



Red Hat Data Grid 7.3

Red Hat Data Grid 7.3 Release Notes

Release Information for Data Grid

Red Hat Data Grid 7.3 Red Hat Data Grid 7.3 Release Notes

Release Information for Data Grid

Legal Notice

Copyright © 2022 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, the Red Hat 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 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

Red Hat Data Grid is a high-performance, distributed in-memory data store. These Release Notes provide cumulative release information for Data Grid 7.3.x. You can find all the features and enhancements in each Data Grid 7.3 version, locate information about current known issues and resolved issues, review the configurations that Red Hat supports, and component versions in Data Grid 7.3.

Table of Contents

PART I. WHAT'S NEW IN DATA GRID 7.3	5
CHAPTER 1. UPDATES AND ENHANCEMENTS IN 7.3.9	6
1.1. LOG4J VERSION 1.X REMOVED	6
1.2. DATA GRID SERVER UPGRADED TO JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.3.10	6
1.3. JBOSS MARSHALLING UPGRADED	6
CHAPTER 2. UPDATES AND ENHANCEMENTS IN 7.3.8	7
2.1. DATA GRID SERVER UPGRADED TO JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.3.4	7
CHAPTER 3. UPDATES AND ENHANCEMENTS IN 7.3.7	8
3.1. DATA GRID SERVER UPGRADED TO JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.9	8
3.2. ROLLING UPGRADE PERFORMANCE IMPROVEMENTS	8
CHAPTER 4. UPDATES AND ENHANCEMENTS IN 7.3.6	9
4.1. DATA GRID SERVER UPGRADED TO JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.7	9
4.2. SPRING VERSION UPGRADES	9
4.3. ABILITY TO START AND STOP RED HAT DATA GRID ENDPOINTS	9
4.4. HOT ROD ACCESS LOGS INCLUDE TASK NAMES	9
4.5. MULTIPLE IMPROVEMENTS TO CROSS-SITE REPLICATION PERFORMANCE	9
CHAPTER 5. UPDATES AND ENHANCEMENTS IN 7.3.5	10
5.1. CLUSTERED MAXIMUM IDLE EXPIRATION	10
5.2. SOCKET TIMEOUT EXCEPTIONS PROVIDE CONNECTION DETAILS	10
5.3. HOT ROD CLIENT CONNECTION POOL CONFIGURATION	10
5.4. DATA GRID DOCUMENTATION	10
CHAPTER 6. UPDATES AND ENHANCEMENTS IN 7.3.4	11
6.1. DATA GRID SERVER UPGRADED TO JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.5	11
6.2. SPRING BOOT VERSION UPGRADES	11
6.3. EXPIRATION FROM PRIMARY OWNERS WITH CLUSTERED CACHES	11
6.4. EXPIRATION COMMANDS IN CROSS-SITE REPLICATION	11
6.5. RED HAT DATA GRID CLI SESSION EXPIRATION	11
CHAPTER 7. UPDATES AND ENHANCEMENTS IN 7.3.3	12
7.1. DATA GRID SERVER UPGRADED TO JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.4	12
7.2. NATIVE S3 PING FOR AMAZON WEB SERVICES (AWS)	12
7.3. CHANGES TO FLOW CONTROL IN JGROUPS TRANSPORT STACKS	12
7.4. REST ENDPOINT AUTHORIZATION	12
7.5. PROTOSTREAM LIBRARY UPDATED TO 4.2.4.FINAL	13
CHAPTER 8. UPDATES AND ENHANCEMENTS IN 7.3.2	14
8.1. AUTOMATICALLY TAKING SITES OFFLINE WITH ASYNCHRONOUS CROSS-SITE REPLICATION	14
8.2. ASYNCHRONOUS CROSS-SITE REPLICATION STATISTICS VIA JMX	14
8.3. DEPENDENCY ALIGNMENT WITH JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.1	14
8.4. SKIPPING LISTENER NOTIFICATIONS	14
CHAPTER 9. UPDATES AND ENHANCEMENTS IN 7.3.1	15
9.1. JAVA 11	15
9.2. OPENJDK 8 FOR MICROSOFT WINDOWS	15
9.3. JGROUPS UPDATED TO 4.0.18.FINAL	15
9.4. JGROUPS DNS_PING	15
9.5. DATA GRID FOR OPENSIFT CONFIGURABLE LOGGING LEVELS	15
9.6. FILE NAME ANALYZER	15

CHAPTER 10. NEW FEATURES AND ENHANCEMENTS IN 7.3.0	16
10.1. DATA GRID FOR OPENSIFT IMPROVEMENTS	16
10.2. FRAMEWORK INTEGRATION	16
10.2.1. Spring Enhancements	16
10.2.1.1. Synchronized Get Operations	16
10.2.1.2. Asynchronous Operations and Timeout Configuration	16
10.2.1.3. Centralized Configuration Properties for Spring Applications	17
10.2.1.4. Ability to Retrieve Cache Names	17
10.2.1.5. Spring Boot Starter	17
10.2.2. Hibernate Second-level (L2) Caching	18
10.2.3. Integration with Red Hat SSO in Embedded Mode	18
10.3. HOT ROD CLIENT IMPROVEMENTS	18
10.3.1. Support for Transactions	18
10.3.2. Java Hot Rod Client Statistics in JMX	18
10.3.3. Java Hot Rod Client Configuration Enhancements	19
10.3.4. New Java Hot Rod Client Implementation Based on Netty	19
10.3.5. Javascript Hot Rod Client Support for JSON Objects	19
10.3.6. Retrieving Cache Names Through Hot Rod	20
10.4. IMPROVEMENTS TO PERSISTENCE	20
10.4.1. Fault Tolerance for Write-Behind Cache Stores	20
10.5. DATA INTEROPERABILITY	21
10.6. DEFAULT ANALYZERS	21
10.7. METRICS IMPROVEMENTS	21
10.8. LOCKED STREAMS	22
10.9. IMPROVEMENTS TO CONFIGURATION	22
10.9.1. Configuration Wildcards	22
10.9.2. Immutable Configuration	22
10.10. IMPROVED SECURITY FOR CACHE OPERATIONS	22
10.11. HTTP/2 SUPPORT	22
CHAPTER 11. TECHNOLOGY PREVIEWS IN DATA GRID 7.3	23
11.1. CROSS-SITE REPLICATION ON RED HAT OPENSIFT	23
11.2. REMOTE ADMINISTRATION WITH C++ AND C# CLIENTS	24
11.3. CLUSTERED COUNTERS WITH C++ AND C# CLIENTS	24
11.4. ADMINISTRATION CONSOLE CAPABILITIES	24
CHAPTER 12. FEATURES AND FUNCTIONALITY DEPRECATED IN DATA GRID 7.3	25
PART II. SUPPORTED CONFIGURATIONS AND COMPONENT VERSIONS	27
CHAPTER 13. SUPPORTED CONFIGURATIONS FOR DATA GRID 7.3	28
CHAPTER 14. DATA GRID COMPONENT VERSIONS	29
PART III. KNOWN AND FIXED ISSUES	30
CHAPTER 15. KNOWN ISSUES FOR DATA GRID 7.3	31
CHAPTER 16. FIXED ISSUES	35
16.1. FIXED IN DATA GRID 7.3.9	35
16.2. FIXED IN DATA GRID 7.3.8	35
16.3. FIXED IN DATA GRID 7.3.7	35
16.4. FIXED IN DATA GRID 7.3.6	35
16.5. FIXED IN DATA GRID 7.3.5	36
16.6. FIXED IN DATA GRID 7.3.4	36

