



Red Hat Data Grid 8.4

Red Hat Data Grid 8.4 Release Notes

Get release information for Data Grid 8.4

Red Hat Data Grid 8.4 Red Hat Data Grid 8.4 Release Notes

Get release information for Data Grid 8.4

Legal Notice

Copyright © 2024 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

Find out about features and enhancements in Data Grid 8.4 as well as known and resolved issues.

Table of Contents

RED HAT DATA GRID	5
DATA GRID DOCUMENTATION	6
DATA GRID DOWNLOADS	7
MAKING OPEN SOURCE MORE INCLUSIVE	8
CHAPTER 1. DATA GRID RELEASE INFORMATION	9
1.1. WHAT'S NEW IN DATA GRID 8.4.8	9
1.2. REMOVAL NOTICE FOR DATA GRID RELEASE 8.4.7	9
Hot Rod .NET client	9
1.3. DEPRECATION NOTICE FOR DATA GRID RELEASE 8.4.7	9
Hot Rod C++ client	9
Hot Rod Node.js client	9
1.4. WHAT'S NEW IN DATA GRID 8.4.7	9
Data Grid 8.4.7 security update	9
Red Hat Insights Java client	9
Improvements to cross-site message handling	10
Ability to configure encryption when using Helm chart	10
Ability to configure security context for StatefulSet pods	10
IBM WebSphere Liberty transaction manager	10
Support for JBoss marshalling	10
1.5. WHAT'S NEW IN DATA GRID 8.4.6	10
Support for count(*) in aggregation queries	10
Enhanced metrics	10
1.6. DATA GRID 8.4.5 SECURITY UPDATE	10
1.7. WHAT'S NEW IN DATA GRID 8.4.4	11
Support for tracing with OpenTelemetry	11
Improved metrics naming for JGroups and cross-site metrics	11
Improved performance of indexed queries	11
Concurrent based methods for NearCache SPI	12
1.8. DEPRECATION NOTICE FOR DATA GRID RELEASE 8.4.3	12
Support for Java 11	12
Support for Java EE dependencies	12
Support for Spring 5.x and Spring Boot 2.x	12
Support for JCache	12
Deprecation of Data Grid modules for Red Hat JBoss EAP	12
Scattered cache mode	12
Adding caches to ignore list using the Data Grid Console or REST API	13
Cache service type	13
Testing Data Grid Server on Windows	13
1.9. WHAT'S NEW IN DATA GRID 8.4.3	13
Cache name added to the index file path	13
Simplified testing with infinispn-dev-mode.xml file	13
Enhanced DNS resolution for Hot Rod client	13
Jakarta EE version of the Infinispan Test Driver	14
Improved reindex performance with SQL query store	14
1.10. WHAT'S NEW IN DATA GRID 8.4.2	14
Support for Spring 6 and Spring Boot 3	14
Enhancements to cluster shutdown	14
Monitoring disabled in Data Grid Helm chart	14

Cache template conversion support	14
Query of entries with BigInteger and BigDecimal values	15
Query results include object version metadata	15
Cache index schema metamodel	15
Enhancements to update schema functionality	15
1.11. WHAT'S NEW IN DATA GRID 8.4.1	15
Migrating segmented cache stores	15
1.11.1. REST API enhancements	15
Comparing cache configurations	15
1.11.2. Resetting Hot Rod protocol version in the declarative configuration	16
1.11.3. Data Grid Console enhancements	16
1.12. WHAT'S NEW IN DATA GRID 8.4.0	16
Jakarta EE APIs	16
MemcachedDecoder performance improvement	16
1.12.1. Hot Rod clients	16
Support for queries with application/x-protostream	16
Client intelligence configuration per cluster	17
1.12.2. Data Grid Server	17
Micrometer metrics	17
Improved metrics during state transfer operations	17
RESP protocol endpoint	17
Dynamic role-based access control (RBAC)	17
Support for masked and external credentials	17
Option to generate JVM heap dumps	18
1.12.3. Data Grid Console	18
Cache creation wizard	18
Cache distribution statistics	18
1.12.4. Query API	18
Hibernate ORM 6 support	18
1.12.5. Data Grid native indexing annotations	18
Index startup mode	19
Dynamic index schema updates	20
Limit for results returned by a query	20
1.12.6. Data Grid Command Line Interface (CLI)	20
Improved security authentication	20
Applying command to all caches	20
1.13. SUPPORTED JAVA VERSIONS IN DATA GRID 8.4	21
Deprecation of Java 11 support	21
Removal of Java 8 support	21
Supported Java versions in Data Grid 8.4	21
Embedded caches	21
Remote caches	21
CHAPTER 2. KNOWN AND FIXED ISSUES	22
2.1. KNOWN ISSUES FOR DATA GRID	22
2.2. FIXED IN DATA GRID 8.4.7	22
2.3. FIXED IN DATA GRID 8.4.6	23
2.4. FIXED IN DATA GRID 8.4.4	24
2.5. FIXED IN DATA GRID 8.4.3	25
2.6. FIXED IN DATA GRID 8.4.2	25
2.7. FIXED IN DATA GRID 8.4.1	26
2.8. FIXED IN DATA GRID 8.4.0	27
2.9. HOST SYSTEM AND DEPENDENCY ISSUES	28

Nashorn JavaScript engine	28
TLS on Red Hat Enterprise Linux 7	28

RED HAT DATA GRID

Data Grid is a high-performance, distributed in-memory data store.

Schemaless data structure

Flexibility to store different objects as key-value pairs.

Grid-based data storage

Designed to distribute and replicate data across clusters.

Elastic scaling

Dynamically adjust the number of nodes to meet demand without service disruption.

Data interoperability

Store, retrieve, and query data in the grid from different endpoints.

DATA GRID DOCUMENTATION

Documentation for Data Grid is available on the Red Hat customer portal.

- [Data Grid 8.4 Documentation](#)
- [Data Grid 8.4 Component Details](#)
- [Supported Configurations for Data Grid 8.4](#)
- [Data Grid 8 Feature Support](#)
- [Data Grid Deprecated Features and Functionality](#)

DATA GRID DOWNLOADS

Access the [Data Grid Software Downloads](#) on the Red Hat customer portal.



