Oracle® Fusion Middleware

Release Notes for Oracle Coherence

12*c* (12.1.3)

E51564-06

March 2016

Oracle Fusion Middleware Release Notes for Oracle Coherence, 12*c* (12.1.3)

E51564-06

Contents

2 Documentation Errata

Preface

This document describes changes and enhancements that have been made to Oracle Coherence for the 12*c* (12.1.3) release.

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 `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

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

- *Oracle Fusion Middleware Developing Applications with Oracle Coherence*
- *Oracle Fusion Middleware Developing Remote Clients for Oracle Coherence*
- *Oracle Fusion Middleware Tutorial for Oracle Coherence*
- *Oracle Fusion Middleware Integrating Oracle Coherence*
- *Oracle Fusion Middleware Administering HTTP Session Management with Oracle Coherence*Web*
- *Oracle Fusion Middleware Managing Oracle Coherence*
- *Oracle Fusion Middleware Administering Oracle Coherence*
- *Oracle Fusion Middleware Securing Oracle Coherence*
- *Oracle Fusion Middleware Java API Reference for Oracle Coherence*
- *Oracle Fusion Middleware .NET API Reference for Oracle Coherence*

- *Oracle Fusion Middleware C++ API Reference for 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 Technical Changes and Enhancements

This chapter describes the changes and enhancements made to the Oracle Coherence product for 12*c* (12.1.3). This document is accurate at the time of publication. Oracle updates the release notes periodically after the software release.

1.1 Download and Install the Latest Software Patch

Go to My Oracle Support to download the latest software patches.

```
https://support.oracle.com
```

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

1. Login to My Oracle Support.
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**.

The list of currently available patches for Oracle Coherence is returned.

1.2 New and Improved Coherence Data Grid Functionality

Oracle Coherence 12*c* (12.1.3) is the latest release of the industry's leading distributed in-memory data grid product. Coherence 12*c* (12.1.3) contains many new features that help reduce complexity, ease integration, and accelerate time to market of scalable solutions.

- Added SSL support for reliable transports `tmb` and `sdmb`. The feature can be enabled through operational override file or system property. To use `tmb` and `sdmb` transports with SSL, an SSL socket provider is required. See "Changing the Reliable Transport Protocol" in Oracle Fusion Middleware Administering Oracle Coherence.
- The library providing Infiniband Message Bus (IMB) support has been updated. IMB is available only for Oracle Exalogic systems. IMB is automatically used as long as TCMP has not been configured with SSL. IMB is the default reliable transport mechanism on Exalogic. See "Changing the Reliable Transport Protocol" in Oracle Fusion Middleware Administering Oracle Coherence.
- Added functionality to JMX Service beans, which displays partition distribution state of the service. See "ServiceMBean" in Oracle Fusion Middleware Managing Oracle Coherence.
- Created the `RemoteMember` class which allows a custom `ProxyLoadBalancer` to redirect a TCP Extend client from one cluster to another. See the `com.tangosol.net.proxy.RemoteMember` class in *Oracle Fusion Middleware Java API Reference for Oracle Coherence*.
- Enhanced `NameService` so it runs on all cluster members. `NameService` now has the ability to lookup `ProxyService` addresses for other members by using the `ProxyService` state information stored in the cluster. This way clients only need to know the configured unicast address for the cluster in terms of which port to connect to for lookup. See "Using the Name Service Acceptor to Connect to a Proxy" in Oracle Fusion Middleware Developing Remote Clients for Oracle Coherence.
- Added support for the `memcached` protocol. Coherence can be used as a drop-in replacement for memcached servers. Any memcached client API that supports the memcached binary protocol can use Coherence distributed caching. Memcached support is provided through a memcached adaptor that is implemented as an extend-like acceptor that runs on a proxy server. For details about configuring Coherence to accept memcached client connections, see "Using Memcached Clients with Oracle Coherence" in *Oracle Fusion Middleware Integrating Oracle Coherence*.
- A provider implementation has been added for "JSR 107: JCACHE - Java Temporary Caching API" (JCache). JCache is the standard API for caching on the Java Platform. The Coherence JCache provider uses existing Coherence technology and can be thought of as a wrapper for the Coherence NamedCache API. This allows Coherence to reuse and expose many of its best-in-class technologies using JCache interfaces. See "Using the Coherence JCache Implementation" in Oracle Fusion Middleware Developing Applications with Oracle Coherence.
- Added the ability to set a "soft limit" (default is 11GB) for journal size in the Elastic Data feature. The size determines when to start removing stale values from the journal. This is not a hard limit on the journal size, which can still grow to the maximum file count (512). See "Configuring the Flash Journal Resource Manager" in Oracle Fusion Middleware Developing Applications with Oracle Coherence.

- Added the Coherence Java VisualVM Plug-in to the current release. The Coherence Java VisualVM Plug-in allows developers to view a dashboard of summarized statistics and cluster information when using JMX to connected to a Coherence cluster. Java VisualVM is a management utility that is included with the JDK and provides the ability to view and interact with MBeans. See "Accessing MBeans Locally Using Java VisualVM" in Oracle Fusion Middleware Managing Oracle Coherence.
- When using Managed Coherence Servers, a Coherence cluster member can now be configured as a management node. The management node aggregates the management information from all other cluster members. The Administration server for the WebLogic domain then integrates this information and it is made available through the domain runtime MBean server. See "Configure a Coherence Cluster Member as a Management Node" in *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server*.
- A custom index implementation which optimizes deserialization has been added to the current release. The optimization avoids repetitive value deserialization for entry processors and aggregators. See the `com.tangosol.util.extractor.DeserializationAccelerator` class in *Oracle Fusion Middleware Java API Reference for Oracle Coherence*.
- The ability to execute non-blocking EntryProcessors asynchronously has been added to the current release. This allows customers to dramatically increase the parallelism of the processor execution on the grid, without having to create a large number of worker threads. See the `com.tangosol.util.processor.AsynchronousProcessor` class in *Oracle Fusion Middleware Java API Reference for Oracle Coherence*.
- The ability to rebuild index data structures asynchronously after a partition transfer or failover has been added to the current release. This significantly reduces the time to recovery. Note: any requests dependent on indices will not execute until the indices are rebuilt.
- Extend clients support both forward and backward compatibility with cluster proxies. That is, extend clients can connect to cluster proxies that have lower or higher version numbers (within a major release). For example, a 12.1.3 extend client can connect to a 12.1.2.0.1 proxy. Extend client backward compatibility is not supported on proxy versions prior to 12.1.2.0.1, including 12.1.2.0.0 and proxy versions 3.7.1 or earlier.

