Oracle® Coherence

Release Notes for Oracle Coherence

14.1.1.2206
F44675-09
August 2023

Oracle Coherence Release Notes for Oracle Coherence, 14.1.1.2206

F44675-09

Primary Author: Oracle Corporation

Contents

Preface

Release Notes for Oracle Coherence summarizes the release information related to new and updated features, known issues and their workarounds, deprecated and removed functionality, and more.

This preface includes the following topics:

Audience

This document is intended for users of Oracle Coherence.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `https://www.oracle.com/corporate/accessibility/`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `https://support.oracle.com/portal/` or visit `Oracle Accessibility Learning and Support` if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Documents

For more information, see the following documents in the Oracle Coherence documentation set:

- *Administering Oracle Coherence*
- *Administering HTTP Session Management with Oracle Coherence*Web*
- *Developing Applications with Oracle Coherence*
- *Developing Remote Clients for Oracle Coherence*
- *Installing Oracle Coherence*
- *Integrating Oracle Coherence*
- *Managing Oracle Coherence*
- *Securing Oracle Coherence*
- *Java API Reference for Oracle Coherence*
- *.NET API Reference for Oracle Coherence*
- *C++ API Reference for Oracle Coherence*
- *REST API for Managing Oracle Coherence*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1
Introduction

You can use the Oracle Coherence Release Notes to learn about important production information such as Coherence certifications, support, and licensing.
This chapter contains the following sections:

Latest Release Information

This document is accurate at the time of publication. Oracle will update the release notes periodically after the software release. You can access the latest information and additions to these release notes at Oracle Help Center.

Purpose of this Document

This document contains the release information for Oracle Coherence.

Oracle recommends you review its contents before installing, or working with the product.

System Requirements and Specifications

Oracle Coherence follows the Fusion Middleware system requirements and certifications for production environments. For more information, see http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html.

For system requirements for installations of development environments, visit:

- Coherence for Java — *Installing Oracle Coherence for Java*.
- C++ Client — *Installing the C++ Client Distribution*.
- .Net Client — *Installing the .NET Client Distribution*.

Certification Information

To see versions of platforms and related software for which Oracle Coherence is certified and supported, go to http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html.

Product Documentation

For complete documentation on Oracle Coherence, go to Oracle Help Center.

Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support .

Licensing Information

Detailed information regarding license compliance for Oracle Fusion Middleware is available at Licensing Information.

Downloading and Applying Required Patches

To download and install the latest software patch:

1. Log in to `My Oracle Support` to download the latest software patches.
2. Click the **Patches & Updates** tab.
3. Under the Patch Search tab, select **Product or Family (Advanced Search)**, and select the **Include all patches in a product family** check box.
4. Enter **Oracle Coherence** as the product, select the platform and release, and click **Search**.

   A list of currently available patches for Oracle Coherence is returned.
5. Select the required patch and click **Download**.

   You can check the `README` file in the patch distribution for up-to-date information on the software fixes provided by the patch.

2

What's New in this Release

Learn about the features, enhancements, and changes made to Oracle Coherence.Oracle updates the release notes periodically after the software release. This document is accurate at the time of publication.
This chapter includes the following sections:

New Features

This section contains new features for Oracle Coherence that are organized by release.

