

Oracle® NoSQL Database

Release Notes



Release 22.2

E91818-29

August 2022

ORACLE®

Copyright © 2011, 2022, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Overview

Oracle NoSQL Database is a distributed key-value store capable of scaling horizontally to handle very large amounts of data.

This Release

This is release 22.2.12 of Oracle NoSQL Database.

We strongly recommend that users of earlier versions upgrade to this release. For a complete listing of all the changes made to Oracle NoSQL Database for this release, including all bug fixes, see the change log.

Supported Java Versions

The Oracle NoSQL Database server is compatible with Java SE 8 (64-bit), and the client is compatible with Java SE 8. Both the client and the server require at least Java SE 8, and should work with more recent Java SE versions. The client and server have been tested and certified against Oracle Java SE 17.0.3.1 and OpenJDK 17.0.2. We encourage you to upgrade to the latest Java release to take advantage of the latest bug fixes and performance improvements. We expect to certify Oracle NoSQL Database with more recent Java versions in a future release.

Attempting to use this release with a version of Java earlier than the required version will produce an error message similar to:

```
Exception in thread "main" java.lang.UnsupportedClassVersionError:  
    oracle/kv/impl/util/KVStoreMain : Unsupported major.minor version  
    52.0
```

Product Upgrade Policy

Release 22.2 supports upgrades from release 22.1, and from releases for the prior two calendar years. To upgrade a store directly to the current release, the store must be running release 20.1 or later.

If you have a store running a 19.x release, we recommend that you upgrade it to the current release by first upgrading to the 21.2 or 20.3 release (or another 21.x or 20.x release) and then upgrading from that to the current release. If you have a store running an 18.x release, we recommend that you upgrade to the 20.3 release (or another 20.x release) and then upgrade from that to the current release. If you have a store running a 4.x release, we recommend that you first upgrade to the 19.5 release (or another 19.x release) and then follow our recommendations for upgrading from 19.x. If you have a store running a release earlier than the 4.x release and you are an Enterprise Edition user, please contact support. If you are a Community Edition user with this issue, please post a question to the NoSQL Database Discussions forum on Oracle Communities.

New Features in Release 22.2

Removed Import/Export Utility

As of release 22.2 the import/export utility is no longer part of the release. It was previously deprecated and its functionality has been moved to the data migrator utility, which is an independent download. Chapter 7 of the Oracle NoSQL Database Administrator's Guide contains details on using the data migrator as well as a comparison between the import/export utility and the data migrator.

Erasing Obsolete User Data Has Been Enabled By Default

Data erasure, added in 21.2, will now be enabled by default. This feature erases obsolete user data within a specified time period after becoming obsolete, without adding significant performance impact to the application.

Two Rep Node parameters control erasure: "enableErasure" is used to enable erasure, which is now enabled by default, and "erasurePeriod" specifies the time required to complete an erasure cycle, which defaults to 6 days.

Note:

Snapshot-based backups create hard links to original files. Until these backups are copied to their target location and the corresponding hard links are removed, erasure will not process obsolete data in those files. This is to preserve the integrity of the backups. Erasure ignores files with hard links to them.

New Features in Release 22.1

JSON MRCounter Types

The MR_COUNTER SQL type can now be used in JSON fields for multi-region tables. The change adds support for merging independent modifications made in multiple regions into a single combined value to schemaless JSON fields.

Elasticity for MR Tables

Stores that contain multi-region tables now support elasticity changes that change the store topology.

Child tables for MR Tables

Multi-region tables now support child tables. Child tables always inherit the regions of the associated top level tables.

KVLocal

Applications can now create and manage a small, standalone store instance using the new KVLocal class.

Deprecating Import/Export Utility

Release 22.1 is the final release to include the import/export utility. It will be removed as of the next quarterly release. We have moved the functionality of the import/export utility over to the data migrator. Chapter 7 of the Oracle NoSQL Database Administrator's Guide contains details on using the data migrator as well as a comparison between the import/export utility and the data migrator. The import/export utility has been deprecated since the 20.3 release and we encourage customers to move to the new data migrator utility now.

New Features in Release 21.2

Unnesting of Arrays and Maps in Queries

The UNNEST clause has been introduced, which converts an array or map into a series of rows. The syntax of the FROM clause has also been extended to allow specifying a list of expressions after the table or tables used by the query, which can influence which indexes the query runtime selects for queries that have UNNEST clauses.

Erasing User Data

Added support for erasing obsolete user data within a specified time period after becoming obsolete, without adding significant performance impact to the application. This new feature, data erasure, runs in the background to zero out obsolete data. All data that is obsolete from the database point of view is removed: i.e. deleted data, older versions of updated records, and expired data. Data erasure also removes corresponding obsolete index records.

Two new Rep Node parameters control erasure: "enableErasure" is used to enable erasure, which is disabled by default, and "erasurePeriod" specifies the time required to complete an erasure cycle.

Note:

Snapshot-based backups create hard links to original files. Until these backups are copied to their target location and the corresponding hard links are removed, erasure will not process obsolete data in those files. Erasure ignores files with hard links to them.

Deprecating Property Graph Package

Starting with release 21.2, the RDF Property Graph package is deprecated and is no longer certified with Oracle NoSQL Database.