1.3 Oracle Coherence 12.1.3 and Java SE 8

Oracle Coherence 12.1.3 was certified on Java SE 7 when it was first made generally available. In addition, Oracle Coherence 12.1.3 has been certified on Java SE 8 as follows:

- Coherence 12.1.3 is certified on Java SE 8 Update 20 (u20) and will also be supported on later updates of Java SE 8.
- Coherence 12.1.3 is certified on Java SE 8u20 on Windows, Linux, and Solaris with 64-bit JVMs.
- Other Oracle Fusion Middleware 11*g* and 12*c* products related to or layered on Coherence are not supported on Java SE 8 unless specifically documented.
- See the Java SE 8 documentation for details about the features and capabilities available in Java SE 8. For specific information about garbage collection tuning

options in Java SE 8, see the *HotSpot Virtual Machine Garbage Collection Tuning Guide*.

- Note the following restrictions and advice when running Coherence 12.1.3, and Coherence 12.1.3 based applications, on Java SE 8:
  - Coherence 12.1.3 does not support application development using the new Java SE 8 features, including the use of Streams, Lambdas and default methods. While applications using these features may compile correctly with Coherence, use of these features with Coherence 12.1.3 may produce unexpected results including: prevention of Coherence data-locality optimizations, which can dramatically reduce performance and efficiency; and decomposition of previously-atomic operations, which can negatively affect the safety and correctness of concurrent operations.
  - Customers should check third party vendor software that they are using for Java SE 8 compatibility. It may be necessary to upgrade to a later version of the software that correctly handles Java SE 8 classes and some software may not yet be compatible. For example, the current version of the open source tool "jarjar" does not work correctly with Java SE 8 yet.

1.4 Oracle Coherence for Java 12*c* (12.1.3)

New features, improvements, and bug fixes have been added to these Oracle Coherence for Java components:

- Management and Monitoring Enhancements and Fixes
- Partitioned Cache Enhancements and Fixes
- TCMP/Cluster Enhancements and Fixes
- Coherence*Extend Framework Enhancements and Fixes
- Coherence*Web Enhancements and Fixes
- Serialization Framework Enhancements and Fixes
- Configuration Enhancements and Fixes
- Managed Coherence Servers Enhancements and Fixes
- Other Enhancements and Fixes

1.4.1 Management and Monitoring Enhancements and Fixes

- Enhanced the Proxy HostIP MBean attribute to show the subport value.
- Fixed an issue with `Filter` execution producing high memory overhead due to unnecessary JMX statistic.
- `ServiceMBean` now has a property `JoinTime` which shows the time a member has joined the given service.
- Moved Hibernate integration features into Coherence Community project `http://java.net/projects/cohhib`.

1.4.2 Partitioned Cache Enhancements and Fixes

- Fixed an issue that a client could use an already released cache reference.
- Fixed an issue where `MultiExtractor` was not using indexes for its `ValueExtractor`.

- Added a custom index implementation, that allows to avoid repetitive value deserialization. For usage details, please see Javadoc for `com.tangosol.util.extractor.DeserializationAccelerator` class.
- Fixed a deadlock issue when using `LocalCache` with map listeners.
- Optimized `ReadWriteBackingMap` to call the `storeAll` method on `CacheStores` in response to a `putAll` operation. This will allow application code in `CacheStores` to be optimized for bulk storage.
- Fixed a possible Guardian time out issue which might occur when restoring orphaned partitions.
- Updated the `ChainedComparator` three parameter constructor so that it no longer ignores the third `Comparator`.
- Fixed a `NullPointerException` occurring when an event is generated for orphaned partitions with JMX disabled across the cluster.
- Fixed a potential race condition between a guarded thread timing out and the thread no longer being guarded.
- Hardened the handling of cache misconfigurations by guarding against NPE and UOE to provide a stabler service.
- Fixed `ReadWriteBackingMap` to honor the unit calculator configured in cache configuration.
- Fixed an issue with `Size` command in console which resulted in no longer results in indefinite `SizeRequest` when it was called on a cache name with no schema defined.
- Introduced `getReadOnlyEntry()` method on the `BackingMapContext` interface, which allows Aggregators and Entry Processors to obtain read-only access to backing map entries without locking the corresponding keys.
- `ReadWriteBackingMap's` Read/Write Threads now contain the cache name as part of the Thread name.
- Improved readability of explain plan and trace query records.
- Fixed an issue where no error was thrown when creating a transactional cache for which there is no scheme definition in cache-config.

1.4.3 TCMP/Cluster Enhancements and Fixes

- Fixed an issue which could cause `PacketReceiver` spin when extracting packets from buffer.
- Hardened the `PacketReciever` against randomly truncated otherwise valid packets.
- Fixed an issue where partitioned cache could become uninterruptable due to prolonged orphaned partitions.
- Removed an inaccurate warning regarding incompatible WKA addresses when starting a cluster with WKA where at least one host name is specified.
- Created a new lighter-weight IMB Implementation with improved fail-over handling.