New and Improved for 14.1.1.2206

- **Java Modules Support** - You can now run Coherence using Java modules. See Using Java Modules to Build a Coherence Application in *Developing Applications with Oracle Coherence*.
- **Core Improvements**
  - **NamedMap API** - A distributed implementation of `java.util.Map` interface. See Performing Basic Cache Operations in *Developing Applications with Oracle Coherence*.
  - **Bootstrap API** - The new bootstrap API enables you to configure and start a Coherence application by building a `com.tangol.net.Coherence` instance and starting this instance. See Using the Bootstrap API in *Developing Applications with Oracle Coherence*.
  - **Repository API** - The Coherence Repository API provides you with a higher-level, DDD-friendly way to access data managed in Coherence. See Using the Repository API in *Developing Applications with Oracle Coherence*.
  - **Caffeine** - Coherence now adds a Caffeine backing map implementation, enabling you to use Caffeine wherever the standard Coherence local cache can be used. See Integrating Caffeine in *Developing Applications with Oracle Coherence*.
  - **Partition Events Logging** - This feature enables logging of partition unavailable duration when the partition events occur. For example, during partition movements between members. See Logging Partition Events in *Developing Applications with Oracle Coherence*.
  - **Non-Blocking Data Sources** - The new `NonBlockingEntryStore` enables cache stores to respond asynchronously when mutations are made to entries. See Non-Blocking Data Sources in *Developing Applications with Oracle Coherence*.
  - **Remote Client MEMBER_JOINED and MEMBER_LEFT MemberEvents** - A proxy now sends `MEMBER_JOINED` and `MEMBER_LEFT` `MemberEvents` to all active services on the proxy when a remote client joins and leaves. This event enables management of a service's server side resources being retained per remote client. If a `MemberListener` is registered on a service and the environment has both remote and

cluster member access to a service, the `MemberListener` may need to account for remote client `MemberEvent(s)`. For example, see Example 8-3 in Listening to Member Events section in *Developing Applications with Oracle Coherence*.

- **Scheduled Backups** - Writing asynchronous backups has been enhanced to enable scheduling of these backups at a time interval after the primary has been written. See Scheduling Backups in *Developing Applications with Oracle Coherence*.
- **Read Locator** - Coherence now allows for certain requests for data to be targeted to non-primary partition owners (backups) to balance request load or reduce latency. See Using the Read Locator in *Developing Applications with Oracle Coherence*.
- **Cache Configuration Override** - Similar to the Coherence Cluster override, you can now specify a cache configuration override to override elements of existing cache configuration with new elements at runtime. See Using Cache Configuration Override in *Developing Applications with Oracle Coherence*.
- **BigDecimal-related aggregators** - These aggregators now support the ability to set BigDecimal properties such as `scale`, `rounding mode`, `stripTrailingZeros`, and `MathContext` (where applicable) for the final result. See *Java API Reference for Oracle Coherence*.
- **Durable Events (Experimental)** - Coherence now supports an experimental feature which allows missed `MapEvents` to be replayed when a client disconnects. See Using Durable Events (Experimental) in *Developing Applications with Oracle Coherence*.

- **Topics Improvements**
  - A number of durability and stability improvements have been applied to make topics more stable during fail-over.
  - Topics now guarantee at least once delivery, where as in previous releases this was not the case. A subscriber that is part of a group can commit a processed message to indicate that processing is complete and it should not be redelivered on fail-over.
  - Topic channels are now fairly allocated to the subscribers in a subscriber group; only a single subscriber receives messages from an allocated channel.
  - Subscribers will be timed-out after a configurable period of inactivity (or failure to heartbeat) causing their channels to be reallocated to remaining subscribers in the same group.
  - Added API methods to determine the number of unreceived elements for a `NamedTopic` subscriber or subscriber group.

  See Using Topics in *Developing Applications with Oracle Coherence*.
- **Persistence**
  - **Persistent Backups** - You can now enable and configure persistent backups which stores backup partitions on a disk, as additional copies of persisted primary one. See Using Persistent Backups in *Developing Applications with Oracle Coherence*.
  - **Parallel Recovery** - The parallel recovery feature enables Coherence to recover data in parallel within a member/process as well as in parallel across the cluster. See Parallel Recovery in *Administering Oracle Coherence*.