NOTE

You must have a Red Hat account to access and download Data Grid software.

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).

CHAPTER 1. DATA GRID RELEASE INFORMATION

Learn about new features and get the latest Data Grid release information.

1.1. WHAT'S NEW IN DATA GRID 8.4.8

The Red Hat Data Grid 8.4.8 release fixes a packaging issue, and includes the Data Grid Server 8.4.8 OpenShift image. For more information, see [RHEA-2024:2103](#).

1.2. REMOVAL NOTICE FOR DATA GRID RELEASE 8.4.7

In Data Grid 8.4.7, the following features and functionality are removed.

Hot Rod .NET client

The Hot Rod .NET client is not provided with Data Grid 8.4.7.

1.3. DEPRECATION NOTICE FOR DATA GRID RELEASE 8.4.7

In Data Grid 8.4.7, the following features and functionality are deprecated and planned to be removed in future Data Grid releases.

Red Hat will provide support for these features and functionalities during the current release lifecycle, but they will no longer receive enhancements and will be eventually removed. It is recommended to transition to alternative solutions to ensure future compatibility.

Hot Rod C++ client

The support for Hot Rod C++ client is deprecated and planned to be removed in future releases.

You can continue using Hot Rod C++ client in combination with the latest Data Grid Server version.

Hot Rod Node.js client

The support for Hot Rod Node.js client is deprecated and planned to be removed in future releases.

You can continue using Hot Rod Node.js client in combination with the latest Data Grid Server version.

1.4. WHAT'S NEW IN DATA GRID 8.4.7

Data Grid 8.4.7 improves usability, increases performance, and enhances security. Find out what's new.

Data Grid 8.4.7 security update

Data Grid 8.4.7 provides a security enhancement to address CVEs. You must upgrade any Data Grid 8.4 deployments to version 8.4.7 as soon as possible.

For more information see the advisory related to this release [RHSA-2024:1509](#).

Red Hat Insights Java client

Red Hat Data Grid 8.4 Update 7 onward contains the Red Hat Insights Java client. The Red Hat Insights Java client is enabled for Data Grid only if Data Grid is installed on Red Hat Enterprise Linux (RHEL), and the RHEL system has Red Hat Insights client installed, configured, and registered. For more information, see the [Client Configuration Guide for Red Hat Insights](#).

The Red Hat Insights dashboard for Runtimes will be available in a future release on [Red Hat Hybrid](#)

[Cloud Console](#). Similar to the [RHEL dashboard](#) which is available on the Red Hat Hybrid Cloud Console, the Runtimes dashboard will show the inventory of the Runtimes installations, CVE details, and help you select the JVM options.

You can opt-out of the Red Hat Insights Java client client by setting the environment variable **RHT_INSIGHTS_JAVA_OPT_OUT** to **true**. For more information, see the [knowledge base article Red Hat Insights for Runtimes](#).

Improvements to cross-site message handling

To prevent the scenario where JGroups site-up events are delivered after the cross-site event resulting in the cross-site events being discarded, Data Grid 8.4.7 cross-site events are retried in case of an exception. Additionally, if cache or cache manager isn't in the running state yet, cross-site events are retried rather than making the site offline.

Ability to configure encryption when using Helm chart

In Data Grid 8.4.7, you can enable TLS when provisioning Data Grid with Helm chart to ensure that communication with Data Grid clusters is encrypted. You can specify TLS configuration for both endpoint, and cluster transport in the Helm configuration.

For more information, see [Configuring encryption](#).

Ability to configure security context for StatefulSet pods

You can now configure the security context for StatefulSet pods by using the **deploy.securityContext** attribute.

For more information see *Data Grid Operator Guide* [Deployment configuration values](#).

IBM WebSphere Liberty transaction manager

You can now use transactional caches inside your IBM WebSphere Liberty applications. Data Grid looks up and registers with the Transaction Manager running in IBM WebSphere Liberty.

Support for JBoss marshalling

JBoss Marshalling was deprecated in Data Grid 8.4.6 and earlier versions. From Data Grid 8.4.7 onwards, it is fully supported.

1.5. WHAT'S NEW IN DATA GRID 8.4.6

Data Grid 8.4.6 improves usability, increases performance, and enhances security. Find out what's new.

Support for **count(*)** in aggregation queries

Data Grid 8.4.6 adds support for the **count(*)** syntax in aggregation queries. You can use the **count(*)** syntax in aggregation queries to retrieve the count of all instances in each group, regardless of specific field values.

Enhanced metrics

Data Grid 8.4.6 includes the following enhancements to metrics:

- Latency metrics between two Data Grid nodes
- Histograms that provide details about latency distribution
- Additional Micrometer meters, such as **Counter**, **FunctionTimer**, and **DistributionSummary**

1.6. DATA GRID 8.4.5 SECURITY UPDATE

Data Grid 8.4.5 provides a security enhancement to address a CVE. You must upgrade any Data Grid 8.4 deployments to version 8.4.5 as soon as possible.

For more information see the advisory related to this release [RHSA-2023:5716](#).

1.7. WHAT'S NEW IN DATA GRID 8.4.4

Data Grid 8.4.4 improves usability, increases performance, and enhances security. Find out what's new.

Support for tracing with OpenTelemetry

Data Grid 8.4.4 now lets you generate tracing spans compatible with the OpenTelemetry standard, allowing you to export, visualize, and analyze tracing data related to the most important cache operations.

You can enable tracing between Data Grid Server and Hot Rod client or REST API to monitor and analyze the flow of requests, and track the execution path across different components.

For more information see the documentation:

- [Enabling Data Grid tracing](#)
- [Hot Rod client tracing propagation](#)
- [Enabling tracing propagation between Data Grid Server and REST API](#)

Improved metrics naming for JGroups and cross-site metrics

In Data Grid 8.4.4, you can enable the **name-as-tags** property for JGroups metrics and cross-site metrics.

Enabling **name-as-tags** simplifies metrics, displaying cluster and site names as tags rather than including them in metric names.

