



Red Hat JBoss Data Grid 6.1

6.1.0 Release Notes

New features, known issues, and resolved issues for Red Hat JBoss Data Grid 6.1
Edition 1

Red Hat JBoss Data Grid 6.1 6.1.0 Release Notes

New features, known issues, and resolved issues for Red Hat JBoss Data Grid 6.1
Edition 1

Gemma Sheldon
Red Hat Engineering Content Services
gsheldon@redhat.com

Legal Notice

Copyright © 2013 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 6.1 Release Notes list and provide descriptions for a series of bugzilla bugs. The bugs highlight either issues that are known problems for the relevant release or bugs that have now been resolved.

Table of Contents

CHAPTER 1. INTRODUCTION TO JBOSS DATA GRID 6.1	3
1.1. ABOUT JBOSS DATA GRID	3
1.2. OVERVIEW	3
1.3. NEW FEATURES AND ENHANCEMENTS	3
1.3.1. Cross Datacenter Replication	3
1.3.2. Distributed Execution and MapReduce Framework	3
1.3.3. Rolling Upgrades for Deployments Accessed Remotely via Hot Rod protocol	3
1.3.4. Contexts and Dependency Injection (CDI)	4
1.3.5. Non-blocking State Transfer	4
1.3.6. Management support for Library mode	4
1.3.7. Querying (Technology Preview)	4
1.3.8. Command Line Interface (Technology Preview)	4
CHAPTER 2. KNOWN ISSUES	5
CHAPTER 3. RESOLVED ISSUES	8
APPENDIX A. REVISION HISTORY	10

CHAPTER 1. INTRODUCTION TO JBOSS DATA GRID 6.1

Welcome to the JBoss Data Grid 6.1. 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 6.1 documentation suite, available at the Red Hat Customer Service Portal's [JBoss Data Grid page](#).

1.1. ABOUT 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.

1.2. OVERVIEW

This document contains information about the new features, as well as the known and resolved issues in JBoss Data Grid version 6.1.0. Customers are requested to read this documentation prior to installing this version.

Customers can provide feedback or log bugs through their Support account. While issues identified will be considered for future releases, we do not guarantee that fixes will be in the next Beta or release.

1.3. NEW FEATURES AND ENHANCEMENTS

1.3.1. Cross Datacenter Replication

This new feature enables backing up of data from one cluster to other clusters, which can be situated in different geographical locations. Available in both Remote Client-Server and Library modes.

1.3.2. Distributed Execution and MapReduce Framework

Distributed execution and MapReduce, which was a Technology Preview feature in version 6.0.1, has been improved and is fully supported in version 6.1.0.

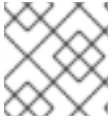
Enhancements include:

- Distributed execution timeout no longer linked to replication timeout.
- MapReduce tasks now have visibility over entries written through to persistent storage.
- Pluggable failover policy for distributed executors.

Available only in Library mode.

1.3.3. Rolling Upgrades for Deployments Accessed Remotely via Hot Rod protocol

This new feature enables you to upgrade your cluster from JBoss Data Grid version 6.1 to the next version without any downtime. Available only for deployments accessed remotely via Hot Rod protocol.

**NOTE**

This feature does not enable a rolling upgrade from version 6.0.1 to 6.1.

1.3.4. Contexts and Dependency Injection (CDI)

The CDI integration in JBoss Data Grid 6.1 allows you to configure and set up your data grid in a manner consistent with the CDI programming model. Further, JBoss Data Grid components can be injected into your CDI beans. This module also allows you to annotate methods in your CDI beans with JSR107 compliant hooks into JBoss Data Grid, to remove boilerplate code.

The CDI module is available in Library mode.

1.3.5. Non-blocking State Transfer

State transfer is the process of adjusting the cache's internal state in response to a cache view change; for example, when a node joins or leaves the cluster. Non-blocking state transfer is a core architectural improvement with the following goals:

- Minimize the interval(s) where the entire cluster can't respond to requests because of a state transfer in progress.
- Minimize the interval(s) where an existing member stops responding to requests because of a state transfer in progress.
- Do not throw any exception during state transfer.

Applies to both Remote Client-Server and Library modes.

1.3.6. Management support for Library mode

A new JBoss Operations Network plugin is available for Library mode.

1.3.7. Querying (Technology Preview)

A Technology Preview of the Query functionality, which enables searching for data stored in the grid using values instead of keys, is available in Library mode.

Features include:

- Keyword, Range, Fuzzy, Wildcard, and Phrase queries.
- Combining queries.
- Sorting, filtering, and pagination of query results.

1.3.8. Command Line Interface (Technology Preview)

A Technology Preview of a command line interface (CLI), which enables you to add or modify data in the grid, is available for both Remote Client-Server and Library modes.

CHAPTER 2. KNOWN ISSUES

The following issues are known to exist in JBoss Data Grid 6.1 and will be fixed in a subsequent release.

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

In 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 behavior persists in JBoss Data Grid 6.1.