16.7. FIXED IN DATA GRID 7.3.3	37
16.8. FIXED IN DATA GRID 7.3.2	37
16.9. FIXED IN DATA GRID 7.3.1	37
16.10. FIXED IN DATA GRID 7.3.0	38
PART IV. MIGRATING TO DATA GRID 7.3	39
CHAPTER 17. CHANGES IN DATA GRID 7.3	40
17.1. JBOSS ENTERPRISE APPLICATION PLATFORM (EAP)	40
17.2. EAP MODULES	40
17.3. CAMEL COMPONENT FOR JBOSS FUSE	40
17.4. CACHE STORE COMPATIBILITY	40
17.5. MEMCACHED STORAGE	40
17.6. MEMCACHED CONNECTOR	41
17.7. SCRIPTS RESPONSE	41
17.8. SERVER THREAD POOLS	41
17.9. SHARED CACHE STORES WITH PASSIVATION	41
17.10. DEFAULT SHARDS WITH AFFINITYINDEXMANAGER	41
17.11. ADVANCEDCACHELOADER CHANGES	41
17.12. DEPRECATIONS IN DATA GRID 7.3	41
PART V. PATCHING DATA GRID SERVERS	42
CHAPTER 18. APPLYING PATCHES TO RED HAT DATA GRID	43
Resolving Errors with the 7.3.8 and 7.3.9 Patch	43
CHAPTER 19. REVERTING PATCHES	45

PART I. WHAT'S NEW IN DATA GRID 7.3

CHAPTER 1. UPDATES AND ENHANCEMENTS IN 7.3.9

1.1. LOG4J VERSION 1.X REMOVED

Data Grid 7.3.9 removes Log4j version 1.x components to address a critical security vulnerability as well as other CVEs of moderate severity. You must upgrade any Data Grid 7.3 deployments to version 7.3.9 as soon as possible.

Any custom components that rely on Log4j version 1.x capabilities, such as a cache loader or a server task implementation, no longer work because Data Grid 7.3.9 does not ship with Log4j version 1.x.

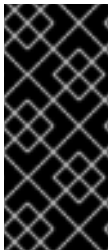
Red Hat recommends you upgrade any deployments from 7.3.x to the latest Data Grid 8 version as soon as possible. The Data Grid team regularly patch security vulnerabilities and actively fix issues on the latest version of the software.

Find the latest Data Grid documentation at: [Data Grid product documentation](#)

For more information about Data Grid version lifecycle and support details, see [Data Grid Product Update and Support Policy](#)

1.2. DATA GRID SERVER UPGRADED TO JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.3.10

Data Grid Server is upgraded to EAP 7.3.10, which provides performance improvements and includes several CVE fixes to enhance security.



IMPORTANT

When patching Data Grid server installations to upgrade to 7.3.9, an issue with the patch results in an error at startup. You must manually edit the configuration after you apply the patch.

See [Resolving Errors with the 7.3.8 and 7.3.9 Patch](#) for instructions.

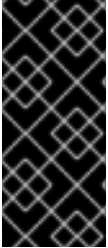
1.3. JBOSS MARSHALLING UPGRADED

Data Grid 7.3.9 upgrades JBoss marshalling to version 2.0.12.Final.

CHAPTER 2. UPDATES AND ENHANCEMENTS IN 7.3.8

2.1. DATA GRID SERVER UPGRADED TO JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.3.4

Data Grid Server is upgraded to EAP 7.3.4, which provides performance improvements and includes several CVE fixes to enhance security.



IMPORTANT

When patching Data Grid server installations to upgrade to 7.3.8, an issue with the patch results in an error at startup. You must manually edit the configuration after you apply the patch.

See [Resolving Errors with the 7.3.8 and 7.3.9 Patch](#) for instructions.

CHAPTER 3. UPDATES AND ENHANCEMENTS IN 7.3.7

3.1. DATA GRID SERVER UPGRADED TO JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.9

Data Grid Server is upgraded to EAP 7.2.9, which provides performance improvements and includes several CVE fixes to enhance security.

3.2. ROLLING UPGRADE PERFORMANCE IMPROVEMENTS

When performing rolling upgrades of Data Grid clusters, target clusters more efficiently retrieve data from source clusters.

CHAPTER 4. UPDATES AND ENHANCEMENTS IN 7.3.6

4.1. DATA GRID SERVER UPGRADED TO JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.7

Data Grid server is upgraded to EAP 7.2.7, which provides performance improvements and includes several CVE fixes to enhance security.

4.2. SPRING VERSION UPGRADES

This release of Data Grid upgrades to Spring versions as follows:

- Spring Boot upgraded to **2.2.5**.
- Spring Session upgraded to **2.2.2**.

4.3. ABILITY TO START AND STOP RED HAT DATA GRID ENDPOINTS

The Red Hat Data Grid Command-Line Interface (CLI) provides the following commands to start and stop Red Hat Data Grid endpoint connectors:

- **:stop-connector**
- **:start-connector**

See [Starting and Stopping Red Hat Data Grid Endpoints](#) for more information.

4.4. HOT ROD ACCESS LOGS INCLUDE TASK NAMES

When executing tasks on Red Hat Data Grid servers via remote Hot Rod clients, task names are now included in the Hot Rod access logs.

See [Access Logs](#) for more information about Hot Rod access logs.

4.5. MULTIPLE IMPROVEMENTS TO CROSS-SITE REPLICATION PERFORMANCE

This release includes several improvements to cross-site replication performance. Red Hat Data Grid now:

- Handles asynchronous replication requests in parallel.
- Uses thread pools for handling asynchronous replication requests.
- Sends operations to backup data across sites from blocking threads.

CHAPTER 5. UPDATES AND ENHANCEMENTS IN 7.3.5

5.1. CLUSTERED MAXIMUM IDLE EXPIRATION

Clustered **max-idle** expiration is enhanced in this release to prevent data loss with clustered cache modes.

As of this release, when clients read entries that have **max-idle** expiration values, Red Hat Data Grid sends touch commands to all owners. This ensures that the entries have the same relative access time across the cluster.

See [Maximum Idle Expiration](#).

5.2. SOCKET TIMEOUT EXCEPTIONS PROVIDE CONNECTION DETAILS

As of this release, when Hot Rod client connections time out, Red Hat Data Grid provide additional detail about connections in exception messages, as in the following example:

Socket timeout exceptions before this release

```
org.infinispan.client.hotrod.exceptions.TransportException::
java.net.SocketTimeoutException: ContainsKeyOperation{ExpirationCache,
key=[B0x033E0131, flags=0] timed out after 60000 ms
```

Socket timeout exceptions as of this release

```
org.infinispan.client.hotrod.exceptions.TransportException::
java.net.SocketTimeoutException: ContainsKeyOperation{ExpirationCache,
key=[B0x033E0131, flags=0, connection=127.0.0.1/127.0.0.1:11222} timed out
after 60000 ms
```

5.3. HOT ROD CLIENT CONNECTION POOL CONFIGURATION

As of this release, Data Grid Hot Rod C++ clients allow you to configure the recover policy when the connection pool is exhausted.

- **configurationBuilder.connectionPool().exhaustedAction(WAIT)** waits indefinitely for a connection to be returned and made available.
- **configurationBuilder.connectionPool().exhaustedAction(EXCEPTION)** throws **NoSuchElementException**.

5.4. DATA GRID DOCUMENTATION

Data Grid server documentation was updated to describe how to collect JMX statistics with Prometheus.

See [Exposing JMX Beans to Prometheus](#).

The Data Grid User Guide was updated to clarify capacity factor configuration for cross-site replication.

See "Capacity Factors" in [Key Ownership](#).

CHAPTER 6. UPDATES AND ENHANCEMENTS IN 7.3.4

6.1. DATA GRID SERVER UPGRADED TO JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.5

Data Grid server is upgraded to EAP 7.2.5, which provides performance improvements and includes several CVE fixes to enhance security.

6.2. SPRING BOOT VERSION UPGRADES

This release of Data Grid upgrades to Spring Boot 2.2.0, which uses Spring 5 dependencies as follows:

- Spring 5 upgraded to **5.2.0**.
- Spring Session upgraded to **2.2.0**.
- Spring Security upgraded to **5.2.0**.