1.4.4 Coherence*Extend Framework Enhancements and Fixes

- Fixed an issue where a ConcurrentModificationException may be thrown when invoking ConditionalRemove on a ReplicatedCache from an Extend client.
- Changed the proxy service worker threads to close Coherence*Extend connections asynchronously when a connection will be closed due to an error.
- Fixed an issue where an Extend client doesn't receive delete events from a replicated cache.
- Fixed an issue in ensureCache which can result in client threads becoming stuck if the ensureCache was unresponsive.
- Added tracking of unauthorized hosts attempting to connect via Coherence*Extend to include logging of a D6 message as well as recording the number of attempts in a new JMX attribute "UnauthorizedConnectionAttempts" on ConnectionManagerMBean.
- Added a new ConnectionManager MBean for the NameService.
- Added ability to define MapTriggerListeners for remote caches by using the <listener> element.
- Fixed an issue causing NamedCache iterators to return duplicate results to Extend clients from Standard edition clusters.
- Hardened the handling of cache deactivation calls.
- Fixed a custom Comparator issue with proxy service load balancing.
- Fixed an InFilter serialization issue.
- Changed the NameService to run on all cluster nodes instead of just proxy server nodes.
- Added additional log messages to indicate that a Coherence*Extend connection has been closed by the suspect protocol.
- Fixed an issue with the dynamic proxy service thread pool size increasing too aggressively under certain load scenarios.
- Resolved an issue where custom proxies were not being called when ensuring and/or destroying a cache.
- Created RemoteMember class which allows a custom ProxyLoadBalancer to redirect an TCP Extend client from one cluster to another.
- Added Guardian support to the NameService to auto-restart the Acceptor.
- Added support for the memcached protocol.

1.4.5 Coherence*Web Enhancements and Fixes

- Fixed NotSerializableException in Coherence*Web for WebLogic Portal components during session invalidation.
- During session invalidation, ConcurrentModificationException is now logged only when the logging level is set to FINEST.
- Fixed an issue where Coherence*Web on Tomcat 7.0 reported incorrect Tomcat version information in the log file.
- Fixed an erroneous clean-up of locally stored session attributes.

- Fixed Session Reaper to ensure that local attributes are cleaned up when the session invalidation is initiated from a different node.

1.4.6 Serialization Framework Enhancements and Fixes

- Improved POF date and time deserialization performance.
- Updated `PofValue.getValue()` API documentation to indicate that the type must be explicitly specified for primitive types.
- Fixed an issue where integer value of -31 is not read properly by the POF serializer.
- Fixed `PofHelper.getPofTypeId()` to return the correct type id for user types that derive from `ArrayList`.
- Improved the POF annotation log message when a requested property index is already in use.

1.4.7 Configuration Enhancements and Fixes

- When using the Coherence Cache Provider with WebLogic Portal, the cache configuration may be set using a system property.
- Fixed an issue where resources containing '`#`' character in filename/path failed to load.
- Fixed the issue where custom MBean registration assumes MBean name to be unique.
- In 12.1.2-0-0, empty `XmlElements` were skipped when considered for injection with `@Injectable`. In 12.1.3 empty `XmlElements` are always processed when an `ElementProcessor` is registered, unless the new `ConditionalElementProcessor` is implemented.
- As of 12.1.3 System Properties and Resource Registry values are considered when attempting to locate named and typed values for injection into `@Injectables.`
- Fixed an issue where `DefaultClusterDependencies.makeClusterName()` throws NPE due to null group address.

1.4.8 Managed Coherence Servers Enhancements and Fixes

- Changed Extend security to use the Subject provided by the client rather than the WebLogic Subject when running an Extend Proxy inside WebLogic Server Managed Servers. This change removes the need to use WebLogic APIs to obtain the client's Subject and instead standard JAAS APIs can be used and hence removes the need for application code to have any dependency on WebLogic Server.
- Fixed an issue where `jdk/log4j` log level is not allowed to be configured because of Coherence log level not being at maximum.
- JCache is not currently designed to work with Managed Coherence Servers.

1.4.9 Other Enhancements and Fixes

- Fixed an issue where `cmd` scripts shipped by Coherence fail to handle `JAVA_HOME` path containing spaces and parenthesis.

- Fixed an issue where log messages that are not in the range of destination logger are enqueued.
- Fixed `CacheFactory.shutdown()` to stop the Logger Thread if there is no running cluster.
- Hardened the transactional cache handling of `javax.transaction.xa.Xid` to allow non-serializable implementations.
- Optimized the transfer logic of backups when backup-count > 1
- Additional permissions added to the security policy for Java security manager.
- The ability to rebuild index data structures asynchronously after a partition transfer or failover has been added to the current release. This significantly reduces the time to recovery. Note: any requests dependent on indices will not execute until the indices are rebuilt.
- Oracle Coherence 12.1.2.0.0 RESTful interfaces now depend on Jersey 1.17.
- Added SSL support for reliable transport `tmb` and `sdmb`. The feature can be enabled through operational override file or system property.
- Changed format of printing PartitionSet from {1, 2, 4, 5, 6, 8} to {1, 2, 4...6, 8}

1.5 Oracle Coherence for .NET 12*c* (12.1.3)

New features, improvements, and bug fixes have been added to these Oracle Coherence for .NET components:

- Coherence*Extend Framework Enhancements and Fixes
- Serialization Framework Enhancements and Fixes
- Partitioned Cache Enhancements and Fixes
- Other Enhancements and Fixes

1.5.1 Coherence*Extend Framework Enhancements and Fixes

- Corrected an issue when iterating the entries, keys, or values of a `ContinuousQueryCache`.
- Fixed an event delivery issue for cache listeners registered with filters that use a `PofExtractor`.
- Added a `SimplePrincipal` which derives from `GenericPrincipal`. `SimplePrincipal` uses the `Identity.Name` property to determine object equality.
- Added ability to define `MapTriggerListeners` for remote caches by using the `<listener>` element.

