Oracle® Coherence
Release Notes for Oracle Coherence
Release 3.5.3
E16628-01

January 2010
Preface ............................................................................................................................................................................ vii
Audience ........................................................................................................................................................................... vii
Documentation Accessibility ........................................................................................................................................ vii
Related Documents ...................................................................................................................................................... viii
Conventions ................................................................................................................................................................ viii

1 Technical Changes and Enhancements

Oracle Coherence for Java 3.5.3 ................................................................................................................................. 1-1
   Coherence*Web Enhancements and Fixes ................................................................................................................ 1-1
   Management Framework Fixes ............................................................................................................................ 1-1
   Partitioned Cache Fixes ...................................................................................................................................... 1-1
   Other Enhancements and Fixes ........................................................................................................................... 1-2
Oracle Coherence for .NET 3.5.3 ............................................................................................................................. 1-2
Oracle Coherence for C++ 3.5.3 ............................................................................................................................... 1-3

2 Documentation Errata

Changes to the Setting Single Server Mode ............................................................................................................ 2-1
Changes to Specifying a Custom Eviction Policy .................................................................................................. 2-2
Changes to init-params and class-name Elements ............................................................................................... 2-2
Changes to Well Known Addresses .................................................................................................................... 2-2
Changes to packet-buffer Description ................................................................................................................ 2-2
Changes to local-scheme Element ....................................................................................................................... 2-3
Changes to tcp-acceptor and tcp-initiator Elements .......................................................................................... 2-3
Changes to JVM Tuning ....................................................................................................................................... 2-3
   Basic Sizing Recommendation .......................................................................................................................... 2-4
   Cache Servers .................................................................................................................................................... 2-4
   TCMP Clients ..................................................................................................................................................... 2-4
   Extends Clients .................................................................................................................................................. 2-4
   Heap Size Considerations ................................................................................................................................ 2-4
   General Guidelines ................................................................................................................................................ 2-5
   Storage Ratios ..................................................................................................................................................... 2-5
   Cache Topologies and Heap Size ....................................................................................................................... 2-5
   Deciding How Many JVMs to Run Per System ................................................................................................ 2-5
   Sizing Your Heap .............................................................................................................................................. 2-6
Index
### List of Examples

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2–1</td>
<td>Disabling Partition Storage</td>
<td>2-7</td>
</tr>
<tr>
<td>2–2</td>
<td>Message Indicating Target Cluster Node is in Garbage Collection Mode</td>
<td>2-8</td>
</tr>
</tbody>
</table>
List of Tables

2–1  Change to Setting Single Server Mode ................................................................. 2-2
2–2  Changes to high-units Description ........................................................................ 2-3
2–3  Changes to low-units Description .......................................................................... 2-3
Preface

This document describes changes and enhancements that have been made to the Oracle Coherence product for the 3.5.3 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 more information, visit the Oracle Accessibility Program Web site at 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.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at http://www.fcc.gov/cgb/consumerfacts/trs.html, and a list of phone numbers is available at http://www.fcc.gov/cgb/dro/trsphonebk.html.
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*

Conventions

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></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><em>italic</em></td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td><strong>monospace</strong></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>
Technical Changes and Enhancements

This chapter describes the changes and enhancements made to the Oracle Coherence product for the 3.5.3 release. This document is accurate at the time of publication. Oracle updates the release notes periodically after the software release.

Oracle Coherence for Java 3.5.3

The following is a list of new features, improvements, and bug fixes in Oracle Coherence for Java 3.5.3.

Coherence*Web Enhancements and Fixes

- Implemented session affinity for Tomcat 5.5.x and higher.
- SPI now honors the session-timeout defined in web.xml.
- Optimized session reaping of expired sessions for large clusters.
- Enhanced ServletRequestListener to properly exit the session at the termination of execution.
- Added support for Tomcat 6.0 using the Servlet 2.5 API.
- Updated SPI implementation to reflect modifications to the WebLogic Server integration.

Management Framework Fixes

- Fixed regression in the refresh-ahead functionality in the JMX framework.
- Fixed ArrayIndexOutOfBoundsException in Reporter that could occur when all (optional) arguments were not provided.
- MBeanConnector now properly copies the JMX security properties.
- Fixed runtime error in the Reporter that occurred when a cache name contains reserved characters.

