Oracle® Coherence
Release Notes for Oracle Coherence
Release 3.7
E21505-03

June 2011
Preface .................................................................................................................................................................................. vii

Audience .................................................................................................................................................................................. vii
Documentation Accessibility .................................................................................................................................................. vii
Related Documents .............................................................................................................................................................. viii
Conventions ............................................................................................................................................................................ viii

1 Technical Changes and Enhancements

Coherence 3.7 Documentation Library ................................................................................................................................. 1-1
Download and Install the Latest Software Patch .................................................................................................................. 1-1
New and Improved Coherence Data Grid Functionality ......................................................................................................... 1-2
Oracle Coherence for Java 3.7 .................................................................................................................................................. 1-3
  Management and Monitoring Enhancements and Fixes .................................................................................................. 1-3
  Partitioned Cache Enhancements and Fixes ....................................................................................................................... 1-3
  TCMP Enhancements and Fixes ........................................................................................................................................ 1-4
  Coherence*Extend Framework Enhancements and Fixes ................................................................................................. 1-4
  Coherence*Web Enhancements and Fixes ......................................................................................................................... 1-5
  Security Framework Enhancements and Fixes ................................................................................................................. 1-5
  Serialization Framework Enhancements and Fixes ........................................................................................................... 1-6
  Transaction Framework Enhancements and Fixes ........................................................................................................... 1-6
  Other Enhancements and Fixes .......................................................................................................................................... 1-6
Oracle Coherence for .NET 3.7 .................................................................................................................................................. 1-7
Oracle Coherence for C++ 3.7 .................................................................................................................................................. 1-8
Changes and Enhancements for 3.7.0.1 ........................................................................................................................................ 1-8
Changes and Enhancements for 3.7.0.2 ........................................................................................................................................ 1-9
Known Problems and Workarounds ......................................................................................................................................... 1-10
  Communications Issues with Cluster Members Under IPv6 .......................................................................................... 1-10
  Memory Usage During Data Grid Operations .................................................................................................................. 1-10
Deprecated Features ............................................................................................................................................................... 1-10
  Support for Oracle Products Running on the Intel Itanium Processor .................................................................................. 1-11

2 Documentation Errata

JVM Heap Size Recommendations ........................................................................................................................................ 2-1
List of Tables

1–1  Deprecated Methods in the CacheFactory API ........................................................................ 1-11
This document describes changes and enhancements that have been made to Oracle Coherence for the 3.7 release.

**Audience**

This document is intended for users of Oracle Coherence.

**Documentation Accessibility**

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at [http://www.oracle.com/accessibility/](http://www.oracle.com/accessibility/).

**Accessibility of Code Examples in Documentation**

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

**Accessibility of Links to External Web Sites in Documentation**

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

**Access to Oracle Support**

Oracle customers have access to electronic support through My Oracle Support. For information, visit [http://www.oracle.com/support/contact.html](http://www.oracle.com/support/contact.html) or visit [http://www.oracle.com/accessibility/support.html](http://www.oracle.com/accessibility/support.html) if you are hearing impaired.
Related Documents

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

- Getting Started with Oracle Coherence
- Developer’s Guide for Oracle Coherence
- Client Guide for Oracle Coherence
- Tutorial for Oracle Coherence
- User’s Guide for Oracle Coherence*Web
- Integration Guide for Oracle Coherence
- Management Guide
- Administrator’s Guide
- Security Guide

Conventions

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>boldface</td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td>italic</td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td>monospace</td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
This chapter describes the changes and enhancements made to the Oracle Coherence product for the 3.7 release. This document is accurate at the time of publication. Oracle updates the release notes periodically after the software release.

- Coherence 3.7 Documentation Library
- Download and Install the Latest Software Patch
- New and Improved Coherence Data Grid Functionality
- Oracle Coherence for Java 3.7
- Oracle Coherence for .NET 3.7
- Oracle Coherence for C++ 3.7
- Changes and Enhancements for 3.7.0.1
- Changes and Enhancements for 3.7.0.2
- Known Problems and Workarounds
- Deprecated Features

**Coherence 3.7 Documentation Library**

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

http://download.oracle.com/docs/cd/E18686_01/index.htm

For the 3.7 release, these new guides have been added to the library:

- *Management Guide* describes key management concepts and provides instructions for using Coherence management features.
- *Security Guide* describes key security concepts and provides instructions for implementing various levels of security for both Coherence clusters and Coherence*Extend clients.
- *Administrator’s Guide* provides key administration concepts and detailed instructions for administering Coherence clusters and caches.