- Fixed a .NET 4.5 framework issue with `SynchronizedDictionary`.
- Hardened the handling of cache deactivation calls.
- Fixed an issue where type -29 is not read properly by the POF serializer.
- Fixed an `InFilter` serialization issue.
- Improved SSL client certificate selection for SSL enabled Coherence*Extend connections.

- Fixed the Coherence for .NET library to look for the correct XML configuration elements for using network filters, like the `gzip` compression filter.

1.5.2 Serialization Framework Enhancements and Fixes

- Updated `PofValue.getValue()` API documentation to indicate that the type must be explicitly specified for primitive types.
- Fixed `PofHelper.getPofTypeId()` to return the correct type id for user types that derive from `ArrayList`.
- Improved the POF annotation log message when a requested property index is already in use.

1.5.3 Partitioned Cache Enhancements and Fixes

- Fixed a deadlock issue when using `LocalCache` with map listeners.

1.5.4 Other Enhancements and Fixes

- The ability to rebuild index data structures asynchronously after a partition transfer or failover has been added to the current release. This significantly reduces the time to recovery. Note: any requests dependent on indices will not execute until the indices are rebuilt.
- Oracle Coherence 12.1.2.0.0 RESTful interfaces now depend on Jersey 1.17.
- Added SSL support for reliable transport `tmb` and `sdmb`. The feature can be enabled through operational override file or system property.
- Changed format of printing PartitionSet from {1, 2, 4, 5, 6, 8} to {1, 2, 4...6, 8}

1.6 Oracle Coherence for C++ 12*c* (12.1.3)

New features, improvements, and bug fixes have been added to these Oracle Coherence for C++ components:

- Coherence*Extend Framework Enhancements and Fixes
- Configuration Enhancements and Fixes
- Partitioned Cache Enhancements and Fixes
- Other Enhancements and Fixes

1.6.1 Coherence*Extend Framework Enhancements and Fixes

- Fixed a typo in the guard macro for `XorFilter.hpp`.
- Updated `TypedMethod.hpp` so that Xcode (`clang`) no longer warns about unused functions in the file.
- Changed `DetachFinalizer::set()` to detach from the old object with the old escaped flag.
- Fixed `LiteSet` so that it can support more than 2 elements.
- Added support for defining class factories with the `<class-factory-name>` element.
- Added support for defining class initialization parameters with the `<init-params>` element.

- Hardened the handling of cache deactivation calls.
- Fixed an `InFilter` serialization issue.
- Fixed `adapter_map/boxing_map` iterator initialization when the map size is 1.
- Added the `NavigableMap` interface and enhanced `TreeMap` to implement `NavigableMap`.
- Added optional parameter, `nPrecision`, to `Float32::parse()` and `Float64::parse()` to allow user specify the precision:
  `Float32::parse(String::View vString, size_t nPrecision)`
  `Float64::parse(String::View vString, size_t nPrecision)`
- Added `AbstractClusterProcessor` to C++.
- Added `add/removeMapListener` API to `ObservableMap`.

1.6.2 Configuration Enhancements and Fixes

- Fixed an issue where resources containing '`#`' character in filename/path failed to load.

1.6.3 Partitioned Cache Enhancements and Fixes

- Fixed a deadlock issue when using `LocalCache` with map listeners.

1.6.4 Serialization Framework Enhancements and Fixes

- Improved the POF annotation log message when a requested property index is already in use.

1.6.5 Other Enhancements and Fixes

- `SimpleDocument` is now cloneable, allowing the invocation of `clone()` on `XmlElement` returned by `CacheFactory::loadXml()`.
- The ability to rebuild index data structures asynchronously after a partition transfer or failover has been added to the current release. This significantly reduces the time to recovery. Note: any requests dependent on indices will not execute until the indices are rebuilt.
- Oracle Coherence 12.1.2.0.0 RESTful interfaces now depend on Jersey 1.17.
- Added SSL support for reliable transport `tmb` and `sdmb`. The feature can be enabled through operational override file or system property.
- Changed format of printing PartitionSet from {1, 2, 4, 5, 6, 8} to {1, 2, 4...6, 8}

1.7 New Features, Enhancements, and Fixes for Coherence 12*c* (12.1.3.0.1)

The following changes and enhancements have been made for the 12*c* (12.1.3.0.1) release.

- New Features for Coherence 12c (12.1.3.0.1)
- Changes and Enhancements for Oracle Coherence for Java 12c (12.1.3.0.1)
- Changes and Enhancements for Oracle Coherence for .NET 12c (12.1.3.0.1)
- Changes and Enhancements for Oracle Coherence for C++ 12c (12.1.3.0.1)

1.7.1 New Features for Coherence 12*c* (12.1.3.0.1)

- In this release, Coherence introduces an extra Tab in the Coherence JVisualVM Plugin that uses graphing functionality to display Elastic Data.
- In this release, Coherence added support for enabling or disabling specified SSL protocols. For more information, see "Controlling Cipher Suite and Protocol Version Usage" in *Securing Oracle Coherence*.

