Oracle® Coherence

Release Notes

Release 3.7.1

E22623-19

December 2016

This document contains the release notes for the Oracle Coherence 3.7.1 release.

Oracle Coherence Release Notes, Release 3.7.1

E22623-19

Contents

2 Documentation Errata

Preface

This document describes changes and enhancements that have been made to Oracle Coherence for the 3.7.1 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 Coherence Getting Started Guide*
- *Oracle Coherence Developer's Guide*
- *Oracle Coherence Client Guide*
- *Oracle Coherence Tutorial*
- *User's Guide for Oracle Coherence*Web*
- *Oracle Coherence Integration Guide*
- *Oracle Coherence Management Guide*
- *Oracle CoherenceAdministrator's Guide*
- *Oracle Coherence Security Guide*
- *Integration Guide for Oracle TopLink with Coherence Grid*

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 the 3.7.1 release. This document is accurate at the time of publication. Oracle updates the release notes periodically after the software release.

Coherence 3.7.1 Documentation Library

The Coherence documentation library can be found at the following URL:

`http://download.oracle.com/docs/cd/E24290_01/index.htm`

Download and Install the Latest Software Patch

For Oracle Coherence, patches contain the "full distribution" kit. That is, the patch contains the base product (in this case 3.7.1) plus the contents of the patch.

Patches are also cumulative. That is, each patch contains the contents of previous patches. For example, patch 5 contains the contents of patch 4, plus the contents of all previous patches.

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.

New and Improved Coherence Data Grid Functionality

Oracle Coherence 3.7.1 is the latest release of the industry's leading distributed in-memory data grid product. Coherence 3.7.1 contains many new features that help reduce complexity, ease integration, and accelerate time to market of scalable solutions.

- Oracle Coherence 3.7.1 and higher is supported and certified on Java SE Development Kit (JDK) 7.
- The release of Coherence 3.7.1 provides customers with greater flexibility in data grid client languages, reduced latency and increased throughput of network communication when deployed on Oracle Exalogic Elastic Cloud and significant performance improvements for seamlessly managing data across memory and disk-based solutions.
- Oracle Coherence 3.7.1 introduces a new REST API, enabling customers to leverage popular client languages, such as PHP and Python, to access, manipulate and report on data stored in the Oracle Coherence data grid.
- Oracle Coherence 3.7.1 leverages the Exabus technology exclusive to Oracle Exalogic Elastic Cloud to dramatically increase scalability and lower the latency of data grid operations. Combined with Exabus technology, Oracle Coherence 3.7.1 provides the next generation of massively scalable, low-latency data and state management for middleware applications.

  - Exabus is the high-speed communication (I/O) fabric that ties all Oracle Exalogic system components together and provides the foundation of Oracle Exalogic's extreme performance, reliability and scalability.
  - Exabus consists of unique hardware, software, firmware, device drivers and management tools and is built on Oracle's QDR InfiniBand technology.
- Oracle Coherence 3.7.1 Elastic Data advancements provide more efficient memory and disk-based storage and improve data access performance, giving customers greater flexibility, expanded capacity and more efficiency of their existing infrastructure.
  - The Elastic Data feature of Oracle Coherence allows organizations to leverage both memory and disk-based devices for data grid storage, enabling near memory speed access to data, regardless of storage medium.
  - Oracle Coherence 3.7.1 includes significant optimizations to the journaling algorithms in Elastic Data, providing dramatically improved performance, reduced storage overhead requirements for server side operations and highly-efficient algorithms for data migration from RAM to disk.
- Oracle Coherence 3.7.1 introduces the ability to analyze the efficiency of queries against data in Oracle Coherence by generating an "explain plan" to describe the calculated effectiveness of each element of an Oracle Coherence query.
  - The Query Explain Plan facilitates troubleshooting and managing queries against large distributed data sets, decreases the cost of adoption, reduces complexity and simplifies configuration of scalable distributed data sets.
- Oracle Coherence 3.7.1 is a component of Oracle Fusion Middleware 11*g* and is tightly integrated with the other products in Oracle's middleware stack and Oracle Exalogic Elastic Cloud.
- Oracle Coherence 3.7.1 removes the requirement for a Java implementation of custom key classes used by Coherence*Extend .NET and C++ clients in many cases. A Java class may still be required for some grid features such as filters or queries.
- Oracle Coherence 3.7.1 improves POF to support the use of object identities and references for user-defined objects. References can be used when a large number of sizeable objects are repeated multiple times or when objects use nested or circular data structures.
- Oracle Coherence 3.7.1 adds POF annotation support to simplify the development of POF-serializable classes by allowing developers to mark their serializable classes, fields, and properties with the `Portable` and `PortableProperty` annotations.

Oracle Coherence for Java 3.7.1

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

Management and Monitoring Enhancements and Fixes

- Added support to configure time zone and format for the JMX reporter `date` field.
- Added "escape" character support to CohQL.
- Fixed a memory leak that occurred when registering a custom mBean multiple times.

Exalogic Enhancements and Fixes

- Optimized Elastic Data to prioritize RAM availability to store the primary copies of data, by moving backup copies to or from RAM to flash when necessary.
- Optimized Elastic Data overflow to flash devices by dramatically increasing concurrency.
- Enhanced the Elastic Data feature to trigger its internal garbage collection as its remaining RAM capacity approaches zero. Also optimized the Elastic Data garbage collection to become less aggressive when the available capacity is plentiful.

Partitioned Cache Enhancements and Fixes

- Fixed an issue where executing single-threaded queries while performing write-behind could stall the cache.
- Fixed an issue that could cause duplicate events to be received by a `MapListener` following a failover with `backup-count` > 1.
- Fixed an issue that could result in an `IllegalStateException` while a `getAll()` operation is executed during re-distribution.
- Fixed an issue where write-behind updates on an idle cache could be stalled until the next cache request.
- Fixed an issue that caused missing backups/events when a non-Observable custom backing-map is used.
- Fixed an issue that could cause an erroneous `RequestPolicyException` (`The current quorum policy prevents PartitionSet{...} from being recovered`).
- Fixed an issue that caused `ArrayIndexOutOfBounds` exception to be thrown during partition distribution.
- Fixed a potential deadlock that could occur as a result of the `getAll()` operation.
- Fixed issue with multi-threaded `putAll` operations in combination with invocations may stall the cache.
- Fixed an issue that could result in `failure to apply delta` warnings.
- Fixed a regression introduced by 3.6.1 patch 3 which caused aggregations or invocations to skip some entries under a heavy load.
- Fixed a deadlock that could occur during heavy eviction.
- Optimized memory utilization during an aggregation eliminating the need to keep all the intermittent data for a specific member alive. WARNING: you may get errors performing an aggregation on a partially upgraded cluster when applying this fix to 3.7.0.
- Fixed an issue that occurred when the departure of the (storage) senior member while a new member joined the partitioned cache service, resulted in a failure to create user map indices.
- Fixed an issue where the delta was not applied when using `backup-count-after-writebehind` and POF.
- Fixed a deadlock that could occur during heavy eviction or write-behind.
- Fixed an issue that prevents keys from one partitioned cache from being used as keys to a remote (Extend) partitioned cache.

- Improved the way in which the `MapListenerSupport` class handles filter-based listeners.
- Fixed an issue where `EntryProcessor` objects could be re-executed if the partition is transferred immediately after it is restored from backup.
- Fixed an issue where removal of an entry with `ConditionalIndex` fails in heterogeneous caches.

TCMP Enhancements and Fixes

- Introduced the ability to specify the local IP address as a subnet mask instead of an explicit IP address.
- Added support for large clusters (>500 nodes) without the need for manual packet size configuration.
- Reduced port allocation requirements for running TCMP over TCP.
- Fixed an issue with over-utilization of the 64-kb buffers for multipoint packets.
- Fixed `MulticastTest` and `DatagramTest` to parse for ipv6 literal addresses such as `-local [fe80::216:3eff:fe35:eba3%eth0]:8888`.

Coherence*Extend Framework Enhancements and Fixes

- Hardened `TcpConnection.getDescription()` method against unexpected errors thrown from the Socket API.
- Fixed a `ClassCastException` that could corrupt operation of the `DaemonPool$Daemon.abandon` method.
- Fixed an issue where client Members were not passed to the `ProxyServiceLoadBalancer.getMemberList()` method.
- Improved performance of the `ContinuousQueryCache get` operation.
- Fixed an issue with the ProxyService "server" load balancing policy that caused client connections to be incorrectly rejected.
- Removed the requirement for a Java implementation of custom key classes used by Coherence*Extend .NET and C++ clients in many use cases. A Java class may still be required for some grid features such as filters or queries.

  Key association is now processed on the Extend client by default. Existing client implementations (including Java clients) that rely on key association on the cluster must set the `defer-key-association-check` parameter to force the processing of key classes on the cluster.

  To force key association processing to be done on the cluster side, instead of by the Extend client, set the `<defer-key-association-check>` element, within a `<remote-cache-scheme>` element, in the client-side cache configuration to `true`, for example:

  ```
  <remote-cache-scheme>
     .
     .
     .
     <defer-key-association-check>true</defer-key-association-check>
  </remote-cache-scheme>
  ```

- Removed the dependency on the optional `Common.Logging` library from `DefaultOperationalContext`.

Coherence*Web Enhancements and Fixes

- Added a configuration option to suppress exceptions thrown during session invalidation.
- Fixed the GlassFish SPI (ActiveCache for GlassFish) so that Coherence classes are no longer present in GlassFish after the Web application is undeployed.
- Fixed `coherence-session-weblogic-compatibility-mode` behavior across Web applications.
- Fixed an issue that caused session attributes to be lost when configuring the `SplitSessionModel` with `coherence-enable-suspect-attributes` disabled.
- Moved the `session-cache-config.xml` file into `coherence-web.jar`. It is no longer inside `coherence-web-spi.war` and `webInstaller.jar`.
- The `coherence-web.jar` file is now distributed in the `coherence/lib` directory.

Serialization Framework Enhancements and Fixes

- Fixed incorrect handling of null values in uniform arrays and collections.
- Fixed a bug in the `PofBufferWriter` class which caused an assertion while serializing collections containing null elements.
- Optimized buffer allocation when using the `SerializationHelper` class.
- Improved POF to support the use of object identities and references for user-defined objects. References can be used when a large number of sizeable objects are repeated multiple times or when objects use nested or circular data structures.
- Fixed an issue that caused POF extraction to fail with some compact "small" POF values, such as `true` and `false`.
- POF annotation support simplifies the development of POF-serializable classes by allowing developers to mark their serializable classes and fields (either directly or by using their accessors) with the `Portable` and `PortableProperty` annotations.

Other Enhancements and Fixes

- Included TopLink Grid within the Coherence distribution.
- Fixed an `ArgumentException` thrown by the `ContinuousQueryCache.IsEventDeferred()` method when the supplied key was already recorded.
- Fixed an `EOFException` that occurred when calling the `BinaryEntry.getOriginalValue` method.
- Fixed an issue where the `coherence.jar` file could remain in an application server's deployment directory after it has been undeployed.
- Fixed an issue that could cause slow access to sorted indexes from range comparison filters.
- Fixed a `NullPointerException` (NPE) that could occur during distribution with `backup-count=0`.
- Fixed an issue causing an `IllegalMonitorStateException` to be thrown by deadlock detection under heavy load.

- Fixed a performance regression when calling addIndex with entries that have duplicate values.
- Fixed an issue where the guard timeout was not being honored when using Coherence*Extend.
- Fixed XML validation issue that occurred when running with FMW applications.
- Removed DTD files from Coherence (XSD is now used to validate configuration files).
- The Coherence distribution now includes the jline.jar file which is used with the query command line utility.
- Fixed an issue where the XML Parser did not handle encoding correctly.
- Added "escape" character support to the QueryHelper-based LikeFilter class.

Oracle Coherence for .NET 3.7.1

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

- Oracle Coherence for .NET supports Microsoft .NET 2.0 and higher.
- ASP.NET Session Provider Memory Leak: The root cause for this issue is a memory leak in the .NET4 framework. This leak is documented in http://support.microsoft.com/kb/2540745/en-gb. Microsoft has also released a patch for this issue.
- Fixed a deadlock in .NET ContinuousQueryCache class.
- Fix a .Net SSL issue during simultaneous write operations.
- Improved POF to support the use of object identities and references for user-defined objects. References can be used when a large number of sizeable objects are repeated multiple times or when objects use nested or circular data structures.
- POF annotation support simplifies the development of POF-serializable classes by allowing developers to mark their serializable classes, fields, and properties with the Portable and PortableProperty attributes.

Oracle Coherence for C++ 3.7.1

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

- Fixed an issue with C++ Coherence*Extend clients generating different POF key values for custom key classes containing 64 bit integers.
- Added support for local execution of the ReflectionExtractor and ReflectionUpdater classes to C++.
- Improved POF to support the use of object identities and references for user-defined objects. References can be used when a large number of sizeable objects are repeated multiple times or when objects use nested or circular data structures.
- POF annotation support simplifies the development of POF-serializable classes by allowing developers to mark their serializable classes and fields (by using their accessors) with the Portable and PortableProperty annotations.

Changes and Enhancements for 3.7.1.1

The following changes and enhancements have been made for the 3.7.1.1 release.

- Added the missing security permissions in `security.policy`.
- Added support for service-level reliable transport selection, for example:

```
<distributed-scheme>
   ...
   <reliable-transport>
      datagram
   </reliable-transport>
   ...
</distributed-scheme>
```

- Resolved file descriptor leak in Exabus SocketBus.
- Fixed an issue to make sure that the transaction timeout value is always reset when a connection is acquired from the pool. This ensures that transactions do not time out unexpectedly.
- Fixed an issue with `EntryProcessor` throwing an exception, which in turn caused the backup node to throw a `NullPointerException`.
- Fixed an issue where 0.0.0.0 (`IP_ANY`) could not be used to listen on all IP addresses.
- Fixed issue that caused an `ConcurrentModficiationException` to be thrown during a partition transfer when Exabus is enabled.
- Fixed an issue with a `ConcurrentModificationException` or `NegativeArraySizeException` being thrown when running index-backed queries.
- Made the internal REST `EntryProcessors` POF-serializable, to support exposing POF classes by using REST.
- Fixed an issue where unindexed filtered invocations employed excessive locking.
- Coherence now ignores a bad host name in authorized hosts, and logs a message.
- Resolved a race condition which could result in a `NullPointerException` during service shutdown.
- Fixed an issue with Refresh Ahead when the object being updated is locked.
- Fixed the incorrect Jersey library versions that appeared in the `coherence-rest.jar` manifest file.
- Fixed an issue with non-monotonic registrations of request SUIDs.
- Modified the XSD to allow any XML element to be overridden by using a system-property.
- Fixed an issue where Coherence cannot read its manifest file to obtain the version number.
- Resolved flow control issue causing stuck nodes for services running on Exabus SocketBus.
- Added the ability to set the service and worker thread priorities.
- Resolved a service shutdown race condition which could result in an `IllegalStateException`.
- Fixed an issue that caused an infinite loop in `TcpAcceptor` class.

- Reduced memory consumption overhead for selective indices.
- Fixed an issue with using `LimitFilters` from Coherence*Extend clients.

Changes and Enhancements for 3.7.1.2

The following changes and enhancements have been made for the 3.7.1.2 release.