Disabling DSA (ssh-dss) SSH Keys in the Diagnostics Utility

The diagnostics utility no longer supports DSA (ssh-dss) SSH keys due to security weaknesses with that algorithm. Attempting to use the diagnostics utility with a DSA (ssh-dss) SSH key will produce an error message similar to:

```
DefaultAuthFuture[ssh-connection]: Failed (IllegalArgumentException) to  
execute: No signer could be located for key type=ssh-dss
```

In a future version, RSA (ssh-rsa) SSH key support may also be removed for similar reasons. It's recommended to migrate to other SSH key algorithms, for example ECDSA or ED25519.

New Features in Release 21.1

MRCounter Type

Added support for the new `MR_COUNTER` SQL type, which can be used with `INTEGER`, `LONG` and `NUMBER` value types in multi-region tables. The new type supports merging independent modifications made in multiple regions into a single combined value.

Left Outer Join

Added support for the `LEFT OUTER JOIN` syntax to be compatible with standard ANSI SQL.

Indexes for Nested Arrays and Maps

This release adds support for indexing arrays and maps that are nested within other arrays or maps.

New SQL Functions

Added new SQL functions to access information about row storage size, index storage size, modification time, containing partition, and containing shard. Also added a function to convert a string to JSON.

New Features in Release 20.3

Universally Unique Identifier

Added SQL support for randomly generated UUIDs, including the `random_uuid()` function, the new "AS UUID" option for STRING data types, and the new "GENERATED BY DEFAULT" option for "STRING AS UUID" data types. These new facilities provide an easy way to create a field that contains a unique ID for each row. UUID fields can be used to create unique primary keys in multi-region tables, since multi-region tables do not support the IDENTITY fields that serve this purpose for local tables.

Deprecating Import/Export Utility

In the Oracle NoSQL Database 20.2 release, we released a new Data Migrator utility. We have moved the functionality of the import/export utility over to the data migrator. The Using Oracle NoSQL Data Migrator section in the *Administrator's Guide* contains details on using the data migrator as well as a comparison between the import/export utility and the data migrator. We are deprecating the import/export utility as of the 20.3 release and encourage customers to use the new data migrator utility.

New Features in Release 20.2

Generic ORDER BY and GROUP BY

In previous releases, ORDER BY and GROUP BY were possible only if there was an index that sorted the rows by the ORDER BY/GROUP BY expressions. Furthermore, it was not possible for queries to include both ORDER BY and GROUP BY. These restrictions are lifted in this release. Note that generic ORDER BY and GROUP BY may consume a lot of driver memory, because of the need to materialize the full query result in driver memory.

SELECT DISTINCT

The DISTINCT keyword is now supported in the SELECT clause. If present, duplicate results will be removed from query result set. As with generic ORDER BY and GROUP BY, SELECT DISTINCT needs to cache the full result set in driver memory.

TTL in Streams API and Multi-Region Tables

Added support for Time to Live (TTL) in the Streams API and in Multi-Region tables. The Rows supplied by the Streams API to subscribers will include the TTL expiration time for the operation from source store. For Multi-Region tables, the TTL expiration time of each row is replicated to other regions, and thus a row in Multi-Region table in any region will expire at the expiration time replicated from the region where and when operation is made.

New Features in Release 20.1

IN Operator

Added support for the IN operator in SQL for Oracle NoSQL Database. The IN operator provides a convenient way to match one of several equality conditions. For example, the query

```
SELECT * FROM Foo WHERE a IN (1, 5, 4)
```

is equivalent to

```
SELECT * FROM Foo WHERE a = 1 OR a = 5 OR a = 4
```

Untyped JSON indexes

Before this release, indexing a field inside a JSON document required that the field have a concrete JSON atomic type - one of integer, long, double, number, string, or boolean. If, for some document, the index field value did not conform to the declared type, index creation or the insertion of that document in a table would fail. This release adds support for the new anyAtomic type as a valid type for indexed JSON fields. A field indexed with the anyAtomic type can be of any valid JSON atomic type and the index will store all the heterogeneous values of that field.

General Availability of Multi-Region Tables

Multi-Region Tables have received several improvements in this release and is now considered a general availability feature.

New additions and improvements:

- Regions can now be added or removed from an existing multi-region table.
- A multi-region table can be dropped in one region and the associated tables can remain in other regions.
- Regions can now be added or removed from an existing multi-region table.
- When creating a multi-region table, the associated tables in remote regions no longer need to be empty.



Note:

Multi-Region tables are now only available in the Enterprise Edition.

New Features in Release 19.5

Multi-Region Tables

Multi-Region Tables are a new feature that lets users create "read-anywhere" and "write-anywhere" tables that live in multiple regions, where each region is a separate Oracle NoSQL Database store. This is a preview release to give users early access to the new feature. It is not recommended to deploy this version of multi-region tables in a production environment. A general availability version will be available in a future release. Data stored in multi-region tables created with this release may need to be discarded when upgrading to the general availability version. Please refer to the Admin Guide and related documents for more information on the new feature and see the Known Issues section below for a list of limitations in this release.

Asynchronous Table API Methods

The table API now has asynchronous methods that applications can use to make calls without using a thread to wait for results, which can improve the efficiency of clients that make many concurrent calls.

The client uses a new network protocol that multiplexes multiple calls on the same socket which supports asynchronous operations and reduces the number of socket connections needed. The new client is only compatible with this version of the server, although the new server continues to support older clients.