1.7.2 Changes and Enhancements for Oracle Coherence for Java 12*c* (12.1.3.0.1)

- Fixed an issue where the JVisualVM Plug-in would not correctly gather data via JMX reports when connected to a WebLogic Server domain runtime MBean server.
- Added support for displaying the cluster name and version when using the JVisualVM Plug-in.
- For Coherence*Web configurations on WebLogic, a new report, `reports/report-web-weblogic.xml`, is available.
- Added support for displaying information about Elastic Data when using the JVisualVM Plug-in.
- Fixed an issue where `ConfigurableCacheMap` methods do not return correct values with `ObservableSplittingBackingCache`.
- Fixed an issue where a client could use an already released cache reference.
- Added support for displaying partition information in graphical format when selecting a service in the JVisualVM Plug-In.
- Fixed a memory leak issue in `CacheFactory.releaseCache()` and `NamedCache.release()`.
- Fixed an issue where off-heap (NIO) backing map could lose cache entries during failover.
- Improved the BetweenFilter's efficiency of applying indexes and evaluating values.
- Fixed an issue where Async EntryProcessors were being deferred after suspend and resume of a service.
- Fixed an issue where scroll pane backgrounds are the incorrect color in the JVisualVM Plug-in on Windows environments.
- Fixed an issue which could cause a server to fail to join a cluster with tmbs protocol.
- Fixed an issue where session attributes are mandated to implement `java.io.Serializable`.
- Fixed a POF serialization issue for negative Dates with nonzero sub-seconds.
- Fixed an issue where internal session attribute names are returned without stripping the `InternalWLSAttribute` prefix.
- Fixed an issue where NearCache event interceptors defined in cache-config are not registered when using `ExtensibleConfigurableCacheFactory`.
- Fixed an issue with `UpdaterProcessor` that could cause missing entry update in `CacheStore`, `BinaryEntryStore`, and `Listeners`.
- Added support for displaying detailed JMX information when selecting members, service details and cache details in the JVisualVM Plug-In.

- Fixed an issue that could prevent the node from starting up when using `<host-filter>` in cache configuration.
- Fail fast cache configuration processing has been removed. Class loading is deferred until configuration entity is instantiated in the server.
- Fixed an issue with InFilter that could result in poor performance if used with an initial filter that reduces the data set significantly.
- Fixed a concurrent access issue with UUIDs.
- Fixed an issue where `<maximum-size>` was not taking effect for `flashjournal-manager`.
- Fixed an issue which could cause an `Assertion` error to occur with the `ReadWriteBackingMap.putAll` operation.
- Fixed an issue with starting a proxy service when the `http-acceptor class-name` was not specified in the configuration.
- Fixed an issue that occurs when under high load using expiry or eviction and results in `AssertionException` errors or deadlock.
- Fixed an issue that occurs when applications using Java Security Manager need extensive security permissions.
- Removed an improper `Service.heartbeat()` call from a non-service thread that resulted in a concurrent `checkGuardables` call by the service thread.
- Fixed an issue where an invocation of `QueryPlus.main` preceded by `QueryHelper.createFilter` caused a `NullPointerException`.
- Added support for displaying tool tips for various fields allowing easier viewing of data when using the JVisualVM Plug-in.
- An instance factory method is not required to return the required exact `ClassType` in its method signature. Runtime type mismatches will result in a `ClassCastException` error.
- Fixed a bug where a value set for the `management-configuration/default-domain-name` in the operational configuration (or overrides) file was being ignored.
- Added two new reports (`reports/report-jcache-statistics.xml` and `reports/report-jcache-configuration.xml`) to the output JCache configuration and statistics information when JCache is being used.
- Fixed an issue with invoking `GuardSupport.heartbeat` during the execution of post commit LiveEvents.
- Added support for displaying information about JCache when using the JVisualVM Plug-in.
- Fixed an issue where the `<cipher-suites>` element in a cluster configuration caused a `SAXParseException`.
- Fixed an issue which could cause the proxy server's `TcpProcessor` thread to get stuck in a CPU busy loop.
- Coherence JCache system property `tangosol.coherence.jcache.configuration.classname` now supports `passthrough` alias.
- Addressed an issue which could cause a `ConcurrentModificationException` error on IBM J9 JVM when running TCMP over TCP.

- Fixed an issue with registering `<serializer>` resources multiple times by overriding the previous registration.
- Fixed an issue when using `MultiExtractor` as an index that may result in inaccurate indexed data.
- Fixed an issue that could cause `NoSuchElementException` errors during Ram journal eviction.
- Fixed an issue that resulted in guardian errors on startup when using `EventInterceptors` and a large number of concurrent requests per member.

1.7.3 Changes and Enhancements for Oracle Coherence for .NET 12*c* (12.1.3.0.1)

- Improved the BetweenFilter's efficiency of applying indexes and evaluating values.
- Fixed a concurrent access issue with UUIDs.
- .NET clients can now configure POF configuration files in the cache configuration rather than the application configuration. Note that a custom serializer must now implement a constructor which initializes the `<init-params>` of the configuration.

1.7.4 Changes and Enhancements for Oracle Coherence for C++ 12*c* (12.1.3.0.1)

- Improved the BetweenFilter's efficiency of applying indexes and evaluating values.
- Fixed an issue where dependent threads of a Coherence service may fail to stop during service shutdown.
- Fixed a concurrent access issue with UUIDs.
- Added SLTPort compatible Coherence for C++ libraries for Solaris SPARC 64 and 32 bit and Intel 64 bit platforms.

1.8 New Features, Enhancements, and Fixes for Coherence 12*c* (12.1.3.0.2)

The following changes and enhancements have been made for the 12*c* (12.1.3.0.2) release.

- Changes and Enhancements for Oracle Coherence for Java 12c (12.1.3.0.2)
- Changes and Enhancements for Oracle Coherence for .NET 12c (12.1.3.0.2)
- Changes and Enhancements for Oracle Coherence for C++ 12c (12.1.3.0.2)

1.8.1 Changes and Enhancements for Oracle Coherence for Java 12*c* (12.1.3.0.2)

- Fixed incorrect caching after `CoherenceBasedCachingProvider.close()`. Setting the system property `tangosol.coherence.jcache.configuration.classname` to a different value after closing and reacquiring via `Caching.getCachingProvider()` was ignored.
- Fixed an issue where repeated calls to NamedCache create, write and destroy causes a Storage instance leak.
- Fixed an issue where an `addIndex()` run in a non-synchronized thread gave incorrect results for a replicated cache.
- Fixed an issue where low-units was ignored by the SerializationCache.