**Download and Install the Latest Software Patch**

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

https://support.oracle.com
See the README file in the patch distribution for up-to-date information on the software fixes provided by the patch.

1. Login to My Oracle Support.

2. Click the Patches & Updates tab.

3. Under the Patch Search tab, select Product or Family (Advanced Search), and select the Include all patches in a product family check box.

4. Enter Oracle Coherence as the product, select the platform and release, and click Search.

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

New and Improved Coherence Data Grid Functionality

Oracle Coherence 3.7 is the latest release of the industry’s leading distributed in-memory data grid product.

- Oracle Coherence 3.7 simplifies the configuration and management of large-scale deployments and provides customers with dramatically more data storage capabilities and intelligent and dynamic load balancing of client connections.

- The new Elastic Data feature of Oracle Coherence 3.7 introduces breakthrough technology that enables organizations to take advantage of both memory and disk-based devices for data grid storage, enabling near memory speed access to data, regardless of storage medium, and greater flexibility in capacity and resource planning.
  - Customers that demand extremely high data capacity combined with low-latency data access requirements, such as financial services firms and online retailers, can use Elastic Data to seamlessly manage data across memory and disk-based devices.
  - Additional benefits of Elastic Data can include both the potential for significantly increased capacity, and smaller or more-easily managed deployments.

- Oracle Coherence 3.7 introduces dynamic load balancing and automatic proxy discovery for Coherence*Extend clients (for example, application servers or desktop applications accessing the Oracle Coherence data grid), allowing customers to more intelligently distribute and manage client connections, while also simplifying the configuration of large-scale environments.

- Through integration with F5 BIG-IP Load Traffic Manager (LTM) and SSL Acceleration technologies, Oracle Coherence 3.7 allows users to take advantage of components of their existing infrastructure to secure, optimize and load balance application traffic generated from Coherence*Extend clients across any number of servers in a cluster, and to hardware-accelerate the related CPU-intensive SSL encryption.

- Coherence configuration files are now described using XML schema definitions (XSD) and are no longer described using Document Type Definition (DTD).

- Oracle Coherence 3.7 introduces native integration with GlassFish Server by using the Coherence*Web SPI for GlassFish (ActiveCache for GlassFish), providing “no code change” installation and configuration of Coherence*Web, making it dramatically easier for GlassFish Server users to scale their applications.
Oracle Coherence 3.7 is a component of Oracle Fusion Middleware 11g and is tightly integrated with the other products in Oracle's middleware stack. Oracle Coherence is also part of Oracle Exalogic Elastic Cloud.

Oracle Coherence for Java 3.7

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

Management and Monitoring Enhancements and Fixes

- Fixed an issue where the `evictions` field of the cache usage report was always blank.
- Fixed a regression in the MBean registry that caused a memory leak when caches were destroyed.

Partitioned Cache Enhancements and Fixes

- The ability to perform "lightweight" transactions within single partition has been added to this release.
- Fixed a race condition that could cause index corruption.
- Fixed an issue that could cause an `EntryProcessor` to be re-executed under certain circumstances in the event of primary server failure.
- Fixed an issue that can cause a filter-based `invoke()` call to hang when the filter throws an exception.
- Fixed an issue causing endless repeating `*Request due to the re-distribution of PartitionSet(...)` during concurrent calls to `destroyCache()`.
- Fixed an issue causing indices built by `IndexAwareExtractor` to not be transferred correctly during failover.
- Fixed an issue that could cause spurious "null" aggregation results during a cache server shutdown.
- Fixed an issue that could result in storage-disabled clients being unable to communicate with storage nodes.
- Fixed issue causing the maintenance of sorted indices to be extremely slow.
- Fixed an issue on sorted indices that could cause a `NullPointerException` during filter-based queries.
- Fixed an issue where a `ConcurrentModificationException` was thrown when a set or map was emptied concurrently.
- Fixed an issue causing `LimitFilter` to return an inconsistent set of results.
- Fixed an issue that could result in indexes (and `keySet` iterators) becoming corrupt or missing entries under heavy eviction.
- Several query-related statistics have been added to `StorageManagerMBean`. These include descriptions of queries that are most expensive in terms of time and processing power.
- Fixed an issue that could lead to `PartitionedCache putAll()` throwing a `ClassCastException` if executed during partition transfer.
TCMP Enhancements and Fixes