String manipulation SQL functions

A set of new functions have been added to SQL queries to help with string manipulation: concatenation, substr, trim, ltrim, rtrim, length, contains, starts_with, ends_with, index_of, replace, reverse, upper, lower.

Deprecating "kvproxy drivers"

In the 19.3 release, Oracle NoSQL Database released a database proxy component that lets Oracle NoSQL Database drivers communicate with the Oracle NoSQL Database cluster, called the "HTTP proxy drivers." Prior to the 19.3 release, there was another set of Oracle NoSQL Database drivers that used a thrift based proxy for communication, called the "kvproxy drivers." These drivers are available in Java, C#, Javascript, and Python. The "HTTP proxy drivers" provide richer capabilities and will be usable for both on-prem and cloud applications. The "kvproxy drivers" have been deprecated in this release and are subject to removal from future versions. They should be used with that possibility in mind. The APIs are different between the "kvproxy drivers" and the "HTTP proxy drivers", so existing applications will need to be modified to use the "HTTP proxy drivers." Deprecated drivers will be clearly marked on the Oracle Technology Network.

New Features in Release 19.3

String pattern matching in queries

Queries now support a new built-in function, `regex_like`, which can be used to specify string pattern matching in queries.

Hybrid Cloud Deployment

Oracle NoSQL Database can now be used with the drivers for the Oracle NoSQL Database Cloud Service by deploying the new Oracle NoSQL Database Proxy. The JAR file for the Oracle NoSQL Database Proxy is included in the Enterprise Edition distribution of Oracle NoSQL Database. All users can download the latest version of the JAR from the Oracle Technology Network.

New Features in Release 19.1

JSON datatype support for Full Text Search

Users can create full text search indexes on a JSON field and on the attributes within that JSON field.

Export/Import Enhancement

The Export/Import tool has been enhanced to allow users to migrate their data into Oracle NoSQL Database using either JSON or MongoDB JSON formats.

Deprecating Export/Import support for Oracle Storage Cloud Services

Export/Import support for the Oracle Storage Cloud Service that is part of Oracle Cloud Infrastructure Classic has been deprecated for use as a source or sink. Support for the cloud storage that is part of the new Oracle Cloud Infrastructure will be provided in the future.

New Features in Release 18.3

Querying GeoJson Data

Introduced support for a number of built-in functions in SQL queries to interpret JSON objects that represent geographical locations as specified in the GeoJSON specification (<https://tools.ietf.org/html/rfc7946>).

Support for Namespaces

Starting this release users can create namespaces. Tables can be created for a particular namespace. If users do not specify a namespace at the time of table creation the tables are placed under the default namespace of `sysDefault`. Prior to this release there was no concept of namespace in Oracle NoSQL Database. All tables created using prior releases will be placed in `sysDefault` namespace on upgrade.

IDENTITY Column Support

Users can create a table with an IDENTITY column for which the system will automatically generate numeric values using a sequence generator.

Support for INSERT and UPSERT in SQL Query

Support for INSERT and UPSERT statements has been added in SQL queries.

Support for Sequence Aggregation Functions in SQL Query

The following new functions have been added to support sequence aggregation operations in SQL queries: `seq_count`, `seq_sum`, `seq_avg`, `seq_min`, and `seq_max`.

Admin Web Service

Oracle NoSQL Database Admin can now be started as a web service that processes Admin CLI commands through a REST API over the HTTP and HTTPS protocol. The input and output to the Admin Web Service is in JSON format. This feature is intended to help DevOps and other NoSQL admins automate administrative operations. See [REST API for Administering Oracle NoSQL Database](#).

Data Migrator - Preview

This release includes a preview version of the Data Migrator utility, which allows users to migrate their data into Oracle NoSQL Database that is either in JSON or MongoDB JSON formats. This utility will be integrated into the Oracle NoSQL Database IMPORT/EXPORT utility in future releases. See the `MIGRATOR-README.txt` found in the `doc` subdirectory of the Oracle NoSQL Database package for its usage.

Monitoring Using ELK Framework

Oracle NoSQL Database can be setup to generate logs that can be used by the popular ELK (Elasticsearch - Logstash - Kibana) framework to visually monitor NoSQL clusters.

Support for Encryption at Rest

Oracle NoSQL Database can be setup to encrypt its data using dm-crypt, a kernel-level disk encryption mechanism available for Linux.

New Features in Release 18.1

Support for Group-by and Aggregate Function

This release introduces support for the group-by clause and the following aggregate functions: count(*), count(expr), sum(expr), avg(expr), min(expr) and max (expr).

Joining Tables in the Same Table Hierarchy

Support for joining tables that belong to the same table hierarchy — parent-child table joins — is now available.

Zone Affinity

Users can specify master affinity for the primary zones that are in close proximity to the client application. This feature allows the system to favor placing master nodes in primary zones that have master affinity set so that write operations are directed to nearby zones for more predictable latency.

Distributions

Oracle NoSQL Database comes in three distributions: Community Edition (CE), Enterprise Edition (EE), and Client.

- The CE version is open source. It ships with source code and is released under the Apache License, Version 2.0 (Apache 2.0). It includes the client and server distribution, but does not include some server features.
- The EE version does not include source code and it ships with an Oracle license. The EE version includes the client and server distribution, and includes several features not found in the CE version: Oracle External Tables support, Oracle Wallet support for external password storage, and support for Kerberos authentication.
- The Client version is open source. It ships with source code and is released under the Apache 2.0 License (Apache 2.0). The Client version only contains the implementation of the client API, which may be used to access servers running under CE or EE.