- The default value for `coherence-session-weblogic-compatibility-mode` system property was incorrect. The default value has been changed to `true`.
- Fixed the issue that caused some Coherence*Web configuration options from being registered when configured by using system properties.
- Fixed .NET `ContinuousQueryCache` missing entries when the cache size is large.
- Installing Coherence for .NET in quiet mode will no longer throw an error if Microsoft Help 2.x Runtime is not present.
- Fixed a regression in `ReadWriteBackingMap.getAll()` logic that could result in insertion of null values into the cache.
- Fixed an issue that caused an assertion to be thrown for performing single key operations while backups are being transferred.
- Preserved the entry expiry time across partition transfer or failover when the system property `tangosol.coherence.distributed.strictexpiry` is specified as `true`.
- Fixed a regression causing incomplete results for queries that use indexes against collection-based extractors.
- Fixed an issue that could result in data-loss with a multi-node failure.
- Fixed an issue that caused `put` requests to be blocked by eviction when storage is slow.
- Resolved an issue where default SSL context was loaded in non-SSL configurations.
- Fixed an issue where the aggregation test hangs during rolling restart for the `tmb`.
- Fixed an issue related to using `ConditionalIndex` with `PofExtractor`.
- Fixed an issue where creating an index while performing queries could stall the cache service.
- Optimized permission checking when the security framework is used such that running a server in a `Subject` context (`Subject.doAs`) will not require repeated authentications.
- Fixed an issue that could raise backing map events out of order when `ObservableHashMap` was used as a backing map.
- Made C++ `LimitFilter::m_vComparator` mutable, to better match how it is used.
- Fixed the issue that caused the domain name of a custom MBean to be ignored and changed to "Coherence". The JMX domain name for custom MBeans registered by applications is now "retained". For more information, see Knowledge Module Doc ID 1392954.1, available from Oracle Support:

  `https://support.oracle.com`

- Added send-timeout support for Extend clients. The `<request-timeout>` value in `<initiator-config>` now limits the time to send the request, in addition to the prior behavior of limiting the time to receive the response.
- Fixed an issue that caused nodes to stop due to "Incompatible BinaryDelta implementation" when a storage-disabled proxy joined a cluster with a different backup count.
- Fixed an issue that could cause an assertion error when encountering extremely long communication delays during service startup.
- Addressed partial network failures when running over Exabus Account for `MessageBus` failure which occurs independent of network/process failure.
- Reduced the cost of removing and recovering from timed-out members.

Changes and Enhancements for 3.7.1.3

The following changes and enhancements have been made for the 3.7.1.3 release.

- Fixed a deadlock that could occur during query evaluation in the presence of heavy concurrent updates.
- Fixed an issue with `ReplicatedCache` that could cause new service members to hold outdated lease information.
- Fixed an issue that caused the aggregation of large data sets to spuriously miss partitions.
- Fixed a regression causing incomplete results for queries that use indexes against collection-based extractors.
- Fixed an issue that could result in an `IllegalStateException` during the `removeIndex` operation.
- Fixed an issue with `PacketPublisher` auto-throttling.
- Coherence now allows the `PacketSpeaker` to be disabled to reduce memory consumption in large WKA clusters.
- Fixed Issue with `IllegalMonitorState` being thrown while performing a query during rapid updates.
- Fixed Coherence*Web to honor the JVM system property `coherence.cache.configuration.path`.
- Fixed a memory leak of pending invocation results, triggered by high `EntryProcessor` load during a failover.
- Resolved an issue in `MessageBus` death-detection which could have lead to the wrong service member being terminated.
- Improved the process of locating the `MANIFEST.MF` file.
- Optimized the `IPMonitor` topology so that a single member on each machine performs outgoing pings.
- Made `PofExtractor` tolerant to null values.
- Fixed an issue in the read-write backing map where `put` requests could be blocked by eviction.
- Removed a potential deadlock which could occur during Coherence*Extend client request timeout.

Changes and Enhancements for 3.7.1.4

The following changes and enhancements have been made for the 3.7.1.4 release.

- For all Extend Clients customers (Java, C++, and .NET), you must upgrade the cluster side **before** upgrading the Coherence*Extend client. This is due to the fix for the problem where incorrect results were being returned to an Extend client using a `LimitFilter` when the Coherence cluster had more than one storage node.
- Fixed an issue that caused the cachestore rolling restart to fail.
- Implemented `hashCode` method on the `PartitionSet` interface.
- Fixed an issue with the Asp.NET `SessionCleanupListener` that caused an `IllegalThreadStateException` to be thrown after a Cluster restart.
- Made changes to Flash and RAM Journal to avoid running out of disk space. For more information, see "Additions and Corrections to Elastic Data Documentation" on page 2-9.
- An exception is no longer thrown when a duplicate `HttpSessionListener` is added in a Web application.
- Fixed registration of custom REST aggregators and processors.
- Fixed an issue which caused the `host-address` element under `authorized-hosts` in operational configuration to be ignored.
- Fixed an issue that could cause a client to be stuck waiting for re-distribution.
- Reduced contention in `PortalCacheProvider.get()` method during heavy concurrent access to the same key.
- Added support for WebSphere Application Server 8 to Coherence*Web.
- Fixed Coherence*Web to honor the JVM system property `coherence.cache.configuration.path`.
- Fixed an issue where incorrect results can be returned to an Extend client using a LimitFilter when the Coherence cluster has more than one storage node.
- Fixed an issue which caused the custom `service-failure-policy` configuration to be ignored.
- Redirect logging to `System.err` if the configured logger is backlogged to avoid an overflow of the log queue.
- Restored the ability to specify the cache name used by the ASP.NET Session Provider with the "`cacheName`" attribute. See "Specifying the Session Cache for Session Management" on page 2-8.
- Fixed missing cache event call-backs when using custom `ExtractorEventTransformer`.
- Fixed a potential `NullReferenceException` during concurrent access to a session.
- Fixed `LockRecursionException` in `AttributeHolder` when accessing session data in Web Session Model.
- Fixed a crashing issue which could occur when registering POF Annotated classes.
- Updated the Coherence for C++ distributions to include the header files for the `NearCache` and `LocalCache` classes.
- Fixed an issue which could cause C++ Class and Allocation `HeapAnalyzers` to crash during initialization.

Changes and Enhancements for 3.7.1.5

The following changes and enhancements have been made for the 3.7.1.5 release.

- Reduced additional memory overhead for running `InvokeAll` operation on `PartitionedCache`.
- Fixed a memory leak caused by Daemon's internal usage of `ThreadGroups`.
- Fix Rolling upgrade issues for 3.7.1.4 and older releases.
- Fixed an issue where an attempt to perform multiple `invokeAll(colKeys, processor)` calls could produce `AssertionException: Invalid heartbeat interval` exceptions.
- Fixed an issue which could cause `entrySet()` method to return a smaller result set if the indexed value is a collection or array.
- Improved accuracy of machine death detection by IP monitor, reduced the possibilities of false positives.
- Added throttling to avoid `PacketPublisher` thread overwhelming the `PacketSpeaker`.
- Fixed an issue where `Request.RequestStatus.WaitForResponse` throws `ArgumentOutOfRangeException` instead of `RequestTimeoutException`.

Changes and Enhancements for 3.7.1.6

The following changes and enhancements have been made for the 3.7.1.6 release.

- Fixed a potential core dump during application shutdown.
- Fixed an issue that caused a cluster to be created if TCMP was disabled.
- Coherence parser can now handle UTF-8 with BOM characters.
- Fixed and issue manifesting in `ClassCastException` with Proxy Server/Extend client when using Optimistic Cache Scheme.
- Fixed an issue where `WindowsFormsCacheListener` could hang processing events.
- Fixed a memory leak in transactional caches with read operations and auto commit enabled.
- Added a Coherence for C++ distribution for Solaris x86 32-bit for use with the STLport library.
- Fixed `NullReferenceException` during deserialize of a user defined object when `<references-enabled>` is set to `true`.
- Fixed the session affinity token configuration in GlassFish SPI.
- Fixed an issue where the `AllFIlter` method was throwing an `ArrayIndexOutOfBoundsException` when initialized with an empty filter array.
- Resolved an issue which prevented an SSL handshake from completing on some IBM JDKs.
- Fixed an issue where Coherence*Extend for C++ clients may not detect a connection being closed by the proxy service.
- Fixed an issue where two members could be started on the same process when a cache server is auto-restarted.

- Resolved an issue in which the packet publisher could overwhelm the packet speaker as a result of moderate to heavy packet loss.
- Fixed an issue where the POF Object References feature may not work due to an uninitialized POF context.
- Connection timeouts are now honored by .NET clients.
- Fixed an issue which could cause Coherence Server OOM for off-heap backup.
- Improved resiliency of an index when a cache entry causes an exception during index update. The index is no longer dropped, so that queries relying on the index continue to execute efficiently. Corrupted entries are excluded from query results.

Changes and Enhancements for 3.7.1.7

The following changes and enhancements have been made for the 3.7.1.7 release.

- Fixed `java.lang.IllegalStateException: invalid minute offset` for non-standard timezone.
- Fixed an issue that could cause `SafeCluster.ensureLocalService()` to throw a `NullPointerException` error
- Fixed an issue that could cause a Coherence*Extend `ContinuousQueryCache` to stop receiving events if the proxy server is restarted.
- Fixed an issue that may cause a data loss during `putAll()` operation from Extend clients that immediately follows a failover or cache creation.
- Fixed an issue where Coherence*Extend clients can receive duplicate events when there is more than one listener configured.
- Fixed an issue which could cause missing `MapListener` events for certain kinds of expiry or eviction changes.
- Fixed an issue with the default `<connect-timeout>` value for Coherence .NET.
- Fixed an issue where calling `NamedCache.invoke` using POF/Extend with objects containing circular references throws `java.lang.StackOverflowError`.
- Fixed an issue where a Coherence for .NET Extend client may make repeated attempts to reconnect to a proxy server which has shut down.
- JMX now reports the correct size for `read-write-backing-map-scheme` cache.
- Fixed "attempt to exit session" `IllegalStateException` during concurrent invalidation by multiple session reapers.
- Changed the Coherence for C++ thread dump handler to run as a detached thread.
- Fixed an issue where a resource lock timeout during the `getAll,putAll removeAll` operations could result in service failure.
- Fixed flash journal max file size configuration so that it is not ignored.
- Fixed an issue where a Coherence for C++ Extend client connection may not failover when a proxy server goes down.
- Fixed an issue where `ClassCastException` could be thrown when `LimitFilter` is used with custom `Comparator`.
- Fixed an issue that caused `$QuorumRollCall$Poll` to throw a `NullPointerException` error.

- Improved resiliency of an index when a cache entry causes an exception during an index update. The index is no longer dropped, so that queries relying on the index continue to execute efficiently. Corrupted entries are excluded from query results.
- Remainders are now handled correctly for nested evolvable POF objects.

Changes and Enhancements for 3.7.1.8

The following changes and enhancements have been made for the 3.7.1.8 release.

- Changes and Enhancements for Oracle Coherence—All Platforms 3.7.1.8
- Changes and Enhancements for Oracle Coherence for Java 3.7.1.8
- Changes and Enhancements for Oracle Coherence for .NET 3.7.1.8
- Changes and Enhancements for Oracle Coherence for C++ 3.7.1.8

Changes and Enhancements for Oracle Coherence—All Platforms 3.7.1.8

> **Reminder:** For all Extend Clients customers (Java, C++, and .NET), you must upgrade the cluster side before upgrading the Coherence*Extend client. This is in compliance with our client and proxy upgrade policy. While there are some patches where it may be possible to get away with not following this policy, it must be followed for this patch due to the fix for the issue where Filter Listeners were not receiving the correct number of Events.

- Fixed an issue with Extend clients receiving duplicate events in some cases for key and filter based listeners.
- Fixed an issue where the `ContinuousQueryCache` might not receive events after the proxy has been restarted.

Changes and Enhancements for Oracle Coherence for Java 3.7.1.8

- Fixed a regression in the initial 3.7.1.8 patch with default distribution.
- Fixed an issue that causes a `NullPointerException` to be thrown during cluster formation, if the `IpMonitor` is disabled.
- Fixed an issue where a deadlock was seen between `put/clear` threads with `ReplicatedCache`.
- Fixed an issue avoiding unnecessary `MemberSet` object creation.
- Updated the Coherence*Web `JspServlet` class to extend `GenericServlet` so that instrumented JSPs work in Tomcat Server.
- Fixed the `IllegalStateException` which was thrown after a cache destroy/recreate operation was executed using an Extend client.
- Fixed a race condition where setting the `ClassLoader` on a service after it has been started would result in the service's serializer being temporarily set to null. Note that it is recommended that the `ClassLoader` only be set prior to starting a service.
- Fixed a potential race condition where a `ConfigurablePofContext` could be used before it is fully initialized.

- Fixed an issue when executing concurrent `EntryProcessors` where a change made by one processor may not be visible to a subsequent processor.
- Fixed an issue that caused excessive memory usage by lightly used near caches after a cluster disconnect and reconnect.
- Fixed an entry expiry issue where the expiry reset `byBinaryEntryStore` is ignored.
- Fixed an issue where a new entry expiry set by a map trigger is ignored.
- Fixed an `OutOfMemoryIssue` that occurs immediately after a cluster member with a `NearCache` configured with `AUTO` or `ALL` joins the cluster.
- Fixed an issue with query operations resulting in `CacheLoader/BinaryEntryStore#load` if entries concurrently expire from the backing map.
- Display of JMX attribute `MaxQueryDescription` (under `StorageManager`) is now truncated to 1024 characters.
- Fixed a `java.lang.ArrayIndexOutOfBoundsException` that can occur when using `LimitFilter` objects.
- The Javadoc documentation for the methods `RequestMaxDuration`, `RequestAverageDuration`, `RequestPendingCount`, `RequestPendingDuration` of classes `ServiceModel` and `ServiceMBean` have been updated.
- Fixed a memory leak caused by stopping and restarting the cluster multiple times in the same JVM.
- Added a default constructor for `CoherenceWebSessionData` for a specific customer requirement.
- Added support for WebSphere Application Server 8.5 to Coherence*Web.

Changes and Enhancements for Oracle Coherence for .NET 3.7.1.8

- Fixed an issue where `UUID.CompareTo()` may return the wrong result for certain IP addresses.
- Coherence for .NET Extend connections now have correct timestamps in the proxy server log messages and connection MBeans.
- Implemented a workaround for the issue where `PofStreamWriter.WriteLocalDateTime()` recorded an incorrect value for certain `DateTime` values. This issue was caused by a bug in `TimeZone.CurrentTimeZone.GetUtcOffset()` in .NET Framework 4.5.

Changes and Enhancements for Oracle Coherence for C++ 3.7.1.8

- Fixed an issue where Coherence for C++ may read freed memory during a network address lookup.
- Updated the Coherence for C++ header files so that applications can be compiled using GCC 4.7.
- Fixed a C++ compiler error in `NativeAtomic64.hpp` on Linux.

Changes and Enhancements for 3.7.1.9

The following changes and enhancements have been made for the 3.7.1.9 release.