6.3. EXPIRATION FROM PRIMARY OWNERS WITH CLUSTERED CACHES

Data Grid expires entries from clustered caches only when the primary owners determines the entries meet the expiration criteria. This change improves performance by reducing duplicate expiration commands.

6.4. EXPIRATION COMMANDS IN CROSS-SITE REPLICATION

When expiring entries, Red Hat Data Grid replicates a command that removes expired entries across clusters. As of this release Red Hat Data Grid no longer replicates this command across sites, which improves performance.

6.5. RED HAT DATA GRID CLI SESSION EXPIRATION

Documentation is added to note the fact that CLI sessions expire after an idle timeout of six minutes. See the [Command-Line Interface \(CLI\)](#) section for more information.

CHAPTER 7. UPDATES AND ENHANCEMENTS IN 7.3.3

7.1. DATA GRID SERVER UPGRADED TO JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.4

Data Grid server is upgraded to EAP 7.2.4, which provides performance improvements and includes several CVE fixes to enhance security.

7.2. NATIVE S3 PING FOR AMAZON WEB SERVICES (AWS)

Changes to the AWS S3 API require Data Grid to use **NATIVE_S3_PING** instead of the JGroups **S3_PING** protocol.

JGroups **S3_PING** no longer works for server discovery on AWS. You should migrate any existing **S3_PING** configurations to **NATIVE_S3_PING**.

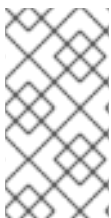
For more information, see the following:

- [Amazon Web Services](#) in the Red Hat Data Grid User Guide.
- [JGroups Subsystem Configuration](#) in the Red Hat Data Grid Server Guide.
- [Amazon S3 Update on SigV2 deprecation](#)

7.3. CHANGES TO FLOW CONTROL IN JGROUPS TRANSPORT STACKS

This release includes changes that improve transport layer performance for Data Grid clusters in Library Mode:

- Data Grid adds JGroups Unicast Flow Control (**UFC**) to default **TCP** transport stacks. **UFC** prevents nodes from sending asynchronous messages to other nodes when the JGroups thread pool for the receiving nodes reaches maximum capacity. When this occurs, receiving nodes start discarding messages but sending nodes keep re-sending them.
- The value for the **max_credits** attribute for **UFC** and **MFC** increases to **3m** from the JGroups default of **2m**. This attribute sets the maximum number of bytes that one node can send to another without an acknowledgment from the receiving node.



NOTE

- **UFC** is already included in the default **UDP** transport stack for Library Mode.
- In Server Mode, Data Grid uses non-blocking flow control (**UFC_NB** and **MFC_NB**) as default.

7.4. REST ENDPOINT AUTHORIZATION

The Data Grid REST endpoint now allows access to caches that are configured for authorization.

This enhancement resolves the issues outlined in this [KCS Solution](#).

7.5. PROTOSTREAM LIBRARY UPDATED TO 4.2.4.FINAL

The protostream library component in Data Grid is updated to 4.2.4.Final.

CHAPTER 8. UPDATES AND ENHANCEMENTS IN 7.3.2

8.1. AUTOMATICALLY TAKING SITES OFFLINE WITH ASYNCHRONOUS CROSS-SITE REPLICATION

Data Grid now applies the **take-offline** configuration when using Cross-Site Replication capabilities with asynchronous replication strategies.

8.2. ASYNCHRONOUS CROSS-SITE REPLICATION STATISTICS VIA JMX

Data Grid adds the following statistics for asynchronous cross-site replication:

- **AsyncXSiteAcksCount**
- **AverageAsyncXSiteReplicationTime**
- **MaximumAsyncXSiteReplicationTime**
- **MinimumAsyncXSiteReplicationTime**

See the [JMX Components](#) documentation for more information.

8.3. DEPENDENCY ALIGNMENT WITH JBOSS ENTERPRISE APPLICATION PLATFORM (EAP) 7.2.1

Core module dependencies for Data Grid 7.3.2 in Server Mode align with EAP 7.2.1.

8.4. SKIPPING LISTENER NOTIFICATIONS

The Hot Rod client now includes a **SKIP_LISTENER_NOTIFICATION** flag so that client listeners do not get notified by the Data Grid server when session IDs change.

This flag resolves issues when using **spring-session** integration with Spring 5. If you are using spring-session with Spring 5, you should upgrade both the Data Grid server and Hot Rod client to 7.3.2.

Likewise, you must upgrade both the Data Grid server and Hot Rod client to 7.3.2 or later before you can set the **SKIP_LISTENER_NOTIFICATION** flag.

For more information, see [Skipping Notifications](#) in the Data Grid User Guide.

CHAPTER 9. UPDATES AND ENHANCEMENTS IN 7.3.1

9.1. JAVA 11

Data Grid now supports Java 11. See the supported configurations at <https://access.redhat.com/articles/2435931>.



NOTE

Data Grid for OpenShift release 7.3.1 is built with Java 8. Data Grid for OpenShift supports Java 11 in version 7.3.2. See [Data Grid for OpenShift Versions](#).

9.2. OPENJDK 8 FOR MICROSOFT WINDOWS

Data Grid now supports OpenJDK 8 on Windows. See the supported configurations at <https://access.redhat.com/articles/2435931>.

9.3. JGROUPS UPDATED TO 4.0.18.FINAL

Data Grid now uses JGroups version 4.0.18.Final. See the component details at <https://access.redhat.com/articles/488833>.

9.4. JGROUPS DNS_PING

Data Grid can now use the JGroups **DNS_PING** protocol for cluster discovery.



NOTE

The Data Grid for OpenShift image uses **openshift.DNS_PING** which provides the same functionality as JGroups **DNS_PING**. By default, you cannot enable JGroups **DNS_PING** with the Data Grid for OpenShift image. However, you can build custom images from the Data Grid for OpenShift image with custom configuration that uses JGroups **DNS_PING**. You can also use JGroups **DNS_PING** if you embed Data Grid in custom OpenShift applications.

For more information, see [JGroups Subsystem Configuration](#).

9.5. DATA GRID FOR OPENSIFT CONFIGURABLE LOGGING LEVELS

This release adds the **LOGGING_CATEGORIES** environment variable that adjusts the categories and levels for which Data Grid captures log messages. See [Monitoring and Logging](#) for more information.

9.6. FILE NAME ANALYZER

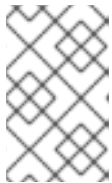
Data Grid now provides a default **filename** analyzer. For more information, see [Default Analyzers](#).

CHAPTER 10. NEW FEATURES AND ENHANCEMENTS IN 7.3.0

10.1. DATA GRID FOR OPENSIFT IMPROVEMENTS

This release includes several improvements for Data Grid for OpenShift, including:

- Full support for the Cache service (**cache-service**).
- A Data Grid service (**datagrid-service**) that provides a full distribution of Data Grid for OpenShift.
- Enhancements to the Data Grid for OpenShift image.
- Monitoring capabilities through integration with Prometheus.
- Library Mode support that allows you to embed Data Grid in containerized applications running on OpenShift. Limitations apply. See [Data Grid for OpenShift](#) for more information.



NOTE

Red Hat does not recommend embedding Data Grid in custom server applications. If you want Data Grid to handle caching requests from client applications, deploy Data Grid for OpenShift.

Visit the [Data Grid for OpenShift Documentation](#) to find out more and get started.

10.2. FRAMEWORK INTEGRATION

This release of Data Grid improves integration with well-known enterprise Java frameworks such as Spring and Hibernate.

10.2.1. Spring Enhancements

This release adds several enhancements to Data Grid integration with the Spring Framework:



NOTE

This release of Data Grid supports specific versions of Spring Framework and Spring Boot. See the supported configurations at <https://access.redhat.com/articles/2435931>.

10.2.1.1. Synchronized Get Operations

Data Grid implements [SPR-9254](#) so that the **get()** method uses a multi-threaded mechanism for returning key values.

For more information, see the description for [get](#) in the **org.springframework.cache** interface.