- Fixed MulticastTest and DatagramTest to parse for IPv6 literal addresses such as -local [fe80::216:3eff:fe35:eba3%eth0]:8888.
- Allow for sub-second IPMonitor timeout.
- Fixed an issue that caused the cluster service to encounter a NullPointerException when removing a slow member.
- Fixed a deadlock condition when exception occurs in Cluster.start.
- Fixed an issue where the departure of the senior member while a new member is joining could result in join-timeout.
- Fixed repeating DiagnosticPacket entries in log files. The pinged member will reply back to the originator of the DiagnosticPacket.
- Resolved an intermittent issue where a new TCMP/SSL connection could trigger a BufferOverflowException.
- Fixed an issue causing WKA (well known address) clusters to combine or panic due to socket address re-use.
- Increased the ping-timeout used by IPMonitor to accommodate some Windows environments.

Coherence*Extend Framework Enhancements and Fixes

- Fixed an issue that caused an invalid SSL session exception with concurrent .NET SSL connections.
- Renamed types and #defines to avoid namespace collisions with windows.h files.

Note: This change will break backwards compatibility and will require a recompile.

- Fixed a race condition in TcpConnection.open() that could cause an error during a reconnection attempt.
- Improved SafeConfigurablePofContext to support custom POF serialization of collection implementations.
- Fixed an issue with String to std::string conversion that caused non-ASCII strings to be truncated.
- Fixed an issue in ConfigurablePofContext that prevented the same POF configuration file from being included multiple times.
- Optimized POF serialization of variable length integer values.
- Added typed get methods to the PofValue class for convenience.
- Unique names can now be specified for extend clients to identify them in the MBean list of client connections and log messages.
- Removed redundant re-resolution of remote addresses in TcpConnection.open().
- Fixed an issue with the LocalCache implementation that could cause a deadlock during concurrent use.
Oracle Coherence for Java 3.7

Technical Changes and Enhancements

- Fixed an issue with the TypeResolver that prevented assemblies with fully-qualified names from being loaded.
- Fixed deadlock issue in Peer.OnServiceStarted() when the client is configured purely programatically.
- Fixed a server-side timeout issue so that the proxy service will enforce task timeout.
- Fixed an ArgumentOutOfRangeException that was thrown when applying a LikeFilter with a long pattern containing an escape character.
- Fixed an issue where abnormal termination of a cache service could cause Coherence*Extend clients to lose events.
- Hardened the proxy service against exceptions thrown by custom proxy implementations.
- Significantly improved the performance of CompositeCache.GetAll().
- Certified Coherence*Extend with the F5 BIG-IP Load Traffic Manager.
- Fixed an issue where an EmbeddedResource was unable to find the Coherence assembly in the GAC.
- Improved the proxy service to support pluggable load balancing strategies and added a default policy that distributes client connections based upon a variety of metrics.
- Added WKA functionality to Coherence*Extend. This feature allows clients to discover and connect to proxy servers that are not explicitly listed in the client's cache configuration file.
- Improved CacheFactory so that it restarts the logger during a restart.

