



# Red Hat Data Grid 8.0

## Red Hat Data Grid 8.0 Release Notes

Data Grid Documentation





## 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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 8.0, learn about current known issues and resolved issues, review the configurations that Red Hat supports, and more.

# Table of Contents

<b>CHAPTER 1. RED HAT DATA GRID</b> .....	<b>3</b>
1.1. DATA GRID DOCUMENTATION	3
1.2. DATA GRID DOWNLOADS	3
<b>CHAPTER 2. DATA GRID RELEASE INFORMATION</b> .....	<b>4</b>
2.1. UPGRADE TO THE LATEST DATA GRID VERSION	4
2.1.1. Apache Log4j security vulnerabilities	4
2.2. DATA GRID 8.0.1 RELEASE INFORMATION	4
2.2.1. Server Patching Tool	4
2.2.2. Server Rolling Upgrade	4
2.2.3. Per-Cache Hot Rod Configuration	5
2.2.4. Creating Caches with Templates via Hot Rod	5
2.2.5. Improved Configuration for Single File Cache Stores	5
2.2.6. Defining External Service Ports	5
2.2.7. Data Grid Servers Warn if Partition Handling Configuration Leads to Degraded Clusters	5
2.3. WHAT'S NEW IN DATA GRID 8.0	6
2.3.1. Data Grid Operator	6
2.3.2. Data Grid Server	6
2.3.3. Management and Administration Console	6
2.3.4. Data Grid REST API	7
2.3.5. Data Grid Command Line Interface (CLI)	7
2.3.6. Monitoring and Observability	7
2.3.7. Marshalling and Data Durability	7
2.3.8. Performance	8
2.4. SUPPORTED JAVA VERSIONS IN DATA GRID 8.0	8
Embedded caches	8
Remote caches	8
Java 8 deprecation	8
<b>CHAPTER 3. DATA GRID FOR OPENSIFT</b> .....	<b>9</b>
3.1. DATA GRID FOR OPENSIFT 8.0 IMAGES	9
3.2. DATA GRID LIBRARY MODE ON OPENSIFT	9
<b>CHAPTER 4. KNOWN AND FIXED ISSUES</b> .....	<b>10</b>
4.1. KNOWN ISSUES FOR DATA GRID	10
4.1.1. Known Issues	10
4.2. FIXED IN DATA GRID 8.0.1	12
4.3. FIXED IN DATA GRID 8.0	13
<b>CHAPTER 5. DEPRECATED FEATURES AND FUNCTIONALITY</b> .....	<b>14</b>
5.1. DEPRECATIONS	14
5.2. REMOVED FEATURES AND FUNCTIONALITY	15
<b>CHAPTER 6. APPENDIX</b> .....	<b>16</b>
6.1. EXTERNALIZING HTTP SESSIONS FROM JBOSS WEB SERVER (JWS) TO DATA GRID	16
6.1.1. Installing the Session Client	16
6.1.2. Configuring the Session Manager	16
6.1.3. HotRodManager Configuration Properties	16



# CHAPTER 1. 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.

## 1.1. DATA GRID DOCUMENTATION

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

- [Data Grid 8.0 Documentation](#)
- [Data Grid 8.0 Component Details](#)
- [Supported Configurations for Data Grid 8.0](#)

## 1.2. 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.

## CHAPTER 2. DATA GRID RELEASE INFORMATION

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

### 2.1. UPGRADE TO THE LATEST DATA GRID VERSION

Red Hat recommends you upgrade any deployments from 8.0.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.

#### 2.1.1. Apache Log4j security vulnerabilities

Data Grid includes Apache Log4j components in:

- Data Grid Server distribution
- Data Grid Server container image for Red Hat OpenShift

Data Grid 8.2.2 fixes [CVE-2021-44228](#) which is a critical security vulnerability. Data Grid 8.2.3 fixes several Log4j CVEs of moderate severity. For more information, see the [Data Grid 8.2 release notes](#) or the [Data Grid Operator 8.2 release notes](#). You can find the latest Data Grid documentation at [Red Hat Data Grid Product Documentation](#).



#### NOTE

Data Grid Operator version 8.2.7 corresponds to Data Grid 8.2.2.

If you cannot upgrade your Data Grid deployment, you must follow the steps to mitigate the critical vulnerability as outlined in the [RHSB-2021-009 Log4Shell - Remote Code Execution](#) security bulletin.