10.2.1.2. Asynchronous Operations and Timeout Configuration

You can now set a maximum time to wait for read and write operations when using Data Grid as a Spring cache provider. The timeout allows method calls to happen asynchronously.

Consider the differences before and after timeouts with the following **put()** method examples:

Before Write Timeouts

```
public void put(Object key, Object value, long lifespan, TimeUnit unit) {
    this.cacheImplementation.put(key, value != null ? value : NullValue.NULL, lifespan, unit);
}
```

After Write Timeouts

```
public void put(final Object key, final Object value) {
    this.nativeCache.put(key, value);
    try {
        if (writeTimeout > 0)
            this.nativeCache.putAsync(key, value != null ? value : NullValue.NULL).get(writeTimeout,
TimeUnit.MILLISECONDS);
        else
            this.nativeCache.put(key, value != null ? value : NullValue.NULL);
    } catch (InterruptedException e) {
        Thread.currentThread().interrupt();
        throw new CacheException(e);
    } catch (ExecutionException | TimeoutException e) {
        throw new CacheException(e);
    }
}
```

If you configure a timeout for write operations, **putAsync** is called, which is a "fire-and-forget" call that does not block other writes.

If you do not configure a timeout, a synchronous **put** is called, which blocks other writes.

Set timeout configuration with **infinispan.spring.operation.read.timeout** and **infinispan.spring.operation.write.timeout**. Read the [Configuring Timeouts](#) in the documentation to learn how.

10.2.1.3. Centralized Configuration Properties for Spring Applications

If you are using Data Grid as a Spring Cache Provider in remote client-server mode, you can set configuration properties in *hotrod-client.properties* on your classpath. Your application can then create a **RemoteCacheManager** with that configuration.

Information about available configuration properties is available in the [org.infinispan.client.hotrod.configuration Package Description](#).

10.2.1.4. Ability to Retrieve Cache Names

The **RemoteCacheManager** class now includes a **getCacheNames()** method that returns cache names as a JSON array of strings, for example, **["cache1", "cache2"]**. This method is included in the **org.springframework.cache.CacheManager** implementation so that you can look up defined cache names when using Data Grid as a Spring cache provider.

Find out more in the Javadocs for [RemoteCacheManager](#).

10.2.1.5. Spring Boot Starter

Data Grid includes a Spring Boot starter to help you quickly get up and running. See [Data Grid Spring Boot Starter](#).

10.2.2. Hibernate Second-level (L2) Caching

Data Grid seamlessly integrates with Hibernate as an (L2) cache provider to improve the performance of your application's persistence layer.

Hibernate provides Object/Relational Mapping (ORM) capabilities for Java and is a fully compliant JPA (Java Persistence API) persistence provider. Hibernate uses first-level (L1) caching where objects in the cache are bound to sessions. As an L2 cache provider, Data Grid acts as a global cache for objects across all sessions.

You can configure Data Grid as the L2 cache in:

- JPA: *persistence.xml*
- Spring: *application.properties*

For complete information on enabling L2 cache, along with information for different deployment scenarios, see [JPA/Hibernate L2 Cache](#) in the documentation.

10.2.3. Integration with Red Hat SSO in Embedded Mode

This release provides support for using Red Hat SSO to secure access to Data Grid in *Library (embedded) Mode*.

See the [secure-embedded-cache](#) quickstart for more information and to deploy and run a sample application that demonstrates integration with Red Hat SSO.

10.3. HOT ROD CLIENT IMPROVEMENTS

10.3.1. Support for Transactions

Java, C++, and C# Hot Rod clients can now start and participate in transactions.

The Java Hot Rod client supports both **FULL_XA** and **NON_XA** transaction modes. C++ and C# Hot Rod clients provide support for **NON_XA** transaction modes only.

For more information, see [Hot Rod Transactions](#).

10.3.2. Java Hot Rod Client Statistics in JMX

The **ServerStatistics** interface now exposes statistics for the Java Hot Rod client through JMX.

You must enable JMX statistics in your Hot Rod client implementation, as in the following example:

```
ConfigurationBuilder builder = new ConfigurationBuilder();
builder.addServer()
    .host("127.0.0.1")
    .port(11222)
    .statistics()
    .enable()
    .jmxEnable();
```

- **enabled()** lets you collect client-side statistics.
- **jmxEnabled()** exposes statistics through JMX.

For more information, see [ServerStatistics](#) in the Javadocs.

10.3.3. Java Hot Rod Client Configuration Enhancements

This release improves configuration for the Java Hot Rod client with the **hotrod-client.properties** file. You can configure near cache settings, cross-site (xsite) properties, settings to control authentication and encryption, and more.

For more information, see the [Hot Rod client configuration summary](#).

10.3.4. New Java Hot Rod Client Implementation Based on Netty

The Hot Rod Java client is built with the Netty framework, providing improved performance and support for executing operations concurrently over the same connection.

In previous releases, application-executed operations were done synchronously or delegated to dedicated thread pools.

As of this release, operations are executed asynchronously and the response is processed in the **HotRod-client-async-pool** thread pool. Operations can be multiplexed over the same connection, which requires fewer connections.



NOTE

Custommarshallers must not rely on any particular thread calling them to unmarshall data.

For more information, see the Javadocs:

- [org.infinispan.client.hotrod.configuration.ConnectionPoolConfigurationBuilder](#)
- [org.infinispan.client.hotrod.configuration.Configuration](#)
- [org.infinispan.client.hotrod.configuration.ConfigurationBuilder](#)
- [ConfigurationChildBuilder](#)

10.3.5. Javascript Hot Rod Client Support for JSON Objects

The node.js Hot Rod client adds support for native **JSON** objects as keys and values. In previous releases, the client supported **String** keys and values only.

To use native **JSON** objects, you must configure the client as follows:

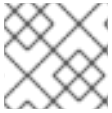
```
var infinispan = require('infinispan');
var connected = infinispan.client(
  {port: 11222, host: '127.0.0.1'}
  , {
    dataFormat : {
      keyType: 'application/json',
      valueType: 'application/json'
    }
  }
);
```

```

    }
  }
);
connected.then(function (client) {
  var clientPut = client.put({k: 'key'}, {v: 'value'});
  var clientGet = clientPut.then(
    function() { return client.get({k: 'key'}); });
  var showGet = clientGet.then(
    function(value) { console.log("get({k: 'key'})=" + JSON.stringify(value)); });
  return showGet.finally(
    function() { return client.disconnect(); });
}).catch(function(error) {
  console.log("Got error: " + error.message);
});

```

You can configure data types for keys and values separately. For example, you can configure keys as **String** and values as **JSON**.



NOTE

Scripts do not currently support native **JSON** objects.

10.3.6. Retrieving Cache Names Through Hot Rod

This release includes the **getCacheNames()** method that returns a collection of caches names that have been defined declaratively or programmatically as well as the caches that have been created at runtime through **RemoteCacheManager**.

The Hot Rod protocol also now includes a **@@cache@names** admin task that returns cache names as a JSON array of strings, for example, ["cache1", "cache2"].

For more information, see:

- [Hot Rod Admin Tasks](#)
- [RemoteCacheManager](#)

10.4. IMPROVEMENTS TO PERSISTENCE

10.4.1. Fault Tolerance for Write-Behind Cache Stores

Data Grid now lets you configure write-behind cache stores so that, in the event a write-behind operation fails, additional operations on the cache are not allowed. Additionally, modifications that failed to write to the cache store are queued until the underlying cache store becomes available.

You can configure fault tolerance with the **connection-attempts**, **connection-interval**, and **fail-silently** declaratively.

To configure fault tolerance programmatically, use the following:

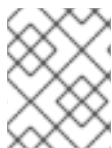
- **connectionAttempts()** and **connectionInterval()** methods in the **PersistenceConfiguration** class.
- **failSilently()** method in the **AsyncStoreConfiguration** class.