- Changes and Enhancements for Oracle Coherence—All Platforms 3.7.1.9

- Changes and Enhancements for Oracle Coherence for Java 3.7.1.9
- Changes and Enhancements for Oracle Coherence for .NET 3.7.1.9
- Changes and Enhancements for Oracle Coherence for C++ 3.7.1.9

Changes and Enhancements for Oracle Coherence—All Platforms 3.7.1.9

- Fixed an issue which caused duplicate delivery of transformed `MapEvents`.
- Introduced a `LISTEN_LOGICAL` strategy for `CachingMap` to listen to all back map events that are not synthetic deletes. A synthetic event could be emitted as a result of eviction or expiration. With this invalidation strategy, it is possible for the front map to contain cache entries that have been synthetically removed from the back (though any subsequent re-insertion will cause the corresponding entries in the front map to be invalidated).

Changes and Enhancements for Oracle Coherence for Java 3.7.1.9

- Fixed an issue which could cause a cluster configured with WKA hostnames to be stopped by the guardian if the DNS becomes unavailable. This issue only affects WKA cluster configurations which rely on symbolic hostnames which require DNS resolution.
- `PofHelper.getPofTypeId()` will now return the correct type ID for user types that derive from `ArrayList`.
- Removed a `NullPointerException` during Coherence*Web initialization in older versions (pre-10.3.3) of WebLogic
- Fixed an issue where the `PofHelper` method `getPofTypeId()` was returning the incorrect `type id` for user types deriving from an `ArrayList`.
- Fixed an issue that could result in a `NullPointerException` being thrown when registering a partial result during server failover.
- Fixed an issue where custom proxies were not being called when ensuring and/or destroying a cache.
- Fixed the display values for `BufferPublishSize` and `BufferReceiveSize` attributes of the `ClusterNodeMBean`.
- Added the `management-config` configuration element `extended-mbean-name` to reflect the member name in global MBean object names.
- Added support for `HttpOnly` cookie attribute in instrumented Coherence*Web web applications.
- Added a Coherence*Web configuration parameter to replace stripped-off suffix with a new affinity suffix.
- Fixed an issue that caused native memory (`DirectByteBuffer`) to be allocated unnecessarily.
- Fixed an issue with invalid reporter configuration leading to an unsafe state.
- Binary keys will be re-used for key listener registrations.

Changes and Enhancements for Oracle Coherence for .NET 3.7.1.9

- Added `ReducerAggregator` to Coherence for .NET and C++ so that the `ReducerAggregator` can be invoked from these languages.

- Fixed the Coherence for .NET library to look for the correct XML configuration elements for using network filters, like the gzip compression filter.
- `PofHelper.getPofTypeId()` will now return the correct type ID for user types that derive from `ArrayList`.

Changes and Enhancements for Oracle Coherence for C++ 3.7.1.9

- Fixed `adapter_map/boxing_map` iterator initialization when the map size is 1.
- Added `ReducerAggregator` to Coherence for .NET and C++ so that the `ReducerAggregator` can be invoked from these languages.
- Improved performance by removing unnecessary mutex locking when accessing registered classes and serializers in the `SystemPofContext`.

Changes and Enhancements for 3.7.1.10

The following changes and enhancements have been made for the 3.7.1.10 release.

- Changes and Enhancements for Oracle Coherence—All Platforms 3.7.1.10
- Changes and Enhancements for Oracle Coherence for Java 3.7.1.10
- Changes and Enhancements for Oracle Coherence for .NET 3.7.1.10

Changes and Enhancements for Oracle Coherence—All Platforms 3.7.1.10

- Fixed an `InFilter` serialization issue.
- Fixed an issue where integer value of -31 is not read properly by the POF serializer.
- Hardened the handling of cache deactivation calls.

Changes and Enhancements for Oracle Coherence for Java 3.7.1.10

- Fixed a custom `Comparator` issue with proxy service load balancing.
- Improved the sending of `DiagnosticPackets` such that they are sent to both advertised and preferred ports.
- Fixed an issue with Elastic Data in which an invalid binary exception is reported during an operation that directly or indirectly calls keySet.
- Hardened the transactional cache handling of `javax.transaction.xa.Xid` to allow non-serializable implementations.
- Added a new session configuration in Coherence*Web which keeps a local instance of a session in addition to flushing the session to the distributed cache.
- Added a new session configuration in Coherence*Web which keeps an local instance of a session in addition to flushing the session to the distribute cache. For more information, see "Getting Concurrent Access to the Same Session Instance" on page 2-13.
- Fixed an issue that could result in false incompatible WKA list warnings when host names are used in the WKA list
- Fixed an issue causing `NamedCache` iterators to return duplicate results to Extend clients from Standard edition clusters.
- Fixed an issue where `RWBM.flush()` could return before all data are flushed.

- Fixed an issue that caused the unit calculator configuration to be ignored by the `ReadWriteBackingMap`.
- Fixed an issue resulting in deadlock between the thread starting the cluster and the thread starting a service.
- Fixed an issue where listeners on Replicated Cache could not receive map events.
- Fixed an issue which could cause client memory leak if partition distribution "stuck".
- Fixed an issue with `Filter` execution producing high memory overhead due to an unnecessary JMX statistic.
- Fixed a potential session initialization issue in WebLogic Server during side-by-side deployment.
- Fixed an issue where integer value of -31 is not read properly by the POF serializer.
- Added support for Coherence*Web with Tomcat 7.*x* application servers.
- Hardened the handling of cache misconfigurations by guarding against Null Pointer Exceptions (NPE) and Unsupported Operation Exceptions (UOE) to provide a more stable service.

Changes and Enhancements for Oracle Coherence for .NET 3.7.1.10

- Fixed an issue where type -29 is not read properly by the POF serializer.
- Improved client certificate selection for SSL enabled Coherence*Extend connections.
- Removed Coherence for .NET installer prerequisite on .NET Framework 2.0.

Changes and Enhancements for 3.7.1.11

The following changes and enhancements have been made for the 3.7.1.11 release.

- Changes and Enhancements for Oracle Coherence—All Platforms 3.7.1.11
- Changes and Enhancements for Oracle Coherence for Java 3.7.1.11
- Changes and Enhancements for Oracle Coherence for .NET 3.7.1.11
- Changes and Enhancements for Oracle Coherence for C++ 3.7.1.11

Changes and Enhancements for Oracle Coherence—All Platforms 3.7.1.11

- Improved `ConnectionException` to return the cause of the exception from the proxy server.

Changes and Enhancements for Oracle Coherence for Java 3.7.1.11

- Reduced contention on the Proxy service for key-based requests that target the same keys.
- Fixed an issue in `ensureCache` which can result in client threads becoming stuck if the `ensureCache` was unresponsive.
- Fixed an issue where an Extend client doesn't receive delete events from a replicated cache.

- Fixed a ConcurrentModificationException that may occur in RefreshableAddressProvider upon retrieving the next address when the last address in the list was rejected.
- Fixed an issue in which we may duplicate the call to the CacheStore on failover for a successfully stored entry.
- Fixed an issue where IllegalStateException was thrown from CoherenceWebSessionContextImpl while retrieving the session.
- Hardened the PacketReciever against randomly truncated, otherwise valid packets.
- Fixed an issue where Coherence*Web on Tomcat 7.0 reported incorrect Tomcat version information in the log file.
- Fixed an issue in which CacheMBean may produce an exception if a non-ConfigurableCacheMap instance is used as the front map.
- Added ability to define MapTriggerListeners for remote caches by using the <listener> element.
- Fixed an erroneous clean-up of locally stored session attributes.
- Fixed a possible Guardian time out issue which might occur when restoring orphaned partitions.
- Fixed Session Reaper to ensure that local attributes are cleaned up when session invalidation is initiated from a different node.
- Fixed a potential AssertionException in Coherence*Web during application undeploy.
- Fixed an issue that could prevent a conditional index from correctly indexing an entry.
- Fixed a potential race condition between a guarded thread timing out and the thread no longer being guarded for ReadWriteBackingMaps.
- Fixed an issue in the asynchronous SSL implementation that may cause a request to hang when SSL communication is enabled.
- Added the "RACK-SAFE" and "SITE-SAFE" support for statusHA and configured them to be displayed only when the partition assignment strategy is not legacy.

Changes and Enhancements for Oracle Coherence for .NET 3.7.1.11

- Added a SimplePrincipal which derives from GenericPrincipal. SimplePrincipal uses the Identity.Name property to determine object equality.
- Fixed a .NET 4.5 framework issue with SynchronizedDictionary.
- Added ability to define MapTriggerListeners for remote caches by using the <listener> element.

Changes and Enhancements for Oracle Coherence for C++ 3.7.1.11

- Added "-mmacosx-version-min=10.4" to the examples compilation commands to address link errors with Xcode 5.
- Changed DetachFinalizer::set() to detach from the old object with the old escaped flag.

- Updated TypedMethod.hpp so that Xcode (clang) no longer warns about unused functions in the file.
- Fixed LiteSet so that it can support more than two elements.
- Fixed an issue with String::substring prematurely ending a substring search.

Changes and Enhancements for 3.7.1.12

The following changes and enhancements have been made for the 3.7.1.12 release.

- Changes and Enhancements for Oracle Coherence—All Platforms 3.7.1.12
- Changes and Enhancements for Oracle Coherence for Java 3.7.1.12
- Changes and Enhancements for Oracle Coherence for .NET 3.7.1.12

Changes and Enhancements for Oracle Coherence—All Platforms 3.7.1.12

- Fixed a serialization buffer allocation performance issue.

Changes and Enhancements for Oracle Coherence for Java 3.7.1.12

- Fixed an issue where a ConcurrentModificationException may be thrown when invoking ConditionalRemove on a ReplicatedCache from an Extend client.
- When using the Coherence Cache Provider with WebLogic Portal, the cache configuration may be set using a system property.
- Changed the proxy service worker threads to close Coherence*Extend connections asynchronously when a connection will be closed due to an error.
- Fixed NotSerializableException in Coherence*Web for WebLogic Portal components during session invalidation.
- Fixed an issue that could cause the index size from getUnits() to become negative.
- Updated PofValue.getValue() API documentation to indicate that the type must be explicitly specified for primitive types.
- Fixed an issue with events not being delivered to pre-3.7.1 Coherence*Extend clients.
- During session invalidation, ConcurrentModificationException is logged only when the logging level is set to FINEST.
- Reduced the contention when ReadWriteBackingMap is subject to high load and evictions.

Changes and Enhancements for Oracle Coherence for .NET 3.7.1.12

- Fixed an event delivery issue for cache listeners registered with filters that use a PofExtractor.
- Updated PofValue.getValue() API documentation to indicate that the type must be explicitly specified for primitive types.
- Fixed an issue with events not being delivered to pre-3.7.1 Coherence*Extend clients.

- Changed `Coherence.dll` for Coherence.NET to have no diagnostic instrumentation overhead. `Coherence.pdb` is still provided for debugging purposes.

Changes and Enhancements for 3.7.1.13

The following changes and enhancements have been made for the 3.7.1.13 release:

- Changes and Enhancements for Oracle Coherence—All Platforms 3.7.1.13
- Changes and Enhancements for Oracle Coherence for Java 3.7.1.13
- Changes and Enhancements for Oracle Coherence for .NET 3.7.1.13
- Changes and Enhancements for Oracle Coherence for C++ 3.7.1.13

Changes and Enhancements for Oracle Coherence—All Platforms 3.7.1.13

- Fixed a concurrent access issue with UUIDs.

Changes and Enhancements for Oracle Coherence for Java 3.7.1.13

- Fixed an issue with `UpdaterProcessor` that could cause missing entry update in `CacheStore`, `BinaryEntryStore`, and `Listeners`.
- Fixed an issue where internal session attribute names are returned without stripping the `InternalWLSAttribute` prefix.
- Fixed a POF serialization issue for negative Dates with nonzero sub-seconds.
- Fixed an issue where session attributes are mandated to implement `java.io.Serializable`.
- Implemented Servlet 3.1 feature which enables an application container to update the session ID after authentication.
- Fixed memory leak in release cache.
- Fixed an issue which could cause server to fail to join the cluster with `tmbs` protocol.
- Fixed an issue where internal session attribute names are returned without stripping the `InternalWLSAttribute` prefix.
- Fixed an issue which could cause `TcpAcceptor` hang when decoding non-Coherence messages.
- Improved POF date and time deserialization performance.
- Fixed an issue where `ReadWriteBackingMap` may unnecessarily log messages regarding an unexpected `EvictionApprover`.
- Fixed an issue where `cmd` scripts shipped by Coherence fail to handle `JAVA_HOME` path containing spaces and parenthesis.
- Fixed memory leak in release cache.
- Fixed an issue where `MultiExtractor` was not using indexes for its `ValueExtractor`.
- Provided limited support for `PofValueParser` and `PofNavigator` when object identity/reference is enabled. If the user performs an unsupported function, then an `UnsupportedOperationException` will be thrown.

- Fixed an issue with RefreshableAddressProvider in which previously unaccepted addresses may not be retried after successful connection.

Changes and Enhancements for Oracle Coherence for .NET 3.7.1.13

- Fixed `msiexec /x coherence-net-3.7.1.*.msi /qb` to no longer show error message dialog.
- Corrected an issue when iterating the entries, keys, or values of a ContinuousQueryCache.
- Provided limited support for PofValueParser and PofNavigator when object identity/reference is enabled. If the user performs an unsupported function, then an UnsupportedOperationException will be thrown.

Changes and Enhancements for Oracle Coherence for C++ 3.7.1.13

- Fixed an issue where dependent threads of a Coherence service may fail to stop during service shutdown.
- Fixed a typo in the guard macro for XorFilter.hpp.

Changes and Enhancements for 3.7.1.14

The following changes and enhancements have been made for the 3.7.1.14 release:

- Changes and Enhancements for Oracle Coherence—All Platforms 3.7.1.14
- Changes and Enhancements for Oracle Coherence for Java 3.7.1.14
- Changes and Enhancements for Oracle Coherence for .NET 3.7.1.14
- Changes and Enhancements for Oracle Coherence for C++ 3.7.1.14

Changes and Enhancements for Oracle Coherence—All Platforms 3.7.1.14

- Disabled POF object references for cache keys to prevent the possibility of two cache entries with the same key. For further details including potential upgrade issues, see "Upgrading to 3.7.1.14 or Higher" on page 1-25.

Changes and Enhancements for Oracle Coherence for Java 3.7.1.14

- Fixed an issue where asynchronous ownership is requested by WLS SPI code for Coherence*Web.
- Improved SSL connect to include handshake in any configured connect timeout.
- Fixed an issue where the ReportTime column is missing in report-cache-size.xml.
- Added codecs for well known class types like ArrayList, HashMap, and so on.
- Disabled POF object references for cache keys to prevent the possibility of two cache entries with the same key. For further details including potential upgrade issues, see "Upgrading to 3.7.1.14 or Higher" on page 1-25.
- Fixed a problem that caused unnecessary "unregister MBean" messages sent to a managed node after a cache is destroyed.
- Fixed an issue when under high load using expiry or eviction resulting in AssertionExceptions or deadlock.

- Fixed the issue that cache entries over high-units are not correctly evicted.
- Fixed an issue where `addIndex()` run in a non-synchronized thread gave wrong results for a replicated cache.
- Fixed an issue that could prevent the service or cluster from restarting after abnormal failure.
- Fixed an issue where exception `null` was seen upon stop of FlashJournalRM Daemon thread during service restart.
- Fixed an issue where `cache-server.sh` and `coherence.sh` scripts were using CRLF characters for line endings.
- Fixed an issue where daemon threads were not using the specified service classloader.
- Fixed an issue where `<cipher-suites>` element in cluster configuration caused `SAXParseException`.
- Fixed an issue where an `ArrayIndexOutofBoundsException` is thrown from `ClusterModel.getMemberIds()` call due to some of cluster members going down during the operation.
- Resolved a stability issue on IBM J9 JVM when running Coherence on TCP.
- Fixed an issue where invocation of `QueryPlus.main` preceded by `QueryHelper.createFilter` caused `NullPointerException`.