### 2.2. DATA GRID 8.0.1 RELEASE INFORMATION

Data Grid 8.0.1 is an errata release that provides enhancements, fixes bugs, and hardens security.

#### 2.2.1. Server Patching Tool

The Data Grid CLI includes a **patch** command that lets you manage and install server patches.

See [Patching Data Grid Servers](#) for instructions on downloading and installing server patches.



#### NOTE

Data Grid 8.0.1 does not include a patch for 8.0.0. For this reason, you need to download the full Data Grid 8.0.1 server distribution.

For instructions on patching Data Grid 8.0.0 server installations, refer to the following article from the Red Hat support team:

<https://access.redhat.com/solutions/4962581>

#### 2.2.2. Server Rolling Upgrade

Data Grid servers expose the ability to perform rolling upgrades via the REST API.



See [Performing Rolling Upgrades](#) for steps on upgrading Data Grid clusters without downtime.

### 2.2.3. Per-Cache Hot Rod Configuration

The Hot Rod client configuration API includes a **RemoteCacheConfigurationBuilder** class that lets you create and configure caches on first access either programmatically or via Hot Rod client properties file.

See the following documentation for more information:

- [Creating Caches on First Access](#)
- [Hot Rod Client Configuration API](#)
- [org.infinispan.client.hotrod.configuration.RemoteCacheConfigurationBuilder](#)

### 2.2.4. Creating Caches with Templates via Hot Rod

The **RemoteCacheManagerAdmin** interface allows you to create caches with the **org.infinispan** cache configuration templates:

- **createCache(String name, DefaultTemplate template)**
- **getOrCreateCache(String name, DefaultTemplate template)**

For example:

```
private void createCacheWithTemplate() {  
    manager.administration().getOrCreateCache("myDistCache", "org.infinispan.DIST_SYNC");  
    System.out.println("Cache created.");  
}
```

See [org.infinispan.client.hotrod.RemoteCacheManagerAdmin](#) for more information.

### 2.2.5. Improved Configuration for Single File Cache Stores

Configuration of Single File cache stores is simplified so you no longer need to set **path** or **relative-to** attributes. The location of Single File caches stores are always relative to the global persistent location.

See the following configuration documentation for more information:

- [Setting a Global Persistent Location for File-Based Cache Stores](#)
- [Single File Cache Store](#)

### 2.2.6. Defining External Service Ports

Data Grid Operator lets you define the port where external services are exposed.

See [Exposing Data Grid to External Clients](#) for more information.

### 2.2.7. Data Grid Servers Warn if Partition Handling Configuration Leads to Degraded Clusters

If you configure distributed caches with one owner (`owners="1"`) then the conflict resolution strategy for partition handling must be `ALLOW_READ_WRITES` (`when-split="ALLOW_READ_WRITES"`), otherwise the cache configuration exception is thrown at server startup:

**ISPN000575:** PartitionStrategy must be `ALLOW_READ_WRITES` when numOwners is 1

In the case of Data Grid service clusters on OpenShift, the Data Grid Operator automatically modifies configuration during upgrade to use `when-split="ALLOW_READ_WRITES"`. This ensures that clusters restart when rolling upgrades take place.

For Data Grid servers running outside of OpenShift, you must modify cache configuration manually.

## 2.3. WHAT'S NEW IN DATA GRID 8.0

Data Grid 8.0 is a new major version with significant changes that focus on improved usability, increased performance, and a cloud-native architecture.

### 2.3.1. Data Grid Operator

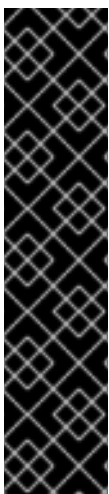
Data Grid 8.0 introduces a fully supported Data Grid Operator that provides operational intelligence to simplify Data Grid cluster deployment and management on Red Hat OpenShift.

Try some of the following documentation to create a Data Grid Operator subscription and get running in OpenShift:

- [Getting Started with Data Grid Operator](#)
- [Running Data Grid for OpenShift](#)

### 2.3.2. Data Grid Server

Data Grid servers are no longer based on Red Hat JBoss Enterprise Application Platform (EAP), which reduces the memory and disk footprint and also simplifies configuration. In addition Data Grid 8.0 servers provide several enhancements and improvements to security, including integration with Red Hat SSO and a smaller attack surface.



#### IMPORTANT

To take advantage of security enhancements to Data Grid Server, as well as simplified configuration and improved usability, you should download and install Data Grid 8.1 if you are starting out.