Coherence*Web Enhancements and Fixes

- Fixed an issue where the split session model did not transition an external attribute to an internal attribute when it was replaced with a "small" value.
- Changed the default value for coherence-enable-suspect-attributes to true.
- Created the Coherence*Web SPI for GlassFish 3.1 (ActiveCache for GlassFish).
- Added a configurable session affinity token to enable session cookie compatibility across different Web containers.
- The local-storage parameter is now disabled by default for Web applications that are instrumented by the Coherence*Web webInstaller.
- Coherence*Web now can be used on WebSphere 7.
- Fixed an issue preventing session information for each session from being displayed in session tab.
- Added WebLogicHttpSessionManagerMBean for the WebLogic Coherence*Web SPI.

Security Framework Enhancements and Fixes

- The transformIdentity method in the IdentityTransformer interface and assertIdentity method in the IdentityAsserter interface have been changed to add a new parameter to specify the invoking service. This change
allows you to use multiple transformers and asserters. Existing custom implementations will have to change the method signatures and recompile.

### Serialization Framework Enhancements and Fixes

- You can now enter identical includes in a POF configuration.
- Fixed an issue where `CompositeKey` had no entry in the `coherence-pof-config.xml` file.
- Added the missing `AbstractEvolvable` class to the API.
- Fixed an issue causing an `IllegalArgumentException` to be thrown from `ExternalizableHelper` when serializing a very large number of objects.
- Removed validation logic in POF date deserialization that limited the year range to between 1582 and 2199.
- Fixed a problem with reading uniform collections into a generic collection.
- The `AbstractInvocable` result is not required to implement the `Serializable` interface.
- The `PofHelper.ResizeArray()` method always creates a new array without first checking the size.

### Transaction Framework Enhancements and Fixes

- Fixed transactional cache memory leak.
- Throw an `UnsupportedOperationException` when a `MapTrigger` is registered against a transactional cache.
- Fixed an issue that could lead to version dependency becoming orphaned if commit/rollback fails after updating entries but before releasing dependency.

### Other Enhancements and Fixes

- An application running in a Coherence cluster can isolate its caches and services to prevent unintentional use by other applications running in the same cluster.
- Added a switch to the `coherence.sh` and `coherence.cmd` files that enables you to specify JMX more easily.
- Introduced XML schema definitions (XSD) and fail-fast validation for Coherence XML configuration files.
- Fixed an issue causing `TcpRing` to throw a `NullPointerException`.
- Created a JMX reporter configuration for Elastic Data.
- Files that have been created by Elastic Data will be deleted from the designated directory if they have not been accessed in more than two hours.
- Upgraded Coherence 3.7 to require the 1.6 JDK update 23.
- Fixed an issue that results in a `NullPointerException` during cluster startup if `port-auto-adjust=false` and the unicast listener port is unavailable.
- Reduced contention on the `BackingMapManagerContext.getBackingMap()` method. Note that the `BackingMapManagerContext.getBackingMap()` method has been deprecated in favor of the `getBackingMapEntry` method. See "Deprecated Features" on page 1-10.
Fixed an issue causing MapEvents not to be delivered to both synchronous and asynchronous map listeners.

Fixed an issue in LimitFilter where setPage did not work properly.

Fixed an issue that could cause a NotFilter query to fail with NullPointerException.

Added a check for Hotspot server JVM on SPARC-Solaris which in prevents node startup if this check fails. (This is to avoid Java bug 7009231 with 32-bit client JVM).

Added a BackingMapContext storage abstraction which can be used to enable data structures like MapIndex to easily access the CacheService. Furthermore, the BackingMapContext is available on BinaryEntry and BackingMapManagerContext.

Fixed an issue that could result in a NullPointerException being thrown after Guardian recovery or termination.

Fixed a NullPointerException issue in Daemon$Guard.recover() after service termination.

Fixed an issue where Nearcache.getAll() returns copies when using invalidation strategy "none".

Fixed an issue that could result in missing events when updating a key to an identical value.

Fixed a deadlock that occurs while destroying a LocalCache.

Fixed a race condition that could result in spurious wrong results appearing in a ContinuousQueryCache.

Removed deprecated static methods from the CacheFactory API.

Fixed an issue where the XML parser did not handle encoding correctly.

Removed deprecated flush-delay element from the local-scheme DTD definition.

Fixed an issue causing service-level guardian configuration to be ignored.

Added support for the listener element under the near-scheme and local-scheme elements in the cache-config.xsd file.