- **Distributed Concurrency** - The Coherence Concurrent module provides distributed implementations of the concurrency primitives from the `java.util.concurrent` package such as executors, atomics, locks, semaphores, and latches. See Implementing Concurreny in a Distributed Network in *Developing Applications with Oracle Coherence*.
- **Serialization/ POF**
  - **Portable Types and POF Maven Plug-in** - This release introduces Portable Types, which provide a way to add support for POF serialization to your classes by using annotations and without the requirement to implement serialization code by hand. See Using Portable Object Format in *Developing Applications with Oracle Coherence*.
  - **POF Configuration Discovery** - It is now possible to make POF configuration files discoverable at runtime by the `ConfigurablePofContext` class instead of needing to put them inside `<include>` elements. See Making POF Configuration Files Discoverable at Runtime in *Developing Applications with Oracle Coherence*.
- **Integrations**
  - **Internal**
    - **CDI Support** - Coherence provides support for Contexts and Dependency Injection (CDI) within the Coherence cluster members to inject Coherence-managed resources, such as NamedMap, NamedCache, and Session instances into CDI managed beans. See Using Contexts and Dependency Injection in *Developing Applications with Oracle Coherence*.
    - **MicroProfile Configuration** - Coherence MicroProfile (MP) Configuration provides support for Eclipse MicroProfile Configuration within Coherence cluster members. See Using Coherence MicroProfile Configuration in *Integrating Oracle Coherence*.
    - **MicroProfile Metrics** - Coherence MicroProfile Metrics provides support for Eclipse MicroProfile Metrics within the Coherence cluster members. See Using Coherence MicroProfile Metrics in *Integrating Oracle Coherence*.
  - **External**
    - **Helidon** - Coherence can be integrated with Helidon through Contexts and Dependency Injection (CDI). See Helidon.
    - **GraphQL Support through Helidon** - Using Helidon integration, you can enable access to Coherence data from GraphQL. See GraphQL.
    - **Kafka** - Coherence can now integrate with Kafka using Kafka Entry Store and Kafka Sink Connector. See Kafka.
    - **Micronaut** - Coherence now provides integration to Micronaut. See Micronaut Coherence.
    - **Hibernate** - Updated support for Coherence integration with Hibernate. See Integrating Hibernate and Coherence in *Integrating Oracle Coherence*.
    - **Spring** - Coherence can be integrated with Spring, which is a platform for building and running Java-based enterprise applications. See Integrating Spring with Coherence in *Integrating Oracle Coherence*.
- **gRPC** - Coherence introduces the ability to use gRPC to access Coherence caches. See Introduction to gRPC.

  - **gRPC Proxy** - A new Coherence gRPC proxy implementation of the services defined within the Coherence gRPC module. See Using the Coherence gRPC Server in *Developing Remote Clients for Oracle Coherence*.
  - **gRPC Java Client** - The Coherence Java gRPC Client is a library that enables a Java application to connect to a Coherence gRPC proxy server. See Using the Coherence Java gRPC Client in *Developing Remote Clients for Oracle Coherence*.
- **Clients**
  - **JavaScript Client** - The Coherence JavaScript Client allows Node applications to act as cache clients to a Coherence Cluster using gRPC framework as the network transport. See Coherence JavaScript Client.
  - **Go Client** - The Coherence Go Client allows native Go applications to act as cache clients to a Coherence cluster using gRPC for the network transport. See Coherence Go Client.
  - **Python Client** - The Coherence Python Client allows Python applications to act as cache clients to an Oracle Coherence cluster using gRPC as the network transport. See Coherence Python Client.
  - **Client for .NET Core** - Coherence introduces a new client supporting .NET Core 3.1 and .NET Standard 2. See Oracle Coherence for .NET.
- **Security: SSL Improvements** - Various SSL improvements that enable more flexible configuration and allow customizations through extensions. See Using Private Key and Certificate Files and Using Custom Keystore, Private Key, and Certificate Loaders in *Securing Oracle Coherence*.
- **Management/Administration**
  - **Coherence Metrics** - Additional dependencies are no longer required when using the coherence-metrics module. See Using Oracle Coherence Metrics in *Managing Oracle Coherence*.
  - **Coherence Management over REST** - The dependencies required to enable Management over REST have been reduced significantly. The only additional dependency required to enable management over REST is `coherence-json.jar`. See REST API for Managing Oracle Coherence.
  - **Health Check API** - A new health check API to enable application code to determine the health of the local Coherence member. See Using the Health Check API in *Managing Oracle Coherence*.
  - **Micrometer Metrics** - The coherence-micrometer module provides integration between Coherence metrics and Micrometer allowing Coherence metrics to be published through any of the Micrometer registries. See Using Coherence Micrometer Metrics in *Managing Oracle Coherence*.
  - **New Reports - Executor, View, Storage, and Proxy Connections** - See Understanding the Executor Report, Understanding the View Report, Understanding the Cache Storage Report, and Understanding the Proxy Connections Report in *Managing Oracle Coherence*.
  - **New ExecutorMBean** - Provides statistics for the executor services that run in a cluster. See ExecutorMBean in *Managing Oracle Coherence*.
  - **New ViewMBean** - Provides statistics for view caches that run in a cluster. See ViewMBean in *Managing Oracle Coherence*.