Changes and Enhancements for Oracle Coherence for .NET 3.7.1.14

- Disabled POF object references for cache keys to prevent the possibility of two cache entries with the same key. For further details including potential upgrade issues, see "Upgrading to 3.7.1.14 or Higher" on page 1-25.
- Fixed an issue where `SynchronizedDictionary.AcquireWriteLock()` may hide thread interrupts.
- Fixed an issue where a client could end up with a thread stuck in a CPU busy loop if an error occurs while processing a message from the proxy server.
- Fixed a concurrent access issue on `CacheFactory` methods.

Changes and Enhancements for Oracle Coherence for C++ 3.7.1.14

- Added `Class::getTypeInfo()` and `Class::typeEquals()` methods.
- Disabled POF object references for cache keys to prevent the possibility of two cache entries with the same key. For further details including potential upgrade issues, see "Upgrading to 3.7.1.14 or Higher" on page 1-25.
- Fixed an issue where `CacheFactory::shutdown()` could hang waiting for the `Logger` thread to close.
- Updated some header files to address GCC 4.8 compilation warnings.

Changes and Enhancements for 3.7.1.15

The following changes and enhancements have been made for the 3.7.1.15 release:

- Changes and Enhancements for Oracle Coherence for Java 3.7.1.15

Changes and Enhancements for Oracle Coherence for Java 3.7.1.15

- Fixed an issue where Coherence*Web does not save sessions to the cache when used with WebLogic servers that have the patch for bug 18057437.
- Fixed an issue where HTTP sessions are not saved when Coherence*Web is being used by PeopleSoft applications.
- Fixed an issue where a NIC with multiple IP addresses can trigger IpMonitor failures.
- Fixed an issue where the SSL configuration password may be logged if an exception occurs due to invalid SSL configuration.
- Fixed an issue where a non-SSL client could trigger a node to restart.
- Fixed an issue where Coherence*web is creating a new session cache (by appending the appversion to the appname) for each version of an application.
- Improved diagnostic logging on port binding failures when auto-adjust is disabled.
- Fixed a potential concurrency issue with `SafeClock.updateSafeTimeMillis()` which could result in thread hangs.
- Hardened tmb against erroneous socket closures caused by interrupts or idle connections.

Changes and Enhancements for 3.7.1.16

The following changes and enhancements have been made for the 3.7.1.16 release:

- Changes and Enhancements for Oracle Coherence for Java 3.7.1.16

Changes and Enhancements for Oracle Coherence for Java 3.7.1.16

- Fixed an issue where a `PartitionedCache` service could be terminated due to a `NullPointerException` being thrown in `PartitionedCache$Storage.moveResourcesToPrimary().`
- Fix an issue where an IllegalArgumentException could be thrown when calling cache destroy while a partition transfer is in process.
- Changed the default value of the system property "`tangosol.coherence.publisher.yieldonflush`" to "`false`" to address Coherence cluster performance issues in some environments.
- Addressed an issue where threads accessing Coherence via Extend could have the interrupt status of a thread erroneously cleared.
- Addressed an issue where an inconsistent configuration of the Coherence reliable transport setting could result in a `NullPointerException.`
- Fixed an issue where a deadlock may occur in an *Extend client when the connection to the proxy server is closed.
- Improved Coherence's management of off-heap `DirectByteBuffers` to more closely couple their life-cycle to that of the garbage collector.
- Added Windows TCP retransmission setting overrides to optimize.reg to help reduce false death detections by allowing the Coherence cluster members to be more tolerant of short network outages.

- Added support for enabling or disabling specific SSL protocols within Coherence's SSL configuration.
- Fixed an issue where an interrupted Cluster start operation can cause the Cluster service to fail to shutdown cleanly.

Known Problems and Workarounds

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

- Upgrading to 3.7.1.14 or Higher
- Building C++ Applications Using Apple OS X Mavericks
- Querying POF Objects

Upgrading to 3.7.1.14 or Higher

To prevent the possibility of two cache entries with the same key, POF object references are disabled for cache keys in release 3.7.1.14. This fix, in some cases, breaks compatibility with earlier 3.7.1.*x* releases. For partitioned caches, the serialized key format in 3.7.1.14 is different than keys that are generated with 3.7.1.13 and earlier when all of the following conditions are true:

- The `PartitionedCache` service is configured to use POF serialization.
- The POF serializer is configured with object references enabled (`<enable-references>` is set to `true`). Object reference support is disabled by default.
- One or more of the caches in the `PartitionedCache` service contain entries with keys that are complex type objects (that is, an object class that does not have built-in POF support).

> **Note:** Note that the 3.7.1.14 key format is now identical to what is used by release 12.1.2 and above. So, 3.7.1.14 Coherence*Extend clients may interact with a 12.1.2 or later version Coherence cluster.

Starting with 3.7.1.14 and later releases, POF object references are disabled for cache key serialization to prevent a condition where key objects which are `.equals()` equivalent but are not the same object (not `==`) can result in different serialized keys. This issue can cause duplicate entries for the same key and can also result in what appears to be missing, phantom, or inaccessible cache entries.

For cases where one or more of the previously mentioned conditions are not true, there are no issues with upgrading to 3.7.1.14. The normal upgrade process can be used. That is, perform a rolling restart, upgrade the cluster before the clients, and so on.

If all the conditions are true, then perform the following upgrade process:

- If the Coherence cluster is 12.1.2 or above, and the upgrade is only being done to Coherence*Extend clients, then all 3.7.1.*x* clients should be upgraded to 3.7.1.14 at the same time. If they are not, then cache entries added by an older 3.7.1.*x* client may not be accessible by 3.7.1.14 (and 12.1.2 and above) clients and the cluster members.
- If the Coherence cluster is 3.7.1.*x* (3.7.1.0 - 3.7.1.13), then the cluster must first be shutdown prior to patching. All data must be reloaded so that the keys are in the correct serialized format. In other words, a rolling restart is not possible. In

addition, all 3.7.1.*x* Coherence*Extend clients must be upgraded. Older 3.7.1.*x* version Coherence*Extend clients may not be able to access data in the cluster and any data inserted by older clients may not be accessible by 3.7.1.14 Coherence*Extend clients and cluster members. In terms of ordering, the 3.7.1.*x* cluster must be upgraded to 3.7.1.14 first then followed by upgrading the 3.7.1.*x* Coherence*Extend clients.

Building C++ Applications Using Apple 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=10.4`.

Querying POF Objects

The use of object identity and references has the following limitations:

- Object references are only supported for user-defined object types.
- Evolvable objects are not able to use object references.
- Objects that have been written out with a POF context that does not support references cannot be read by a POF context that supports references. The opposite is also true.
- POF objects that use object identity and references cannot be queried using POF extractors. Instead, use the `ValueExtractor` API to query object values or disable object references.

Deprecated Features

The following features have been deprecated or removed from the Coherence 3.7.1 release.

- DTD files have been removed from the Coherence distribution. XSD is now used to validate configuration files.
- Support for the symmetric and PKCS encryption filters has been removed in this release of Oracle Coherence. Compression filters and custom network filters are still supported. You can find documentation for network filters in the 3.7.0 release of the *Developer's Guide for Oracle Coherence*.

2 Documentation Errata

This chapter describes changes, enhancements, and corrections made to the Oracle Coherence documentation library for the 3.7.1 release.

The Coherence documentation library can be found at the following URL:

`http://download.oracle.com/docs/cd/E24290_01/index.htm`

This chapter contains the following sections:

Serializing Cache Keys

This section provides addendum instructions to the POF documentation. Release 3.7.1.14 introduces a change which, in some cases, breaks compatibility with earlier 3.7.1.*x* releases. Specifically, the format of serialized keys has been changed in a partitioned cache when all of the following conditions are true:

- The `PartitionedCache` service is configured to use POF serialization
- The POF serializer is configured with object references enabled (object reference support is disabled by default)
- One or more of the caches in the `PartitionedCache` service contain entries with keys that are complex type objects (That is, an object class created by a customer and not one of the class types that have built-in POF support like `String`, or `Int`).

If all of the above conditions are true, then those cache entries will have serialized keys that are different than keys generated by 3.7.1.13 and earlier 3.7.1 releases. Note that the 3.7.1.14 key format is now identical to what is used by 12.1.2 and above releases. So, 3.7.1.14 Coherence*Extend clients may interact with a 12.1.2 or later version Coherence cluster.

What is the change exactly?

Starting with 3.7.1.14, and for 12.1.2 and later releases, POF object references are disabled for cache key serialization.

Why was this change made?

This is to prevent a condition where key objects which are .equals() equivalent, but are not the same object (not "==") may result in different serialized keys. This issue could cause duplicate entries for the same key. Or could result in what appears to be missing cache entries or "phantom" or inaccessible entries.

What is the upgrade path for this change?

For cases where one or more of the above 3 conditions are NOT true, there are no issues with upgrade. The normal upgrade process can be used, i.e. rolling restart, cluster upgrade before clients, etc.. This should be the case for the vast majority of our customers.

If the 3 above conditions ARE true, then the following must be done:

A. If the Coherence cluster is 12.1.2 or above, and the upgrade is only being done to Coherence*Extend clients, then ALL 3.7.1.x clients should be upgraded to 3.7.1.14 at the same time. If they are not, then cache entries added by an older 3.7.1.x client may not be accessible by 3.7.1.14 (and 12.1.2 and above) clients and the cluster members.

or,

B. If the Coherence cluster is 3.7.1.x (3.7.1.0 - 3.7.1.13), then the cluster must first be shutdown prior to patching. All data must be reloaded so that the keys will be in the correct serialized format. In other words, a rolling restart is not possible. Additionally all 3.7.1.x Coherence*Extend clients MUST be upgraded. Older 3.7.1.x version Coherence*Extend clients may not be able to access data in the cluster and any data inserted by older versioned clients may not be accessible by 3.7.1.14 Coherence*Extend clients and cluster members. In terms of ordering the 3.7.1.x cluster must be upgraded to 3.7.1.14 first. Followed by upgrading the 3.7.1.x Coherence*Extend clients.

Incorrect Interface in the REST Documentation

The REST documentation in *Oracle Coherence Client Guide* refers to a `com.tangosol.util.EntryAggregator` interface that is used to create custom aggregators. This interface does not exist in the distribution. The correct API should be `com.tangosol.util.InvocableMap.EntryAggregator`.

POF Serialization on a Proxy Server

The *Oracle Coherence Client Guide* incorrectly documents POF serialization behavior on a proxy: "The proxy deserializes POF data put into the cache, and serialize data it returns to the client." The proxy no longer performs deserialization and POF data remains serialized.

POF Object References are Not Evolvable

POF supports the use of object identities and references for objects that occur more than once in a POF stream. POF object references are not supported for `Evolvable` objects. The `Evolvable` interface is implemented by classes that require forwards- and backwards-compatibility of their serialized form. For more information on the `Evolvable` interface, see the `Evolvable` interface in the *Java API Reference for Oracle Coherence*.

Note these additional limitations on using POF object references:

- Object references are only supported for user-defined object types.
- Object references are not supported for keys.

- Objects that have been written out with a POF context that does not support references cannot be read by a POF context that supports references. The opposite is also true.
- POF objects that use object identity and references cannot be queried using POF extractors. Instead, use the `ValueExtractor` API to query object values or disable object references.

For more information on POF object references, see "Using POF Object References" in *Oracle Coherence Developer's Guide*.

Support for Coherence on Virtualized Environments

Coherence Administrator's Guide incorrectly states that Coherence is supported on all virtualized environments. The correct statement is that Oracle Coherence follows the support policies of Oracle Fusion Middleware. See the following link for Oracle Fusion Middleware supported virtualization and partitioning technologies:

http://www.oracle.com/technetwork/middleware/ias/oracleas-supported-virtualization-089265.html

Configuring the Name of the Cache File for Coherence*Web

The context parameter `coherence-cache-configuration-path` allows Coherence*Web users to configure the name of the cache file in the `web.xml` file.

The following example illustrates a sample `coherence-cache-configuration-path` configuration. In the example, *`your-cache-config-name.xml`* represents the name of the cache configuration file in the *`your_app`* directory that you want Coherence*Web to use.

```
...
<context-param>
   <param-name>coherence-cache-configuration-path</param-name>
   <param-value>c:\your_app\your-cache-config-name.xml</param-value>
</context-param>
...
```

Coherence*Web and WebLogic Server in the Tutorial

In Step 6 of "Create the Counter Web Application" in *Oracle Coherence Tutorial* lists the name of the manifest file as being `manifest.mf`. This is an error. The name of the manifest file is case-sensitive and must be in upper case: `MANIFEST.MF`.

If you are using the 10.3.6 version of WebLogic Server with Coherence 3.7.1, then in the section "Start a Cache Server", you must add the `-Dtangosol.coherence.cluster=CoherenceCluster` system property to the startup command.

In the 10.3.6 version of WebLogic Server, a cluster name check has been added. The `tangosol.coherence.cluster` property must be added to the cache server because you are declaring the cluster name in the WebLogic Server application. If the Coherence servers are started in standalone mode, they must pass this property, otherwise the cluster will not form between the WLS servers and the standalone cache server.

Custom Eviction Policies Must Extend AbstractEvictionPolicy Class or Implement EvictionPolicy Interface

Custom eviction policies must extend the `com.tangosol.net.cache.AbstractEvictionPolicy` class or implement the `com.tangosol.net.cache.ConfigurableCacheMap.EvictionPolicy` interface. The documentation for the `<local-scheme>` element in the *Oracle Coherence Developer's Guide* incorrectly states to implement the `com.tangosol.net.cache.LocalCache.EvictionPolicy` interface.

IMB Reliable Transport Excluded From Documentation

The InfiniBand Message Bus (IMB) protocol can be enabled for reliable transport. IMB uses an optimized protocol based on native InfiniBand verbs. For details about configuring reliable transport, see "unicast-listener Subelements" in *Oracle Fusion Middleware Developing Applications with Oracle Coherence*.

Incorrect Large Cluster Recommendation

The Large Cluster Configuration Recommendation section in *Oracle Fusion Middleware Administering Oracle Coherence* contains an error regarding setting the partition count. The first bullet in the section should be replaced as follows:

Distributed caches on large clusters of more than 16 cache servers require more partitions to ensure optimal performance. The default partition count is 257 and should be increased relative to the number of cache servers in the cluster and the amount of data being stored in each partition.

Incorrect AbstractMultiplexingBackingMapListener Implementation Example

Example 21-17 in *Oracle Coherence Developer's Guide* contains an error that affects performance. The example should *not* call the `sleep` method on the current thread. The example should be written as follows:

```
import com.tangosol.net.BackingMapManagerContext;
import com.tangosol.util.MapEvent;

public class VerboseBackingMapListener extends
AbstractMultiplexingBackingMapListener {

        public VerboseBackingMapListener(BackingMapManagerContext context) {
                super(context);
        }

        @Override
        protected void onBackingMapEvent(MapEvent mapEvent, Cause cause) {
                System.out.printf("Thread: %s Cause: %s Event: %s\n",
                Thread.currentThread().getName(), cause, mapEvent);
        }
}
```

Incorrect Default Value for low-units Element

In *Oracle Coherence Developer's Guide*, the default value of the `<low-units>` element is incorrectly documented as 75% of the `<high-units>` value. The correct default value is 80% of the `<high-units>` element.