If you have any existing Data Grid Server deployments you should upgrade from 8.0 to 8.1 as a security best practice.

See the following:

- [Upgrading Data Grid](#)
- [Data Grid Server Guide](#)

See [Getting Started with Data Grid Server](#) and create a running Data Grid cluster in a few minutes.

### 2.3.3. Management and Administration Console

Data Grid 8.0 includes a new console with improved management and administration capabilities.

To access the console, start Data Grid server and go to:

**127.0.0.1:11222/console/**

You can also access the console from Data Grid clusters running on OpenShift.

### 2.3.4. Data Grid REST API

Data Grid 8.0 introduces REST API v2 with 50% faster response rates compared to v1 along more extensive capabilities.

Data Grid REST API v2 provides operations that let you:

- Access data and manipulate objects such as counters.
- Perform management tasks, like gracefully shutting down Data Grid clusters or transferring cache state to backup locations when using cross-site replication.
- Monitor cluster and server health and retrieve statistics.

Data Grid REST API v2 also automatically converts between storage formats such as JSON, XML, Protobuf, and plain text for increased inter-operability.

The Data Grid engineering team develop and maintain comprehensive [REST API Documentation](#)

### 2.3.5. Data Grid Command Line Interface (CLI)

In 8.0, Data Grid gives you a new CLI with intuitive commands for remotely accessing data and managing clusters.

The CLI uses familiar Bash commands for navigating, such as **cd** and **ls**. It also provides command history and auto-completion for ease of use. Additionally the CLI provide help text and man pages for commands with clear examples.

Try the docs: [Getting Started with the Data Grid CLI](#).

### 2.3.6. Monitoring and Observability

Data Grid 8.0 is compatible with the Eclipse MicroProfile Metrics API and provides a **/metrics** endpoint for integration with Prometheus.

Data Grid metrics include:

- Gauges that return values, such as JVM uptime or average number of seconds for cache operations.
- Histograms that show how long read, write, and remove operations take, in percentiles.

Data Grid 8.0 also offers improved statistics and management operations via JMX and updates to logging with coarse-grained logging categories and support for logs in JSON format.

### 2.3.7. Marshalling and Data Durability

In 8.0, Data Grid marshalling capabilities are refactored so that user objects are isolated from internal objects, which improves efficiency and makes it easy to change between different Marshaller implementations.

Data Grid also now uses a ProtoStream API to store user types as Protocol Buffers, which is a platform-independent, language-neutral format that is highly customizable. As a result of this change, Data Grid 8.0 can also ensure binary compatibility of your data with future versions.

Find out about Data Grid [Marshaller Implementations](#).

### 2.3.8. Performance

Data Grid 8.0 brings a number of performance improvements, notably:

- Faster and more efficient cross-site replication operations.
- Reduced thread usage through non-blocking internal calls.
- Automatic detection of Hot Rod protocols.

## 2.4. SUPPORTED JAVA VERSIONS IN DATA GRID 8.0

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

### Embedded caches

Red Hat supports Java 8 and Java 11 when using Data Grid for embedded caches in custom applications.

### Remote caches

Red Hat supports Java 11 only for Data Grid Server installations. For Hot Rod Java clients, Red Hat supports Java 8 and Java 11.

### Java 8 deprecation

Support for Java 8 is deprecated in Data Grid 8.2 and currently planned for removal in Data Grid 8.4.

Users with embedded caches in custom applications should plan to upgrade to Java 11 or to Java 17 when support becomes available.

Hot Rod Java clients running in applications that require Java 8 can continue using older versions of client libraries. Red Hat supports using older Hot Rod Java client versions in combination with the latest Data Grid Server version.

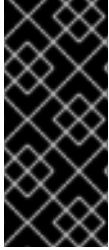
### Additional resources

- [Supported Configurations for Data Grid 8.0](#)
- [Data Grid Deprecated Features and Functionality](#)

## CHAPTER 3. DATA GRID FOR OPENSIFT

### 3.1. DATA GRID FOR OPENSIFT 8.0 IMAGES

Data Grid 8.0 includes two container images, the Data Grid Operator image and Data Grid server image.



#### IMPORTANT

Red Hat supports Data Grid for OpenShift 8.0 only through Data Grid Operator subscriptions.

Likewise Red Hat does not support customization of any 8.0 images from the Red Hat Container Registry.

Data Grid for OpenShift images are hosted on the Red Hat Container Registry, where you can find health indexes for the images along with information about each tagged version.