When you set **name-as-tags** to **false**, metrics are named based on the channel, resulting in multiple metrics for the same purpose:

```
# TYPE vendor_jgroups_xsite_frag4_get_number_of_sent_fragments gauge
# HELP vendor_jgroups_xsite_frag4_get_number_of_sent_fragments Number of sent fragments
vendor_jgroups_xsite_frag4_get_number_of_sent_fragments{cluster="xsite",node="..."} 0.0
# TYPE vendor_jgroups_cluster_frag4_get_number_of_sent_fragments gauge
# HELP vendor_jgroups_cluster_frag4_get_number_of_sent_fragments Number of sent fragments
vendor_jgroups_cluster_frag4_get_number_of_sent_fragments{cluster="cluster",node="..."} 2.0
```

When you set **name-as-tags** to **true**, metrics are simplified, and cluster and site names appear as tags:

```
# TYPE vendor_jgroups_frag4_get_number_of_sent_fragments gauge
# HELP vendor_jgroups_frag4_get_number_of_sent_fragments Number of sent fragments
vendor_jgroups_frag4_get_number_of_sent_fragments{cache_manager="default",cluster="xsite",node="..."} 0.0
vendor_jgroups_frag4_get_number_of_sent_fragments{cache_manager="default",cluster="cluster",node="..."} 2.0
```

In addition to simplified metrics, when you change the cluster name and site name, there is no need to update Grafana dashboards, as the metric names remain consistent.

Improved performance of indexed queries

You can now limit the required accuracy of hit counts by setting **hit-count-accuracy** attribute. When dealing with large data sets, precise hit counts can impact performance. Setting a limit to the hit count accuracy, lets you achieve faster query responses while ensuring that the provided hit counts remain sufficiently accurate for your application's needs.

The default accuracy of the **hit-count-accuracy** attribute is limited to **10000**. This means that for any query, Data Grid provides exact hit count up to maximum of 10000. If the effective hit count is higher than 10000, Data Grid returns a lower bound estimate of the count.

For more information see [Creating Ickle queries - Hit count accuracy](#) .

Concurrent based methods for NearCache SPI

The **NearCache** SPI has been updated to have concurrent based methods. This enhancement addresses an issue where concurrent accesses could lead to stale entries being saved in the near cache and resulted in inconsistent data.

1.8. DEPRECATION NOTICE FOR DATA GRID RELEASE 8.4.3

In Data Grid 8.4.3, the following features and functionality are deprecated and planned to be removed in future Data Grid releases.

Red Hat will provide support for these features and functionalities during the current release lifecycle, but they will no longer receive enhancements and will be eventually removed. It is recommended to transition to alternative solutions to ensure future compatibility.

Support for Java 11

Support for Java 11 is deprecated and planned to be removed in Data Grid version 8.5. Users of Data Grid 8.5 must upgrade their applications at least to Java 17.

You can continue using older Hot Rod Java client versions in combination with the latest Data Grid Server version. However, if you continue using older version of the client you will miss fixes and enhancements.

Support for Java EE dependencies

Support for Java EE dependencies is deprecated and planned to be removed in Data Grid version 8.5. Transition to Jakarta EE dependencies and APIs to align with the evolving Java enterprise ecosystem.

Support for Spring 5.x and Spring Boot 2.x

Support for Spring Boot 2.x and Spring 5.x is deprecated and planned to be removed in version Data Grid 8.5. Migrate to newer versions of Spring Boot and Spring framework for compatibility with future Data Grid releases.

Support for JCache

Support for JCache (JSR 107) is deprecated and planned to be removed in Data Grid version 8.5. As an alternative use other caching API developments in the Jakarta EE ecosystem.

Deprecation of Data Grid modules for Red Hat JBoss EAP

The Data Grid modules for Red Hat JBoss EAP applications that were distributed as a part of the Data Grid release are deprecated and planned to be removed in Data Grid version 8.5.

JBoss EAP users can use the **infinispan** subsystem that is integrated within the JBoss EAP product release without the need to separately install Data Grid modules.

Scattered cache mode

Scattered cache mode is deprecated and planned to be removed in Data Grid version 8.5. As an alternative to scattered caches, you can use distributed caches instead.

Adding caches to ignore list using the Data Grid Console or REST API

The ability to add caches to the ignore list using the Data Grid Console or REST API, which allows temporarily excluding specific caches from client requests, is deprecated. This feature is planned to be removed in future releases.

Cache service type

The **Cache** service type is deprecated and planned to be removed in Data Grid 8.5. The **Cache** service type was designed to provide a convenient way to create a low-latency data store with minimal configuration. Use the **DataGrid** service type to automate complex operations such as cluster upgrades and data migration.

Testing Data Grid Server on Windows

The support for Data Grid Server on Windows Server 2019 is deprecated and planned to be removed in Data Grid 8.5. However, the Data Grid team will continue testing C++ Hot Rod client with Windows Server 2019.

1.9. WHAT'S NEW IN DATA GRID 8.4.3

Data Grid 8.4.3 improves usability, increases performance, and enhances security. Find out what's new.

Cache name added to the index file path

The cache name is now included as part of the default index path. The index path specifies the filesystem path for the index when the storage is set to 'filesystem', and it can be a relative or absolute path.

When you set a custom value for the index path, ensure there are no conflicts between caches that use the same indexed entities.

Simplified testing with `infinispan-dev-mode.xml` file

Data Grid Server distribution now includes **`infinispan-dev-mode.xml`** file that provides you with necessary configuration to simplify development experience with Data Grid and Quarkus projects. The **`infinispan-dev-mode.xml`** configuration file enables **BASIC** client authentication for the Hot Rod client and cross-site configuration.