For more information, see the documentation:

- [Write-Through And Write-Behind Caching](#)
- [org.infinispan.configuration.cache.AsyncStoreConfiguration](#)
- [org.infinispan.configuration.cache.PersistenceConfiguration](#)
- [org.infinispan.notifications.cachelistener.annotation.PersistenceAvailabilityChanged](#)
- [Configuration Schema](#)

10.5. DATA INTEROPERABILITY

Data Grid provides transcoding capabilities that convert data between formats that is suitable for different endpoints. For example, you can write ProtoBuf-encoded data through the Hot Rod endpoint and then retrieve it as a JSON document through the REST endpoint.



NOTE

Compatibility mode is now deprecated and will be removed from Data Grid. You should use protocol interoperability capabilities instead.

10.6. DEFAULT ANALYZERS

Data Grid includes a set of default analyzers that convert input data into one or more terms that you can index and query.

For more information, see [Analysis](#).

10.7. METRICS IMPROVEMENTS

This release exposes new operations and metrics through JMX:

ClusteredLockManager component

- **forceRelease** forces locks to be released.
- **isDefined** returns **true** if the lock is defined.
- **isLocked** returns **true** if the lock exists and is acquired.
- **remove** removes locks from clusters. Locks must be recreated to access again.

Passivation component

- **passivateAll** passivates all entries to the **CacheStore**.

CacheStore component

- **NumberOfPersistedEntries** returns the number of entries currently persisted, excluding expired entries.

For more information, see [jmxComponents](#).

10.8. LOCKED STREAMS

The **invokeAll()** method in the **LockedStream** interface now provides a way to execute code for entries where locks are held for the respective keys, allowing you to not have to acquire locks and be able to guarantee the state for values.

For more information, see [org.infinispan.LockedStream](#).

10.9. IMPROVEMENTS TO CONFIGURATION

10.9.1. Configuration Wildcards

You can now use wildcards in configuration template names so that Data Grid applies the template to any matching caches.

For more information, see [Cache configuration wildcards](#).

10.9.2. Immutable Configuration

You can now create immutable configuration storage providers to prevent the creation or removal of caches.

Use the **immutable-configuration-storage** parameter when configuring Data Grid declaratively or programmatically use the **IMMUTABLE** configuration store in the global state.

- [Obtaining Caches](#)
- [org.infinispan.configuration.global.GlobalStateConfiguration](#)
- [Configuration Schema](#)

10.10. IMPROVED SECURITY FOR CACHE OPERATIONS

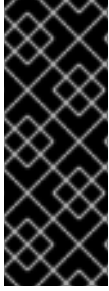
The **AdvancedCache** interface now includes a **withSubject()** method that performs operations with the specified subject when authorization is enabled on caches.

See the Javadocs for the [AdvancedCache](#) interface.

10.11. HTTP/2 SUPPORT

Data Grid now provides support for HTTP/2 with the REST endpoint.

CHAPTER 11. TECHNOLOGY PREVIEWS IN DATA GRID 7.3



IMPORTANT

Technology Preview features or capabilities are not supported with Red Hat production service-level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them for production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information see [Red Hat Technology Preview Features Support Scope](#).

11.1. CROSS-SITE REPLICATION ON RED HAT OPENSIFT

Red Hat Data Grid for OpenShift gives you cross-site replication capabilities to backup data across clusters in running different data centers.

Supported Architecture and Capabilities

Cross-Site replication on Red Hat OpenShift currently is supported with the following architecture:

- Data Grid deployed as a **StatefulSet** with the `datagrid-service` service template.
- Single master site.
- Single backup site.
- **NodePort** service that exposes a port for Data Grid nodes to communicate and perform cross-site replication via the JGroups **RELAY2** protocol.
- Data Grid does not provide any controls for concurrent modifications in different sites. You must configure your applications to handle concurrency as required or implement avoidance strategies.
- Data Grid JMX components for monitoring and administration. See the [JMX Components](#).

Currently Unsupported Features and Capabilities

Data Grid does not currently support the following features and capabilities with cross-site replication on Red Hat OpenShift:

- Schema replication.
- Data indexing.
- Dynamic cache creation.

Resources and Documentation

[standalone.xml for cross-site replication](#) provides a recommended configuration for Red Hat Data Grid for OpenShift.

[Cross-Site Replication: Red Hat Data Grid for OpenShift](#) on GitHub provides documentation and a quickstart tutorial.

[Cross-Site Replication](#) documentation provides additional detail, including procedures for transferring state from one site to another.

11.2. REMOTE ADMINISTRATION WITH C++ AND C# CLIENTS

C++ and C# Hot Rod clients now offer an implementation of the **RemoteCacheManagerAdmin** interface that lets you perform administrative operations remotely.

11.3. CLUSTERED COUNTERS WITH C++ AND C# CLIENTS

C++ and C# Hot Rod clients now provide capabilities for remotely working with clustered counters.

11.4. ADMINISTRATION CONSOLE CAPABILITIES

The Data Grid administration console now lets you configure and manage endpoint configuration and provides the following capabilities for manipulating data:

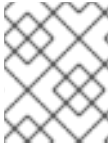
- Querying data
- Creating and updating cache entries
- Deleting cache entries

Additionally, the administration console includes a basic protobuf schema editor.

CHAPTER 12. FEATURES AND FUNCTIONALITY DEPRECATED IN DATA GRID 7.3

This release of Data Grid deprecates various features and functional components.

Deprecated functionality continues to be supported until Data Grid 7.x end of life. Deprecated functionality will not be supported in future major releases and is not recommended for new deployments.



NOTE

This information is current as of the time of writing. You can review all the deprecated code for the next major release with [JDG-1978](#).

Hot Rod v1.x

Hot Rod protocol version 1.x will not be supported in the next major release.

RocksDB Replaces LevelDB Cache Store

As of this release, the LevelDB cache store is deprecated and replaced with the RocksDB cache store. If you have data stored in a LevelDB cache store, the RocksDB cache store converts it to the SST-based format on the first run.

RocksDB provides superior performance and reliability, especially in highly concurrent scenarios. Find out more about the [RocksDB cache store](#) in the documentation.

Compatibility Mode

Compatibility mode is deprecated and will be removed in the next major release. To access a cache from multiple endpoints, you should store data in binary format and configure the **MediaType** for keys and values. See the following topics for more information:

- [Protocol Interoperability](#)
- [MediaType](#)

If you want to store data as unmarshalled objects, you should configure keys and values to store object content as follows:

```
<encoding>
  <key media-type="application/x-java-object"/>
  <value media-type="application/x-java-object"/>
</encoding>
```

Clustered Executor Replaces Distributed Executor API

Data Grid replaces Distributed Executor with Clustered Executor, which is a utility for executing arbitrary code in the cluster. See:

- [Cluster Executor](#)
- [org.infinispan.manager.ClusterExecutor](#)

RemoteCache getBulk()

The **getBulk()** method is deprecated in the **RemoteCache** interface. See: [org.infinispan.client.hotrod.RemoteCache](#).

Agroal PooledConnectionFactory Replaces c3p0/HikariCP JDBC PooledConnectionFactory

The JDBC **PooledConnectionFactory** provides connection pools that you configure with **c3p0.properties** and **hikari.properties**. In the next major release, Data Grid provides a **PooledConnectionFactory** that you configure only with an agroal compatible properties file. See [Agroal project](#).

CLI Loader

infinispan-persistence-cli is now deprecated and will be removed in the next major version.

Deprecated Classes

- **org.infinispan.lifecycle.AbstractModuleLifecycle**
- **org.infinispan.lifecycle.Lifecycle**

Eager Near Caching Residual Code

Residual code artifacts for eager near caching functionality will be removed in the next major version.

PART II. SUPPORTED CONFIGURATIONS AND COMPONENT VERSIONS

CHAPTER 13. SUPPORTED CONFIGURATIONS FOR DATA GRID 7.3

Red Hat supports specific hardware and software combinations for Red Hat Data Grid 7.3, available at <https://access.redhat.com/articles/2435931>.