Incorrect Value for service-component Element

Table A-49 in the "Operational Configuration Elements" appendix in *Oracle Coherence Developer's Guide* incorrectly lists `DistributedCache` as a legal value for the `<service-component>` element. The correct value should be `PartitionedService.PartitionedCache`.

Incorrect Subelements Listed for front-scheme Element

Table B-23 in the "Cache Configuration Elements" appendix in *Oracle Coherence Developer's Guide* incorrectly lists `<external-scheme>`, `<paged-external-scheme>`, `<flashjournal-scheme>`, and `<ramjournal-scheme>` as being valid subelements of the `<front-scheme>` element. In the Coherence 3.7.1.*x* releases, the only valid subelements of the `<front-scheme>` element are `<local-scheme>` and `<class-scheme>`.

Incorrect Command Line to Start Cache Server

"Deploying and Running Applications Out-of-Process" in the *User's Guide for Oracle Coherence*Web*. incorrectly lists the application WAR file (`c:/application.war`) in the command line to start a cache server. The corrected command line should be as follows:

```
java -server -Xms512m -Xmx512m  -cp <Coherence installation
dir>/lib/coherence.jar:<Coherence installation dir>/lib/coherence-web.jar
-Dtangosol.coherence.management.remote=true
-Dtangosol.coherence.cacheconfig=session-cache-config.xml
-Dtangosol.coherence.session.localstorage=true com.tangosol.net.DefaultCacheServer
```

Corrections to WebLogic Portal—Coherence*Web Integration Instructions

This section describes corrections to the instructions to integrate WebLogic Portal with Coherence*Web that appear in the "Using Coherence*Web with WebLogic Portal" chapter in the *User's Guide for Oracle Coherence*Web*.

Start a Cache Server Section:

In the "Start a Cache Server" section, there is an error in the cache server startup command in Step 1. The command should be the following:

```
java -server -Xms512m -Xmx512m -cp <Coherence installation
dir>/lib/coherence.jar:<Coherence installation dir>/lib/coherence-web-spi.war
-Dtangosol.coherence.management.remote=true
-Dtangosol.coherence.cacheconfig=C:\Coherence371\coherence\session-cache-config.xm
l -Dtangosol.coherence.session.localstorage=true
com.tangosol.net.DefaultCacheServer
```

The command assumes that you have extracted the `session-cache-config.xml` file from the `coherence-web.jar` file and stored it in the `C:\Coherence371\coherence\` directory.

In 3.7.1, the session-cache-config.xml file has been moved from the coherence-web-spi.war file to the coherence-web.jar file. The coherence-web.jar file can be found in the coherence\lib directory.

Deploying Coherence Cache Provider for Out-of-Process Topology Section:

This section provides revised instructions for deploying the Coherence cache provider for out-of-process topologies.

1. Obtain the session cache and portal cache configuration files:
   - Extract the session-cache-config.xml file from the coherence-web.jar file.
   - Extract the portal-cache-config.xml file from the coherence-wlp.jar file.
2. Merge the contents of the session-cache-config.xml and portal-cache-config.xml files so that a set of Coherence grid nodes can support both types of objects: session and P13N cache.

   You will need two copies of this file: one that will be used by the Coherence cache servers and one that will be used by the WebLogic Portal managed servers. The contents of the two files will be identical, except for the values of the local-storage properties in the portal cache and the session cache sections.

   Example D–1 in Appendix D, "Cache Configuration for WebLogic Portal and Oracle Coherence" illustrates a merged configuration file that can be used for the WebLogic Portal managed servers. The local-storage values for the session cache and portal cache are set to false. The configuration file for the Coherence cache server can be the same as Example D–1, except the values for the local-storage system property in the session cache configuration section and portal cache configuration section must be set to true.
3. Use the merged configuration file for the WebLogic Portal managed server as the cache configuration file for the servers. For each application that will run on the WebLogic Portal managed server, copy the file to the application's APP-INF/classes directory.
4. Set the merged configuration file for the Coherence caches as the cache configuration for the Coherence grid JVMs. For example, in the following system property to locate the cache configuration file, portal-session-cache-config.xml represents the merged file:

   ```
   -Dtangosol.coherence.cacheconfig=D:\Coherence1212\coherence\temp\portal-session
   -cache-config.xml
   ```
5. Add the system properties to disable local storage for the WebLogic Portal managed servers. For example, you can add the following properties to your Coherence cache server startup script:

   ```
   -Dtangosol.coherence.weblogic.localstorage=false
   -Dtangosol.coherence.distributed.localstorage=false
   -Dtangosol.coherence.session.localstorage=false
   ```
6. Copy the coherence.jar and coherence-wlp.jar files to the APP-INF/lib directory of your EAR file.

   See the deployment instructions in "Using Coherence*Web with WebLogic Portal—Main Steps".
7. Start the WebLogic Portal managed servers.

Incorrect Initialization Parameter Listed for PofAnnotationSerializer Class

The "Enabling Automatic Indexing" section in *Coherence Client Guide* incorrectly lists a `PofAnnotationSerializer` class initialization parameter type as `boolean`. The correct parameter type is `bool`.

Incorrect Macro Name for the Reporter file-name Element

The documentation incorrectly lists `sequence` as a valid macro for the `<file-name>` element that is located in a report configuration file. The correct macro name is `batch`.

Using the id Attribute in Host-Based Authorization

The "Using Host-Based Authorization" section in the *Coherence Security Guide* does not include instructions for using the `id` attribute with the `<host-address>` and `<host-range>` elements. The `id` attribute is used to uniquely identify multiple address or range elements, for example:

```
<authorized-hosts>
   <host-address id="1">192.168.0.5</host-address>
   <host-address id="2">192.168.0.6</host-address>
   <host-range id="1">
      <from-address>192.168.0.10</from-address>
      <to-address>192.168.0.20</to-address>
   </host-range>
   <host-range id="2">
      <from-address>192.168.0.30</from-address>
      <to-address>192.168.0.40</to-address>
   </host-range>
</authorized-hosts>
```

Specifying the Session Cache for Session Management

In the "Setting Up Coherence Session Management" section of the *Coherence Client Guide*, you must also specify a session cache if you do not want to use the default cache. If a session cache is not specified, then the default cache, `aspnet-session-storage`, is used. In the following example, the cache name is specified as `my-session-cache`.

```
...
<providers>
    <add name="CoherenceSessionProvider"
              type="Tangosol.Web.CoherenceSessionStore, Coherence"
              cacheName="my-session-cache"
              model="split"
              externalAttributeSize="512"/>
</providers>
...
```

Incorrect Schema Versions

The documentation provides the wrong schema version in the public URL for schema locations. The correct URLs are as follows:

- Operational Configuration:

  `http://xmlns.oracle.com/coherence/coherence-operational-config/1.1/coherence-operational-config.xsd`

- Coherence Cache Configuration:

  http://xmlns.oracle.com/coherence/coherence-cache-config/1.1/coherence-cache-config.xsd

- POF Configuration:

  http://xmlns.oracle.com/coherence/coherence-pof-config/1.1/coherence-pof-config.xsd

Additions and Corrections to Elastic Data Documentation

There are several additions and corrections which should be made to the Elastic Data documentation in the *Coherence Developer's Guide*.

- In the section "Using a Journal Scheme for Backup Storage", the line which reads: "By default, a distributed scheme that is configured to use a RAM journal as a backing map also uses a RAM journal for backup storage" should be changed to: "By default, Flash Journal is used as the backup storage."
- In the section "Configuring the Flash Journal Resource Manager", the following bullet should be added to the list:
  - Journal also allows you to set a soft limit (called `high-journal-size`) on the size. This allows the compaction (GC) thread to tune itself to prune stale values from the journal so that it doesn't grow beyond the soft limit. This is not a hard-limit, and the journal could still grow beyond that up to the max file count of 512. The default `high-journal-size` element is 11GB.

    It is generally recommended to just set the `high-journal-size` element and let the system auto-tune itself, for example:

    ```
    <?xml version='1.0'?>

    <coherence xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://xmlns.oracle.com/coherence/coherence-operational-config"
    xsi:schemaLocation="http://xmlns.oracle.com/coherence/
    coherence-operational-config coherence-operational-config.xsd">
       <cluster-config>
          <journaling-config>
              <flashjournal-manager>
                 <maximum-value-size>64K</maximum-value-size>
                 <maximum-file-size>8M</maximum-file-size>
                 <block-size>512K</block-size>
                 <maximum-pool-size>32M</maximum-pool-size>
                 <directory>/coherence_storage</directory>
                 <async-limit>32M</async-limit>
                 <high-journal-size>11GB</high-journal-size>
              </flashjournal-manager>
          </journaling-config>
       </cluster-config>
    </coherence>
    ```

- In the table for "flashjournal-manager" in the "Operational Configuration Elements" appendix, add a row for the `<high-journal-size>` subelement:

Table 2–1 high-journal-size Subelement for the flashjournal-manager Element

Element	Required/ Optional	Description
`<high-journal-size>`	Optional	Specifies the soft limit on the journal size. This allows the compaction (GC) thread to tune itself to prune stale values from the journal to keep it within the soft limit. This is not a hard-limit and the journal could still grow beyond that up to the max file count (512). The default `high-journal-size` is 11GB. It can also be configured using the system property `-Dtangosol.coherence.flashjournal.highjournalsize`

Coherence*Web Session Locking Context Parameter Values for Session Locking Modes

"Session Locking Modes" in *Oracle Coherence User's Guide for Oracle Coherence*Web* describes the locking modes that are available in Coherence*Web and the context parameter values that are required to specify each mode. Table 2–2 presents a summary of the session parameter values for each session locking mode.

Table 2–2 Values of Session Locking Context Parameters for Session Locking Modes

Session Locking Modes	coherence-session-thread-locking	coherence-session-app-locking	coherence-session-member-locking	coherence-session-locking
Last Write Wins Locking (default)	false (default)	false (default)	false (default)	false (default)
Optimistic Locking	false	false	false	true
Member Locking	false	false	true	(true)
Application Locking	false	true	(true)	(true)
Thread Locking	true	(true)	(true)	(true)

* (true) means that the context parameter values are changed automatically by the super context parameter value.

Incorrect Default Value for coherence-session-thread-locking Parameter

The description of the `coherence-session-thread-locking` parameter in the "Coherence*Web Context Parameters" appendix in *User's Guide for Oracle Coherence*Web* incorrectly lists the default value as `true`. The correct default value should be `false`.

Incorrect Value for the tangosol.coherence.management.remote Property

In "Stopping a Cluster Member from being Managed Remotely" in *Oracle Coherence Management Guide*. the example at the end of the section lists an incorrect value of `true` for the `tangosol.coherence.management.remote` property. The correct value for the property should be as follows:

```
-Dtangosol.coherence.management.remote=false
```

Incorrect Information for Optimistic Locking for Coherence*Web

The 3.7.1 version of the *User's Guide for Oracle Coherence*Web* contains the following incorrect information for Optimistic Locking:

In the section, "Optimistic Locking":

- *Coherence*Web and the Coherence*Web SPI are configured with Optimistic Locking by default.* This is not true. The default locking for Coherence *Web and Coherence*Web SPI is Last Write Wins Locking.
- *When Coherence*Web detects a concurrent modification, an exception,* `ConcurrentModificationException` *is thrown to the application. Therefore, an application must be prepared to handle this exception in an appropriate manner.* This is not true. The exception `ConcurrentModificationException` is thrown when the session is flushed to the cache, which is after the Servlet request has finished processing.
- *The Optimistic Locking mode can be configured by setting the coherence-session-member-locking context parameter to false.* The `coherence-session-member-locking` context parameter is already `false` by default. To set optimistic locking mode, set `coherence-session-locking` context parameter to `true`.

In Appendix A, "Coherence*Web Context Parameters":

- The description of the `coherence-session-member-locking` context parameter contains a cross reference to the Optimistic Locking section. This cross reference belongs with the `coherence-session-locking` context parameter.

Incorrect Information Regarding Sticky Sessions and Distribution Controller for Coherence*Web

The description of the `coherence-distribution-class` context parameter in the 3.7.1 and earlier versions of "Coherence*Web Context Parameters" in *User's Guide for Oracle Coherence*Web* states that this parameter requires `coherence-sticky-sessions` optimization to be enabled. This is not correct. The `coherence-distribution-class` context parameter is not dependent on sticky sessions.

Incorrect Example for Specifying Cluster Member Authorized Hosts

The example listed in "Specify Cluster Member Authorized Hosts" in *Oracle Coherence Security Guide* is incorrect. In the example, you must use the `id` attribute in `host-address` and `host-range` to uniquely identify multiple elements. The corrected example follows:

```
<?xml version='1.0'?>

<coherence xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="http://xmlns.oracle.com/coherence/coherence-operational-config"
   xsi:schemaLocation="http://xmlns.oracle.com/coherence/
   coherence-operational-config coherence-operational-config.xsd">
   <cluster-config>
      <authorized-hosts>
         <host-address id="1">192.168.0.5</host-address>
         <host-address id="2">192.168.0.6</host-address>
         <host-range id="1">
            <from-address>192.168.0.10</from-address>
            <to-address>192.168.0.20</to-address>
         </host-range>
         <host-range id="2">
            <from-address>192.168.0.30</from-address>
            <to-address>192.168.0.40</to-address>
         </host-range>
      </authorized-hosts>
   </cluster-config>
</coherence>
```

Coherence*Web Support for IBM WebSphere Application Server

Support for WebSphere Application Server 8 was added to Coherence*Web in release 3.7.1.4. Support for WebSphere Application Server 8.5 was added in release 3.7.1.8.

Corrections and Clarifications to Configuring Coherence*Web for WebLogic Portal

There were errors in the "Deploying Coherence Cache Provider for Out-of-Process Topology" section of *User's Guide for Oracle Coherence*Web*. Corrections and clarifications have been made to the instructions on how to deploy the Coherence cache provider for out-of-process topology in the WebLogic Server Portal. Use the following instructions instead.

1. Obtain the session cache and portal cache configuration files:

   - Extract the `default-session-cache-config.xml` file from the `coherence-web.jar` file.

   - Extract the `portal-cache-config.xml` file from the `coherence-wlp.jar` file.

2. For the Coherence Cache Server:

   a. Merge the contents of the `default-session-cache-config.xml` and `portal-cache-config.xml` files so that a set of Coherence grid nodes can support both types of objects: session and P13N cache.

   You will need two copies of this file: one that will be used by the Coherence cache servers and one that will be used by the WebLogic Portal managed servers. The contents of the two files will be identical, except for the values of the `local-storage` properties in the portal cache and the session cache sections.