To start the Data Grid Server with this configuration, complete the following steps:

1. Copy the **`server/`** directory and rename the directory to **`server2/`**.
2. Start the first server using the **`infinispan-dev-mode.xml`** file.

```
./bin/server.sh -c infinispan-dev-mode.xml
```

3. In another terminal, start the second server using the same configuration.

```
./bin/server.sh -c infinispan-dev-mode.xml -s server2 -Dinfinispan.site.name=site2 -o 100 -Djgroups.mcast_port=46656
```

4. See the Data Grid logs to verify that Data Grid Servers have successfully formed cross-site view.

Enhanced DNS resolution for Hot Rod client

The Hot Rod client now supports alternate resolution strategies for connecting to Data Grid Server. You can configure the Hot Rod client with a single internal service DNS name that contains DNS entries for all running pods in the Data Grid cluster. Then the Hot Rod client can retrieve and use all the DNS records associated with that name, connecting to any available Data Grid Server instance.

Jakarta EE version of the Infinispan Test Driver

Data Grid distributes **infinispan-server-testdriver-core-jakarta** for users who are developing applications using the Jakarta EE platform.

Improved reindex performance with SQL query store

Data Grid 8.4.3 includes notable performance improvements specifically for reindex operations with SQL query store.

1.10. WHAT'S NEW IN DATA GRID 8.4.2

Data Grid 8.4.2 improves usability, increases performance, and enhances security. Find out what's new.

Support for Spring 6 and Spring Boot 3

Data Grid 8.4.2 adds support for Spring 6 and Spring Boot 3. Both Spring 6 and Spring Boot 3 require Java 17.

If you want to use the latest version of Spring or Spring Boot, update your artifacts as follows:

Spring 6

- Embedded caches: **infinispan-spring6-embedded**
- Remote caches: **infinispan-spring6-remote**

Spring Boot 3

- Embedded caches: **infinispan-spring-boot3-starter-embedded**



NOTE

Data Grid continues support for previous versions. You can continue using Spring 5 and Spring Boot 2.x in Data Grid 8.4.

Enhancements to cluster shutdown

Data Grid 8.4.2 includes several enhancements to prevent data loss and ensure consistency when you use the **shutdown cluster** command:

- Data Grid restricts access to the data stored in the cluster and modifications of the cluster state until the cluster is fully operational again.
- Improved error messages indicating that data stored in the cluster cannot be modified until the cluster is fully operational again.
- Improved log messages indicating which nodes already joined the cluster and which are still missing.

For more details see *Using the Data Grid Command Line Interface* [Shutdown and restart of Data Grid clusters](#).

Monitoring disabled in Data Grid Helm chart

Monitoring is disabled by default in Data Grid Helm chart. This prevents deployment failure in case a user does not have authorization to manipulate **ServiceMonitors** granted by an admin.

Cache template conversion support

The cache configuration conversion tool now supports converting cache templates between different configuration formats, such as XML, JSON, and YAML.

For more information see *Using the Data Grid REST API* [Converting Cache Configurations between XML, JSON and YAML](#).

Query of entries with **BigInteger** and **BigDecimal** values

Data Grid 8.4.2 now supports queries on cache entries that use **BigInteger** and **BigDecimal** values. Use **@ProtoField** annotations to annotate these fields in your Java classes and generate a Protostream schema that includes the **CommonTypesSchema**.

Query results include object version metadata

When performing a projection query, Data Grid let's you specify the project version of the queried entity. The version metadata is included in the projection results, so you can determine which version of the entity was used to generate the projection results.

Cache index schema metamodel

You can retrieve a full index schema metamodel of all indexes defined on a cache using REST API. The full index schema metamodel includes detailed information about the index configuration, including entity names, Java classes, index names, and field specifications.

For more information see *Using the Data Grid REST API* [Get Indexes Metamodel](#).

Enhancements to update schema functionality

From now you can update list of indexed entities in an existing cache with minimal downtime. Data Grid lets you restructure entity relationships, such as moving nested entities to the root level, without the need to create new caches.

1.11. WHAT'S NEW IN DATA GRID 8.4.1

Data Grid 8.4.1 improves usability, increases performance, and enhances security. Find out what's new.

Migrating segmented cache stores

The **StoreMigrator.java** utility now supports migration of data from segmented cache stores.

- Use the **source.segment_count** property to configure the number of segments for the source cache stores.

migrator.properties

```
source.segment_count=256
```

1.11.1. REST API enhancements

Data Grid gives you access to remote caches and containers with HTTP over REST.

Comparing cache configurations

You can verify whether two cache configurations are semantically equivalent.

- Invoke a **POST** request with a **multipart/form-data** body containing two configurations and the **?action=compare** parameter.

```
POST /rest/v2/caches?action=compare
```

Data Grid responds with **204 (No Content)** in case the configurations are equal, and **409 (Conflict)** in case they are different.

For more information see *Data Grid REST API Guide* [Comparing cache configurations](#).

1.11.2. Resetting Hot Rod protocol version in the declarative configuration

In case your declarative configuration forces a specific version of the Hot Rod protocol, you can reset the version back to its default value of **AUTO**. When you set **infinispan.client.hotrod.protocol_version** to **AUTO**, the Hot Rod client automatically uses the latest available version.

1.11.3. Data Grid Console enhancements

Data Grid Console 8.4.1 includes the following enhancements and bug fixes:

- Distributed caches have two owners by default
- Improved error handling with indexing configuration and running queries
- Improved user experience with Global statistics page

1.12. WHAT'S NEW IN DATA GRID 8.4.0

Data Grid 8.4 improves usability, increases performance, and enhances security. Find out what's new.

Jakarta EE APIs

Data Grid 8.4 distributes Jakarta EE 9+ based jars. If your application requires Jakarta specific dependencies, append the artifacts with **-jakarta**, for example:

pom.xml

```
<dependency>
  <groupId>org.infinispan</groupId>
  <artifactId>infinispan-client-hotrod-jakarta</artifactId>
</dependency>
```

For information about Jakarta EE see [About Jakarta EE](#).

MemcachedDecoder performance improvement

MemcachedDecoder no longer blocks Netty threads: preventing deadlocks when the thread pool is exhausted.

1.12.1. Hot Rod clients

Hot Rod is a custom binary TCP protocol that provides high-performance data access to client applications in different programming languages.

Support for queries with application/x-protostream

Hot Rod Node.js client in Data Grid 8.4 adds support for queries with **application/x-protostream** media type. Configure Hot Rod JS client to have **application/x-protostream** media type for values in your caches to be able to perform queries on them.