CHAPTER 14. DATA GRID COMPONENT VERSIONS

Component details for Red Hat Data Grid 7.3 are available on the [Customer Portal](#).

You can also find Data Grid component versions from your project as follows:

1. Download the archive that contains the Data Grid sources, **jboss-datagrid-`{version}`-sources.zip**, from the [Product Downloads](#) page.
2. Extract the archive to your file system.
3. Open a terminal window and change to the top-level directory that contains the **pom.xml** file.
4. Run **mvn dependency:tree** to retrieve information about dependencies.

PART III. KNOWN AND FIXED ISSUES

CHAPTER 15. KNOWN ISSUES FOR DATA GRID 7.3

Hot Rod Java Client Security Exception on Java 8 when Using Kerberos Authentication

Issue: [JDG-4224](#)

Description: The following exception is thrown when using the Hot Rod Java client on Java 8 with GSSAPI security mechanisms:

```
org.infinispan.client.hotrod.exceptions.HotRodClientException: javax.security.sasl.SaslException:
ELY05123: No security layer selected but message length received
```

This issue occurs due to stricter requirements for Kerberos authentication with the Elytron subsystem in EAP 7.3.

Workaround: Add the **wildfly.sasl.relax-compliance** property to the **sasl** security realm in your Data Grid Server configuration:

```
<authentication security-realm="ApplicationRealm">
  <sasl server-context-name="hotrod-service" server-name="node0" mechanisms="GSSAPI"
  qop="auth" strength="high medium low">
    <policy>
      <no-anonymous value="true" />
    </policy>
    <property name="wildfly.sasl.relax-compliance">true</property>
  </sasl>
</authentication>
```

Internal Serialization Library Prevents Conversion from JSON to Java Object

Issue: [JDG-3965](#)

Description: If you attempt to store data as unmarshalled, Plain Old Java Objects (POJO) on Data Grid Server, and then read and write data in JSON format, the following exception occurs:

```
com.fasterxml.jackson.databind.exc.InvalidDefinitionException: Illegal type
(com.example.MyClass) to deserialize: prevented for security reasons
```

Workaround: Specify user classes with the following system property with Data Grid Server:

```
-Djackson.deserialization.whitelist.packages=com.example
```

Specify fully qualified class names as follows:

```
-Djackson.deserialization.whitelist.packages=com.example.MyClass
```

Outdated Logging Configuration on Some Versions of the OpenShift ConfigMap Quickstart

Issue: [JDG-4024](#)

Description: The Data Grid **ConfigMap** quickstart, [_Customizing Data Grid Service Deployments](#), does not work as expected for versions 7.3.4, 7.3.5, and 7.3.6.

Data Grid for OpenShift 7.3.4 and later uses an updated logging formatter which is not compatible with the logging formatter in the **ConfigMap** quickstart. As a result it is not possible to start customized Data Grid server images with the quickstart.

Workaround: Use the 7.3.7 tags for the quickstart or work with the **7.3.x** branch.

MySQL and PostgreSQL cache store drivers not available with the Data Grid for OpenShift image for OpenShift Container Platform on IBM Z or IBM Power

Issue: [JDG-3376](#)

Description: The Data Grid for OpenShift image for OpenShift Container Platform on IBM Z or IBM Power does not provide drivers for MySQL and PostgreSQL cache stores.

Workaround: There is no workaround for this issue.

Connection error events from probes with the Data Grid for OpenShift image for OpenShift Container Platform on IBM Z or IBM Power

Issue: [JDG-3395](#)

Description: Data Grid for OpenShift for OpenShift Container Platform on IBM Z or IBM Power contains **Liveness probe failed:** and **Readiness probe failed:** error messages.

Workaround: Ignore the error messages. They stop after the image is ready.

ClassCastException Occurs with Data Grid Command Line Interface

Issue: [JDG-3348](#)

Description: If you stop a controller node in a Data Grid cluster configured for cross-site replication and then run the **site** command with the Data Grid CLI, the following exception occurs:

```
org.infinispan.remoting.responses.CacheNotFoundResponse cannot be cast to  
org.infinispan.remoting.responses.SuccessfulResponse
```

Workaround: Wait until the Data Grid node is completely removed from the cluster view before running the **site** command.

Data Grid 7.3 Certification with Red Hat Fuse 6 Not Possible Due to Test Failures

Issue: [JDG-2758](#)

Description: Data Grid 7.3 certification with Red Hat Fuse 6 not possible due to test failures.

Workaround: There is no workaround for this issue.

Cannot use Data Grid 7.3 with Red Hat Fuse 6 and 7 in Conjunction with Java 11

Issue: [JDG-2800](#)

Description: Red Hat Fuse 6 and 7 are not compatible with Java 11. As a result, you cannot use Data Grid 7.3 with Red Hat Fuse 6 and 7 in conjunction with Java 11.

Workaround: There is no workaround for this issue.

Data Grid 7.3 with EAP 6 and MySQL 5 Can Result in Unexpected Behavior

Issue: [JDG-2871](#)

Description: Running Data Grid with EAP 6 and MySQL 5 can result in unexpected behavior.

Workaround: There is no workaround for this issue.

Data Grid 7.3 Libraries Do Not Deploy Successfully on EAP 6 or Oracle WebLogic

Issue: [JDG-2559](#)

Description: Data Grid 7.3 libraries do not deploy successfully on EAP 6 or Oracle WebLogic.

Workaround: There is no workaround for this issue.

Rolling Upgrades Not Successful from Data Grid 6.6.2

Issue: [JDG-2832](#)

Description: Attempting to perform a rolling upgrade from Data Grid 6.6.2 to Data Grid 7.3 results in exceptions and data does not migrate successfully.

Workaround: There is no workaround for this issue.

Externalizing HTTP Sessions from JBoss Web Server to Data Grid Does Not Work with the FINE Persistence Strategy

Issue: [JDG-2796](#)

Description: Setting the **persistenceStrategy** attribute to a value of **FINE** causes HTTP session externalization to behave unexpectedly.

Workaround: Set the **persistenceStrategy** attribute to a value of **COARSE**.

RocksDB Cache Store Not Supported on Red Hat Enterprise Linux 6 or Microsoft Windows Platforms

Issue: [JDG-2761](#)

Description: It is not currently possible to use a RocksDB Cache Store on Red Hat Enterprise Linux 6 or Microsoft Windows platforms.

Workaround: There is no workaround for this issue.

GLIBC_2.14 Error with RocksDB Cache Store On Red Hat Enterprise Linux (RHEL) 6

Issue: [JDG-2546](#)

Description: The following error was encountered when creating a RocksDB cache store on RHEL 6: **/lib64/libc.so.6: version `GLIBC_2.14' not found.**

Workaround: There is no workaround for this issue.

SKIP_CACHE_LOAD Flag Has No Effect if Authentication is Enabled

Issue: [JDG-1424](#)

Description: In Remote Client-Server mode, if you set the **SKIP_CACHE_LOAD** flag in the cache store configuration and enable authentication on Hot Rod clients, all entries are retrieved from the cache, including evicted entries.

Workaround: There is no workaround for this issue.

Cluster Actions Disabled on Data Grid Administration Console in Reload-Required State

Issue: [JDG-1843](#)

Description: Actions available for the Data Grid cluster are not available in the Administration Console if you choose to restart the cluster after changing the configuration. In this case, the cluster is in the **Reload-Required** state.

Reload and **Stop** actions are available for each node in the cluster.

Workaround: Reload at least one node in the cluster to restore actions at the cluster level.

Errors Occur When Changing the Eviction Strategy from the Data Grid Administration Console

Issue: [JDG-1804](#)

Description: If Data Grid is running in domain mode and you change the eviction strategy in the configuration through the Administration Console but do not restart to apply the changes, an error occurs.

Workaround: Restart the server after changing to the eviction strategy.

Intermittent Data Loss Occurs During Rolling Upgrades Between Clusters

Issue: [JDG-991](#)

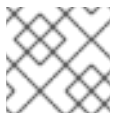
Description: When performing a rolling upgrade of Data Grid, all migrated data can be deleted from the target cluster after the nodes in the source cluster are taken offline.