   Example F–1 in Appendix F, "Cache Configuration for WebLogic Portal and Oracle Coherence" illustrates a merged configuration file that can be used for the WebLogic Portal managed servers. The `local-storage` values for the session cache and portal cache are set to `false`. The configuration file for the Coherence cache server can be the same as Example F–1, except the values for the `local-storage` system property in the session cache configuration section and portal cache configuration section must be set to `true`.

   b. Set the merged configuration file for the Coherence caches as the cache configuration for the Coherence grid JVMs. For example, in the following system property to locate the cache configuration file, `portal-session-cache-config.xml` represents the merged file:

   ```
   -Dtangosol.coherence.cacheconfig=D:\Coherence1212\coherence\temp\portal-ses
   sion-cache-config.xml
   ```

3. For the Portal Server:

   a. Use the merged configuration file for the WebLogic Portal managed server as the cache configuration file for the servers. For each application that will run on the WebLogic Portal managed server, copy the file to the application's `APP-INF/classes` directory.

   b. Add the system properties to disable local storage for the WebLogic Portal managed servers and optionally specify the cache configuration file. For example, you can add the following properties to your Coherence cache server startup script:

```
-Dtangosol.coherence.weblogic.localstorage=false
-Dtangosol.coherence.distributed.localstorage=false
-Dtangosol.coherence.session.localstorage=false
-Dtangosol.coherence.cacheconfig=D:\Coherence1212\coherence\temp\portal-ses
sion-cache-config.xml
```

4. Copy the `coherence.jar` and `coherence-wlp.jar` files to the `APP-INF/lib` directory of your EAR file.

   See the deployment instructions in "Using Coherence*Web with WebLogic Portal—Main Steps" on page 10-3.

5. Start the WebLogic Portal managed servers.

Getting Concurrent Access to the Same Session Instance

A cache delegator class is a class that is responsible for manipulating (getting, putting, or deleting) any data in the distributed cache. Use the `<coherence-cache-delegator-class>` context parameter in the `web.xml` file to specify the name of the class responsible for the data manipulation.

One of the possible values for the context parameter is the `com.tangosol.coherence.servlet.LocalSessionCacheDelegator` class. This class indicates that the local cache should be used for storing and retrieving the session instance before attempting to use the distributed cache. This delegator is useful for applications that require concurrent access to the same session instance.

> **Note:** This feature must be enabled when working with PeopleSoft applications.

To enable the `LocalSessionCacheDelegator` cache delegator, the following items must be configured in `web.xml`:

- The `coherence-cache-delegator-class` context parameter with the value set to `com.tangosol.coherence.servlet.LocalSessionCacheDelegator`.
- The `coherence-preserve-attributes` context parameter set to `true` to allow nonserializable objects to be stored in the session object.
- The `coherence-distributioncontroller-class` context parameter with the value set to `com.tangosol.coherence.servlet.AbstractHttpSessionCollection$HybridController`. This value forces all sessions and serializable attributes to be managed in a distributed manner. All session attributes that do not implement the `Serializable` interface will be kept local. Note that the use of this context parameter also requires `coherence-sticky-sessions` optimization to be enabled.

Example 2–1 illustrates a sample configuration for the cache delegator.

Example 2–1 Configuring Cache Delegator in the web.xml File

```
...
 <context-param>
    <param-name>coherence-cache-delegator-class</param-name>
    <param-value>com.tangosol.coherence.servlet.LocalSessionCacheDelegator
</param-value>
 </context-param>
 <context-param>
    <param-name>coherence-preserve-attributes</param-name>
    <param-value>true</param-value>
 </context-param>
 <context-param>
    <param-name>coherence-distributioncontroller-class</param-name>

<param-value>com.tangosol.coherence.servlet.AbstractHttpSessionCollection$HybridCo
ntroller</param-value>
 </context-param>
...
```

Also, when using `LocalSessionCacheDelegator` as the cache delegator, you should not configure a near cache in the `session-cache-config.xml` file. This is because local session instances are used. Example 2–2 illustrates a sample `session-cache-config.xml` file without a near cache configuration. The `scheme-name` under `cache-mapping` has been changed from `session-storage` to `session-distributed`. The `near-scheme` section has been removed, but the `local-scheme` and `distributed-scheme` remain. This is noted in **`bold`** font in the example.

Example 2–2 session-cache-config.xml File Without a Near Cache Configuration

```
<?xml version="1.0"?>
<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
<!--                                                                   -->
<!--      Cache configuration descriptor for Coherence*Web             -->
<!--                                                                   -->
<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->

<cache-config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xmlns="http://xmlns.oracle.com/coherence/coherence-cache-config"

xsi:schemaLocation="http://xmlns.oracle.com/coherence/coherence-cache-config
coherence-cache-config.xsd">

  <caching-scheme-mapping>
    <!--
    The clustered cache used to store Session management data.
    -->
    <cache-mapping>
      <cache-name>session-management</cache-name>
      <scheme-name>replicated</scheme-name>
    </cache-mapping>

    <!--
    The clustered cache used to store ServletContext attributes.
    -->
    <cache-mapping>
      <cache-name>servletcontext-storage</cache-name>
      <scheme-name>replicated</scheme-name>
    </cache-mapping>
    <!--
    The clustered cache used to store Session attributes.
    -->
<!--
session-near has been changed to session-distributed
-->
    <cache-mapping>
      <cache-name>session-storage</cache-name>
      <scheme-name>session-distributed</scheme-name>
    </cache-mapping>

    <!--
    The clustered cache used to store the "overflowing" (split-out due to size)
    Session attributes. Only used for the "Split" model.
    -->
    <cache-mapping>
      <cache-name>session-overflow</cache-name>
      <scheme-name>session-distributed</scheme-name>
    </cache-mapping>

    <!--
    The clustered cache used to store IDs of "recently departed" Sessions.
    -->
    <cache-mapping>
      <cache-name>session-death-certificates</cache-name>
      <scheme-name>session-certificate</scheme-name>
    </cache-mapping>

    <!--
    The local cache used to store Sessions that are not yet distributed (if
    there is a distribution controller).
    -->
    <cache-mapping>
      <cache-name>local-session-storage</cache-name>
      <scheme-name>unlimited-local</scheme-name>
    </cache-mapping>

    <!--
    The local cache used to store Session attributes that are not distributed
    (if there is a distribution controller or attributes are allowed to become
    local when serialization fails).
    -->
    <cache-mapping>
      <cache-name>local-attribute-storage</cache-name>
      <scheme-name>unlimited-local</scheme-name>
    </cache-mapping>
  </caching-scheme-mapping>

  <caching-schemes>
    <!--
    Replicated caching scheme used by the Session management and ServletContext
    attribute caches.
    -->
    <replicated-scheme>
      <scheme-name>replicated</scheme-name>
      <service-name>ReplicatedSessionsMisc</service-name>
      <request-timeout>30s</request-timeout>
      <backing-map-scheme>
        <local-scheme>
```

```
          <scheme-ref>unlimited-local</scheme-ref>
        </local-scheme>
      </backing-map-scheme>
      <autostart>true</autostart>
    </replicated-scheme>

<!--
The near-scheme section has been removed
-->
    <local-scheme>
      <scheme-name>session-front</scheme-name>
      <eviction-policy>HYBRID</eviction-policy>
      <high-units>1000</high-units>
      <low-units>750</low-units>
    </local-scheme>

    <distributed-scheme>
      <scheme-name>session-distributed</scheme-name>
      <scheme-ref>session-base</scheme-ref>
      <backing-map-scheme>
        <local-scheme>
          <scheme-ref>unlimited-local</scheme-ref>
        </local-scheme>
        <!-- for disk overflow use this backing scheme instead:
        <overflow-scheme>
          <scheme-ref>session-paging</scheme-ref>
        </overflow-scheme>
        -->
      </backing-map-scheme>
    </distributed-scheme>

    <!--
    Distributed caching scheme used by the "recently departed" Session cache.
    -->
    <distributed-scheme>
      <scheme-name>session-certificate</scheme-name>
      <scheme-ref>session-base</scheme-ref>
      <backing-map-scheme>
        <local-scheme>
          <eviction-policy>HYBRID</eviction-policy>
          <high-units>4000</high-units>
          <low-units>3000</low-units>
          <expiry-delay>86400</expiry-delay>
        </local-scheme>
      </backing-map-scheme>
    </distributed-scheme>

    <!--
    "Base" Distributed caching scheme that defines common configuration.
    -->
    <distributed-scheme>
      <scheme-name>session-base</scheme-name>
      <service-name>DistributedSessions</service-name>
      <thread-count>0</thread-count>
      <lease-granularity>member</lease-granularity>
      <local-storage
system-property="tangosol.coherence.session.localstorage">false</local-storage>
      <partition-count>257</partition-count>
      <backup-count>1</backup-count>
      <backup-storage>
```

```
        <type>on-heap</type>
      </backup-storage>
      <request-timeout>30s</request-timeout>
      <backing-map-scheme>
        <local-scheme>
          <scheme-ref>unlimited-local</scheme-ref>
        </local-scheme>
      </backing-map-scheme>
      <autostart>true</autostart>
    </distributed-scheme>

    <!--
    Disk-based Session attribute overflow caching scheme.
    -->
    <overflow-scheme>
      <scheme-name>session-paging</scheme-name>
      <front-scheme>
        <local-scheme>
          <scheme-ref>session-front</scheme-ref>
        </local-scheme>
      </front-scheme>
      <back-scheme>
        <external-scheme>
          <bdb-store-manager/>
        </external-scheme>
      </back-scheme>
    </overflow-scheme>

    <!--
    Local caching scheme definition used by all caches that do not require an
    eviction policy.
    -->
    <local-scheme>
      <scheme-name>unlimited-local</scheme-name>
      <service-name>LocalSessionCache</service-name>
    </local-scheme>

    <!--
    Clustered invocation service that manages sticky session ownership.
    -->
    <invocation-scheme>
      <service-name>SessionOwnership</service-name>
      <request-timeout>30s</request-timeout>
    </invocation-scheme>
  </caching-schemes>
</cache-config>
```

Errors in the Merged Session Cache/Portal Cache File for Coherence*Web

There are syntax errors in the sample merged `session-cache-config.xml` and `portal-cache-config.xml` configuration file listed in the "Cache Configuration for WebLogic Portal and Oracle Coherence" appendix of the *User's Guide for Oracle Coherence*Web*. Please use the following sample file instead.