You can now use the **application/x-protostream** media type for all the Hot Rod operations supported in JS client. For more information see [Hot Rod Node.JS client guide](#).

Client intelligence configuration per cluster

When you configure Hot Rod client to use multiple Data Grid clusters you can use different client intelligence for each of the clusters.

For more information see [Hot Rod Java client guide](#).

1.12.2. Data Grid Server

Data Grid Server runs in a dedicated Java Virtual Machine (JVM) and provides scalable and fault tolerant clusters for remote caches.

Micrometer metrics

Data Grid 8.4 uses Micrometer library for collecting metrics. Data Grid Server exposes statistics through the **metrics** endpoint. You can collect the statistics with any monitoring tools, for example with Prometheus. Micrometer metrics replace SmallRye Metrics implementation of the MicroProfile Metrics.

Additional resources

- [Configuring Data Grid metrics](#)
- [Micrometer application monitoring](#)

Improved metrics during state transfer operations

When the cluster topology changes Data Grid exposes metrics for the number of segments and the number of transactional segments. Data Grid generates time metrics that are compatible with the REST API and the JMX API.

RESP protocol endpoint

Data Grid Server includes an experimental module that implements the RESP3 protocol. The RESP endpoint allows Redis clients to connect to one or more Data Grid backed RESP servers and perform cache operations.

Data Grid 8.4 supports a subset of Redis commands that are typically used for cache operations. For a list of supported commands refer to the *Using the RESP protocol endpoint with Data Grid* documentation. The Data Grid team plans to implement additional commands in subsequent releases.



IMPORTANT

RESP protocol endpoint is available as a technology preview feature.

Additional resources

- [Using the RESP protocol endpoint with Data Grid](#)
- [Red Hat Technology Preview Features Support Scope](#)

Dynamic role-based access control (RBAC)

Role-based access control (RBAC) capabilities use different permissions levels to restrict user interactions with Data Grid. You can modify user roles and assign them different permissions either programmatically or using the Data Grid CLI and Data Grid Console.

For more information see [Security authorization with role-based access control](#)

Support for masked and external credentials

Data Grid Server lets you use masked passwords and external commands to add passwords to Data Grid Server configuration.

Option to generate JVM heap dumps

You can generate a dump of all the live objects that are on the JVM heap. Data Grid generates a heap dump file in HPROF format in the server data directory and responds with the full path of the file in JSON format.

1.12.3. Data Grid Console

Data Grid Console is a graphical user interface for monitoring and managing remote caches that includes the following features and enhancements in 8.4:

Cache creation wizard

Open the **Cache Setup** tab and let the Data Grid Console wizard guide you through the process of creating and configuring caches. The wizard includes instructions and tooltips that help you to create caches and to enable Data Grid capabilities and features, such as indexing, security or persistence. Once you complete all the steps you can download your configuration in JSON, YAML, or XML format and reuse the file.

Cache distribution statistics

The Data Grid Console introduces **Metrics** tab on the **Cache details** page. You can view and filter data for your nodes, such as the number of entries per node or memory usage.

1.12.4. Query API

Data Grid lets you index caches and perform relational or full-text queries with in the Ickle query language.

Hibernate ORM 6 support

The Hibernate ORM second-level cache (2LC) implementation in Data Grid has been upgraded to work with Hibernate 6.

Add the following dependency:

pom.xml

```
<dependency>
  <groupId>org.infinispan</groupId>
  <artifactId>infinispan-hibernate-cache-v60</artifactId>
  <version>${version.infinispan}</version>
</dependency>
```



NOTE

Data Grid 8.4 does not support Hibernate 5 due to Jakarta EE migration.

1.12.5. Data Grid native indexing annotations

This release introduces new indexing annotations: **@Basic**, **@Keyword**, **@Text**, and **Embedded**. These new annotations replaced Hibernate Search annotations which means that you are no longer required to annotate your Java classes with the **@ProtoDoc** annotation for remote caches.

With the new native indexing annotations you can annotate entities directly. Moreover, you can use these annotations the same way for both embedded and remote queries.

Each of the new annotations supports a set of attributes that further describe how the entity is indexed.

Data Grid indexing annotations

@Basic

Use this annotation for numbers and short **Strings** that don't require any text transformation or processing. Supports the **sortable**, **searchable**, and **projectable** attributes.

@Keyword

Use this annotation for **Strings**. Supports the following attributes: **searchable**, **projectable**, **sortable**, and **normalizer**.

@Text

Use this annotation for larger **Strings**. Supports **searchable**, **projectable**, and **analyzer** attributes.

@Embedded

Use this annotation for embedded indexes and specify the entity structure. The **NESTED** structure preserves the original object relationship structure while the **FLATTENED** structure makes the leaf fields multivalued of the parent entity.

Poem.java

```
@Indexed
public class Poem {

    private Author author;
    private String description;
    private Integer year;

    @Embedded(includeDepth = 2, structure = Structure.NESTED)
    public Author getAuthor() {
        return author;
    }

    @Text(projectable = true, analyzer = "whitespace", termVector = TermVector.WITH_OFFSETS)
    public String getDescription() {
        return description;
    }

    @Basic(projectable = true, sortable = true, indexNullAs = "1800")
    public Integer getYear() {
        return year;
    }
}
```

Index startup mode

You can configure Data Grid to automatically perform operations to ensure that the index is consistent with data in the cache.

Startup modes

none

By default no indexing operation takes place.

purge

Clear the index when the cache starts. The cache will be unavailable during this operation.

reindex

Rebuild the index when the cache starts.

auto

Automatically clear or reindex the cache. If data is volatile and the index is persistent then the cache is cleared when it starts. If data is persistent and the index is volatile then the cache is reindexed when it starts.

Purge mode configuration

```
<distributed-cache>
  <indexing storage="filesystem" startup-mode="purge">
    <!-- Additional indexing configuration goes here. -->
  </indexing>
</distributed-cache>
```

Dynamic index schema updates

You can update schema without reindexing and downtime as long as your changes are compatible with the previous schema. Data Grid preserves the existing indexing and updates only the schema.