  - **New HealthMBean** - Provides information about health checks configured in a cluster. See HealthMBean in *Managing Oracle Coherence*.
  - **ServiceMBean** - Additional attributes added to the ServiceMBean track Persistent Backups storage utilization. See ServiceMBean in *Managing Oracle Coherence*.
  - **StorageManagerMBean** - Additional attributes added to StorageManagerMBean to show Indexing Total Millis and Index Total Units. See StorageManagerMBean in *Managing Oracle Coherence*.
  - **Persistence and Persistence Detail Reports** - Additional columns added to these reports. See Understanding the Persistence Detail Report and Understanding the Persistence Report in *Managing Oracle Coherence*.
  - **WKA Improvements** - Added the ability to provide a comma-separated list of addresses when specifying a Well Known Address (WKA). See Using Well Known Addresses.
  - **Coherence Operator** - A number of major enhancements have been made to the Coherence Operator. See Coherence Operator.
  - **Coherence VisualVM Plug-in** - Updates have been made to support new functionalities added in 14.1.1.2206. See Coherence VisualVM Plugin Releases.
  - **Coherence CLI** - Updates have been made to support new functionalities added in 14.1.1.2206. See Coherence CLI Release.
- **Federation** - Additional attributes added to Origin and Destination MBeans to show errors and replication estimation. See OriginMBean and DestinationMBean in *Managing Oracle Coherence*.
- **Examples** - Coherence guides and tutorials are now hosted on the Coherence GitHub Repository and are documented here: Examples - Guides & Tutorials Overview.

New and Improved for 14.1.1.0

- **Topics** - Topics introduces a publish and subscribe messaging functionality in Oracle Coherence. See Using Topics in *Developing Applications with Oracle Coherence*.
- **OpenTracing** - The OpenTracing tool helps you diagnose unexpected latencies involved in request processing. See Distributed Tracing in *Developing Applications with Oracle Coherence*.
- **Asynchronous Persistence Mode** - Asynchronous persistence mode allows the storage servers to persist data asynchronously. See Using Asynchronous Persistence in *Developing Applications with Oracle Coherence*.
- **Oracle GraalVM Enterprise Edition Certification** - Oracle Coherence 14.1.1.0 is certified to run on Oracle GraalVM Enterprise Edition, a high performance runtime platform built on Oracle's enterprise-class Java SE. See Running Oracle WebLogic Server and Coherence on GraalVM Enterprise Edition.
- **Polyglot Coherence Applications** - These applications allow writing server side objects such as `EntryProcessor`, `Filter`, `Aggregator`, and so on in JavaScript, and running Coherence using Oracle GraalVM Enterprise Edition. See Developing Polyglot Coherence Applications in *Developing Applications with Oracle Coherence*.

Deprecated Features

Learn about the deprecated and desupported features of Oracle Coherence.

This section includes the following topics:

Deprecated Features for 14.1.1.2206

A brief description of the deprecated features for 14.1.1.2206.