Oracle Coherence for .NET 3.7

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

- Added proper operational configuration support to .NET Extend services.

- The methods IdentityTransformer.transformIdentity and IdentityAsserter.assertIdentity have been changed to add a new parameter of the invoking service. Existing custom implementations will have to change the method signatures and recompile.

- Introduced user-defined indexes, which can be used to control what entries are added to an index. User-defined indexes are typically used to reduce the memory and processing overhead required to maintain an index.

- The transformIdentity method in the IdentityTransformer class and assertIdentity method in the IdentityAsserter class have been changed
to add a new parameter to specify the invoking service. This change allows you to use multiple transformers and asserters. Existing custom implementations will have to change the method signatures and recompile.

**Oracle Coherence for C++ 3.7**

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

- The `NestedPofReader/Writer` methods have been ported to C++.
- The methods `IdentityTransformer::transformIdentity` and `IdentityAsserter::assertIdentity` have been changed to add a new parameter for the invoking service. This change allows you to use multiple transformers and asserters. Existing custom implementations will have to change the method signatures and recompile.

**Changes and Enhancements for 3.7.0.1**

These improvements, and bug fixes have been made for the Oracle Coherence 3.7.0.1 release:

- Fixed potential deadlock when shutting down cluster members that use a journal cache.
- Fixed an NPE that could occur during distribution with `backup-count=0`.
- Fixed the cause of the `NullPointerException` being thrown by the `CacheStore` during a `LoadAll` operation.
- Fixed an issue where the departure of the (storage) senior while a new member joins the partitioned cache service could result in a failure to create user map indices.
- Fixed a deadlock that could occur during heavy eviction.
- Fixed an issue causing an `IllegalMonitorStateException` to be thrown by deadlock detection under heavy load.
- Fixed a performance regression that occurred when calling the `addIndex()` method with entries that have duplicate values.
- Removed the use of the ephemeral listen port for TCMP over TCP.
- Fixed an issue which caused the `entrySet()` and `values()` methods to miss large cache entries.
- Added support for large clusters (more than 500 nodes) without the need for manual packet size configuration.
- Fixed an issue where configuring `SplitSessionModel` with `coherence-enable-suspect-attributes` context parameter set to false, caused session attributes to be lost.
- Optimized memory utilization during an aggregation, which eliminates the need to keep all the intermittent data for a specific member alive.

---

**WARNING:** You may receive errors when performing an aggregation on a partially upgraded cluster when applying this fix to version 3.7.0.
Changes and Enhancements for 3.7.0.2

These improvements, and bug fixes have been made for the Oracle Coherence 3.7.0.2 release:

- Fixed an issue where single-threaded queries which cause a read-through of recently expired entries could stall the cache.
- Fixed an issue that could delay access to sorted indexes which are called from range comparison filters.
- Improved the default IP selection so that it prefers "localhost" when given multiple IPs with the equal MTUs.
- Fixed an issue with the ProxyService "server" load balancing policy that caused client connections to be incorrectly rejected.
- Fixed an issue with a few out of band (OOB) events, such as those made by the write-behind thread, that might cause corresponding updates to become vulnerable and queries to stall until the next cache operation is performed.
- Fixed an issue that could cause partitions to become unable to be transferred as a result of multiple node failures during a backup transfer.
- Fixed EOFException that occurred when calling the BinaryEntry.getOriginalValue operation.
- Fixed an issue that could cause duplicate events to be received by a MapListener following a failover with backup-count greater than 1.
- Improved MapListenerSupport handling of filter-based listeners.
- Fixed a ClassCastException that could occur during the DaemonPool$Daemon.abandon operation.
- Fixed an issue that could result in an IllegalStateException while a getAll() operation is executed during re-distribution.
- Fixed an issue that caused JournalService to deadlock on startup.
- Fixed a potential deadlock that could occur as a result of a get() operation.
- Fixed an issue where a client member was not passed to the ProxyServiceLoadBalancer.getMemberList() method.
- Fixed issue with multi-threaded putAll operations which could stall the cache when issued in combination with invocations.
- Fixed an issue that could result in failure to apply delta warnings.
- Fixed an issue where a ClosedChannelException could occur if the guardian interrupts an Elastic Data cache operation.
- Added a configuration option to suppress exceptions thrown during session invalidation.
- Fixed a rare transient regression in aggregation logic introduced by 3.6.1 patch 3.
Known Problems and Workarounds