You can update schema in one of the following ways:

- Call the **updateIndexSchema()** method to programmatically to update schema from a Hot Rod Java client.

```
remoteCacheManager.administration().updateIndexSchema("MyCache");
```

TIP

For remote caches you can also update schema from Data Grid Console using the **update-schema** command.

- Update the schema with **POST** request.

```
POST .../v2/caches/{myCache}/search/indexes?action=updateSchema
```

Limit for results returned by a query

Data Grid adds **default-max-results** property that limits the number of results returned by a query. The default value of **default-max-results** property is **100**. Limiting the number of results returned by a query significantly improves performance of queries that don't have an explicit limit set.

1.12.6. Data Grid Command Line Interface (CLI)

The Data Grid CLI lets you script and automate administrative operations for remote caches.

Improved security authentication

You can use Data Grid CLI to connect to Data Grid Server using client certificate authentication.

Applying command to all caches

Data Grid CLI lets you use **--all-caches** or **-a** option to apply your command to all caches. For example, if you execute the following command, Data Grid shows the backup location status for all caches:

```
[//containers/default]> site status --all-caches --site=NYC
```

For more information see [Using the Data Grid command line interface](#).

1.13. SUPPORTED JAVA VERSIONS IN DATA GRID 8.4

Red Hat supports different Java versions, depending on how you install Data Grid.

Deprecation of Java 11 support

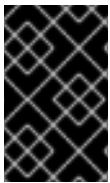
In Data Grid 8.4, support for Java 11 is deprecated and is planned to be removed in future release. Users of Data Grid 8.5 must upgrade their applications at least to Java 17.

You can continue using older Hot Rod Java client versions in combination with the latest Data Grid Server version. However, if you continue using older version of the client you will miss fixes and enhancements.

Removal of Java 8 support

Data Grid 8.4 removes support for Java 8. Support for Java 8 was deprecated and planned for removal since Data Grid 8.2.

Hot Rod Java clients running in applications that require Java 8 can continue using older versions of client libraries.



IMPORTANT

Users of Data Grid 8.4 must upgrade their applications at least to Java 11. However, considering the deprecation of Java 11 and its removal in Data Grid 8.5, the Data Grid team recommends upgrading directly to Java 17.

Supported Java versions in Data Grid 8.4

Embedded caches

Red Hat supports Java 11 and Java 17 when using Data Grid for embedded caches in custom applications.

Remote caches

Red Hat supports Java 11 and Java 17 for Data Grid Server installations. For Hot Rod Java clients, Red Hat supports Java 11 and Java 17.

Red Hat supports Java 11 and Java 17 for Data Grid Server, Hot Rod Java clients, and when using Data Grid for embedded caches in custom applications.



NOTE

When running Data Grid Server on bare metal installations, the JavaScript engine is not available with Java 17.

Additional resources

- [Supported Configurations for Data Grid 8.4](#)
- [Data Grid Deprecated Features and Functionality](#)

CHAPTER 2. KNOWN AND FIXED ISSUES

Learn about known issues in Data Grid and find out which issues are fixed.

2.1. KNOWN ISSUES FOR DATA GRID

For issues that affect Data Grid clusters that you manage with Data Grid Operator, you should refer to the [Data Grid Operator 8.4 release notes](#).

JGroups address defaults to an external IP

Issue: [JDG-6053](#)

Description: In bare metal deployments, when JGroups bind to an external IP without authentication configured by default, the connection is not secure, posing a risk of unauthorized access or manipulation.

Workaround: Secure the connection in one of the following ways:

- Configure JGroups security to control the network so only authorized nodes can join. For more information see, [Encrypting cluster transport](#).
- Use the **-Djgroups.bind.address=<internal-network>** parameter when starting Data Grid Server to set the JGroups address to secure internal network.

Inconsistent transactions when network partitions occur

Issue: [JDG-3935](#)

Description: In scenarios where a network partition occurs for a Data Grid cluster, transactions are rolled back after the partition is healed.

Workaround: There is no workaround for this issue.

Data Grid conflict resolution performance

Issue: [JDG-3636](#)

Description: In some test cases, Data Grid partition handling functionality took longer than expected to perform conflict resolution.

Workaround: There is no workaround for this issue.

2.2. FIXED IN DATA GRID 8.4.7

Data Grid 8.4.7 includes the following notable fixes:

- [JDG-6894](#) Response time between nodes is not correctly computed
- [JDG-6811](#) **SSLSocket** blocks on **close()**
- [JDG-6872](#) Avoid individual GET by Key in distributed query aggregation
- [JDG-6680](#) Attempt to iterate over cluster cache second time fails with segment already completed
- [JDG-6732](#) Cache startup failures on individual nodes can cause other caches to enter **DEGRADED** mode on restart

- [JDG-6652](#) Cache stats in REST should be clustered stats
- [JDG-6736](#) Check if **MetricsRegistry** is enabled before trying to register metrics
- [JDG-6591](#) Data Grid 8 automatic load cache from Oracle database limitation numeric values longer than 10
- [JDG-6824](#) Embedded JCache listener does not correctly return the previous value
- [JDG-6861](#) HR Rolling Upgrades fail with `__protobuf_metadata` cache
- [JDG-6840](#) Hot Rod Client **RemoteCache#computeAsync()** can lose the result of remapping processing when executed concurrently against a new entry
- [JDG-6751](#) Keycloak integration broken with ACL call
- [JDG-6738](#) **NullPointerException** in case of failover brings to HR Client hanging
- [JDG-6678](#) Put operation fails on cache with **remote-store** configured with uri property
- [JDG-6847](#) **SSLContext** ignored by **RemoteStore**
- [JDG-6889](#) Shared Store Publisher should fetch data from remote nodes with **SKIP_CACHE_LOAD**
- [JDG-6793](#) **NullPointerException** in MBeans without getters

2.3. FIXED IN DATA GRID 8.4.6

Data Grid 8.4.6 includes the following notable fixes:

- [JDG-5435](#) Avoid executing hybrid queries if **count** targets aggregable fields
- [JDG-6585](#) IRAC state transfer might lead to data loss
- [JDG-6593](#) Soft Index File Store (SIFS) **ExpirationCompaction** on logFile might fail to update the index
- [JDG-6636](#) Index might receive **NegativeArraySizeException**
- [JDG-6586](#) **PreferConsistencyStrategy** throws **UnsupportedOperationException** with zero-capacity nodes
- [JDG-6540](#) Reduce OpenTelemetry log spam by logging the **DEBUG** message only once, when OpenTelemetry is not found on the client
- [JDG-6301](#) Deadlock occurs during expiration of entries with local cache and JDBC cache store
- [JDG-6489](#) Incorrect memory management when using Control Groups version 2
- [JDG-6644](#) SIFS index corruption might occur if a shutdown is initiated while compactor is processing a file
- [JDG-6682](#) **ClassCastException** occurs when executing **PutAll** operations during rebalancing

- [JDG-6681](#) JDBC **PooledConnectionFactory** retains stale connections after database restart
- [JDG-6633](#) Backup write operations should ignore previous value when replicating from the primary owner
- [JDG-6635](#) SIFS expiration compaction writes WARN message for concurrent write to logFile
- [JDG-6655](#) Remote cache store ignores defined configuration properties unless the **blocking** property is explicitly set
- [JDG-4534](#) Remote site goes offline due to an issue during the initial state transfer for cross-site replication
- [JDG-6581](#) Near Cache with Bloom filter enabled doesn't work properly when configured with JBoss marshalling
- [JDG-6548](#) Exceptional HEAD requests return a non-zero Content-Length header
- [JDG-6458](#) Data Grid Console shows an error for non-admin users accessing a cache to which they have permission
- [JDG-6653](#) Hot Rod client repeatedly retries iteration operation against a failed server even after removal from topology
- [JDG-6634](#) SIFS removing a nonexistent key causes index to be incorrect

2.4. FIXED IN DATA GRID 8.4.4

Data Grid 8.4.4 includes the following notable fixes:

- [JDG-6447](#) Hot Rod query exception is missing stack trace to the caller
- [JDG-6446](#) Exclusion of 'byteman-bmunit' from JGroups dependencies
- [JDG-6443](#) Calling a Prometheus metrics endpoint collects both local and other node's metrics
- [JDG-6428](#) Queued Hot Rod client operations should retry if the connection pool terminates
- [JDG-6416](#) InvalidatedNearRemoteCache can persist stale values in case of concurrent access
- [JDG-6391](#) RemoteStore leaks out netty event loop client threads
- [JDG-6389](#) An exception in Hibernate Search occurs when using search#hitCount() if the returned SearchResultTotal is not exact
- [JDG-6382](#) Startup failure occurs when dynamically created caches that use dynamic RBAC are enabled
- [JDG-6378](#) Query module deadlock caused by multiple threads waiting on the Class initialization monitor
- [JDG-6358](#) State transfer never completes

- [JDG-6278](#) Cache template parsing fails when the template is defined after the configuration element
- [JDG-6256](#) Circular actions during indexing might throw an exception
- [JDG-6161](#) Attributes with nested properties in the JGroups configuration are not applied

2.5. FIXED IN DATA GRID 8.4.3

Data Grid 8.4.3 includes the following notable fixes:

- [JDG-6197](#) REST API fails to load the data distribution for local secured caches in Data Grid Console
- [JDG-6194](#) The JSON configuration parser does not replace properties when retrieving values by name
- [JDG-6160](#) It is not possible to override the `tracing.propagation_enabled` property using the HotRod properties
- [JDG-6125](#) Empty authorization roles serialized as JSON cannot be parsed
- [JDG-6095](#) Remote `getCacheNames` does not exclude secured caches based on user roles when listing available caches
- [JDG-6251](#) The `CacheLoaderInterceptor` leaves orphaned pending operations when encountering an availability exception
- [JDG-6235](#) Incorrect index statistics are displayed after clearing or rebuilding index
- [JDG-6231](#) The `Compactor.java` class fails to properly close files, resulting in heap accumulation
- [JDG-6229](#) Missing `CacheUpdateConfigurationAttributeTask` in the `ServerAdminOperationsHandler`
- [JDG-6228](#) Protobuf metadata cache fails to provide validation error messages when the Protobuf schema is replaced
- [JDG-6227](#) Deadlock during expiration of entries with local cache mode
- [JDG-6226](#) `ManagedConnectionFactory DataSource` is initialized at component start time
- [JDG-6223](#) Incorrect detection of IPv6 multihomed addresses by the Data Grid Server
- [JDG-6222](#) Failed cache creation with backup and merge policy
- [JDG-6218](#) `NoClassDefFoundError` when using JDK 17 and JBossMarshalling
- [JDG-6217](#) Missing `infinispan-hibernate-cache-v62` in the Data Grid BOM file

2.6. FIXED IN DATA GRID 8.4.2

Data Grid 8.4.2 includes the following notable fixes:

- [JDG-6057](#) RESP endpoint cannot parse request larger than packet size

- [JDG-6015](#) Not possible to update Helm Chart `deploy.security.batch` on upgrade
- [JDG-5997](#) Changing index schema incompatibly, for example, making a non-sortable field sortable, results in `ISPN014827`
- [JDG-5988](#) Helm Chart: DNS discovery fails when pods are not marked as ready
- [JDG-5972](#) REST API blocks key and entry retrieval, causing non-blocking threads to wait for stream elements
- [JDG-5943](#) Adding entry with `putAll` and `replaceWithVersion` causes a timeout
- [JDG-5927](#) SQL Cache store initiation fails on SQL type `CHAR`
- [JDG-5811](#) Data Grid Console: missing default value for indexing startup mode in the Cache creation wizard
- [JDG-5732](#) After the `shutdown cluster` command, caches remain accessible before all nodes rejoin the cluster
- [JDG-3967](#) Missing error message about incomplete restart of a cluster after the `shutdown cluster` command
- [JDG-5549](#) REST and CLI commands add schema definition for invalid files