BZ#808623 - Some entries not available during view change

In rare circumstances, when a node leaves the cluster, instead of going directly to a new cluster view that displays all nodes save the note that has departed, the cluster splits into two partitions, which then merge after a short amount of time. During this time, some nodes do not have access to all the data that previously existed in the cache. After the merge, all nodes regain access to all the data, but changes made during the split may be lost or be visible only to a part of the cluster.

Normally, when the view changes because a node joins or leaves, the cache data is rebalanced on the new cluster members. However, if the number of nodes that leaves the cluster in quick succession equals or is greater than the value of numOwners, keys for the departed nodes are lost. This occurs during a network split as well - regardless of the reasons for the partitions forming, at least one partition will not have all the data (assuming cluster size is greater than numOwners).

While there are multiple partitions, each one can make changes to the data independently, so a remote client will see inconsistencies in the data. When merging, JBoss Data Grid does not attempt to resolve these inconsistencies, so different nodes may hold different values even after the merge.

This behavior persists in JBoss Data Grid 6.1.

BZ#818863 - Reopened: Tests for UNORDERED and LRU eviction strategies fail on IBM JDK

The LinkedHashMap implementation in IBM's JDK sometimes behaves erratically when extended (as is done by the eviction strategy code). This wrong behavior is exposed by the JBoss Data Grid Test Suite. It is recommended, if using eviction, to use LIRS eviction strategy, which is not affected by this issue. Only LRU eviction strategy is affected.

This behavior persists in JBoss Data Grid 6.1.

BZ#847809 - Cluster with non-shared JDBC cache store has too many entries after node failure

During a node restart, JBoss Data Grid may not start correctly because of duplicate entries in a cache. As a workaround, use a shared cache store instead of local cache stores. Using this workaround, JBoss Data Grid works correctly across restarts and the cache store does not contain duplicate entries.

The root cause of this issue is unknown.

This behavior persists in JBoss Data Grid 6.1.

BZ#854665 - Coordinator tries to install new view after graceful shutdown

In JBoss Data Grid, when gracefully stopping a coordinator node, the node itself would log an attempt to install a new clustering view which would fail. This is harmless, since the new coordinator would in fact perform the proper view installation.

This behavior persists in JBoss Data Grid 6.1.

BZ#881080 - Silence SuspectExceptions

SuspectExceptions are raised when nodes are shutting down, which is normal and expected, since a suspect node is an unresponsive node. As a result, a SuspectException ERROR is written in the logs. There is currently no fix for this issue. The SuspectExceptions will not be logged in a future version of JBoss Data Grid. The SuspectExceptions are harmless and do not affect data integrity.

This behavior persists in JBoss Data Grid 6.1.

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

Using special characters in directory paths can cause problems when starting JBoss Data Grid server. The JBoss Data Grid server either fails to start or a configuration file for logging capabilities cannot be loaded properly. Special characters include spaces, # (hash sign), ! (exclamation mark), % (percentage sign), \$ (dollar sign). This bug causes the JBoss Data Grid server to fail to start properly. Avoiding the use of special characters will stop this problem occurring, allowing JBoss Data Grid server to start properly.

This behavior persists in JBoss Data Grid 6.1.

BZ#888429 - JGroups: SuccessfulResponse not serializable exception on shutdown

A race condition in the shutdown sequence causes an attempt to deserialize an incoming response after the externalizer table, which is used for deserialization of messages, has been stopped. Consequently, a SuccessfulResponse not serializable exception is thrown on shutdown. The issue does not cause data corruption and only happens during shutdown, therefore, it does not affect availability.

This behavior persists in JBoss Data Grid 6.1.

BZ#903308 - IllegalStateException on shutdown

Occasionally, while a coordinator node is being shut down, an asynchronous listener thread might attempt to retrieve the state of the cluster and fail with an IllegalStateException. As a result, the IllegalStateException is logged by the coordinator node while it is shutting down.

The IllegalStateException printed by the coordinator node during shutdown can be ignored. The issue does not affect data integrity in any way, and the node will safely shut down.

This behavior persists in JBoss Data Grid 6.1.

BZ#919034 - JON server plugin, xsite MBean - status shouldn't probably be operation but metric

The JBoss Operations Network Server plug-in exposes a non-functional "status" operation on a remote site. Consequently, invoking the operation returns an error. As a workaround, use the CLI "site --status \${SITE}" command instead.

As a result of this bug, it is not possible to obtain the status of a remote site using the JBoss Operations Network plug-in however, it is possible when using the CLI.

This behavior persists in JBoss Data Grid 6.1.

BZ#919423 - Carmart quickstart has low max-entries for eviction

The readme file for the Carmart example suggests configuring the cache with eviction and max-entries equal to 4. As a consequence, when opening car's details in the application, the resulting page might be empty as some required cache entries had been already evicted from the cache. This problem occurs only in Remote Client-Server mode.

It is recommended that eviction not be used when configuring the quickstart in Remote Client-Server mode (exclude `<eviction/>` element). This will allow all cache entries to remain in the cache and details for all cars to be available to the application.

This behavior persists in JBoss Data Grid 6.1.

CHAPTER 3. RESOLVED ISSUES

The following issues have been resolved in JBoss Data Grid 6.1.