- [Data Grid Container Images](#)

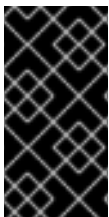
### 3.2. DATA GRID LIBRARY MODE ON OPENSIFT

Embedding Data Grid in custom OpenShift applications, or running in Library Mode, is intended for specific uses only.

- Using local or distributed caching in custom Java applications to retain full control of the cache lifecycle. Additionally, when using features that are available only with embedded Data Grid such as distributed streams.
- Reducing network latency to improve the speed of cache operations. Although it is worth noting that the Hot Rod protocol provides near-cache capabilities that achieve equivalent performance of a standard client-server architecture.

Embedding Data Grid in OpenShift also has some limitations:

- Persist cache stores are not currently supported.
- Only **DNS\_PING** is supported for the cluster discovery protocol.
- **TCP** is supported for the ping port.
- **UDP** is not supported with embedded Data Grid.



#### IMPORTANT

Red Hat highly discourages embedding Data Grid to build custom caching servers to handle remote client requests. You should create Data Grid service clusters with the Data Grid Operator instead to get full Red Hat support and benefit from regular updates with performance improvements and security fixes.

## CHAPTER 4. KNOWN AND FIXED ISSUES

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

### 4.1. KNOWN ISSUES FOR DATA GRID

#### 4.1.1. Known Issues

##### Global Persistent Locations Do Not Retrieve Server Data Path Property

**Issue:** [JDG-3826](#)

**Description:** With Data Grid server deployments, if you set the path of the global persistent location relative to the **infinispan.server.data.path** property, Data Grid does not set that path in the configuration and the following exception is written to the server log file:

```
org.infinispan.commons.CacheConfigurationException: javax.xml.stream.XMLStreamException:
ParseError at [row,col]:[1,130]
Message: Missing required property '/user/path/to/server/data' for attribute 'relative-to'
```

**Workaround:** Set the path in the global persistent location instead of using the property, for example:

```
<global-state>
  <persistent-location path="tmp/example" relative-to=/user/path/to/server/data"/>
</global-state>
```

##### JDBC String-Based Cache Store Functional Tests Hang with Sybase

**Issue:** [JDG-1492](#)

**Description:** When running functional tests for the JDBC String-Based cache store with Sybase databases, cache stores become unresponsive and the functional tests fail. This issue occurs because Sybase databases do not support batch updates to binary columns.

**Workaround:** Set a value of **true** for the property **infinispan.jdbc.upsert.disabled** on the JDBC String-Based cache store. This property executes write operations individually instead of in batches.

##### Default Service for Data Grid on OpenShift Missing Port Name

**Issue:** [JDG-3643](#)

**Description:** The default network service that Data Grid Operator creates for Data Grid clusters on OpenShift is missing a port name. As a result, the Prometheus console attempts to collect Data Grid metrics from the **8888/metrics** target as well as **11222/metrics**.

**Workaround:** There is no workaround for this issue.

##### Data Grid Operator Does Not Encrypt Endpoints by Default

**Issue:** [JDG-3591](#)

**Description:** Data Grid Operator does not automatically encrypt Data Grid endpoints with OpenShift service certificates.

**Workaround:** Set the **spec.security.endpointEncryption** configuration in your **Infinispan CR**. See [Securing Data Grid Connections](#).

### Cannot Access Caches If Names Contain the "/" Character

**Issue:** [JDG-3494](#)

**Description:** If cache names contain the "/" character, it is not possible to access them remotely via the CLI or REST API.

**Workaround:** There is no workaround for this issue.

### Data Grid Hot Rod JavaScript Client SSL Test Failures

**Issue:** [HRJS-88](#)

**Description:** The Hot Rod JS client behaves unexpectedly and the following exceptions are logged when using encrypted connections to Data Grid server:

```
[org.infinispan.server.hotrod.HotRodHeader] ISPN006010: Conditional operation 'FOO' should be used with transactional caches, otherwise data inconsistency issues could arise under failure situations
```

```
[org.infinispan.server.hotrod.HotRodHeader] ISPN006011: Operation 'FOO' forced to return previous value should be used on transactional caches, otherwise data inconsistency issues could arise under failure situations
```

```
[org.infinispan.SECURITY] ISPN000946: Using OpenSSL Provider ...
```

**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.

### Data Grid Does Not Passivate JWS Sessions Correctly

**Issue:** [JDG-2796](#)

**Description:** When externalizing sessions from JBoss Web Server (JWS), sessions are not passivated correctly if using the **FINE** persistence strategy.