This section includes the following topics:

VisualVM Plug-In Shipped with Installer

The Coherence VisualVM plug-in shipped with the installer is deprecated. You should use the open-source plug-in available through the VisualVM tool. For more information about this plug-in, see Using the Coherence VisualVM Plug-In in *Managing Oracle Coherence*.

Coherence Metrics 'vendor:coherence_' Prefix

When retrieving metrics from the Coherence metrics endpoint in text (Prometheus) format, the metrics name prefix `vendor:coherence_` is deprecated. The default name format has been changed to remove `vendor:` so that all metric names just have the prefix `coherence_`.

> **Note:**
>
> To ensure compatibility with the existing Grafana Dashboards, this new format has not yet been enabled by default. The legacy `vendor:` prefix is still generated for metric names.

Examples Shipped with the Installer

The examples shipped with the installer are now deprecated. Use the Coherence guides and tutorials hosted on GitHub. See Running the Coherence Examples.

Deprecated Features for 14.1.1.0

A brief description of the deprecated features for 14.1.1.0.

This section includes the following topics:

WebLogic Server Multitenant Functionality

Coherence support for WebLogic Server Multitenancy feature is deprecated because this feature is now not supported in WebLogic Server. See Removed Functionality and Components in *What's New in Oracle WebLogic Server*.

Reduced HTTP Server Support

For production environments, Coherence REST supports implementations only for the Netty HTTP server. Support for the following servers is deprecated:

- Simple HTTP Server
- Jetty HTTP Server
- Grizzly HTTP Server

For more information about Netty HTTP Server, see Using Netty HTTP Server in *Developing Remote Clients for Oracle Coherence*.

Classes in <com.oracle.common.base>

The classes in `<com.oracle.common.base>` have been deprecated. These classes are now relocated to an area within the coherence namespace (`<com.oracle.coherence>`).

3

Known Issues and Workarounds

Learn about the known issues at the time of release.
This chapter includes the following sections:

Changing the Partition Count When Using Active Persistence

Issue

The partition count cannot be changed when using active persistence. If you change a services partition count, then on restart of the services all active data is moved to the persistence trash and must be recovered after the original partition count is restored. Data that is persisted can only be recovered to services running with the same partition count.

Ensure that the partition count is not modified if active persistence is being used. If the partition count is changed, then a message similar to the following is displayed when the services are started:

```
<Warning> (thread=DistributedCache:DistributedCachePersistence, member=1):
Failed to recover partition 0 from SafeBerkeleyDBStore(...); partition-count
mismatch 501(persisted) != 277(service); reinstate persistent store from
trash once validation errors have been resolved
```

The message indicates that the change in the partition-count is not supported and the current active data has been copied to the trash directory.

Workaround

To recover the data:

1. Shutdown the entire cluster.
2. Remove the current active directory contents for the cluster and service affected on each cluster member.
3. Copy (recursively) the contents of the trash directory for each service to the active directory.
4. Restore the partition count to the original value.
5. Restart the cluster.

Binary Incompatibility with Older Versions

There is a binary incompatibility between Oracle Coherence release 14.1.1.0 and Oracle Coherence release 14.1.1.2206. This incompatibility requires upgrading users to recompile your applications against 14.1.1.2206. It is a binary incompatibility only; not an API or functional incompatibility.

When you run Coherence-based applications prior to release 14.1.1.2206 with the `coherence.jar` file from that release, you may encounter the following exception at runtime:

```
java.lang.ClassNotFoundException: com.tangosol.net.NamedCache$Option
```

For example, from `Session.getCache("<cache name>")` calls. This exception occurs because the nested class `Option` moved, in a refactoring, from `NamedCache` to its supertype `NamedMap` between 14.1.1.0 and 14.1.1.2206. This refactoring preserves the API compatibility for the previous code using `NamedCache.Option`, but requires recompilation of that code against the 14.1.1.2206 `coherence.jar`.