Workaround: There is no workaround for this issue.

NullPointerException Occurs When Reading Data from Persistent Storage in Data Grid 7.0 and Earlier

Issue: [JDG-968](#)

Description: If you store data in a cache store with Data Grid 7.0 and earlier and then attempt to read that data with Data Grid 7.1 or later, an error occurs and it is not possible to read the data.



NOTE

This issue does not apply when upgrading from Data Grid 7.1 to a later version.

Workaround: There is no workaround for this issue.

CHAPTER 16. FIXED ISSUES

16.1. FIXED IN DATA GRID 7.3.9

Red Hat Data Grid 7.3.9 includes the following notable fixes:

- [JDG-4160](#) Memory leak on HotRodSourceMigrator
- [JDG-4550](#) Cache using single file store fails to start when security manager is enabled
- [JDG-4857](#) Initial server list switch should increment topology age
- [JDG-4956](#) Memory leak on Hot Rod clients when adding entries with large value size
- [JDG-4387](#) Simple cache with statistics enabled results in NullPointerException in EvictionManagerImpl
- [JDG-4767](#) Hot Rod client cluster switch happens too soon

16.2. FIXED IN DATA GRID 7.3.8

Red Hat Data Grid 7.3.8 includes the following notable fix:

- [JDG-4151](#) ConcurrentModificationException occurs when using the EntryMergePolicyFactoryRegistry class to register custom EntryMergePolicyFactory implementations

16.3. FIXED IN DATA GRID 7.3.7

Red Hat Data Grid 7.3.7 includes the following notable fixes:

- [JDG-3700](#) Unexpected behavior with applications externalizing sessions from EAP 7.3.
- [JDG-3848](#) Issues can occur with SingleFileStore cache stores when nodes leave clusters.
- [JDG-3818](#) Registered Listeners do not behave as expected and cause out-of-memory errors to occur.
- [JDG-3498](#) Performing remote tasks with compute operations results in error messages when eviction is used.
- [JDG-3497](#) Performing remote tasks with compute operations results in error messages when off-heap storage is used.

16.4. FIXED IN DATA GRID 7.3.6

Red Hat Data Grid 7.3.6 includes the following notable fixes:

- [JDG-2644](#) Data Grid administration console displays the first node in a cluster.
- [JDG-3527](#) Client requests can get routed to the wrong Data Grid server when using binary storage.

- [JDG-3522](#) SQL server exceptions occur when purging data from JDBC string based cache stores.
- [JDG-3450](#) Classloading issues occur when using annotation generated marshallers in deployed server tasks.
- [JDG-3529](#) JGroups subsystem for cluster transport did not persist properties for RELAY protocol.

16.5. FIXED IN DATA GRID 7.3.5

Red Hat Data Grid 7.3.5 includes the following notable fixes:

- [JDG-3355](#) RpcManager:SitesView attribute is empty or contains an incomplete view of XSiteRepl members.
- [JDG-3366](#) Using transactional caches causes performance degradation of expiration operations.
- [JDG-3354](#) Maximum idle configuration causes data to expire incorrectly when node failover occurs with clustered cache modes.
- [JDG-3413](#) Invalidation commands load previous values from cache stores unnecessarily.
- [JDG-3428](#) JDBC string-based cache store examples are incorrect.
- [JDG-3357](#) HotRod clients register timeout handlers for operations using socket timeouts.
- [JDG-3309](#) Caches configured with JPA cache stores using asynchronous mode, modifications to cache entries result in DEBUG exceptions and operations to remove entries from the cache store do not succeed.
- [JDG-2532](#) Administration console does not load when starting servers with the clustered.xml file.
- [JDG-3416](#) Stored indexes are not replicated when nodes join clusters.

16.6. FIXED IN DATA GRID 7.3.4

Red Hat Data Grid 7.3.4 includes the following notable fixes:

- [JDG-3200](#) When expiring entries, Data Grid replicated a command to remove expired entries across sites.
- [JDG-3149](#) Deploying Data Grid for OpenShift with Prometheus monitoring enabled resulted in pod startup failure. WFLYCTL0079 and WFLYLOG0078 exceptions were written to logs.
- [JDG-3264](#) FD_ALL sent out messages to non-members, which generated the warning messages in logs.
- [JDG-3167](#) JMX MBean for clustered cache statistics stopped working with the exception eviction type.
- [JDG-3194](#) ISPN004034 messages written to logs when querying near cache configurations.

- [JDG-3214](#) The administration console became unresponsive and exceptions occurred in Data Grid log files if you configured authorization for cache containers.
- [JDG-3324](#) *Wrong site status for cache 'null' exception occurred with cross-site replication
- [JDG-3185](#) Cache entry creation date is set to the Data Grid server start date if preload is enabled in persistence configuration

16.7. FIXED IN DATA GRID 7.3.3

Red Hat Data Grid 7.3.3 includes the following notable fixes:

- [JDG-3137](#) Parsing errors occur with the `jboss-cli.xml` file
- [JDG-3114](#) RocksDB ReadOptions memory leak
- [JDG-2960](#) Adding proto files with CLI commands fail if the target node is in a cluster
- [JDG-2834](#) Starting multiple caches with cache stores results in threading issues for custom cache stores
- [JDG-2148](#) Client logs unnecessary warning messages for `@Indexed` annotations
- [JDG-2117](#) Wrong values when a transactional cache stops during operations
- [JDG-3106](#) Operator fails to mount volumes that contain keystores
- [JDG-2968](#) Security update required for the ReflectionUtil class

16.8. FIXED IN DATA GRID 7.3.2

Red Hat Data Grid 7.3.2 includes the following notable fixes:

- [JDG-2673](#) Stale Reads Occur for Transactional Invalidation Caches with Shared Stores
- [JDG-2922](#) TCP: connection close can block when `send()` block on full TCP send-window
- [JDG-2854](#) Distinguishing multiple server Store configurations is impossible
- [JDG-2836](#) Counter manager configs not applied on server
- [JDG-2835](#) Counter client does not use CH to locate a counter
- [JDG-2817](#) Stackoverflow error with `computeIfAbsent`
- [JDG-2679](#) Shared stores should throw exception when cache is local
- [JDG-2897](#) `openshift.KUBE_PING` doesn't work with OCP4.1
- [JDG-2909](#) Session session bug, remove when session id has changed

16.9. FIXED IN DATA GRID 7.3.1

Red Hat Data Grid 7.3.1 includes the following notable fixes:

- [JDG-2561](#) Metrics not Available with Remote Client-Server Spring Boot Sample Application

- [JDG-2528](#) Quickstart Exceptions Occur with Custom Classpaths
- [JDG-2534](#) Failure Loading WildFly Modules Module Loading Failures Occur When Running Data Grid on JDK 8
- [JDG-272](#) Command Line Interface Script is Named ispn-cli.sh
- [JDG-2529](#) Administration Console Becomes Unresponsive After Creating Caches
- [JDG-2528](#) Quickstart Projects Fail With Custom Classpaths
- [JDG-2504](#) Exception Occurs When Running on IBM JDK
- [JDG-2518](#) Cache Instances Cannot Start with Insufficient Hash Space Segments

16.10. FIXED IN DATA GRID 7.3.0

[List of All Issues Resolved in Red Hat Data Grid 7.3.0 GA](#)

Features and enhancements, documentation improvements, bug fixes.

PART IV. MIGRATING TO DATA GRID 7.3

Migrating to Data Grid 7.3 involves reviewing product changes so you can adapt your existing configuration and usage to ensure a successful upgrade.

CHAPTER 17. CHANGES IN DATA GRID 7.3

17.1. JBOSS ENTERPRISE APPLICATION PLATFORM (EAP)

The Data Grid server is based on EAP 7.2 in this release. Make sure you consult the right version of the EAP documentation and that any underlying EAP configuration or setup is supported in 7.2.