2.7. FIXED IN DATA GRID 8.4.1

Data Grid 8.4.1 includes the following notable fixes:

- [JDG-5725](#) Creating Data Grid cluster backup over REST fails
- [JDG-5726](#) Retrieving Data Grid Server configuration with GET request over the browser fails
- [JDG-5724](#) Retrieving diagnostic reports for multiple Data Grid Server instances fails
- [JDG-5825](#) GET command blocks text/plain caches
- [JDG-5802](#) Data Grid Console: Queries return error when they run after the entries expire
- [JDG-5842](#) Data Grid Console: Metrics for misses and retrievals are updated only after the page is reloaded
- [JDG-5831](#) Data Grid Console: Cache creation wizard let users create indexed caches with invalid media type
- [JDG-5832](#) Data Grid Console: Data distribution chart resizing issue
- [JDG-5625](#) Updates to the Data Grid Console logo
- [JDG-5686](#) Thread Locking when doing Access Control List (ACL) permission check
- [JDG-5805](#) Getting statistics with `AdvancedCache.getStats()` fails
- [JDG-5801](#) The default value of datasource connection pool `maxSize` causes `NullPointerException` (NPE)

- [JDG-5799](#) JSON characters should be escaped with two equality checks
- [JDG-5800](#) Using Java 17 with Red Hat Single Sign-On configured on a Data Grid cluster throws an exception
- [JDG-5821](#) Incompatibility between Data Grid Server 8.3 and Hot Rod client 8.4
- [JDG-5790](#) `CacheInputStream` escapes the value for every byte in the value

2.8. FIXED IN DATA GRID 8.4.0

Data Grid 8.4.0 includes the following notable fixes:

- [JDG-4863](#) Option to delete Protobuf schema from Data Grid CLI or the Console
- [JDG-5439](#) Topology update with multithread client is not secure
- [JDG-5320](#) Missing support for embedded query with Protobuf-encoded caches
- [JDG-5316](#) `SoftIndexFileStore` increases the size of the data folder
- [JDG-5318](#) `SoftIndexFileStore` loads incorrect size of the data that is stored
- [JDG-5221](#) `ArrayIndexOutOfBoundsException` when using indexed field in a remote query with `startOffset=10` and `maxResults=-1`
- [JDG-5325](#) RELAY2 does not reconnect after a `UnknownHostException`
- [JDG-5219](#) Invalid Protobuf schema causes `PUT` operation to fail without specific error message
- [JDG-5253](#) Creating two caches that have name difference only in special characters and have statics enabled results in second cache being degraded
- [JDG-4766](#) Client should not close the server connection after timeout expires
- [JDG-5544](#) Potential deadlock in index writer threads under heavy load
- [JDG-5542](#) REST keys call fails if a cache has a value media-type that is not convertible to JSON
- [JDG-5553](#) Bulk operations do not work properly with embedded key SQL store
- [JDG-5641](#) Stores that don't support expiration should not be configured to have expiration
- [JDG-5575](#) If a Protobuf entry contains number with fractional portion it fails to persist this value for MicrosoftSQL 2019 server or Postgresql 13
- [JDG-5531](#) `YamlConfigurationReader` throws `NullPointerException` for valid cache configuration
- [JDG-5622](#) `NullPointerException` when computing Netty global connections
- [JDG-5624](#) `SoftIndexFileStore` does not remove log files with expiration
- [JDG-5327](#) `ConcurrentModificationException` for a joining node with `CrossSiteReplication`

- [JDG-5232](#) Invalid YAML representation of a cache
- [JDG-5442](#) Configuring memory max-size as non-byte value causes failure on CacheManager restart
- [JDG-5411](#) If a security realm name contains special characters the HTTP library fails to parse it
- [JDG-3394](#) Hot Rod client throws an exception if a remote-task is executed in TaskExecutionMode.ALL_NODES
- [JDG-5629](#) A site sends empty batches with IRAC to the remote sites

2.9. HOST SYSTEM AND DEPENDENCY ISSUES

In some cases Data Grid deployments can encounter errors that are caused by the host system or external dependency. This section provides details about any such known issues as well as troubleshooting and workaround procedures.

Nashorn JavaScript engine

If your Data Grid Server uses JavaScript to automate tasks, you must install the Nashorn JavaScript engine to ensure that these scripts can run on Data Grid 8.4. This is because OpenJDK 17 has removed support for the Nashorn JavaScript engine, its APIs, and the **jjs** tool.

For a bare-metal Data Grid Server, you can install Nashorn from the Maven central repository by issuing the following command in your Data Grid CLI:

```
bin/cli.sh install org.openjdk.nashorn:nashorn-core:15.4 \
    org.ow2.asm:asm:7.3.1 \
    org.ow2.asm:asm-util:7.3.1
```

On OpenShift, you can create an **Infinispan** Custom Resource (CR) that sets the Data Grid Operator to install Nashorn for your Data Grid cluster. For example:

```
apiVersion: infinispan.org/v1
kind: Infinispan
metadata:
  name: infinispan
spec:
  replicas: 2
  dependencies:
    artifacts:
      - maven: org.openjdk.nashorn:nashorn-core:15.4
      - maven: org.ow2.asm:asm:7.3.1
      - maven: org.ow2.asm:asm-util:7.3.1
  service:
    type: DataGrid
```

TLS on Red Hat Enterprise Linux 7

RHEL 7 provides a version of the OpenSSL library that does not yet offer support for TLSv1.3. However Data Grid Server 8.4 enables TLSv1.3 and TLSv1.2 by default, which causes errors with client connections for encrypted Hot Rod and REST endpoints.

Data Grid Server also logs messages such as the following:


```
WARN [org.infinispan.HOTROD] ISPN004098: Closing connection due to transport error  
org.infinispan.client.hotrod.exceptions.TransportException:: ISPN004077:  
Closing channel due to error in unknown operation.
```

If you install Data Grid Server on RHEL 7 you should use the native Java SSL library by disabling OpenSSL with the following JVM option:

```
-Dorg.infinispan.openssl=false
```

Additional resources

- [Securing Applications with TLS in RHEL](#)