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 6.6.0 Release Notes list and provide descriptions for a series of bugzilla bugs. The bugs highlight issues that are known problems and resolved issues for the release.

Table of Contents

CHAPTER 1. INTRODUCTION TO RED HAT JBOSS DATA GRID 6.6	3
1.1. ABOUT RED HAT JBOSS DATA GRID	3
1.2. OVERVIEW	3
CHAPTER 2. NEW FEATURES AND ENHANCEMENTS	4
2.1. CONTINUOUS QUERIES	4
2.2. QUERY AGGREGATION OPERATIONS	4
2.3. CLUSTERED EXPIRATION EVENTS	4
2.4. CROSS SITE FAILOVER FOR HOT ROD CLIENT	4
CHAPTER 3. DEPRECATED FEATURES	5
3.1. MAP/REDUCE	5
3.2. C++ HOTROD CLIENT ON RHEL 5	5
3.3. SPRING FRAMEWORK 3.2	5
3.4. NOTIFYINGFUTURE	5
CHAPTER 4. SUPPORTED CONFIGURATIONS	6
4.1. SUPPORTED CONFIGURATIONS	6
CHAPTER 5. COMPONENT VERSIONS	7
5.1. COMPONENT VERSIONS	7
CHAPTER 6. KNOWN AND RESOLVED ISSUES	8
6.1. KNOWN ISSUES	8
6.2. RESOLVED ISSUES	9
APPENDIX A. REVISION HISTORY	12

CHAPTER 1. INTRODUCTION TO RED HAT JBOSS DATA GRID

6.6

Welcome to Red Hat JBoss Data Grid 6.6.2. 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 6.6.0. Customers are requested to read this documentation prior to installing this version.

[Report a bug](#)

CHAPTER 2. NEW FEATURES AND ENHANCEMENTS

2.1. CONTINUOUS QUERIES

Red Hat JBoss Data Grid (JDG) 6.6 introduces the ability to specify Continuous Queries, in which the result set is kept up to date as entries in the cache change, without having to rerun the query. This feature is available in both Library and Client-Server modes.

[Report a bug](#)

2.2. QUERY AGGREGATION OPERATIONS

JDG 6.6 adds aggregation operations such as sum, average, min/max, count, group-by, and order-by to the Query DSL. This enhancement is available in both Library and Client-Server modes.

[Report a bug](#)

2.3. CLUSTERED EXPIRATION EVENTS

JDG 6.6 features support for listeners to view clustered, lifespan-based expiration events. This enables you to implement custom logic triggered by expiration of an entry when its lifespan has elapsed. This feature is available in both Library and Client-Server modes.

[Report a bug](#)

2.4. CROSS SITE FAILOVER FOR HOT ROD CLIENT

In JDG 6.6, if cross datacenter replication is configured for JDG Server, then a Java Hot Rod client application can be configured to failover to the backup JDG cluster if the primary cluster becomes unavailable. Switching between sites can also be done programmatically. This enhancement is applicable to Client-Server mode.

[Report a bug](#)

CHAPTER 3. DEPRECATED FEATURES

3.1. MAP/REDUCE

The native Map/Reduce API in JDG 6 has been deprecated in version 6.6.0 and is expected to be removed in version 7.0.0. JDG 7.0.0 will support the Map/Reduce use case via a new Distributed Streams feature, which is based on the Java 8 Stream API.

[Report a bug](#)

3.2. C++ HOTROD CLIENT ON RHEL 5

The C++ HotRod Client on Red Hat Enterprise Linux (RHEL) 5 has been deprecated in JBoss Data Grid 6.6.0, and is expected to be removed in version 7.0.0; the C++ HotRod Client for RHEL 6 and RHEL 7 will continue to be supported in JDG 7.0.0.

[Report a bug](#)

3.3. SPRING FRAMEWORK 3.2

The `infinispan-spring3` module has been deprecated in JBoss Data Grid 6.6.0 as the Spring Framework 3.2.x line is reaching End-Of-Life at the end of 2016, and is expected to be removed in version 7.0.0. The integration with Spring Framework 4.x will continue to be supported in version 7.0.0, via the `infinispan-spring4` module.