NOTE

This release of Data Grid supports the **org.infinispan.extension** module with EAP 7.2 only.

For information about supported versions of EAP in this release, see the supported configurations at <https://access.redhat.com/articles/2435931>.

17.2. EAP MODULES

Data Grid now provides Library and Java Hot Rod Client modules for EAP as a single package. In previous releases, Data Grid provided Library modules and Java Hot Rod Client modules as separate packages. Download the modules for EAP from the [customer portal](#).



NOTE

The Data Grid EAP modules are now located in the **system/add-ons/ispn** directory.

17.3. CAMEL COMPONENT FOR JBOSS FUSE

For Apache Camel integration, the Red Hat Fuse team actively develop and maintain a **camel-infinispan** component that supersedes the **camel-jbosmdatagrid** component. Red Hat recommends that you use the **camel-infinispan** component that is available with Red Hat Fuse 7.3 and later.

Refer to [Red Hat Fuse documentation](#) for more information.

17.4. CACHE STORE COMPATIBILITY

Data Grid 7.3 introduces changes to internal marshalling functionality that are not backwardly compatible with previous versions of Data Grid. As a result, Data Grid 7.3.x and later cannot read cache stores created in earlier versions of Data Grid. Additionally, Data Grid no longer provides some store implementations such as JDBC Mixed and Binary stores.

Use **StoreMigrator.java** to migrate cache stores. This migration tool reads data from cache stores in previous versions and rewrites the content for compatibility with the current marshalling implementation.

For more information, see [Store Migrator](#).

17.5. MEMCACHED STORAGE

The Data Grid Memcached endpoint no longer stores keys as **java.lang.String**. For better compatibility Data Grid stores keys as **byte[]** arrays that represent UTF-8 encoded strings.

If you use the Memcached endpoint, you should reload data in the cache to store keys as **byte[]**. Either perform a rolling upgrade or load the data from an external source.

17.6. MEMCACHED CONNECTOR

The default **memcache-connector** is disabled for security considerations. To enable the **memcache-connector** you must configure the endpoint subsystem. See [Memcached Connector Configuration](#).

17.7. SCRIPTS RESPONSE

Distributed scripts with a text-based data type no longer return **null** when the result from each server is null. The response is now a JSON array with each individual result, for example: **[null, null]**.

17.8. SERVER THREAD POOLS

The threads that handle the child Netty event loops have been renamed from *-ServerWorker to *-ServerIO.

17.9. SHARED CACHE STORES WITH PASSIVATION

You cannot configure cache stores as shared and then enable passivation for the stores. This configuration can result in data inconsistencies because it is not possible to synchronize data across nodes. For this reason, Data Grid now throws an exception at startup if it detects this configuration.

17.10. DEFAULT SHARDS WITH AFFINITYINDEXMANAGER

The default number of shards is now **4**. In previous releases, the default number of shards was equal to the number of segments in the cache.

17.11. ADVANCEDCACHELOADER CHANGES

The **AdvancedCacheLoader** SPI allows Reactive Streams-based **publishKeys** and **publishEntries** methods that improve performance, threading, and ease of use. This change affects custom **CacheLoader** implementations.

17.12. DEPRECATIONS IN DATA GRID 7.3

This release deprecates features and functionality that affect migration to Data Grid 7.3. Notably compatibility mode is deprecated by configuring the **MediaType** for key/value pairs and storing data in binary format. The LevelDB cache store is also deprecated and replaced with the RocksDB.

For the complete list, see [Features and Functionality Deprecated in Data Grid 7.3](#) .

PART V. PATCHING DATA GRID SERVERS

Red Hat Data Grid server uses the patching functionality in JBoss Enterprise Application Platform (EAP) so that you can apply changes from errata releases without the need to completely replace an existing installation.

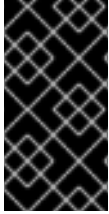
Patches are distributed for cumulative updates within a release version. The base version for this release is 7.3.1. Due to a technical issue, it is not possible to patch Data Grid 7.3.0.

You can apply 7.3.x patches to the base version or on top of other patches.

You cannot apply 7.3.x patches to any other Data Grid release version. Likewise you cannot apply patches from other release versions to the 7.3 release.

Data Grid provides patches for server instances only (Remote Client-Server mode). All other distributions, such as EAP modules, clients, and Data Grid Library mode, are provided as full releases.

CHAPTER 18. APPLYING PATCHES TO RED HAT DATA GRID



IMPORTANT

In Server Mode, it is not possible to patch Data Grid 7.3.1 or later on top of 7.3.0.

Data Grid 7.3.1 server is provided as a complete distribution. Data Grid 7.3.2 server, as well as each subsequent 7.3.x version, is provided as a patch.

To apply a patch to Data Grid, do the following:

1. Download the patch from the Red Hat Customer Portal at <https://access.redhat.com/downloads/>
2. Stop the server instance that you want to patch if it is running. To avoid issues with classloading, you should not apply patches to Data Grid while the server is running.

Either use the Administration Console to stop the server or enter **Ctrl-C** in the terminal where Data Grid is running.

3. Open a terminal and change to the **RHDG_HOME** directory.

```
$ cd RHDG_HOME
```

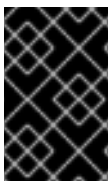
4. Apply the patch as follows:

```
$ bin/cli.sh "patch apply /path/to/jboss-datagrid-7.3.x-server-patch.zip"
```

5. Start the server with either the **standalone.sh** or **domain.sh** script, for example:

```
$ bin/standalone.sh -c clustered.xml
```

Resolving Errors with the 7.3.8 and 7.3.9 Patch



IMPORTANT

When patching Data Grid server installations to upgrade to 7.3.8 or 7.3.9, an issue with the patch results in an error at startup. You must complete the following procedure after you apply the patch otherwise you cannot start Data Grid.

1. Open your server configuration file for editing.
2. Remove **http-remoting-connector** from the remoting subsystem.

Before

```
<subsystem xmlns="urn:jboss:domain:remoting:4.0">
  <http-connector name="http-remoting-connector" connector-ref="default" security-
  realm="ApplicationRealm"/>
</subsystem>
```

After

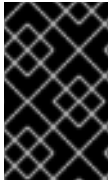
```
█ <subsystem xmlns="urn:jboss:domain:remoting:4.0"/>
```

3. Save and close your server configuration.

Refer to the following for more information: [RHDG server fail to start after patch 7.3.8 is applied](#) (Red Hat Knowledgebase)

CHAPTER 19. REVERTING PATCHES

You can roll back patches to revert the Red Hat Data Grid server to the previously installed version.



IMPORTANT

You should roll back patches only after applying a patch that results in unexpected behavior or undesirable effects. Rolling back patches is not intended for general uninstall functionality.

To revert a Data Grid patch, do the following:

1. Stop the server instance that you want to roll back if it is running. Either use the Administration Console to stop the server or enter **Ctrl-C** in the terminal where Data Grid is running.

2. Open a terminal and change to the **RHDG_HOME** directory.

```
$ cd RHDG_HOME
```

3. Find the ID of the patch that you want to roll back.

```
$ bin/cli.sh "patch history"
```

4. Roll back the server version as follows:

```
$ bin/cli.sh "patch rollback --patch-id=PATCH_ID --reset-configuration=false"
```



WARNING

Use caution when specifying the **reset-configuration** option.

--reset-configuration=false does not revert the server configuration. Because applying patches can change the server configuration, it is possible that the server does not restart if you roll back the patch but do not roll back the configuration. In this case, you should verify the server configuration and manually adjust it as needed before starting the server.

--reset-configuration=true reverts the server configuration to the pre-patch state. Any changes to the server configuration after the patch was applied are removed.

If conflicts exist when you attempt to roll back the patch, the operation fails and warnings occur. Enter **patch --help** to list available arguments that you can use to resolve the conflicts.

5. Start the server with either the **standalone.sh** or **domain.sh** script.