**Workaround:** There is no workaround for this issue.

### Remote Cache Stores Cannot Use Preload with EAP Session Externalization

**Issue:** [JDG-3504](#)

**Description:** JBoss Enterprise Application Platform (EAP) throws the following exception if remote cache store configurations pre-load data into memory:

```
ERROR [org.jboss.msc.service.fail] (ServerService Thread Pool -- 68) MSC000001: Failed to start service org.wildfly.clustering.infinispan.cache.web.jws-session-dist-async.war: org.jboss.msc.service.StartException in service org.wildfly.clustering.infinispan.cache.web.jws-session-dist-async.war: org.infinispan.commons.CacheException: Unable to invoke method public void org.infinispan.persistence.manager.PreloadManager.start() on object of type PreloadManager
```

**Workaround:** Ensure that the remote cache store configuration uses **preload="false"**.

## Data Grid Server Allows REST and Hot Rod Operations from Unauthenticated Users

**Issue:** [JDG-3624](#)

**Description:** It is possible to perform operations to access data and control Data Grid nodes with unauthenticated requests from both REST and Hot Rod endpoints.

**Workaround:** Configure endpoint authentication before you start Data Grid server instances. See [Configuring Endpoint Authentication Mechanisms](#) for more information.

## Errors Occur Using Oracle Database 19c JDBC Drivers with JDK 11

**Issue:** [JDG-3558](#)

**Description:** Data Grid has tested using JDBC drivers, **ojdbc10.jar**, with JDK 11 to create persistent cache stores with Oracle 19c database. Test results indicate that Oracle Database 19c JDBC drivers are not compatible with JDK 11.

**Workaround:** Use **ojdbc8.jar** with JDK 8.

## JBoss Marshalling with Hot Rod Clients in EAP

**Issue:** [JDG-3552](#)

**Description:** **NoClassDefFoundError: sun/reflect/ReflectionFactory** exceptions occur if you package the Data Grid Hot Rod Java client as a WAR file in an JBoss Enterprise Application Platform (EAP) deployment and use JBoss Marshalling.

**Workaround:** Include **WEB-INF/jboss-deployment-structure.xml** in your WAR file as follows:

```
<jboss-deployment-structure xmlns="urn:jboss:deployment-structure:1.1">
  <deployment>
    <dependencies>
      <system export="true">
        <paths>
          <path name="sun/reflect"/>
        </paths>
      </system>
    </dependencies>
  </deployment>
</jboss-deployment-structure>
```

## 4.2. FIXED IN DATA GRID 8.0.1

Data Grid 8.0.1 includes the following fixes:

- [JDG-3784](#) Package `org.infinispan.context` is missing from javadocs.
- [JDG-3767](#) Server does not load user classes with custom root directories.
- [JDG-3748](#) Remote `JCacheManager` does not apply configuration from URI.
- [JDG-3631](#) Single File cache store directory not relative to the global persistent location.
- [JDG-3612](#) Configuration schema do not include the default values.
- [JDG-3598](#) Exceptions occur when performing rolling upgrades of Data Grid clusters.



- [JDG-3599](#) Data Grid Server log files display incorrect version.
- [JDG-3578](#) NullPointerException occurs when installing continuous query listeners.

### 4.3. FIXED IN DATA GRID 8.0

Data Grid 8.0 includes the following fixes:

- [JDG-2992](#) Pessimistic locks can be lost if transactions timeout or are removed.
- [JDG-2861](#) Cache timeout during rebalancing operation.
- [JDG-2612](#) RemoteCache API documentation not clear if lifespan expiration uses Seconds or UnitTime.

## CHAPTER 5. DEPRECATED FEATURES AND FUNCTIONALITY

Support for deprecated functionality is not available beyond the release in which it is deprecated.



### IMPORTANT

Red Hat does not recommend including, enabling, or configuring deprecated functionality in new deployments.

## 5.1. DEPRECATIONS

Data Grid 8.0 deprecates the following features and functionality:

### Memcached Endpoint Connector

As of this release, Data Grid no longer supports the Memcached endpoint. The Memcached connector is deprecated and planned for removal in a future release.



### NOTE

If you have a use case or requirement for the Memcached connector, contact your Red Hat support team to discuss requirements for a future Data Grid implementation of the Memcached connector.