Workaround

To run a Coherence-based application prior to Oracle Coherence 14.1.1.2206 with that release, you should first recompile the application against 14.1.1.2206. Recompiling will avoid encountering a binary incompatibility exception at runtime.

4

Bugs Fixed and Enhancements in this Release

Learn about the bugs fixed and enhancements in this release.
This chapter includes the following section:

Oracle Coherence for Java

New features, improvements, and bug fixes added to Oracle Coherence for Java components.

Enhancements and Fixes for 14.1.1.2206

- Removed support for Log4j version 1.x.
- Fixed an issue where a Coherence `LifecycleListener` discovered using `ServiceLoader` can be registered twice and hence received events multiple times.
- Fixed an issue where `TcpRing.close.keys()` may throw an unhandled `ClosedSelectorException` which can cause the `Cluster` service to terminate unexpectedly.
- Added support for scale, stripping of trailing zeros, `MathContext`, and rounding mode to the `BigDecimal` aggregators.
- Fixed an issue with Coherence docker image where the system property, `coherence.serializer`, does not take effect when used to specify a default serializer.
- Added functionality to allow the list of included POF configuration files to be discoverable at runtime using Java `ServiceLoader`.
- Added the ability to show the Coherence version without starting a cluster using `java -jar coherence.jar --version`.
- Fixed an issue where the cluster service thread may be blocked on a member that is assuming the JMX cluster member role.
- Fixed an issue where catching an entry processor event of type `EXECUTED` due to a call to `invokeAll` would pass an empty entry set event if at least one entry was successfully processed.
- Added a REST endpoint to `ClusterMemberResource` to return the response of the `reportEnvironment` MBean operation of the `ClusterNodeMBean`, providing details about the Java environment and system properties.
- Fixed an issue where the Coherence `JsonSerializer` could not serialize a string made up of a single backslash.
- Fixed an issue in Topics where seeking to the tail for a subscriber did not actually move the subscriber's position.

- Fixed an issue in `NamedTopic Subscriber` where a `LockContentionException` could be thrown when many subscribers are being created and polling for messages at the same time.
- Fixed an issue in `Subscriber` Topic where `CompletableFuture` returned from calls to `Subscriber.receieve()` could fail to complete on an empty topic when the `Subscriber` was created with the `CompleteOnEmpty` option enabled.
- Enabled the use of custom namespace handlers in the operational configuration file.
- Fixed an issue when there are multiple cache factories in a JVM; the cache factory that created the proxy service may not be the one used by it, resulting in error finding the caches.
- Fixed an issue where a `PartitionedCache` service may terminate due to an unhandled `ClassCastException` in `PartitionedService$PartitionRecoverRequest$Poll.onResponse`.
- Fixed an issue where multiple Extend client sessions may be unable to access their caches created by different cache factories because the wrong cache factory and proxy were used on the server side.
- Fixed a thread safety issue that can disrupt a joining member from joining the service in very rare cases.
- Moved Coherence metrics functionality into the `coherence.jar` file, removing the requirement to use `coherence-metrics.jar` and removing the requirement for additional third-party dependencies when enabling Coherence metrics.
- Fixed an issue with serializing classes annotated with `PortableType` that did not specify an ID.
- Reduced the CPU utilization in sending backup messages.
- Changed the default for Java lambda serialization mode to `STATIC` for Coherence production mode; See About Lambdas in a Distributed Environment to evaluate whether to alter your environment to work with `STATIC` mode or to explicitly configure lambda serialization to the `DYNAMIC` mode.
- Fixed an issue where `CacheStore.eraseAll()` had no path to be called on the `NamedCache` bulk operations such as `invokeAll()`. With this fix, `CacheStore.eraseAll()` is called when `NamedCache.invokeAll()` is invoked with a "remove" processor, or when `NamedCache.keySet().removeAll()` or `NamedCache.entrySet().removeAll()` are called.
- Fixed an issue in which `PortableTypeGenerator` was not generating the relevant code for `RawDate` and `RawDateTime`.
- Improved cache operations to use an interruptible lock so that operations can be interrupted after the specified timeout.
- Improved `NearCache getOrDefault` and `computeIfAbsent` to utilize the front map.
- Removed the Coherence JVisualVM plug-in from distribution. Use the new Open Source VisualVM plug-in from `https://github.com/oracle/coherence-visualvm`.
- Improved the ordering of startup messages in `DefaultCacheServer` to show the list of services last.
- Fixed an issue where the `ValueUpdater` interface was not serializable and could cause certain lambda entry processors to fail to execute.