Example 2–3 Cache Configuration for WebLogic Portal Managed Server

```
<?xml version="1.0"?>

<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
```

```
<!--                                                                       -->
<!--        Coherence BEA WebLogic Portal CacheProvider:                   -->
<!--                  Cache Configuration Descriptor                       -->
<!--                                                                       -->
<!--       (See http://e-docs.bea.com/wlp/docs81/perftune/apenB.html)      -->
<!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->

<cache-config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xmlns="http://xmlns.oracle.com/coherence/coherence-cache-config"

xsi:schemaLocation="http://xmlns.oracle.com/coherence/coherence-cache-config
coherence-cache-config.xsd">

  <caching-scheme-mapping>

    <cache-mapping>
      <!--
      Cache:
        adServiceCache

      Use:
        Used to store the results of searches for content rendered in a
        placeholder (ads). Used by the AdHelper to increase the speed of ad
        queries.

      Key:
        An ad query (java.lang.String).

      Value:
        An array of com.bea.p13n.content.Content objects.
      -->
      <cache-name>adServiceCache:*</cache-name>
      <scheme-name>portal-local</scheme-name>
      <init-params>
        <init-param>
          <param-name>local-size-limit</param-name>
          <param-value>32</param-value>
        </init-param>
        <init-param>
          <param-name>local-expiry</param-name>
          <param-value>5m</param-value>
        </init-param>
      </init-params>
    </cache-mapping>

    <cache-mapping>
      <!--
      Cache:
        binaryCache.[repository name]

      Use:
        Used to cache binary property values for the BEA Repository.

      Key:
        Node ID + property ID (java.lang.String).

      Value:
        A byte array associated with the binary property.
      -->
      <cache-name>binaryCache.*</cache-name>
      <scheme-name>portal-local</scheme-name>
      <init-params>
        <init-param>
          <param-name>local-size-limit</param-name>
          <param-value>256</param-value>
        </init-param>
      </init-params>
    </cache-mapping>

    <cache-mapping>
      <!--
      Cache:
        CategoryCache

      Use:
        Used to cache the root com.beasys.commerce.ebusiness.catalog.Category,
        the total number of categories in the product catalog (java.lang.Integer)
        and the
com.beasys.commerce.ebusiness.catalog.service.category.CategoryInfo
        for each category.

      Key:
        The key for the root Category and the total number of categories is a
        java.lang.String. The key for a given CategoryInfo object is a
        com.beasys.commerce.ebusiness.catalog.CategoryKey.

      Value:
        The value for the root Category is a
com.beasys.commerce.ebusiness.catalog.Category.
        The value for the total number of categories is a java.lang.Integer.
        The value for the category info objects is a
        com.beasys.commerce.ebusiness.catalog.service.category.CategoryInfo.
      -->
      <cache-name>CategoryCache:*</cache-name>
      <scheme-name>portal-local</scheme-name>
      <init-params>
        <init-param>
          <param-name>local-size-limit</param-name>
          <param-value>1000</param-value>
        </init-param>
        <init-param>
          <param-name>local-expiry</param-name>
          <param-value>1d</param-value>
        </init-param>
      </init-params>
    </cache-mapping>

    <cache-mapping>
      <!--
      Cache:
        CategoryTreeCache

      Use:
        Used to cache portlet category trees.

      Key:
        A webapp name.

      Value:
        A CategoryTree object.
```

```
  -->
  <cache-name>CategoryTreeCache:*</cache-name>
  <scheme-name>portal-replicated</scheme-name>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    communitiesEntityPropertyCache

  Use:
    Used to cache community membership capability information for users
    accessing communities.

  Key:
    A composite key of community definition ID and user name.

  Value:
    A java.util.Map of community membership capabilities.
  -->
  <cache-name>communitiesEntityPropertyCache:*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-size-limit</param-name>
      <param-value>1024</param-value>
    </init-param>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>1d</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    communitiesMemberActiveCache

  Use:
    Used to cache information about active status for community members.

  Key:
     A user name.

  Value:
    A java.lang.String representing the user's community member record
    active status.
  -->
  <cache-name>communitiesMemberActiveCache:*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-size-limit</param-name>
      <param-value>1024</param-value>
    </init-param>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>1d</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    complexProducerPortletHandleToIdCache

  Use:
    Used to cache the primary instance ID of portlets.

  Key:
    A remote portlet handle.

  Value:
    A remote portlet primary instance ID.
  -->
  <cache-name>complexProducerPortletHandleToIdCache:*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-size-limit</param-name>
      <param-value>1000</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    complexProducerPortletIdToDefinitionLabel

  Use:
    Used to cache the definition label of portlets.

  Key:
    A remote portlet primary instance ID.

  Value:
    A remote portlet definition label.
  -->
  <cache-name>complexProducerPortletIdToDefinitionLabel:*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-size-limit</param-name>
      <param-value>1000</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    credentialEntryCache

  Use:
    Used to cache credential vault entries with encrypted credential.

  Key:
    A
com.bea.p13n.security.management.credentials.internal.CredentialEntryLocator.

   Value:
    A com.bea.p13n.security.management.credentials.CredentialEntry.
   -->
   <cache-name>credentialEntryCache:*</cache-name>
   <scheme-name>portal-local</scheme-name>
  </cache-mapping>

  <cache-mapping>
   <!--
   Cache:
    documentContentCache

   Use:
    Used to cache the document bytes for the DocumentManager. It is not
    used by the content repositories.

   Notes:
    Deprecated cache.
   -->
   <cache-name>documentContentCache:*</cache-name>
   <scheme-name>portal-local</scheme-name>
   <init-params>
    <init-param>
     <param-name>local-expiry</param-name>
     <param-value>1m</param-value>
    </init-param>
   </init-params>
  </cache-mapping>

  <cache-mapping>
   <!--
   Cache:
    documentIdCache

   Use:
    Used to caches the results of document searches (ids only) for the
    DocumentManager. It is not used by the content repositories.

   Notes:
    Deprecated cache.
   -->
   <cache-name>documentIdCache:*</cache-name>
   <scheme-name>portal-local</scheme-name>
   <init-params>
    <init-param>
     <param-name>local-expiry</param-name>
     <param-value>1m</param-value>
    </init-param>
   </init-params>
  </cache-mapping>

  <cache-mapping>
   <!--
   Cache:
    documentContentCache

   Use:
    Used to cache the results of document searches for the DocumentManager.
    It is not used by the content repositories.

  Notes:
    Deprecated cache.
  -->
  <cache-name>documentMetadataCache:*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>1m</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    discountAssocCache

  Use:
    Used to cache computed discount associations (applicable to individual
    customers or to customer segments).

  Key:
    A com.beasys.commerce.ebusiness.customer.CustomerPk.

  Value:
    A com.bea.commerce.ebusiness.discount.association.DiscountAssociation.
  -->
  <cache-name>discountAssocCache:*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>1h</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    discountCache

  Use:
    Used to cache computed discount definitions (applicable to individual
    customers or to customer segments).

  Key:
    A com.bea.commerce.ebusiness.discount.mgmt.QualificationDiscountId.

  Value:
    A java.util.Set of
    com.bea.commerce.ebusiness.discount.mgmt.QualificationDiscountDef
    objects.
  -->
  <cache-name>discountCache:*</cache-name>
  <scheme-name>portal-partitioned</scheme-name>
  <init-params>
    <init-param>
      <param-name>back-size-limit</param-name>
      <param-value>10</param-value>
    </init-param>
    <init-param>
      <param-name>back-expiry</param-name>
      <param-value>5m</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    entityIdCache

  Use:
    Used to cache the numeric ids for entity property locators.

  Key:
    A com.bea.p13n.property.PropertyLocator.

  Value:
    An entity ID (java.lang.Long).
  -->
  <cache-name>entityIdCache:*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-size-limit</param-name>
      <param-value>1024</param-value>
    </init-param>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>1d</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    entityPropertyCache

  Use:
    Used to caches property values for users and groups.

  Key:
    A com.bea.p13n.property.PropertyLocator.

  Value:
    A com.bea.p13n.property.EntityPropertyCache.
  -->
  <cache-name>entityPropertyCache:*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-size-limit</param-name>
      <param-value>500</param-value>
    </init-param>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>10m</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    globalDiscountAssocCache

  Use:
    Used to cache computed global discount associations. This is the set of
    discount associations that is applicable to all users.

  Key:
    A com.beasys.commerce.ebusiness.customer.CustomerPk.

  Value:
    A com.bea.commerce.ebusiness.discount.association.DiscountAssociation.
  -->
  <cache-name>globalDiscountAssocCache:*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>1h</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    globalDiscountCache

  Use:
    Used to cache computed global discount definitions. This is the set of
    global discounts that is applicable to all users.

  Key:
    A global discount set name (java.lang.String).

  Value:
    A java.util.Set of
    com.bea.commerce.ebusiness.discount.mgmt.QualificationDiscountDef
    objects.
  -->
  <cache-name>globalDiscountCache:*</cache-name>
  <scheme-name>portal-partitioned</scheme-name>
  <init-params>
    <init-param>
      <param-name>back-size-limit</param-name>
      <param-value>10</param-value>
    </init-param>
    <init-param>
      <param-name>back-expiry</param-name>
      <param-value>5m</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    jndiNameCache

  Use:
    Used to cache the JNDI names of entity property managers and unified
    user profile managers.

  Key:
    An entity ID (java.lang.Long).

  Value:
    A JNDI name (java.lang.String).
  -->
  <cache-name>jndiNameCache:*</cache-name>
  <scheme-name>portal-replicated</scheme-name>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    netuix.community.definition.cache

  Use:
    Used to cache community definitions.

  Key:
    A composite key of webapp name, portal path, and desktop path for a
    community.

  Value:
    A CommunityDefinition object.
  -->
  <cache-name>netuix.community.definition.cache:*</cache-name>
  <scheme-name>portal-replicated</scheme-name>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    netuix.community.id.to.path.cache

  Use:
    Used to cache community webapp names, desktop paths, and portal
    paths.

  Key:
    A CommunityDefinitionId.

  Value:
    The community webapp name, portal path, and desktop path.
  -->
  <cache-name>netuix.community.id.to.path.cache:*</cache-name>
  <scheme-name>portal-replicated</scheme-name>
</cache-mapping>
```

```
<cache-mapping>
  <!--
  Cache:
    netuix.notification.global

  Use:
    Used to cache notifications targeted to a user, but not targeted to
    an individual web application.

  Key:
    A user name.

  Value:
    A java.util.List of Notification objects.
  -->
  <cache-name>netuix.notification.global:*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-size-limit</param-name>
      <param-value>1024</param-value>
    </init-param>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>1d</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    nodeCache.[repository name]

  Use:
    Used to cache BEA Repository nodes.

  Key:
    A node ID (java.lang.String).

  Value:
    A com.bea.content.Node.
  -->
  <cache-name>nodeCache.*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-size-limit</param-name>
      <param-value>50</param-value>
    </init-param>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>1m</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    nodePathCache.[repository name]

  Use:
    Used to cache a list of nodes for the BEA Repository based on a path.

  Key:
    A path (java.lang.String).

  Value:
    A com.bea.content.Node.
  -->
  <cache-name>nodePathCache.*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-size-limit</param-name>
      <param-value>50</param-value>
    </init-param>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>1m</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    p13nSecurityResourceCache

  Use:
    Used to cache the logical hierarchical elements of WLP security
    policies to speed performance of navigating the hierarchy.

  Key:
    A com.bea.p13n.entitlements.resource.P13nCachedResource.

  Value:
    A com.bea.p13n.entitlements.resource.P13nCachedResource.
  -->
  <cache-name>p13nSecurityResourceCache:*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-size-limit</param-name>
      <param-value>1024</param-value>
    </init-param>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>1d</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    portalApp_Quiescence_Decision_Cache
  Use:
    When Maintenance Mode is enabled, used to cache the access decisions
    made per user per feature area of the Portal Administration Tool.

  Key:
    A com.bea.p13n.management.quiescence.QuiescenceDecisionCacheKey.

  Value:
    A com.bea.p13n.management.quiescence.QuiescenceDecision.
  -->
  <cache-name>portalApp_Quiescence_Decision_Cache:*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-size-limit</param-name>
      <param-value>1024</param-value>
    </init-param>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>1d</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    portalContentUriCache

  Use:
    Used to cache portal content URIs for a combination of webapp, portal,
    locale and optional user name.

  Key:
    A compound key consisting of the webapp, portal, locale and optional
    user name.

  Value:
    A portal content URI.
  -->
  <cache-name>portalContentUriCache:*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-size-limit</param-name>
      <param-value>500</param-value>
    </init-param>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>0</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    portalControlTreeCache

  Use:
    Used to cache portal control trees for a combination of webapp,
    portal, desktop, locale and optional user name.

  Key:
    A compound key consisting of the webapp, portal, locale and optional
    user name.

  Value:
    A portal control tree.
  -->
  <cache-name>portalControlTreeCache:*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-size-limit</param-name>
      <param-value>500</param-value>
    </init-param>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>0</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    PortalFrameworkServiceLevelManager/[webapp]

  Use:
    Used to cache the state of a portlet instance, suspended or active.

  Key:
    A java.lang.String (identifies the portlet instance).

  Value:
    A com.bea.netuix.servicelevel.PortletServiceLevelDescription.
  -->
  <cache-name>PortalFrameworkServiceLevelManager/*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-size-limit</param-name>
      <param-value>1024</param-value>
    </init-param>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>1d</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    portalLayoutDefinitionCache

  Use:
    Used to cache LayoutDefinition objects.
    Key:
      A LayoutDefinitionId.

    Value:
      A LayoutDefinition.
    -->
    <cache-name>portalLayoutDefinitionCache:*</cache-name>
    <scheme-name>portal-local</scheme-name>
    <init-params>
      <init-param>
        <param-name>local-size-limit</param-name>
        <param-value>1024</param-value>
      </init-param>
      <init-param>
        <param-name>local-expiry</param-name>
        <param-value>1d</param-value>
      </init-param>
    </init-params>
  </cache-mapping>

  <cache-mapping>
    <!--
    Cache:
      portalLocalizationLocaleCache

    Use:
      Used to cache collections of language, character encoding, country
      and variant.

    Key:
      A java.lang.String.

    Value:
      A java.lang.Set of LocalizationLocale objects.
    -->
    <cache-name>portalLocalizationLocaleCache:*</cache-name>
    <scheme-name>portal-local</scheme-name>
    <init-params>
      <init-param>
        <param-name>local-size-limit</param-name>
        <param-value>500</param-value>
      </init-param>
      <init-param>
        <param-name>local-expiry</param-name>
        <param-value>0</param-value>
      </init-param>
    </init-params>
  </cache-mapping>

  <cache-mapping>
    <!--
    Cache:
      portalLocalizationResourceCache

    Use:
      Used to cache localization resources.

    Key:
      A
com.bea.netuix.application.localization.identifier.LocalizationIntersectionId.
```

```
    Value:
      A com.bea.netuix.application.localization.definition.LocalizationResource.
    -->
    <cache-name>portalLocalizationResourceCache:*</cache-name>
    <scheme-name>portal-partitioned</scheme-name>
    <init-params>
      <init-param>
        <param-name>back-size-limit</param-name>
        <param-value>500</param-value>
      </init-param>
      <init-param>
        <param-name>back-expiry</param-name>
        <param-value>0</param-value>
      </init-param>
    </init-params>
  </cache-mapping>

  <cache-mapping>
    <!--
    Cache:
      portalMarkupDefinitionCache

    Use:
      Used to cache markup definition information.

    Key:
      A com.bea.netuix.application.identifier.MarkupDefinitionId.

    Value:
      A com.bea.netuix.application.definition.MarkupDefinition.
    -->
    <cache-name>portalMarkupDefinitionCache:*</cache-name>
    <scheme-name>portal-partitioned</scheme-name>
    <init-params>
      <init-param>
        <param-name>back-size-limit</param-name>
        <param-value>500</param-value>
      </init-param>
      <init-param>
        <param-name>back-expiry</param-name>
        <param-value>1m</param-value>
      </init-param>
    </init-params>
  </cache-mapping>

  <cache-mapping>
    <!--
    Cache:
      portalThemeDefinitionCache

    Use:
      Used to cache ThemeDefinition objects.

    Key:
      A ThemeDefinitionId.

    Value:
      A ThemeDefinition.
    -->
    <cache-name>portalThemeDefinitionCache:*</cache-name>
    <scheme-name>portal-replicated</scheme-name>
  </cache-mapping>

  <cache-mapping>
    <!--
    Cache:
      PortletCategoryCache

    Use:
      Used to cache PortletCategoryDefinition objects.

    Key:
      A PortletCategoryDefinitionId.

    Value:
      A PortletCategoryDefinition.
    -->
    <cache-name>PortletCategoryCache:*</cache-name>
    <scheme-name>portal-replicated</scheme-name>
  </cache-mapping>

  <cache-mapping>
    <!--
    Cache:
      portletControlTreeCache

    Use:
      Used to cache portlet control trees for floating portlets.

    Key:
      A composite key of portlet instance ID and locale.

    Value:
      A portlet control tree.
    -->
    <cache-name>portletControlTreeCache:*</cache-name>