In addition, example code is provided in its own package, which can be used with any of the distributions.

Building from Source Code in the Community Edition

If you have downloaded the Community Edition, you can use the included Ant build script to rebuild JAR files if you make source changes.

You need Apache Ant version 1.9.0 or later in order to do builds. You can download Ant from [here](#).

You also need Apache Ivy for loading dependencies. You can download Ivy from [here](#).

Make sure to add the ivy JAR file to your `~/ .ant/lib` directory or specify the ivy directory using the `ant -lib` option.

The default Ant target will rebuild all JAR files.

Known Issues

Topics

- [Topology Changes May Fail During Software Upgrades](#)
- [Enterprise Manager plug-in not compatible with EM 13.4.0.0 and later](#)
- [Limitations on Multi-Region Tables in This Release](#)
- [Updating Java Memory Settings after Release 18.1 Workaround](#)
- [Out-of-Order Processing During Streams API and Partition Migration](#)
- [Hive Used with Oracle Big Data SQL is Incompatible with Java 9, 10 and 11](#)
- [Import/Export to OCI using Migrator Tool requires Java 8](#)
- [Pre 19.1 Non-Java Drivers Still Require Java 8](#)
- [IDENTITY Column Definition Missing in Export Package](#)
- [Export Hangs When Disk is Full at Sink](#)
- [Need a Minimum of 5 GB of Free Disk Space to Deploy a Storage Node That Hosts an Admin](#)
- [Users Must Manage Admin Directory Size, Can Put All Admins Into "RUNNING,UNKNOWN" State](#)
- [Store With Full Text Search May Become Unsynchronized](#)
- [Data Verifier is Disabled By Default](#)
- [Subscription Cannot Connect and Fails With InternalException](#)

Topology Changes May Fail During Software Upgrades

Making modifications to the store topology that include partition migration may fail if the modifications are performed while the store is being upgraded to a new software version. If you run a plan to deploy a new topology and the plan fails with problems during partition migration, check if the nodes of the store are running different software versions, and upgrade any nodes running old versions before retrying the plan.

Modifying a topology using one of the following topology commands can result in the need for partition migration. Deploying the resulting topology with the 'plan deploy-topology' command can then fail if the plan is performed during a store software version upgrade. The topology commands that can produce partition migrations are:

- topology change-refactor
- topology contract
- topology rebalance
- topology redistribute

Other topology commands do not produce partition migration and do not cause this problem.

If a topology deployment fails, you can tell if it is related to partition migrations during a software version upgrade by looking for errors like the following:

```
Plan 24 ended with errors. Use "show plan -id 24" for more information
Plan Deploy Topo
Id:                24
State:             ERROR
Attempt number:    1
Started:           2020-04-10 15:19:59 UTC
Ended:             2020-04-10 15:24:48 UTC
Plan failures:
  Failure 1: 17/MigratePartition PARTITION-2 from rg1 to rg2
  failed. target=rg2-rn1 state=ERROR java.lang.Exception:
  Migration of PARTITION-2 failed. Giving up after 10 attempt(s)
```

If you see a plan failure involving partition migrations like this, particularly if there are similar failures for all partition migration tasks, use the 'ping' or 'verify topology' commands to display information about the store and check to see if different storage nodes are running different major or minor software versions. If so, upgrade the nodes running the older software to the latest version before retrying the 'plan deploy-topology' command.

Enterprise Manager plug-in not compatible with EM 13.4.0.0 and later

Oracle NoSQL's Enterprise Manager (EM) plug-in is compatible with EM versions up to and including EM version 13.3.0.0. Because of architectural changes in EM's plug-in support, the plugin is not compatible with EM version 13.4.0.0 and subsequent versions.

[KVSTORE-141]

Limitations on Multi-Region Tables in This Release

The Multi-Region Tables feature in this release has the following limitations:

- Specifying a non-zero TTL when inserting or updating a row in a Multi-Region table is only supported after upgrading the driver, and may fail until the local store has been completely upgraded. In addition, TTL expiration times will be lost when rows are replicated to a remote region if the multi-region agent or store for that region have not been upgraded. *[#28165]*
- Only one service agent is supported for each remote region. *[#28166]*
- Elasticity operations must not be performed on stores that contain multi-region tables. *[#28164]*
- If the multi-region agent is unable to replicate data from a remote region for a long period of time, either due to network failure, a store failure, or a failure of the agent for other reasons, that may prevent table entries deleted in that remote region during the failure period from being deleted in the local region. *[#28136]*
- The import, export, and snapshot commands should not be used to restore multi-region tables. The commands do not currently account for region information or modification times, so using these commands to restore a multi-region table to the contents from an earlier time may produce inconsistent results. *[KVSTORE-444]*
- Multi-region tables created using the 19.5 release should not be used in this release or later releases. Conflict resolution for data in tables created in the 19.5

release may not be resolved correctly in later releases, meaning that multi-region tables may not contain the latest entry updated in a remote table. Note that upgrading of multi-region tables to later releases also received limited testing, so there may be other issues that have not been detected.

- Only one XRegion Service agent is allowed for a store. *[KVSTORE-984]*
- Initialization of multi-region table may lose deletion in remote region. *[KVSTORE-986]*

We expect all of these limitations to be removed in the future releases.