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

Communications Issues with Cluster Members Under IPv6

When using Internet Protocol version 6 (IPv6) on older Linux kernels, members of a cluster may have problems joining and communicating with each other if they are co-located on the same host. To resolve this issue, either upgrade to a newer version of the Kernel or use IPv4 or run all members on separate hosts. For more information about the underlying issue, see:

https://bugzilla.kernel.org/show_bug.cgi?id=7432

Additionally, there are issues with running IPv6 on Windows XP because of the experimental IPv6 stack in that version. The fix will be made available in JDK 7 and is scheduled to be backported to JDK 6u27. For more information, see:


Memory Usage During Data Grid Operations

Data grid operations (such as queries and aggregations) require heap space and can exhaust the available heap memory if the result set is significantly large. When performing data grid operations, the impact on memory usage should be understood and planned for accordingly. See Coherence Administrator’s Guide for details and recommendations.

Deprecated Features

The following features have been deprecated or removed from the Coherence 3.7 release.

- The BackingMapManagerContext.getBackinMap() method, which allows you to get a reference to the backing map associated with an Entry, has been deprecated in favor of the getBackinMapEntry method. The getBackinMapEntry method implicitly locks the specified cache entry so callers may use it to atomically access, insert, update, modify, or remove cache entries from within the context of an EntryProcessor invocation. Operating on the entries returned by this method differs from operating directly against the backing-map, as the returned entries provide an isolated, read-consistent view.

- Network filters are deprecated and will be desupported. Current encryption filter implementations must be migrated to use SSL. There is no replacement for the compression filter.

- Coherence configuration files are described using XML schema definitions (XSD) and are no longer described using Document Type Definition (DTD). DTD files are deprecated and will be desupported. All configuration files that currently use the DTD files must be updated to reference and adhere to the appropriate XSD files.

- The VersionedBackingMap class has been deprecated since Coherence 3.2. It has been removed from the code base.

- The CacheFactory methods in Table 1–1, which have been deprecated since Coherence 3.3, have been removed from the API.
Support for Oracle Products Running on the Intel Itanium Processor

Oracle has discontinued all software development on the Intel Itanium microprocessor for products that have not yet been released. Oracle will continue to support existing versions of Oracle software products that already run on Itanium.

The following versions of the Intel Itanium microprocessor are affected:

Table 1–1    Deprecated Methods in the CacheFactory API

<table>
<thead>
<tr>
<th>Deprecated Method</th>
<th>Replacement Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>static NamedCache getDistributedCache()</td>
<td>Use getService(String).ensureCache(null, null) instead.</td>
</tr>
<tr>
<td>static NamedCache getDistributedCache(String sName)</td>
<td>Use getCache(String) instead.</td>
</tr>
<tr>
<td>static NamedCache getDistributedCache(String sName, ClassLoader loader)</td>
<td>Use getCache(String, ClassLoader) instead.</td>
</tr>
<tr>
<td>static CacheService getDistributedCacheService(String sName)</td>
<td>Use getService(String) instead.</td>
</tr>
<tr>
<td>static CacheService getDistributedCacheService(String sName, BackingMapManager manager)</td>
<td>No replacement.</td>
</tr>
<tr>
<td>static InvocationService getInvocationService(String sName)</td>
<td>Use getService(String) instead.</td>
</tr>
<tr>
<td>static CacheService getLocalCacheService(String sName)</td>
<td>Use getService(String) instead.</td>
</tr>
<tr>
<td>static CacheService getLocalCacheService(String sName, BackingMapManager manager)</td>
<td>No replacement.</td>
</tr>
<tr>
<td>static NamedCache getOptimisticCache()</td>
<td>Use getService(String).ensureCache(null, null) instead.</td>
</tr>
<tr>
<td>static NamedCache getOptimisticCache(String sName)</td>
<td>Use getCache(String) instead.</td>
</tr>
<tr>
<td>static NamedCache getOptimisticCache(String sName, ClassLoader loader)</td>
<td>Use getCache(String, ClassLoader) instead.</td>
</tr>
<tr>
<td>static CacheService getOptimisticCacheService(String sName)</td>
<td>Use getService(String) instead.</td>
</tr>
<tr>
<td>static CacheService getOptimisticCacheService(String sName, BackingMapManager manager)</td>
<td>No replacement.</td>
</tr>
<tr>
<td>static NamedCache getReplicatedCache()</td>
<td>Use getService(String).ensureCache(null, null) instead.</td>
</tr>
<tr>
<td>static NamedCache getReplicatedCache(String sName)</td>
<td>Use getCache(String) instead.</td>
</tr>
<tr>
<td>static NamedCache getReplicatedCache(String sName, ClassLoader loader)</td>
<td>Use getCache(String, ClassLoader) instead.</td>
</tr>
<tr>
<td>static CacheService getReplicatedCacheService(String sName)</td>
<td>Use getService(String) instead.</td>
</tr>
<tr>
<td>static CacheService getReplicatedCacheService(String sName, BackingMapManager manager)</td>
<td>No replacement.</td>
</tr>
<tr>
<td>static NamedCache getReplicatedCache(String sName)</td>
<td>Use getCache(String) instead.</td>
</tr>
<tr>
<td>static NamedCache getReplicatedCache(String sName, ClassLoader loader)</td>
<td>Use getCache(String, ClassLoader) instead.</td>
</tr>
<tr>
<td>static CacheService getReplicatedCacheService(String sName)</td>
<td>Use getService(String) instead.</td>
</tr>
<tr>
<td>static CacheService getReplicatedCacheService(String sName, BackingMapManager manager)</td>
<td>No replacement.</td>
</tr>
</tbody>
</table>
■ HP-UX Itanium
■ HP OpenVMS Itanium
■ Linux Itanium
■ Microsoft Windows Itanium (64-bit)
This chapter describes changes, enhancements, and corrections made to the Oracle Coherence documentation library for the 3.7 release. This chapter describes the following issues:

- JVM Heap Size Recommendations
- TCMP-Supported Communication Protocols and Schemes

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

http://download.oracle.com/docs/cd/E18686_01/index.htm

**JVM Heap Size Recommendations**

The "Sizing Your Heap" section in the Coherence Administration Guide recommends (on Oracle 1.6 JVM or JRockit) a conservative 4GB heap size. The recommendation is based on throughput, latency, and stability for most applications. However, larger heaps, up to 20GB, are suitable for some applications where the simplified management of fewer, larger JVMs outweighs the performance benefits of many smaller JVMs. A core-to-heap ratio of roughly 1 core: 2 GB is ideal, with some leeway to manage more GB per core (3 or even 4 GB). Every application is different and garbage collection (GC) times must be monitored accordingly.

**TCMP-Supported Communication Protocols and Schemes**

The "Understanding TCMP" section in the Oracle Coherence Developer’s Guide does not provide a complete list of the communication protocols and schemes that are supported by TCMP. TCMP can be configured to use:

- A combination of UDP/IP multicast and UDP/IP unicast. This is the default configuration.
- UDP/IP unicast only (that is, no multicast). See "Disabling Multicast Communication" for details. This configuration is used for network environments that do not support multicast or where multicast is not optimally configured.
- TCP/IP only (no UDP/IP multicast or UDP/IP unicast). See "Using the TCP Socket Provider" for details. This configuration is used for network environments that favor TCP/IP.
- SSL over TCP/IP. See "Using the SSL Socket Provider" for details. This configuration is used for network environments that require highly secure communication between cluster members.
Index

A
AbstractEvolvable class, 1-6
AbstractInvocable interface, 1-6
ActiveCache for GlassFish, 1-5
aggregations, 1-3
ArgumentOutOfRangeException class, 1-5
assertIdentity method, 1-5, 1-7, 1-8