- Fixed an issue in Coherence*Web where a `NullPointerException` may be thrown when performing operations such as `AbstractHttpSessionModel.setAttribute()` on a session that is being invalidated.
- Fixed an issue where loaded entry from `CacheStore` had inconsistent `entry-event` type and the `synthetic` flag.
- Added the `reportEnvironment` operation to the `ClusterNode` MBean to provide details about the Java environment and system properties.
- Added default implementations to the `CacheLoader` and `CacheStore` interfaces for `loadAll()`, `storeAll()`, and `eraseAll()`.
- Improved persistence to recover data in parallel within a member/process, in addition to in parallel across the cluster. This feature enables the cluster, and more importantly the associated data, to be made available as quickly as possible.

> **Note:**
>
> Fixes from release 14.1.1.0.1 through 14.1.1.0.10 are also included. These fixes are available to view in the README document in the 14.1.1.2206 Feature Pack patch.

Enhancements and Fixes for 14.1.1.0

- Fixed an issue where links returned by Management over REST may have incorrectly URL encoded path separators.
- Fixed an issue where a `NullPointerException` may be thrown when resubmitting asynchronous cache requests internally.
- Added restricted reflection access to a small subset of core JDK classes.
- Fixed an issue where indexes are not recovered from persistence after a cluster restart.
- Reverted changes to `report-cache-effectiveness.xml` due a performance regression.
- Fixed an issue where service statistics from management over REST do not aggregate across all services. Queries such as `coherence/management/services;field=StatusHA` now return the correct result.
- Updated Coherence REST examples to use Oracle JET instead of Bootstrap/AngularJS and jQuery.
- Fixed an issue that would prevent Coherence Metrics from initializing due to a ClassCastException.
- Fixed an issue where federation members may get stuck in the **YIELDING** state due to changes to cache entries, which expired before the changes were federated to destination participants.
- Fixed an issue where a `ConfigurableCacheFactory` is initialized in a WebLogic server even if Coherence is not enabled.
- Fixed an issue where a ClusterService is initialized when `coherence-metrics.jar` is added to the classpath.
- Fixed an issue where the `getOrDefault` method of `InvocableMap` (and consequently `NamedCache`) and `AsyncNamedCache` did not conform to the `java.util.Map` contract. The

default value should only be returned when the key is not present in the cache. However, the default value was also being returned when the key was present and mapped to null.