Updating Java Memory Settings after Release 18.1 Workaround

Starting with release 18.3, the Java heap overhead is explicitly accounted for via the new Storage Node parameter named `jvmOverheadPercent`, with a default value of 25%. If you are running a store using a version earlier than 18.3, and the store was configured with the workarounds suggested in the [Memory Allocation Algorithm Fails to Account for Java Memory Overhead Can Produce OutOfMemoryErrors](#) section of the 18.1 release notes, then you should make the following changes during the upgrade to an 18.3 or later release. The changes to make depends on whether you followed the first or second set workarounds, based on whether your configuration has more than 48 GiB of memory per RN.

If you used the first set of instructions in the release notes because your configuration no more than 48 GiB of memory per RN, then immediately *before* upgrading the store to release 18.3 or a later release, run the following Admin CLI commands:

1. `change-policy -params rnHeapPercent=68`
2. For each storage node, replacing `snX` as appropriate:

```
plan change-parameters -service snX -wait -params rnHeapPercent=68
```

3. *After* the upgrade, run the following Admin CLI command for each storage node, replacing `snX` as appropriate:

```
plan change-parameters -service snX -wait -params memoryMB=0
```

You are done.

If you used the second set of instructions in the release notes because your configuration has more than 48 GiB of memory per RN, then run the following Admin CLI commands *after* the upgrade:

1. `change-policy -params systemPercent=10`
2. For each storage node, replacing `snX` as appropriate:

```
plan change-parameters -service snX -wait -params systemPercent=10  
memoryMB=0
```

You are done.

Note that making changes to multiple Storage Nodes to update Java memory settings may result in warnings in the debug logs regarding mismatched cache sizes such as:

```
2019-11-14 15:26:40.762 UTC WARNING - [rg1-rn3] JE: Mismatched cache
sizes, feeder:516738252 replica: 375809638 feeder off-heap: 0 replica
off-heap: 0
```

Once the changes are completed for all Storage Nodes, these warnings should not continue to be reported, and the temporary ones should be harmless.

[#27855]

Out-of-Order Processing During Streams API and Partition Migration

When an application uses the Streams API with a subscription that has multiple subscribers, and an elasticity operation is performed that involves a partition migration, the application may need to coordinate operations across subscribers. An elasticity change can cause the events being delivered for a given key to switch to a different subscriber. The Streams API delivers events in the proper order to the two subscribers, but it is up to the application to make sure that the subscribers perform actions for those events in the correct order. We hope to remove the need for this coordination in a future release.

[#27541]

Hive Used with Oracle Big Data SQL is Incompatible with Java 9, 10 and 11

Oracle NoSQL Database supports Oracle Big Data SQL using Apache Hive (TM) 1.2.1 and Hadoop-2.3.0-cdh5.1.0. The following warnings are generated when you use Java 9, 10, or 11 to start Hive:

```
Logging initialized using configuration in file:/scratch/kmtest/
release/hadoop/hive/conf/hive-log4j.properties
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by
org.apache.hadoop.security.authentication.util.KerberosUtil (file:/
scratch/kmtest/release/hadoop/hadoop-2.6.0-cdh5.4.8/share/hadoop/
common/lib/hadoop-auth-2.6.0-cdh5.4.8.jar) to method
sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of
org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further
illegal reflective access operations
WARNING: All illegal access operations will be denied in a future
release
Exception in thread "main" java.lang.RuntimeException:
java.lang.RuntimeException: Unable to instantiate
org.apache.hadoop.hive.ql.metadata.SessionHiveMetaStoreClient
    at
org.apache.hadoop.hive.ql.session.SessionState.start(SessionState.java:
522)
    at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:677)
    at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:621)
    ...
Caused by: java.lang.RuntimeException: Unable to instantiate
org.apache.hadoop.hive.ql.metadata.SessionHiveMetaStoreClient
```

```
    at
org.apache.hadoop.hive.metastore.MetaStoreUtils.newInstance(MetaStoreUtils.java:1523)
    at
org.apache.hadoop.hive.metastore.RetryingMetaStoreClient.<init>(RetryingMetaStoreClient.java:86)
    at
org.apache.hadoop.hive.metastore.RetryingMetaStoreClient.getProxy(RetryingMetaStoreClient.java:132)
```

The incompatibility warnings are produced because this version of Hive is incompatible with module support introduced in Java 9. Use Java 8 to bring up Hive for Big Data SQL queries with Oracle NoSQL Database 19.1.

[#27565]

Import/Export to OCI using Migrator Tool requires Java 8

If you are using the Oracle NoSQL Database on-premises migrator tool and you want to import/export data to OCI *classic* object store then you must use Java 8. Other versions of Java are not supported.

Pre 19.1 Non-Java Drivers Still Require Java 8

Due to an incompatibility issue in the proxy, the Oracle NoSQL non-Java drivers released prior to 19.1 must continue to use Java 8 to run the proxy when connecting to an Oracle NoSQL Database 19.1 server that uses Java 10 or 11.