- Fixed an issue where `NamedCache.lock(key, -1)` does not wait for the lock to be available.
- Fixed the poor performance of asynchronous index rebuilds during failover.
- Fixed an issue causing a slow memory leak for near caches with a high churn rate.
- Coherence JCache POF ids were moved from 700-799 in 12.1.3.0.1 or earlier to 610-699 to avoid collision with Transaction POF ids.
- Fixed an issue with the "partitioned" setting of the backup map. When explicitly set to `false`, it was also setting the in-memory backup partitioned setting to `false`.
- Fixed an issue where `ClusterMBean.logClusterState()` is not generating thread dumps if the `extendedmbeanname` system property is set to `true`.
- Added support for displaying information about front caches when using the JVisualVM plug-in.
- Fixed an issue where Exception null was seen upon stop of FlashJournalRM Daemon thread during service restart.
- Fixed an issue which could cause a `NoSuchElementException` during Ram Journal eviction.
- Added diagnostic information to the `IllegalStateException` thrown by `PacketBundle.write()`.
- Replaced fail-fast cache configuration validation check of the proxy service local address provider validation check when the proxy service is started.
- Fixed an issue where transaction caches may return incorrect `size()` values during failover.
- Removed non-printable characters from the Windows registration entries file, `optimize.reg`.
- Fixed an issue where daemon threads were not using the specified service classloader.
- Fixed an issue which could cause the proxy service thread count to grow unnecessarily large.
- Re-introduced the 12.1.2 behavior where a PartitionListener is only instantiated on storage enabled servers.

1.8.2 Changes and Enhancements for Oracle Coherence for .NET 12*c* (12.1.3.0.2)

- Fixed an issue with high CPU usage being consumed while handling exceptions in `Receive()`.
- Removed "DPR" from the Coherence for .NET version string.
- Message logging can now be enabled by either setting the `COHERENCE_MESSAGING_DEBUG` environment variable to `true` or by adding a `<COHERENCE_MESSAGING_DEBUG>true</COHERENCE_MESSAGING_DEBUG>` element in the `<coherence>` section of the `App.config` file.

1.8.3 Changes and Enhancements for Oracle Coherence for C++ 12*c* (12.1.3.0.2)

- Fixed an issue where `CacheFactory::shutdown()` could hang waiting for the Logger thread to close.

- Changed `FinalView` `InKeySetFilter.f_vSetKeys` to `MemberView` `m_vSetKeys` as `InKeySetFilter::ensureConverted()` requires that `m_vSetKeys` be modifiable.
- Resolved strict-aliasing compilation warnings occurring from the -O3 optimization option.
- Fixed an issue with the memory Allocator always running with heap padding diagnostics enabled.
- Improved the performance of the 32-bit C++ clients to more closely match that of 64-bit clients.

1.9 New Features, Enhancements, and Fixes for Coherence 12c (12.1.3.0.3)

The following changes and enhancements have been made for the 12*c* (12.1.3.0.3) release.

- Changes and Enhancements for Oracle Coherence for Java 12c (12.1.3.0.3)
- Changes and Enhancements for Oracle Coherence for .NET 12c (12.1.3.0.3)
- Changes and Enhancements for Oracle Coherence for C++ 12c (12.1.3.0.3)

1.9.1 Changes and Enhancements for Oracle Coherence for Java 12c (12.1.3.0.3)

- Fixed an issue where `IndexOutOfBoundsException` was being thrown on `ChainedRequest` modification.
- Fixed an issue where `ensureCache` unnecessarily performs access control checks in the absence of a security manager.
- Removed resolving Coherence configuration element names against system properties. For example, system property `autostart` will not override Coherence configuration element `autostart`.
- Fixed an issue where the `ReportTime` column is missing in `report-cache-size.xml`.
- Addressed an issue where the proxy could get stuck in a loop processing extremely small messages and consume excessive CPU.
- Addressed an issue where using the TMB reliable transport over the Windows loopback interface resulted in extremely high latencies.
- Fixed the issue that front map of the NearCache is not cleared if all storage nodes are stopped.
- Entry processor cache update with synthetic set to `true` is now reflected in `CacheEvent.isSynthetic()`.
- Fixed a bug in `NotFilter` where the incorrect results are returned if the inner filter of the `NotFilter` is an `AndFilter` and one of the inner filters of the `AndFilter` uses an index but the other does not.
- Fixed an issue where the front map of the NearCache is not cleared on cache destroy.
- Added Codecs for well known class types `(ArrayList`, `HashMap`, and so on).
- Enabled `TCP_NODELAY` in the JDK `HttpServer` for Coherence REST by default to improve performance.

- Fixed an issue where the default request and connect timeouts for extend connections were being set to infinity instead of the documented 30 seconds.
- Enhanced security to disallow DTD and external entities in the XML data sent by user to Coherence REST by default. User can to enable them by setting the boolean value system properties `coherence.rest.xml.allowDTD` and `coherence.rest.xml.allowExternalEnti`.
- Fixed an issue that caused the `coherence-tx.jar` to not be embedded within the file `coherence-tx.rar`.
- Fixed an issue where Coherence REST could return incorrect JSON string format with no quote around the string.
- Fixed an issue in `ValueChangeEventFilter` to correctly extract the value used by the filter.
- Fixed an issue where cache size value may be incorrect during server failover.
- Coherence JCache `PutIfAbsent` intermittently failed on an entry that was just expired by failing to perform the put.
- Fixed a problem causing a slow memory leak for map listeners that receive local-only (in-process) events.
- Fixed an issue where a Distributed Scheme without backingmap scheme causes a `NullPointerException` during a put call.
- Addressed an issue where large multicast packets would be continuously dropped in Exalogic environments.
- Fixed an issue where a warning was being thrown stating absence of Coherence `IdentityAsserter` to assert identity token.
- Fixed an issue where backward-only `SimpleMapIndex` with `KeyExtractor` is very slow.
- Fixed an issue where `PacketSpeaker` was becoming unresponsive while trying to acquire lock on `UdpSocket`.
- Fixed an issue with Coherence REST where `PUT`, `GET`, and `DELETE` on a single entry may not work when the cache is configured with a cache store.