[BZ#862594](#) - Using write-behind store with eviction creates a time window in which entries might not be found

If a cache entry is in the process of being written to the underlying store (somewhere in the async queue) and the entry is being loaded from the store at the same time, the entry was already removed from the container and if not yet written on the store it might not be found. Consequently, cache entries might not be found in a short time window between the moment they are evicted and the moment they are written to the underlying cache store.

This was to have been fixed with a new write-behind cache store implementation. However, the actual fix has not yet been thoroughly tested, and as such this issue is still considered unresolved.

[BZ#881073](#) - Asynchronous FileCacheStore doesn't work in library mode in Tomcat: Error while reading from file

A NullPointerException is thrown when reading objects back from file to a cache. As a result, the Asynchronous FileCacheStore does not work correctly in Tomcat container. There is no workaround for this issue at this stage, and the root cause is unknown.

[BZ#894081](#) - Quickstart problems/improvements

A Maven shade plug-in, which handles creating a resulting jar file for hotrod-endpoint quickstart, is not able to process required dependencies and fails. As a result, the hotrod-endpoint example cannot be built. As a workaround, configure maven-assembly-plugin so that it creates the resulting jar file with all required dependencies. The hotrod-endpoint example can now be built and started.

[BZ#894333](#) - Implement Drop in source migrator for JDG 6.0.x

While JBoss Data Grid 6.1 has introduced support for rolling upgrades via Hot Rod, JBoss Data Grid 6.0 is lacking functionality which would enable a rolling upgrade from that version to a newer version. It is therefore not possible to perform a rolling upgrade from JBoss Data Grid 6.0 to JBoss Data Grid 6.1.

The JBoss Data Grid 6.0 instance should be configured with a persistent cache store so that all states are persisted to it. After shutting down the JBoss Data Grid 6.0 instance, a JBoss Data Grid 6.1 instance, configured to retrieve the persistent state from the same cache store, can be started in its place. An upgrade from JBoss Data Grid 6.0 to JBoss Data Grid 6.1 is not a seamless operation but requires a complete shutdown of the system in order to be performed.

[BZ#902462](#) - ReplaceCommand fails in REST server

An issue in the implementation of the storage of entries written via the REST protocol produces wrong results when the entries are compared. As a result, the replace operation invoked via the REST protocol will fail and the cache will still contain the old entry. There is no workaround available for this issue.

[BZ#904139](#) - JON jdg server plugin - Operations on caches / managers are disabled / unavailable = grey color, not clickable in GUI.

The Infinispan subsystem of JBoss Data Grid Server only exposes metrics and statistics to the JBoss Operations Network plug-in. It does not yet expose the operations which are possible on a cache's components. Therefore, it is not possible to invoke operations on cache containers and caches via the JBoss Operations Network JBoss Data Grid Server plug-in.

There is currently no way to invoke the missing operations via the JBoss Operations Network JBoss Data Grid Server plug-in. However some of these operations can be invoked by connecting to the JBoss Data Grid server via JMX or via the JBoss Data Grid CLI.

[BZ#904140](#) - RHQ library plugin + JON, problems with invoking operations on cache / cache manager

An issue in the JBoss Operations Network JBoss Data Grid library plug-in causes all operations invoked on cache managers/caches to return a NullPointerException. The operation is correctly executed, however an error is always reported by the plug-in.

[BZ#907499](#) - NPE after ReplaceCommand

Occasionally during a conditional replace operation performed via the REST interface, a NullPointerException is thrown. This causes a NullPointerException ERROR to be reported during the replace operation. The actual outcome of the replace operation in this case is undetermined, and there is currently no workaround for this issue.

APPENDIX A. REVISION HISTORY

Revision 1.1-10.400 Rebuild with publican 4.0.0	2013-10-31	Rüdiger Landmann
Revision 1.1-10 Published with new product name.	Mon Aug 05 2013	Misha Husnain Ali
Revision 1.1-9 Updated holder and year entities.	Wed Mar 20 2013	Misha Husnain Ali
Revision 1.1-8 Corrected one typo.	Fri Mar 15 2013	Gemma Sheldon
Revision 1.1-7 Updated with 2 new flagged bugs, 1 requires CCFR information. Removed BZ#906008 as no longer a bug.	Thur Mar 14 2013	Gemma Sheldon
Revision 1.1-6 Removed all references to 6.1 Beta. Re-checked Bugs for inclusion.	Wed Mar 13 2013	Gemma Sheldon
Revision 1.1-5 Removed list of unsupported packages.	Mon Mar 11 2013	Gemma Sheldon
Revision 1.1-4 Changed word "prohibited" to "Unsupported" for unsupported material. Added first run of Bugs for Known and Resolved issues.	Mon Mar 11 2013	Gemma Sheldon
Revision 1.1-3 Added prohibited material.	Fri Mar 08 2013	Misha Husnain Ali
Revision 1.1-2 Added introductory matter.	Fri Jan 11 2013	Misha Husnain Ali
Revision 1.1-1 New build for 6.1.	Tue Sep 25 2012	Misha Husnain Ali