IDENTITY Column Definition Missing in Export Package

The Import/Export utility does not export the `IDENTITY` column property for a table into the export package DDL file (`tableSchema.ddl`). This is a bug and will be fixed in a future release. The user will notice the missing `IDENTITY` column property only during an import into an existing table using the export package. Here are possible scenarios:

1. If the import table already exists and is non-empty, and the `IDENTITY` column is defined as `GENERATED ALWAYS`, the Oracle NoSQL Database will return an error saying that users cannot supply a value for `GENERATED ALWAYS`.
2. If the import table already exists and is non-empty, and the `IDENTITY` column is defined as `GENERATED BY DEFAULT`, the Import/Export utility will return an error saying that the record is already present. The user can choose to overwrite the records by setting the import config file option `overwrite` to `true`.
3. If the import table exists and is empty, and the `IDENTITY` column is defined as `GENERATED ALWAYS`, the Oracle NoSQL Database will return an error saying that users cannot supply a value for `GENERATED ALWAYS`.
4. If the import table exists and is empty, and the `IDENTITY` column defined as `GENERATED BY DEFAULT`, the import will succeed, taking the values from the export package. The user can then set the `START WITH` value to the next value in the sequence using the `ALTER TABLE` command.
5. If the import table does not exist, then import will create the table using the DDL in the export package that had the missing `IDENTITY` column property, thus losing knowledge of the original `IDENTITY` column. This problem will be fixed in a future release. The import will succeed as per the semantics of a table without an `IDENTITY` column.

For all of these options, you can add or modify the `IDENTITY` column property using the `ALTER TABLE` command. See `IDENTITY` column documentation for more details.

[#27562]

Export Hangs When Disk is Full at Sink

During an export, the Import/Export tool will hang if the sink runs out of disk space. This issue will be fixed in a future release. Users must restart the export after freeing up disk space at sink. The user will see a `java.io.IOException: No space left on device` if they had started export in `-verbose` mode.

```
java -jar /home/jinzha/mywork/kv/lib/kvtool.jar export -helper-hosts
192.168.56.1:5000 \
-store kvstore -export-all -config /home/jinzha/mywork/export.cfg -
verbose
Enter command: export
2019-04-22 23:55:16.316 UTC Start migration with configuration:
{
  "configFileVersion" : 1,
  "abortOnError" : true,
  "source" : {
    "type" : "nosqldb",
    "helperHosts" : [ "192.168.56.1:5000" ],
    "storeName" : "kvstore"
  },
  "sink" : {
    "type" : "file",
    "format" : "binary",
    "path" : "/home/jinzha/mywork/data"
  }
}
2019-04-22 23:55:16.338 UTC TaskWaiter thread spawned.
2019-04-22 23:55:16.693 UTC Exporting table schema: users.
TableVersion: 1
2019-04-22 23:55:16.695 UTC Creating a new RecordStream for
SchemaDefinition. File segment number: 1. Chunk sequence: abcdefghijkl
2019-04-22 23:55:16.701 UTC WriteTask worker thread spawned for
SchemaDefinition
2019-04-22 23:55:16.704 UTC [binary]: Exported 1 record from
tableSchema: 0min 0sec 361ms
2019-04-22 23:55:16.729 UTC Exporting store data with configuration:
consistency=null; requestTimeout=0ms
2019-04-22 23:55:16.773 UTC Creating a new RecordStream for users.
File segment number: 1. Chunk sequence: abcdefghijkl
2019-04-22 23:55:16.788 UTC WriteTask worker thread spawned for users
2019-04-22 23:55:18.954 UTC Exception exporting users. Chunk sequence:
abcdefghijkl
java.io.IOException: No space left on device
  at java.io.FileOutputStream.writeBytes(Native Method)
  at java.io.FileOutputStream.write(FileOutputStream.java:326)
  at
oracle.kv.util.expimp.utils.exp.LocalStoreOutput.exportDataStream(Local
StoreOutput.java:211)
  at
```

```
oracle.kv.util.expimp.utils.exp.LocalStoreOutput.doExport(LocalStoreOutput.java:149)
    at
oracle.kv.util.expimp.utils.exp.AbstractStoreOutput$WriteTask.call(AbstractStoreOutput.java:639)
    at
oracle.kv.util.expimp.utils.exp.AbstractStoreOutput$WriteTask.call(AbstractStoreOutput.java:620)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)
2019-04-22 23:56:06.705 UTC Exception exporting SchemaDefinition. Chunk sequence: abcdefghijkl
java.io.IOException: No space left on device
    at java.io.FileOutputStream.writeBytes(Native Method)
    at java.io.FileOutputStream.write(FileOutputStream.java:326)
    at
oracle.kv.util.expimp.utils.exp.LocalStoreOutput.exportDataStream(LocalStoreOutput.java:211)
    at
oracle.kv.util.expimp.utils.exp.LocalStoreOutput.doExport(LocalStoreOutput.java:149)
    at
oracle.kv.util.expimp.utils.exp.AbstractStoreOutput$WriteTask.call(AbstractStoreOutput.java:639)
    at
oracle.kv.util.expimp.utils.exp.AbstractStoreOutput$WriteTask.call(AbstractStoreOutput.java:620)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)
2019-04-22 23:56:16.708 UTC [binary]: Writing continue.., wait 1 minutes
```

[#27574]

Need a Minimum of 5 GB of Free Disk Space to Deploy a Storage Node That Hosts an Admin

If a Storage Node that hosts an admin is deployed on a system with less than 5 GB of free disk space, the following exception will occur:

```
Connected to Admin in read-only mode
(JE 18.1.8) Database AdminSchemaVersion not found. (18.1.3)
```

Make sure you have at least 5 GB of free disk space to successfully deploy a storage node. This same problem will occur when deploying KVLite. We expect to remove this restriction in a future release. [#26818]

Users Must Manage Admin Directory Size, Can Put All Admins Into "RUNNING,UNKNOWN" State

Every Admin is allocated a maximum of 3 GB of disk space by default, which is sufficient space for the vast majority of installations. However, under some rare circumstances you might want to change this 3 GB limit, especially if the Admin is sharing a disk with a Storage Node. For more information, see [Managing Admin Directory Size](#).