[Report a bug](#)

3.4. NOTIFYINGFUTURE

The `NotifyingFuture` interface is being deprecated in JBoss Data Grid 6.6.0 in favor of the standard Java 8 `CompletableFuture`, and is expected to be removed in JBoss Data Grid 7.0. `NotifyingFuture` is currently returned by all async methods in both `Cache` and `RemoteCache`, and will require applications using these methods to use the new class once available.

[Report a bug](#)

CHAPTER 4. SUPPORTED CONFIGURATIONS

4.1. 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 5. COMPONENT VERSIONS

5.1. 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 6. KNOWN AND RESOLVED ISSUES

6.1. KNOWN ISSUES

BZ-1200822 - JSR-107 Support for clustered caches in HotRod implementation

When creating a new cache (which is not defined in server configuration file) in HotRod implementation of JSR-107, the cache is created as `local` only in one of the servers. This behavior requires class `org.jboss.as.controller.client.ModelControllerClient` to be present on the classpath.

As a workaround use a clustered cache defined in the server configuration file. This still requires `cacheManager.createCache(cacheName, configuration)` to be invoked before accessing the cache for the first time.

BZ-1204813 - JSR-107 Support for `cacheResolverFactory` annotation property

JCache annotations provides a way to define a custom `CacheResolverFactory`, used to produce `CacheResolver`; this class's purpose is to decide which cache is used for storing results of annotated methods; however, the support for specifying a `CacheResolver` is not provided yet.

As a workaround, define a CDI `ManagedCacheResolver` which will be used instead.

BZ-1223290 - JPA Cache Store not working properly on Weblogic

A JPA Cache Store deployed to WebLogic servers throws a `NullPointerException` after the following error message:

```
Entity manager factory name (org.infinispan.persistence.jpa) is already registered
```

This is a known issue in Red Hat JBoss Data Grid 6.6.0, and no workaround exists at this time.

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 in Red Hat JBoss Data Grid 6.6.0, and no workaround exists at this time.

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 if the authentication is backed by the LDAP server.

This is a known issue in Red Hat JBoss Data Grid 6.6.0 and a workaround is to use the `PLAIN` authentication over end-to-end SSL encryption.

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.

This is a known issue in Red Hat JBoss Data Grid 6.6.0. As a workaround, enable write skew check and the `REPEATABLE_READ` isolation level. This results in concurrent replace operations working as expected.

BZ-1293575 - Rolling upgrade fails with keySet larger than 2 GB

Rolling upgrades fail if the key set is larger than 2 GB of memory. The process fails when calling `recordKnownGlobalKeyset` because the keys cannot be dumped into a single byte array in the source cluster.

This is a known issue in Red Hat JBoss Data Grid 6.6.0, and no workaround exists at this time.

BZ-1300133 - JMX attribute evictions is always zero in Statistics and ClusterCacheStats MBeans

The `evictions` attribute of `Statistics` and `ClusterCacheStats` components of the Cache MBean return zero even though some eviction operations have been successfully performed. This issue only affects statistics, not the actual eviction process.

This is a known issue in Red Hat JBoss Data Grid 6.6.0, and no workaround exists at this time.

BZ-1273411 - Cannot access cache with authorization enabled when using REST protocol

When authorization is configured for a cache, then any access to the cache via REST endpoint results in a security exception. A user is not able to access the cache since the security `Subject` representing the user is not properly defined, and the user cannot be authorized to access the cache.

This is a known issue in Red Hat JBoss Data Grid 6.6.0, and no workaround exists at this time.

[Report a bug](#)

6.2. RESOLVED 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.

This is resolved as of Red Hat JBoss Data Grid 6.6.0. The content of `infinispan-embedded` and `infinispan-remote` were restructured so that only one of the jar files is needed for any use case. There is no need to use them together anymore.

BZ-1255783 - Server should enable writeSkew for some configurations by default

The server configuration did not allow users to specify write skew check option. This behavior could lead to inconsistencies on failure scenarios when using conditional operations with optimistic locking caches.