### JBoss Marshalling

JBoss Marshalling is a Serialization-based marshalling library and was the default marshaller in previous Data Grid versions. You should not use serialization-based marshalling with Data Grid but instead use Protostream, which is a high-performance binary wire format that ensures backwards compatibility.

### Externalizers

The following interfaces and annotations are now deprecated:

- **org.infinispan.commons.marshall.AdvancedExternalizer**
- **org.infinispan.commons.marshall.Externalizer**
- **@SerializeWith**



### NOTE

Data Grid ignores **AdvancedExternalizer** implementations when persisting data unless you use JBoss Marshalling.

### Total Order Transaction Protocol

The **org.infinispan.transaction.TransactionProtocol#TOTAL\_ORDER** protocol is deprecated. Use the default 2PC protocol instead.

### Lucene Directory

The functionality to use Data Grid as a shared, in-memory index for Hibernate Search queries is now deprecated.

## Custom Interceptors

The functionality to create custom interceptors with the **AdvancedCache** interface is now deprecated.

## 5.2. REMOVED FEATURES AND FUNCTIONALITY

Data Grid 8.0 no longer includes the following features and functionality that was either deprecated in a previous release or replaced with new components:

- Uberjars (replaced with Maven dependencies and individual JAR files)
- EAP Modules (replaced by the EAP Infinispan subsystem)
- Cassandra Cache Store
- Apache Spark Connector
- Apache Hadoop Connector
- Apache Camel component: **jboss-datagrid-camel-library** is replaced by the **camel-infinispan** component in Red Hat Fuse 7.3 and later.
- REST Cache Store
- REST API v1 (replaced by REST API v2)
- Compatibility Mode
- Distributed Execution
- CLI Cache Loader
- LevelDB Cache Store
- **infinispan-cloud** (replaced by default configuration in **infinispan-core**)
- **org.infinispan.atomic** package
- **getBulk()** methods in the **RemoteCache** API for Hot Rod clients
- JDBC PooledConnectionFactory via C3P0 and HikariCP connection pools
- OSGI support
- **infinispan.server.hotrod.workerThreads** system property
- JON Plugin

## CHAPTER 6. APPENDIX

Additional information for Data Grid 8.0.

### 6.1. EXTERNALIZING HTTP SESSIONS FROM JBOSS WEB SERVER (JWS) TO DATA GRID

Externalize HTTP sessions from JBoss Web Server (JWS) to Data Grid via the Apache Tomcat `org.apache.catalina.Manager` interface.

#### 6.1.1. Installing the Session Client

1. Download the `redhat-datagrid-8.0.0.Final-tomcat<$version>-session-client.zip` archive from the [Data Grid Software Downloads](#) on the Red Hat customer portal.
2. Copy the contents of the `lib/` directory in the archive into `$CATALINA_HOME`.

#### 6.1.2. Configuring the Session Manager

1. Open either `$CATALINA_HOME/conf/context.xml` or `/WEB-INF/context.xml` for editing.
2. Specify the `HotRodManager` class for the Session Manager and define configuration as appropriate.
3. Save and close `context.xml`.

#### Example Configuration

```
<Manager className="org.wildfly.clustering.tomcat.hotrod.HotRodManager"
  configurationName="mycache"
  persistenceStrategy="FINE"
  maxActiveSessions="100"
  server_list="127.0.0.1:11222;127.0.0.1:11223;127.0.0.1:11224"/>
```

#### 6.1.3. HotRodManager Configuration Properties

Property	Description
<code>configurationName</code>	Specifies a cache instance or template defined in <code>\$RHDG_HOME/server/conf/infinispan.xml</code> . Your application then creates a cache instance that uses the configuration from the named cache.
<code>persistenceStrategy</code>	Defines how sessions map to entries in the cache.  <b>COARSE</b> stores all attributes of a session in a single cache entry. This is the default.  <b>FINE</b> stores session attributes in separate cache entries.

Property	Description
<b>maxActiveSessions</b>	Defines the maximum number of sessions to store in the cache. The default is no maximum (limitless).

To configure Hot Rod clients, specify properties without the **infinispan.client.hotrod.** prefix. For more information, see the [Hot Rod Client Configuration API](#).

You can also specify common attributes for the Session Manager. Refer to the appropriate version of the Apache Tomcat documentation as follows:

- [Common Attributes for Tomcat 7.0 Manager](#)
- [Common Attributes for Tomcat 8.0 Manager](#)
- [Common Attributes for Tomcat 8.5 Manager](#)
- [Common Attributes for Tomcat 9.0 Manager](#)