B
BackingMapContext interface, 1-7
BackingMapManagerContext interface, 1-7
BackingMapManagerContext.getBackingMap method, 1-6, 1-10
BinaryEntry interface, 1-7
BufferOverflowException class, 1-4

C
cache usage report, 1-3
cache-config.xsd file, 1-7
CacheFactory class, 1-5, 1-7, 1-10
CacheService interface, 1-7
ClassCastException class, 1-3
client connections, 1-4
Cluster.start method, 1-4
Coherence*Extend, 1-5
clients, 1-5
Coherence*Web SPI for GlassFish 3.1, 1-5
coherence.cmd file, 1-6
coherence-enable-suspect-attributes parameter, 1-5
coaherence-pof-config.xml file, 1-6
coherence.sh file, 1-6
CompositeCache.GetAll method, 1-5
CompositeKey class, 1-6
ConcurrentModificationException class, 1-3
ConfigurablePofContext class, 1-4
ContinuousQueryCache class, 1-7
custom proxy implementations, 1-5
deprecated methods, 1-10
destroyCache method, 1-3
DiagnosticPacket class, 1-4
documentation library, 1-1

E
Elastic Data, 1-6
EmbeddedResource class, 1-5
EntryProcessor class, 1-3
ExternalizableHelper class, 1-6

F
F5 BIG-IP Load Traffic Manager, 1-5
flush-delay element, 1-7

G
getBackingMap method, 1-10
getBackingMapEntry method, 1-10
guardian configuration, 1-7

I
IBM WebSphere 7, 1-5
Identity Asserter interface, 1-5, 1-7, 1-8
IdentityTransformer interface, 1-5, 1-7, 1-8
IllegalArgumentException class, 1-6
IndexAwareExtractor interface, 1-3
indexes, 1-3
invoke method, 1-3
IPMonitor, 1-4
IPv6 literal addresses, 1-4

J
JMX, 1-6
reporter, 1-6
join-timeout, 1-4

K
keySet interators, 1-3
lightweight transactions, 1-3
LikeFilter class, 1-5
LimitFilter class, 1-3
LimitFilter setPage method, 1-7
listener element, 1-7
load balancing, 1-5
LocalCache class, 1-4, 1-7
local-scheme element, 1-7
local-storage parameter, 1-5
log messages, 1-4

map listeners, 1-7
MapEvent class, 1-7
MapIndex interface, 1-7
MapTrigger interface, 1-6
MBean registry, 1-3
MulticastTest, 1-4
My Oracle Support, 1-1

namespace collisions, 1-4
Nearcache.getAll method, 1-7
near-scheme element, 1-7
NestedPofReader method, 1-8
NestedPofWriter method, 1-8
.NET Extend services, 1-7
NotFilter class, 1-7
NullPointerException class, 1-3, 1-4, 1-6, 1-7

Peer.OnServiceStarted method, 1-5
Ping-timeout, 1-4
POF
  configuration, 1-6
  configuration file, 1-4
  date deserialization, 1-6
  serialization, 1-4
  serialization, custom, 1-4
PofHelper.ResizeArray method, 1-6
PofValue interface, 1-4
port-auto-adjust parameter, 1-6
proxy service, 1-5
putAll method, 1-3

query-related statistics, 1-3

SafeConfigurablePofContext class, 1-4
Serializable interface, 1-6
session affinity, 1-5
session cookies, 1-5
software patches, 1-1

split session model, 1-5
SSL
  connection, 1-4
  session exception, 1-4
SSL connection, 1-4
storage-disabled clients, 1-3
StorageManagerMBean MBean, 1-3
String conversions, 1-4

task timeout, 1-5
TCMP connection, 1-4
TcpConnection.open method, 1-4
TcpRing, 1-6
timeout, server-side, 1-5
transactional caches, 1-6
transformIdentity method, 1-5, 1-7, 1-8
TypeResolver class, 1-5

UnsupportedOperationException class, 1-6

VersionedBackingMap class, 1-10

WebLogicHttpSessionManagerMBean class, 1-5
windows.h files, 1-4
WKA clusters, 1-4

XML schema definition (XSD) files, 1-6, 1-10