- Fixed an issue where the `AsynNamedCache.getAll()` method did not conform to the `com.tangosol.net.cache.CacheMap` contract in that the returned Map contained entries for all of the requested keys instead of containing entries for only the requested keys that were present in the cache.
- Fixed a typographical error in the management over REST API for `/management/coherence/cluster/members/{memberId}/platform/g1SurvivorSpace`.
- Fixed an issue with the Persistence examples to ensure MBean registration is complete before starting the example.
- Fixed an issue where a `NullPointerException` may be thrown in `reportLastOwnership` during simultaneous shutdown.
- Fixed an issue where federation destination members may use heap memory inefficiently for storing cache entries, resulting in larger heap sizes than origin members for the same amount of cache data.
- Fixed an issue where an `OutOfMemoryError` could be thrown for a `ReadWriteBackingMap` with registered `MapListeners`.
- Fixed an issue where keys are deserialized if returned as keys in the Map that is returned by an `EntryProcessor.processAll` invocation.
- Fixed an issue where Coherence fails to start when `coherence-metrics.jar` is added to the classpath and the `extendedmbeanname` property is set to true.
- Fixed an issue where `getAll()` on a `NearCache` may return incorrect results for an *Extend client.
- Fixed an issue in `LimitFilter` when used in deserialization via the WebLogic T3 protocol that could allow for arbitrary code execution.
- Fixed an issue where `getAll()` on a `NearCache` may return incorrect results for an *Extend client.
- Fixed an issue where an exception may not be thrown if a `ContinuousQueryCache` is in a disconnected state and fails to re-synchronize.
- Updated the SFTP Snapshot Archiver example to use Apache Mina SSHD.
- Fixed an issue where some of the default `InvocableMap` methods do not cause read-through from a `CacheStore` when they should.
- Fixed an issue where there may be repeated logging of a message related to an unexpected `EvictionApprover` during partition transfer.
- Allow persistent writes to be performed asynchronously.
- Fixed an issue where a listener on a 12.2.1.0.x, 12.2.1.1.x, 12.2.1.2.x or 12.2.1.3.x Extend client `NearCache` may miss some events.
- Fixed an issue where a `NullPointerException` may be thrown by Coherence*Web due to access to a session which is no longer valid when there is a high rate of sessions being invalidated.
- Lowered the severity of the log message, which states that event interceptors are already registered when restarting a service.

- Fixed an issue where a service may terminate due to `IllegalArgumentException: unknown extent identifier` being thrown during a rolling restart of an active persistence enabled cluster.
- Fixed an issue where Berkeley DB configuration specified as `init-params` on the `bdb-store-manager` configuration element was being ignored.
- Fixed an issue where connection migration may occur in a loop, without resolution.
- Fixed an issue where the Maven enforcer `dependencyConvergence` rule would generate warnings concerning Jackson modules.
- Fixed an issue where federation may end up in a high CPU usage busy-loop during a `ReplicateAll` operation.
- Fixed an issue where persistence data loss may occur on cluster restart if a client service had ensured a cache before any storage member was started and there was no cache update operation while the cluster was running, prior to the restart.
- Fixed an issue with management over REST API return values for MBean attributes, ensuring that the `ObjectName` key property with same name as attribute does not override its value.
- Fixed an issue where the cluster service could be terminated due to an unhandled `IllegalAccessException` being thrown by the `SlabBufferManager`.
- Fixed an issue where Coherence*Web threads may be stuck at `com.tangosol.util.SegmentedConcurrentMap.lock`.
- Fixed a rare issue where destroying and recreating a partitioned cache may incorrectly throw an exception.
- Fixed an issue where the service thread may be blocked in the journal congestion state during a rolling restart.
- Added `AverageReapQueueWaitDuration` to the `SessionReaper` statistics which indicates the average time a reap task spends in queue, prior to being invoked.
- Enhanced the Coherence*Web session reaper to operate on the back map when a `NearCache` is used to store HTTP sessions.
- Enhanced federation to allow synthetic updates in a federation interceptor. The synthetic update will have different behavior depending on the federation event type: **COMMITTING_LOCAL** - the change will not be federated to other clusters and `CacheStores` will not be called. **REPLICATING** - setting the change synthetic is not allowed. An `UnsupportedOperationException` will be thrown. **COMMITTING_REMOTE** - the change will be applied as a synthetic change. `CacheStores` will not be triggered and federation will not forward the change to other clusters from this destination cluster.
- Changed the Coherence REST example to use Oracle JET.
- Federation internal caches are now excluded from persistence snapshots.
- Fixed an issue where partition distribution may take longer than expected with the error message:

  ```
  Unreconcilable ownership conflict; conceding the ownership
  ```

5
Documentation Changes

The documentation errata lists any corrections to the Oracle Coherence documentation. The Coherence documentation can be found at the following URL:

https://docs.oracle.com/en/middleware/fusion-middleware/coherence/index.html

There are currently no corrections to the Oracle Coherence Feature Pack documentation.