1.9.2 Changes and Enhancements for Oracle Coherence for .NET 12c (12.1.3.0.3)

- Fixed the issue that front map of the NearCache is not cleared if all storage nodes are stopped.
- Fixed an issue where the front map of the NearCache is not cleared on cache destroy.
- Fixed an issue where .NET clients could get a "`System.ArgumentOutOfRangeException: Specified argument was out of the range of valid values`" exception on port parameter when using the Name Service.
- Fixed an issue with read and write lock acquisition.
- Fixed an issue where `SynchronizedDictionary.AcquireWriteLock()` may hide thread interrupts.

1.9.3 Changes and Enhancements for Oracle Coherence for C++ 12c (12.1.3.0.3)

- Added `Class::getTypeInfo()` and `Class::typeEquals()` methods.

- Fixed the issue that front map of the NearCache is not cleared if all storage nodes are stopped.
- Fixed an issue where the front map of the NearCache is not cleared on cache destroy.
- Fixed an issue with getting debug stack traces.

1.10 New Features, Enhancements, and Fixes for Coherence 12c (12.1.3.0.4)

The following changes and enhancements have been made for the 12*c* (12.1.3.0.4) release.

- Changes and Enhancements for Oracle Coherence for Java 12c (12.1.3.0.4)
- Changes and Enhancements for Oracle Coherence for .NET 12c (12.1.3.0.4)
- Changes and Enhancements for Oracle Coherence for C++ 12c (12.1.3.0.4)

1.10.1 Changes and Enhancements for Oracle Coherence for Java 12c (12.1.3.0.4)

- Fixed a memory leak in the client/proxy with repeated create/destroys of uniquely named caches.
- Fixed an issue where thread starvation could occur when changing from blocking synchronization to non-blocking concurrency control.
- Fixed an issue where filter related calls could yield incorrect results.
- Fixed an issue where `UnsupportedOperationException` was being thrown while calling `entry.setValue` from an entry processor.
- Fixed a memory leak in `ContinuousQueryCache`.
- Fixed an issue where binary keys are being decorated twice in rare scenarios.
- Fixed an issue with the JVisualVM Plugin where the Machine tab and Load Average graph were blank when connected to a cluster using IBM JDK on AIX.
- Fixed the issue that client could get `EOFException` when deserializing invocation result during rolling restart.
- Fixed the issue that cache entries over high-units are not correctly evicted.
- Fixed the issue that a non-serializable exception could cause service termination.
- Improved quorum policy handling to allow for more timely response to timeout-based death detection.
- Updated the refresh data on the Cluster Overview screen in the JVisualVM Plugin to reflect the actual date the JMX model was refreshed.
- Fixed an issue where calling `BackingMapContext.getReadOnlyEntry` from an aggregator fails with an `IllegalArgumentException`.
- Fixed an issue where an undeploy/deploy of a Coherence REST application could encounter a `RequestPolicyException: No storage-enabled nodes exist`.
- Added the ability to sort data tables by any column by clicking on the column header in JVisualVM Plugin.
- Addressed an issue where severe clock skew could result in failure to join the cluster.

- Fixed an issue where unnecessary `unregister MBean` messages are sent to a managed node after a cache is destroyed.
- Fixed an issue where a deadlock may occur during service shutdown when a concurrent thread holds the backing map monitor and is both blocked and can not be interrupted.
- Fixed an issue where `ensureCache` unnecessarily performs access control checks in the absence of a security manager.
- Improved SSL connect to include the handshake in any configured connect timeout.

1.10.2 Changes and Enhancements for Oracle Coherence for .NET 12c (12.1.3.0.4)

- Fixed a concurrent access issue on `CacheFactory` methods.
- Fixed a memory leak in `ContinuousQueryCache`.

1.10.3 Changes and Enhancements for Oracle Coherence for C++ 12c (12.1.3.0.4)

- Fixed an issue where a `LimitFilter` with a `PofExtractor` could throw an `UnsupportedOperationException`.
- Fixed an issue with UTF-8 string validation.
- Fixed an memory leak in `ContinuousQueryCache`.

1.11 Known Problems and Workarounds

This section describes issues that are know at the time of release.

- WebLogic Server Domain Size Limitations
- Building C++ Applications with OS X Mavericks
- Elastic Data Display in Coherence JVisualVM Plugin

1.11.1 WebLogic Server Domain Size Limitations

If you are using Managed Coherence Servers with relatively large clusters, you may encounter Administration Server Memory Consumption issues. For more information on this problem and possible workarounds, see "Administration Server Memory Consumption and JMX Notifications" in *Oracle Fusion Middleware Release Notes for Oracle WebLogic Server Release 12c (12.1.2)*.

1.11.2 Building C++ Applications with OS X Mavericks

When building C++ applications with Apple OS X 10.9 (Mavericks), you must compile with the command "`g++`" (as opposed to "`CC`") and you must use the parameter "`-mmacosx-version-min=10.7`".

1.11.3 Elastic Data Display in Coherence JVisualVM Plugin

There is an intermittent issue with the graphing functionality for Elastic Data in the Coherence JVisualVM Plugin: after a couple of hours of a graph being displayed, small values disappear from the graphs. This mostly affects the Elastic Data tab, but can also affect the data in other tabs.

To workaround this issue, download and install the latest 1.3.6, 1.3.7, or 1.3.8 version of VisualVM which is available at the following URL:

`http://visualvm.java.net/download.html`

1.12 Deprecated Features

This section describes features that have been deprecated in the 12*c* (12.1.3) release.