If Admins run out of disk space, then there will be entries in the Admin logs saying "Disk usage is not within je.maxDisk or je.freeDisk limits and write operations are prohibited" and the output of the ping command will show all the Admins in the "RUNNING,UNKNOWN" state. Follow the procedure described in [Managing Admin Directory Size](#) to bring the Admins back to the "RUNNING,MASTER" or "RUNNING,REPLICA" state.

Below is sample output of the ping command and log entries that indicate that Admin ran out of disk space.

```
kv-> ping
Connected to Admin in read-only mode
Pinging components of store kvstore based upon topology sequence #106
90 partitions and 3 storage nodes
Time: 2018-04-03 08:20:22 UTC   Version: 18.3.0
Shard Status: healthy:3 writable-degraded:0 read-only:0 offline:0
total:3
Admin Status: read-only
Zone [name=Houston id=zn1 type=PRIMARY allowArbiters=false
masterAffinity=false]
  RN Status: online:9 offline:0 maxDelayMillis:0 maxCatchupTimeSecs:0
Storage Node [sn1] on localhost:10000
  Zone: [name=Houston id=zn1 type=PRIMARY allowArbiters=false
masterAffinity=false]
  Status: RUNNING   Ver: 18.3.0 2018-04-03 05:36:25 UTC   Build id:
ec627ef967d6 Edition: Enterprise
    Admin [admin1]           Status: RUNNING,UNKNOWN
    Rep Node [rg1-rn1]       Status: RUNNING,REPLICA
sequenceNumber:93 haPort:10011 delayMillis:0 catchupTimeSecs:0
    Rep Node [rg2-rn1]       Status: RUNNING,REPLICA
sequenceNumber:93 haPort:10012 delayMillis:0 catchupTimeSecs:0
    Rep Node [rg3-rn1]       Status: RUNNING,MASTER
sequenceNumber:92 haPort:10013
Storage Node [sn2] on localhost:11000
  Zone: [name=Houston id=zn1 type=PRIMARY allowArbiters=false
masterAffinity=false]
  Status: RUNNING   Ver: 18.3.0 2018-04-03 05:36:25 UTC   Build id:
ec627ef967d6 Edition: Enterprise
    Admin [admin2]           Status: RUNNING,UNKNOWN
    Rep Node [rg1-rn2]       Status: RUNNING,REPLICA
sequenceNumber:93 haPort:11021 delayMillis:0 catchupTimeSecs:0
    Rep Node [rg2-rn2]       Status: RUNNING,MASTER
sequenceNumber:93 haPort:11022
```

```
Rep Node [rg3-rn2]      Status: RUNNING,REPLICA sequenceNumber:92
haPort:11023 delayMillis:0 catchupTimeSecs:0
Storage Node [sn3] on localhost:12000
Zone: [name=Houston id=zn1 type=PRIMARY allowArbiters=false
masterAffinity=false]
Status: RUNNING Ver: 18.3.0 2018-04-03 05:36:25 UTC Build id:
ec627ef967d6 Edition: Enterprise
Admin [admin3]         Status: RUNNING,UNKNOWN
Rep Node [rg1-rn3]     Status: RUNNING,MASTER sequenceNumber:93
haPort:12011
Rep Node [rg2-rn3]     Status: RUNNING,REPLICA sequenceNumber:93
haPort:12012 delayMillis:0 catchupTimeSecs:0
Rep Node [rg3-rn3]     Status: RUNNING,REPLICA sequenceNumber:92
haPort:12013 delayMillis:0 catchupTimeSecs:0
```

```
2018-04-03 08:18:52.254 UTC SEVERE [admin1] JE: Disk usage is not within
je.maxDisk or je.freeDisk limits and write operations are prohibited:
maxDiskLimit=2,097,152 freeDiskLimit=5,368,709,120
adjustedMaxDiskLimit=2,097,152 maxDiskOverage=83,086
freeDiskShortage=-6,945,071,104 diskFreeSpace=12,313,780,224
availableLogSize=-83,086 totalLogSize=2,180,238 activeLogSize=2,180,238
reservedLogSize=0 protectedLogSize=0 protectedLogSizeMap={}
```

```
2018-04-03 08:19:34.808 UTC SEVERE [admin2] JE: Disk usage is not within
je.maxDisk or je.freeDisk limits and write operations are prohibited:
maxDiskLimit=2,097,152 freeDiskLimit=5,368,709,120
adjustedMaxDiskLimit=2,097,152 maxDiskOverage=97,346
freeDiskShortage=-6,944,923,648 diskFreeSpace=12,313,632,768
availableLogSize=-97,346 totalLogSize=2,194,498 activeLogSize=2,194,498
reservedLogSize=0 protectedLogSize=0 protectedLogSizeMap={}
```

```
2018-04-03 08:19:36.063 UTC SEVERE [admin3] JE: Disk usage is not within
je.maxDisk or je.freeDisk limits and write operations are prohibited:
maxDiskLimit=2,097,152 freeDiskLimit=5,368,709,120
adjustedMaxDiskLimit=2,097,152 maxDiskOverage=101,698
freeDiskShortage=-6,944,923,648 diskFreeSpace=12,313,632,768
availableLogSize=-101,698 totalLogSize=2,198,850 activeLogSize=2,198,850
reservedLogSize=0 protectedLogSize=0 protectedLogSizeMap={}
```