This issue is resolved as of Red Hat JBoss Data Grid 6.6.0, where write skew check is automatically enabled when using a combination of optimistic locking, synchronous cache mode, and `REPEATABLE_READ` isolation level.

BZ-1276035 - Default cache resources in JBoss ON are not available

The metrics for local caches were not properly registered within the server. As a result, local cache resources were not available in the JON UI. This issue was only related to local caches and did not affect clustered caches.

This issue is resolved as of Red Hat JBoss Data Grid 6.6.0.

BZ-1284846 - Custom audit logger is ignored in server mode

In Red Hat JBoss Data Grid the custom audit logger option was ignored in server mode, and the default audit logger was not turned on.

This issue is resolved as of Red Hat JBoss Data Grid 6.6.0. Users can now set up their own audit logger, and if no audit logger is provided, then the default audit logger is used.

BZ-1222995 - JSR-107 - Provide mechanism to handle expiration events

When using a cache configured with expiration in remote JCache implementation, expiration events were not handled in clustered environment on cache `c1`, if the insertion of an entry was performed by cache `c2`. As a consequence, cache entry listener registered on cache `c1` was not notified of the expiration event.

The issue is resolved as of Red Hat JBoss Data Grid 6.6.0, and cache entry listeners receive expiration events correctly in clustered environment.

BZ-1274155 - Client keeps using old view after merging of split brains

When using the HotRod client in JBoss Data Grid's Remote Client-Server mode, there was a possibility that an outdated view was used when the cluster healed from network partition. This could lead to data inconsistencies when performing operations by clients that received different views.

The issue is resolved as of Red Hat JBoss Data Grid 6.6.0. The HotRod client correctly receives an updated view after the partition is healed.

BZ-1273137 - HotRod Client does not handle failover silent when nodes are stopped if using role based authentication

In Red Hat JBoss Data Grid, when roles based authentication was enabled, the HotRod client did not fail over; instead, it logged the exception when the JDG server to which the client was connected and went offline.

This issue is resolved as of Red Hat JBoss Data Grid 6.6.0. If the JDG server shuts down, then the HotRod client will fail over to another JDG server in the cluster silently, with no exception logged and the HotRod client will continue to work properly.

BZ-1272052 - Expiration does not work correct in C/S mode for a cache if putIfAbsent(K,V) is used

Expiration settings of a cache were ignored in JBoss Data Grid server if `putIfAbsent` operation was invoked via HotRod client. This could lead to wrong assumptions about presence of an entry in the cache, which never expired.

The issue is resolved as of Red Hat JBoss Data Grid 6.6.0. The entry correctly expires after predefined time period.

BZ-1267599 - SASL failure on jgroups merge event

In Red Hat JBoss Data Grid, when node authentication was enabled and a JDG node left the cluster it was not able to rejoin the cluster.

This issue is resolved as of Red Hat JBoss Data Grid 6.6.0. When node authentication is turned on, then the JDG node can rejoin the cluster without any issue.

BZ-1247380 - HotRod size method doesn't check BULK_READ permission

When `size()` is performed using the HotRod client, a MapReduce task is invoked internally to determine the total number of entries present in the cache. Previously this operation did not require any permissions; however, a check has been added to ensure the role invoking `size()` has the `BULK_READ` permission.

This issue is resolved as of Red Hat JBoss Data Grid 6.6.0.

BZ-1249009 - ClassCastException on transactional cache

There was a possibility of a transactional cache erroneously throwing `ClassCastException` when an entry was obtained while located on a remote node.

This issue is resolved as of Red Hat JBoss Data Grid 6.6.0.

[Report a bug](#)

APPENDIX A. REVISION HISTORY

Revision 6.6.0-2 Added deprecation notice of NotifyingFuture.	Fri 7 Jul 2017	John Brier
Revision 6.6.0-1 Added deprecation notice of Spring Framework.	Tue 26 Jan 2016	Christian Huffman
Revision 6.6.0-0 Initial Draft for 6.6.0. Included all Known and Resolved Issues. Added deprecation notes.	Thu 7 Jan 2016	Christian Huffman