- XmlConfigurable Interface
- NIO Memory Manager

1.12.1 XmlConfigurable Interface

The `com.tangosol.run.xml.XmlConfigurable` interface has been deprecated in the Coherence 12*c* (12.1.3) release. This interface is no longer required due to injection support. For a different technique to initialize implementations, see Section 2.1, "Replacement for the Deprecated XmlConfigurable Interface."

The following list describes the interfaces, classes, and methods affected by the deprecation of the `com.tangosol.run.xml.XmlConfigurable` interface.

Interfaces being deprecated:

This interface will no longer be required due to injection support.

- `com.tangosol.run.xml.XmlConfigurable`

Methods on Interfaces being deprecated:

This method is no longer required due to injection support.

- `com.tangosol.util.Controllable.configure(XmlElement xml)`

Classes being deprecated:

These classes are replaced by `ParameterizedBuilder` implementations.

- `com.tangosol.io.ConfigurableSerializerFactory`
- `com.tangosol.net.ConfigurableAddressProviderFactory`

Interfaces that will be affected:

All implementations of these interfaces will be configured through injection.

- `com.tangosol.net.BackingMapManagerContext`
- `com.tangosol.net.CacheService`
- `com.tangosol.net.Cluster`
- `com.tangosol.net.InvocationService`
- `com.tangosol.net.NameService`
- `com.tangosol.net.PartitionedService`
- `com.tangosol.net.ProxyService`
- `com.tangosol.net.Service`
- `com.tangosol.util.Service`

Classes that will be affected:

These classes will be configured through injection.

- com.tangosol.coherence.reporter.ColumnView
- com.tangosol.coherence.reporter.ReporterColumnView
- com.tangosol.coherence.rest.config.RestConfig
- com.tangosol.coherence.servlet.AbstractHttpSessionCollection
- com.tangosol.coherence.servlet.CoherenceHttpSessionCollection
- com.tangosol.coherence.servlet.SplitHttpSessionCollection
- com.tangosol.coherence.transaction.internal.ClusterWrapper
- com.tangosol.coherence.transaction.internal.ServiceWrapper
- com.tangosol.coherence.weblogic.PortalCacheProvider
- com.tangosol.io.bdb.BerkeleyDBBinaryStoreManager
- com.tangosol.io.pof.ConfigurablePofContext
- com.tangosol.net.ConfigurableAddressProvider
- com.tangosol.net.SocketOptions
- weblogic.servlet.internal.session.WebLogicSPIHttpSessionIdGenerator

Packages that will be affected:

All classes will be configured through injection.

- com.tangosol.coherence.reporter.locator

Classes that will be affected:

No change required, as these classes are already deprecated.

- com.tangosol.net.CompressionFilter (already deprecated)
- com.tangosol.net.DefaultConfigurableCacheFactory (already deprecated)
- com.tangosol.net.security.AbstractEncryptionFilter (already deprecated)
- com.tangosol.net.security.ClusterEncryptionFilter (already deprecated)

Classes that will be affected:

These classes either do not support XML configuration, perform passthrough, or throw UnsupportedOperationException errors.

- com.tangosol.io.journal.AbstractJournalRM
- com.tangosol.io.journal.FlashJournalRM
- com.tangosol.io.journal.RamJournalRM
- com.tangosol.net.WrapperCacheService
- com.tangosol.net.WrapperInvocationService
- com.tangosol.net.WrapperService

Project Resources / XSDs that will be affected:

- coherence-operational-config-base.xsd

1.12.2 NIO Memory Manager

The NIO backing map implementation that can store information in memory (the NIO Memory Manager) has been deprecated in the current release. Note that the NIO backing map implementation which stores information in memory-mapped files (NIO File Manager) is still valid.

Instead of using the NIO Memory Manager, use the `JournalBinaryStore` class. The `JournalBinaryStore` class implements the binary store used by flash journals and RAM journals in the Elastic Data feature. See "Journaling Overview" in *Oracle Fusion Middleware Developing Applications with Oracle Coherence*.

2 Documentation Errata

This chapter describes changes, enhancements, and corrections made to the Oracle Coherence documentation for 12*c* (12.1.3).

The Coherence documentation for 12*c* (12.1.3) can be found at the following URL:

`http://docs.oracle.com/middleware/1213/coherence/index.html`

This chapter contains the following sections:

- Replacement for the Deprecated XmlConfigurable Interface

2.1 Replacement for the Deprecated XmlConfigurable Interface

The `com.tangosol.run.xml.XmlConfigurable` interface has been deprecated in the Coherence 12*c* (12.1.3) release. Coherence used this interface to inject XML parameters into instances of custom classes.

> **Note:** For information on the interfaces, classes, and methods affected by the deprecation of the `XmlConfigurable` interface, see Section 1.12.1, "XmlConfigurable Interface.".

In the Coherence 12*c* (12.1.3) release, you can initialize parameters by writing XML which nests `<instance>` and `<class-scheme>` (or any other custom namespace) inside of `<param-value>` elements.

For example, given the following Java code:

```
public class MyClass
  {
  public MyClass(String s, OtherClass o, int i) { ... }
  }

public class OtherClass
  {
  public OtherClass(String s) { ... }
  }
```

You can initialize the `MyClass` and `OtherClass` classes by writing the following XML. In the XML, the `MyClass` class is initialized with the string `Hello World` and the integer `42`. The instance of the `OtherClass` class which appears in the `MyClass` class, is initialized with the string `Goodbye World`.

```
<instance>
  <class-name>MyClass</class-name>
    <init-params>
    <init-param>
      <param-value>Hello World</param-value>
    </init-param>
    <init-param>
      <param-value>
        <instance>
          <class-name>OtherClass</class-name>
            <init-params>
              <init-param>
                <param-value>Goodbye World</param-value>
              </init-param>
            </init-params>
        </instance>
      </param-value>
    </init-param>
    <init-param>
      <param-value>42</param-value>
    </init-param>
  </init-params>
</instance>
```