[#26922]

Store With Full Text Search May Become Unsynchronized

A store that has enabled support for Full Text Search may, on rare occasions, encounter a bug in which internal components of a master Replication Node become unsynchronized, causing updates from that Replication Node to stop flowing to the Elasticsearch engine. This problem will cause data to be out of sync between the store and Elasticsearch.

When the problem occurs, the Elasticsearch indices stop being populated. The problem involves the shutdown of the feeder channel for a component called the TextIndexFeeder, and is logged in the debug logs for the Replication Node. For example:

```
2018-03-16 11:23:46.055 UTC INFO [rg1-rn1] JE: Inactive channel:
TextIndexFeeder-rg1-rn1-b4e92291-3c73-4128-9557-62dbd4e9ac78 (2147483647)
forced close. Timeout: 10000ms.
```

```

2018-03-16 11:23:46.059 UTC INFO [rg1-rn1] JE: Shutting down feeder
for replica TextIndexFeeder-rg1-rn1-
b4e92291-3c73-4128-9557-62dbd4e9ac78 Reason: null write time: 32ms
Avg write time: 100us
2018-03-16 11:23:46.060 UTC INFO [rg1-rn1] JE: Feeder Output for
TextIndexFeeder-rg1-rn1-b4e92291-3c73-4128-9557-62dbd4e9ac78 soft
shutdown initiated.
2018-03-16 11:23:46.064 UTC WARNING [rg1-rn1] internal exception
Expected bytes: 6 read bytes: 0
com.sleepycat.je.utilint.InternalException: Expected bytes: 6 read
bytes: 0
    at
com.sleepycat.je.rep.subscription.SubscriptionThread.loopInternal(Subsc
riptionThread.java:719)
    at
com.sleepycat.je.rep.subscription.SubscriptionThread.run(Subscrip
tionThread.java:180)
Caused by: java.io.IOException: Expected bytes: 6 read bytes: 0
    at
com.sleepycat.je.rep.utilint.BinaryProtocol.fillBuffer(BinaryProtocol.j
ava:446)
    at
com.sleepycat.je.rep.utilint.BinaryProtocol.read(BinaryProtocol.java:46
6)
    at
com.sleepycat.je.rep.subscription.SubscriptionThread.loopInternal(Subsc
riptionThread.java:656)
    ... 1 more

2018-03-16 11:23:46.064 UTC INFO [rg1-rn1]
SubscriptionProcessMessageThread soft shutdown initiated.
2018-03-16 11:23:46.492 UTC INFO [rg1-rn1] JE: Feeder output for
TextIndexFeeder-rg1-rn1-b4e92291-3c73-4128-9557-62dbd4e9ac78 shutdown.
feeder VLSN: 4,066 currentTxnEndVLSN: 4,065

```

If the `TextIndexFeeder` channel is shutdown, then the user can restore it by creating a dummy full text search index. Here is an example of how you can do that.

Assuming that Elasticsearch is already registered, execute the following commands from the Admin CLI:

```

execute 'CREATE TABLE dummy (id INTEGER,title STRING,PRIMARY KEY (id))'
execute 'CREATE FULLTEXT INDEX dummytextindex ON dummy (title)'
execute 'DROP TABLE dummy'

```

Note that *dummy* is the name of a temporary table that should not exist previously.

Creating a full text search index reestablishes the channel from the store to Elasticsearch and ensures that data is synced up to date. [\[#26859\]](#)

Data Verifier is Disabled By Default

The data verifier is turned off by default. In some cases, the data verifier was using a lot of I/O bandwidth and causing the system to slow down. Users can turn on the data verifier by issuing the following two commands from the Admin CLI:

```
plan change-parameters -wait -all-rns -params
"configProperties=je.env.runVerifier=false"
change-policy -params "configProperties=je.env.runVerifier=false"
```

Note that, if the store has services with preexisting settings for the `configProperties` parameter, then users will need to get the current values and merge them with the new setting to disable the verifier:

```
show param -service rg1-rn1
show param -policy
```

For example, suppose `rg1-rn1` has set the following cleaner parameter:

```
kv-> show param -service rg1-rn1
[...]
configProperties=je.cleaner.minUtilization=40
```

When updating the `configProperties` parameter, the new setting for the verifier should be added, separating the existing settings with semicolons:

```
plan change-parameters -wait -all-rns -params
"configProperties=je.cleaner.minUtilization=40;je.env.runVerifier=false"
```

[KVSTORE-639]

Subscription Cannot Connect and Fails With InternalException

If a master transfer occurs due to a failure after the publisher is started and before a subscriber connects, an `InternalException` can occur when the subscriber tries to connect. The exception message will read "Failed to connect, will retry after sleeping 3000 ms". Restart the publisher to work around this problem. *[#27723]*