    <scheme-name>portal-local</scheme-name>
    <init-params>
      <init-param>
        <param-name>local-size-limit</param-name>
        <param-value>500</param-value>
      </init-param>
      <init-param>
        <param-name>local-expiry</param-name>
        <param-value>0</param-value>
      </init-param>
    </init-params>
  </cache-mapping>

  <cache-mapping>
    <!--
    Cache:
      portletPreferencesCache

    Use:
      Used to cache portlet preferences.

    Key:
      A portlet preference identifier.
    Value:
      A com.bea.portlet.prefs.PortletPreferences.
    -->
    <cache-name>portletPreferencesCache:*</cache-name>
    <scheme-name>portal-local</scheme-name>
    <init-params>
      <init-param>
        <param-name>local-size-limit</param-name>
        <param-value>500</param-value>
      </init-param>
      <init-param>
        <param-name>local-expiry</param-name>
        <param-value>1m</param-value>
      </init-param>
    </init-params>
  </cache-mapping>

  <cache-mapping>
    <!--
    Cache:
      ProductItemCache

    Use:
      Used to cache the total number of product items in the catalog as well
      as the product items themselves.

    Key:
      The key for the total number of product items is a java.lang.String.
      The key for the product items is a
      com.beasys.commerce.ebusiness.catalog.ProductItemKey.

    Value:
      The value for the total number of product items is a java.lang.Integer.
      The value for the product item is a
      com.beasys.commerce.ebusiness.catalog.ProductItem.
    -->
    <cache-name>ProductItemCache:*</cache-name>
    <scheme-name>portal-near</scheme-name>
    <init-params>
      <init-param>
        <param-name>back-expiry</param-name>
        <param-value>6h</param-value>
      </init-param>
    </init-params>
  </cache-mapping>

  <cache-mapping>
    <!--
    Cache:
      profileTypeCache

    Use:
      Used to cache user profile types that are used to look up the
      appropriate user manager profile manager when retrieving a user
      profile.

    Key:
      A user name (java.lang.String).
    Value:
      A profile type (java.lang.String).
    -->
    <cache-name>profileTypeCache:*</cache-name>
    <scheme-name>portal-replicated</scheme-name>
  </cache-mapping>

  <cache-mapping>
    <!--
    Cache:
      propertyKeyIdCache

    Use:
      Used to cache the unique ID associated with a property set type,
      property set and property name combination.

    Key:
      A compound key consisting of the property set type, property set, and
      property name.

    Value:
      A java.lang.Long.
    -->
    <cache-name>propertyKeyIdCache:*</cache-name>
    <scheme-name>portal-replicated</scheme-name>
  </cache-mapping>

  <cache-mapping>
    <!--
    Cache:
      proxyPortletCache

    Use:
      Used to cache ProxyPortlets.

    Key:
      A java.lang.String representing the portlet instance ID.

    Value:
      Information from the consumer registry and about the proxy portlet
      instance.
    -->
    <cache-name>proxyPortletCache:*</cache-name>
    <scheme-name>portal-local</scheme-name>
    <init-params>
      <init-param>
        <param-name>local-size-limit</param-name>
        <param-value>100</param-value>
      </init-param>
      <init-param>
        <param-name>local-expiry</param-name>
        <param-value>0</param-value>
      </init-param>
    </init-params>
  </cache-mapping>

  <cache-mapping>
    <!--
    Cache:
      registrationHandleCache
    Use:
      Used to cache whether or not a particular WSRP registration handle is
      valid.

    Key:
      A WSRP consumer registration handle.

    Value:
      A java.lang.Boolean.
    -->
    <cache-name>registrationHandleCache:*</cache-name>
    <scheme-name>portal-replicated</scheme-name>
  </cache-mapping>

  <cache-mapping>
    <!--
    Cache:
      remoteProducerInfoCache

    Use:
      Used to cache the metadata for WSRP producers added to a WSRP consumer
      application.

    Key:
      The name of the consumer web application (java.lang.String).

    Value:
      A java.util.HashMap containing WSRP producer metadata. This map is
      keyed by the producer handle of each producer.
    -->
    <cache-name>remoteProducerInfoCache:*</cache-name>
    <scheme-name>portal-replicated</scheme-name>
  </cache-mapping>

  <cache-mapping>
    <!--
    Cache:
      repo.explicitPropertyCache

    Use:
      Used to cache explicit property information for all WLP repositories.

    Key:
      The name of the repository (java.lang.String)

    Value:
      A java.util.Collection of repository property definition information
      for explicit properties in that WLP repository.
    -->
    <cache-name>repo.explicitPropertyCache:*</cache-name>
    <scheme-name>portal-replicated</scheme-name>
  </cache-mapping>

  <cache-mapping>
    <!--
    Cache:
      repo.nodeIdCache.[repository name]

    Use:
      Used to cache node information for a specific WLP repository
      instance.

    Key:
      A node ID.

    Value:
      A repository node data object.
    -->
    <cache-name>repo.nodeIdCache.*</cache-name>
    <scheme-name>portal-local</scheme-name>
    <init-params>
      <init-param>
        <param-name>local-size-limit</param-name>
        <param-value>1024</param-value>
      </init-param>
      <init-param>
        <param-name>local-expiry</param-name>
        <param-value>1d</param-value>
      </init-param>
    </init-params>
  </cache-mapping>

  <cache-mapping>
    <!--
    Cache:
      repo.nodePathCache.[repository name]

    Use:
      Used to cache node information for a specific WLP repository
      instance.

    Key:
      A node path.

    Value:
      A repository node data object.
    -->
    <cache-name>repo.nodePathCache.*</cache-name>
    <scheme-name>portal-local</scheme-name>
    <init-params>
      <init-param>
        <param-name>local-size-limit</param-name>
        <param-value>1024</param-value>
      </init-param>
      <init-param>
        <param-name>local-expiry</param-name>
        <param-value>1d</param-value>
      </init-param>
    </init-params>
  </cache-mapping>

  <cache-mapping>
    <!--
    Cache:
      repo.typeBinaryCache.[repository name]

    Use:
      Used to cache node binary property information for a specific WLP
      repository instance.
    Key:
      A composite key of node UID and binary property UID.

    Value:
      A byte array representing property information.
    -->
    <cache-name>repo.typeBinaryCache.*</cache-name>
    <scheme-name>portal-local</scheme-name>
    <init-params>
      <init-param>
        <param-name>local-size-limit</param-name>
        <param-value>1024</param-value>
      </init-param>
      <init-param>
        <param-name>local-expiry</param-name>
        <param-value>1d</param-value>
      </init-param>
    </init-params>
  </cache-mapping>

  <cache-mapping>
    <!--
    Cache:
      repo.typeIdCache.[repository name]

    Use:
      Used to cache node type information for a specific WLP repository
      instance.

    Key:
      A type ID.

    Value:
      A repository type data object.
    -->
    <cache-name>repo.typeIdCache.*</cache-name>
    <scheme-name>portal-replicated</scheme-name>
  </cache-mapping>

  <cache-mapping>
    <!--
    Cache:
      repo.typeNameCache.[repository name]

    Use:
      Used to cache node type information for a specific WLP repository
      instance.

    Key:
      A type name.

    Value:
      A repository type data object.
    -->
    <cache-name>repo.typeNameCache.*</cache-name>
    <scheme-name>portal-replicated</scheme-name>
  </cache-mapping>

  <cache-mapping>
```

```
  <!--
  Cache:
    repositoryConfigCache

  Use:
    Used to cache repository configuration information.

  Key:
    A repository name.

  Value:
    A RepositoryConfig.
  -->
  <cache-name>repositoryConfigCache:*</cache-name>
  <scheme-name>portal-replicated</scheme-name>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    searchCache

  Use:
    Used to cache an array of IDs for nodes that satisfy a content search.

  Key:
    A com.beasys.commerce.foundation.expression.Search.

  Value:
    An array of node IDs that satisfy the query.
  -->
  <cache-name>searchCache:*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-size-limit</param-name>
      <param-value>20</param-value>
    </init-param>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>1m</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    typeCache.[repository name]

  Use:
    Used to cache type information.

  Key:
    An ObjectClass ID.

  Value:
    An ObjectClass.
  -->
  <cache-name>typeCache.*</cache-name>
  <scheme-name>portal-replicated</scheme-name>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    typeNameCache.[repository name]

  Use:
    Used to cache type ID information.

  Key:
    An ObjectClass name.

  Value:
    An ObjectClass ID.
  -->
  <cache-name>typeNameCache.*</cache-name>
  <scheme-name>portal-replicated</scheme-name>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    versionCache

  Use:
    Used to cache individual versions of a Node in the BEA Content
    Repository.

  Key:
    A com.bea.content.ID.

  Value:
    A com.bea.content.virtual.version.Version.
  -->
  <cache-name>versionCache:*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-size-limit</param-name>
      <param-value>1024</param-value>
    </init-param>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>1d</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    virtualNodeCache

  Use:
    Used to cache a node from a repository that has versioning support
    enabled.

  Key:
    A com.bea.content.ID.

  Value:
    A com.bea.content.virtual.VirtualNode.
  -->
  <cache-name>virtualNodeCache:*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-size-limit</param-name>
      <param-value>1024</param-value>
    </init-param>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>1d</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    wlp.urlCompression.compressed

  Use:
    Used to map compressed URL IDs to the expanded URL.

  Key:
    A numeric compressed URL ID.

  Value:
    An expanded URL.
  -->
  <cache-name>wlp.urlCompression.compressed:*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-size-limit</param-name>
      <param-value>1024</param-value>
    </init-param>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>1d</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Cache:
    wlp.urlCompression.expanded

  Use:
    Used to map expanded URLs to compressed URL IDs.

  Key:
    An expanded URL.

  Value:
    A compressed URL ID.
  -->
  <cache-name>wlp.urlCompression.expanded:*</cache-name>
  <scheme-name>portal-local</scheme-name>
  <init-params>
    <init-param>
      <param-name>local-size-limit</param-name>
      <param-value>1024</param-value>
    </init-param>
    <init-param>
      <param-name>local-expiry</param-name>
      <param-value>1d</param-value>
    </init-param>
  </init-params>
</cache-mapping>

<cache-mapping>
  <!--
  Default cache mapping.
  -->
  <cache-name>*</cache-name>
  <scheme-name>portal-local</scheme-name>
</cache-mapping>

<!--
The clustered cache used to store Session management data.
-->
<cache-mapping>
  <cache-name>session-management</cache-name>
  <scheme-name>replicated</scheme-name>
</cache-mapping>

<!--
The clustered cache used to store ServletContext attributes.
-->
<cache-mapping>
  <cache-name>servletcontext-storage</cache-name>
  <scheme-name>replicated</scheme-name>
</cache-mapping>

<!--
The clustered cache used to store Session attributes.
-->
<cache-mapping>
  <cache-name>session-storage</cache-name>
  <scheme-name>session-near</scheme-name>
</cache-mapping>

<!--
The clustered cache used to store the "overflowing" (split-out due to size)
Session attributes. Only used for the "Split" model.
-->
<cache-mapping>
  <cache-name>session-overflow</cache-name>
  <scheme-name>session-distributed</scheme-name>
</cache-mapping>

<!--
The clustered cache used to store IDs of "recently departed" Sessions.
-->
<cache-mapping>
```

```
    <cache-name>session-death-certificates</cache-name>
    <scheme-name>session-certificate</scheme-name>
  </cache-mapping>

  <!--
  The local cache used to store Sessions that are not yet distributed (if
  there is a distribution controller).
  -->
  <cache-mapping>
    <cache-name>local-session-storage</cache-name>
    <scheme-name>unlimited-local</scheme-name>
  </cache-mapping>

  <!--
  The local cache used to store Session attributes that are not distributed
  (if there is a distribution controller or attributes are allowed to become
  local when serialization fails).
  -->
  <cache-mapping>
    <cache-name>local-attribute-storage</cache-name>
    <scheme-name>unlimited-local</scheme-name>
  </cache-mapping>

</caching-scheme-mapping>


<caching-schemes>

  <local-scheme>
    <!--
    Cache scheme definition used by all local caches that require size
    limitation and/or expiry eviction policies.
    -->
    <scheme-name>portal-local</scheme-name>
    <service-name>PortalLocalCache</service-name>

    <eviction-policy>HYBRID</eviction-policy>
    <high-units>{local-size-limit 100}</high-units>
    <expiry-delay>{local-expiry 1h}</expiry-delay>
  </local-scheme>

  <local-scheme>
    <!--
    Cache scheme definition used by all near cache front maps that require
    size limitation and/or expiry eviction policies.
    -->
    <scheme-name>portal-front</scheme-name>
    <service-name>PortalLocalCache</service-name>

    <eviction-policy>HYBRID</eviction-policy>
    <high-units>{front-size-limit 100}</high-units>
    <expiry-delay>{front-expiry 0}</expiry-delay>
  </local-scheme>

  <local-scheme>
    <!--
    Cache scheme definition used by all partitioned cache backing maps that
    require size limitation and/or expiry eviction policies.
    -->
    <scheme-name>portal-back</scheme-name>
```

```
      <service-name>PortalLocalCache</service-name>

      <eviction-policy>HYBRID</eviction-policy>
      <high-units>{back-size-limit 1000}</high-units>
      <expiry-delay>{back-expiry 1h}</expiry-delay>
    </local-scheme>

    <replicated-scheme>
      <!--
      Replicated caching scheme.
      -->
      <scheme-name>portal-replicated</scheme-name>
      <service-name>PortalReplicatedCache</service-name>

      <serializer>
        <instance>
          <class-name>com.tangosol.io.DefaultSerializer</class-name>
        </instance>
      </serializer>

      <backing-map-scheme>
        <local-scheme>
          <scheme-ref>portal-back</scheme-ref>
        </local-scheme>
      </backing-map-scheme>

      <autostart>true</autostart>
    </replicated-scheme>

    <distributed-scheme>
      <!--
      Partitioned caching scheme.
      -->
      <scheme-name>portal-partitioned</scheme-name>
      <service-name>PortalDistributedCache</service-name>

      <serializer>
        <instance>
          <class-name>com.tangosol.io.DefaultSerializer</class-name>
        </instance>
      </serializer>

      <local-storage
system-property="tangosol.coherence.weblogic.localstorage">false</local-storage>

      <backing-map-scheme>
        <local-scheme>
          <scheme-ref>portal-back</scheme-ref>
        </local-scheme>
      </backing-map-scheme>

      <autostart>true</autostart>
    </distributed-scheme>

    <near-scheme>
      <!--
      Near caching scheme.
      -->
      <scheme-name>portal-near</scheme-name>
```

```
  <front-scheme>
    <local-scheme>
      <scheme-ref>portal-front</scheme-ref>
    </local-scheme>
  </front-scheme>

  <back-scheme>
    <distributed-scheme>
      <scheme-ref>portal-partitioned</scheme-ref>
    </distributed-scheme>
  </back-scheme>

  <invalidation-strategy>{near-strategy present}</invalidation-strategy>
</near-scheme>

<invocation-scheme>
  <!--
  Invocation service scheme.
  -->
  <scheme-name>portal-invocation-service</scheme-name>
  <service-name>PortalInvocationService</service-name>

  <serializer>
    <instance>
      <class-name>com.tangosol.io.DefaultSerializer</class-name>
    </instance>
  </serializer>
</invocation-scheme>


<!--
Replicated caching scheme used by the Session management and ServletContext
attribute caches.
-->
<replicated-scheme>
  <scheme-name>replicated</scheme-name>
  <service-name>ReplicatedSessionsMisc</service-name>
  <request-timeout>30s</request-timeout>
  <backing-map-scheme>
    <local-scheme>
      <scheme-ref>unlimited-local</scheme-ref>
    </local-scheme>
  </backing-map-scheme>
  <autostart>true</autostart>
</replicated-scheme>

<!--
Near caching scheme used by the Session attribute cache. The front cache
uses a Local caching scheme and the back cache uses a Distributed caching
scheme.
-->
<near-scheme>
  <scheme-name>session-near</scheme-name>
  <front-scheme>
    <local-scheme>
      <scheme-ref>session-front</scheme-ref>
    </local-scheme>
  </front-scheme>
  <back-scheme>
    <distributed-scheme>
```

```
          <scheme-ref>session-distributed</scheme-ref>
        </distributed-scheme>
      </back-scheme>
      <invalidation-strategy>present</invalidation-strategy>
    </near-scheme>

    <local-scheme>
      <scheme-name>session-front</scheme-name>
      <eviction-policy>HYBRID</eviction-policy>
      <high-units>1000</high-units>
      <low-units>750</low-units>
    </local-scheme>

    <distributed-scheme>
      <scheme-name>session-distributed</scheme-name>
      <scheme-ref>session-base</scheme-ref>
      <backing-map-scheme>
        <local-scheme>
          <scheme-ref>unlimited-local</scheme-ref>
        </local-scheme>
        <!-- for disk overflow use this backing scheme instead:
        <overflow-scheme>
          <scheme-ref>session-paging</scheme-ref>
        </overflow-scheme>
        -->
      </backing-map-scheme>
    </distributed-scheme>

    <!--
    Distributed caching scheme used by the "recently departed" Session cache.
    -->
    <distributed-scheme>
      <scheme-name>session-certificate</scheme-name>
      <scheme-ref>session-base</scheme-ref>
      <backing-map-scheme>
        <local-scheme>
          <eviction-policy>HYBRID</eviction-policy>
          <high-units>4000</high-units>
          <low-units>3000</low-units>
          <expiry-delay>86400</expiry-delay>
        </local-scheme>
      </backing-map-scheme>
    </distributed-scheme>

    <!--
    "Base" Distributed caching scheme that defines common configuration.
    -->
    <distributed-scheme>
      <scheme-name>session-base</scheme-name>
      <service-name>DistributedSessions</service-name>
      <thread-count>0</thread-count>
      <lease-granularity>member</lease-granularity>
      <local-storage
system-property="tangosol.coherence.session.localstorage">false</local-storage>
      <partition-count>257</partition-count>
      <backup-count>1</backup-count>
      <backup-storage>
        <type>on-heap</type>
      </backup-storage>
      <request-timeout>30s</request-timeout>
      <backing-map-scheme>
        <local-scheme>
          <scheme-ref>unlimited-local</scheme-ref>
        </local-scheme>
      </backing-map-scheme>
      <autostart>true</autostart>
    </distributed-scheme>

    <!--
    Disk-based Session attribute overflow caching scheme.
    -->
    <overflow-scheme>
      <scheme-name>session-paging</scheme-name>
      <front-scheme>
        <local-scheme>
          <scheme-ref>session-front</scheme-ref>
        </local-scheme>
      </front-scheme>
      <back-scheme>
        <external-scheme>
          <bdb-store-manager/>
        </external-scheme>
      </back-scheme>
    </overflow-scheme>

    <!--
    Local caching scheme definition used by all caches that do not require an
    eviction policy.
    -->
    <local-scheme>
      <scheme-name>unlimited-local</scheme-name>
      <service-name>LocalSessionCache</service-name>
    </local-scheme>

    <!--
    Clustered invocation service that manages sticky session ownership.
    -->
    <invocation-scheme>
      <service-name>SessionOwnership</service-name>
      <request-timeout>30s</request-timeout>
    </invocation-scheme>

  </caching-schemes>
</cache-config>
```