Partitioned Cache Fixes

- Fixed Partition does not exist at PartitionSplittingBackingMap errors.
- Fixed IllegalStateException in AbstractSparseArray$Crawler.remove.
• Fixed a bug that could potentially cause an `invokeAll()` to deadlock the worker threads.
• Fixed a bug causing the `invokeAll()` call, when used in conjunction with read-through, to fail to update indexes.
• Fixed a bug causing transformed events to be delivered multiple times if listeners are registered from multiple nodes.

Other Enhancements and Fixes

• Fixed `IllegalStateException` in `ReplicatedCache` that could occur during concurrent `ensureCache()` calls.
• Fixed `IllegalArgumentException` when POF is enabled globally in conjunction with the security framework.
• `PofExtractor` now behaves correctly when using `ContinuousQueryCache` with `MapEventFilter`.
• Enhanced membership information consistency to prevent unwarranted node restarts.
• Enhanced visibility of serialization and deserialization exceptions originating from `Peer.assertIdentityToken()` and `generateIdentityToken()`.
• Added the ability to configure Guardian timeouts on a per-service basis.
• Enhanced `PofExtractor` to be tolerant of null values.
• Fixed `NullPointerException` in `DaemonPool.checkHungTasks`.
• Fixed `IllegalArgumentException` caused by recursion in `OldCache` eviction.
• Integrated `AbstractProcessor` with the Guardian to prevent long running processors from being terminated.
• `SegmentedConcurrentMap.lock()` now ensures the load factor.
• Enhanced `SafeTime` to handle repetitive small system clock regressions.
• Fixed `IllegalStateException` when using a custom eviction policy that extends `AbstractEvictionPolicy`.
• `ReplicatedCache.invoke()` now respects the `ClassLoader` associated with the cache.
• Fixed the thread-safety issue in `SimpleMapIndex`.

Oracle Coherence for .NET 3.5.3

The following is a list of new features, improvements, and bug fixes in Oracle Coherence for .NET 3.5.3:

• Enhanced visibility of serialization and deserialization exceptions originating from `Peer.AssertIdentityToken()` and `GenerateIdentityToken()`.
• Fixed potential deadlock in `LocalCache` during concurrent access.
• Fixed regression in `LocalCache.GetKeys()`, `GetEntries()`, and `GetValues()` that may result in `InvalidOperationException` during concurrent access.
• Fixed `LocalCache` performance regression.
Enhanced the clarity of error message with invalid POF configuration.

Fixed regression when using ContinuousQueryCache in combination with MapListener and InFilter would only invoke the first registered listener.

Enhanced SafeTime to handle repetitive small system clock regressions.

Fixed the thread-safety issue in SimpleMapIndex.

Oracle Coherence for C++ 3.5.3

The following is a list of new features, improvements, and bug fixes in Oracle Coherence for C++ 3.5.3:

- Enhanced visibility of serialization and deserialization exceptions originating from Peer::assertIdentityToken() and generateIdentityToken().
- Enhanced SafeTime to handle repetitive small system clock regressions.
- Fixed the thread-safety issue in SimpleMapIndex.
This chapter describes changes, enhancements, and corrections made to the Oracle Coherence documentation library for the 3.5.3 release. The library can be found at the following URL:

http://download.oracle.com/docs/cd/E14526_01/index.htm

- Changes to the Setting Single Server Mode
- Changes to Specifying a Custom Eviction Policy
- Changes to init-params and class-name Elements
- Changes to Well Known Addresses
- Changes to packet-buffer Description
- Changes to local-scheme Element
- Changes to tcp-acceptor and tcp-initiator Elements
- Changes to JVM Tuning
- Changes to Features By Edition
- Changes to Local Cache Example
- Changes to Supported Environments for Coherence for C++
- Changes to the Logger Configuration Example for .NET Clients
- Changes to .NET Installation
- Changes to Tomcat Server Type Aliases for Coherence*Web
- Deprecation of the Coherence*Web APIs
- Disable Page Tag Pooling when Deploying on IBM WebSphere

Changes to the Setting Single Server Mode

There has been a change in the text of the Setting Single Server Mode chapter of the Developer’s Guide for Oracle Coherence.
Changes to Specifying a Custom Eviction Policy

There has been an addition to the description of Specifying a Custom Eviction Policy in the Developer’s Guide for Oracle Coherence:

If using Coherence 3.5, you may run into an IllegalStateException using this example. As a workaround, modify the eviction policy to implement LocalCache.EvictionPolicy. For example:

```
public class SimpleEvictionPolicy
extends AbstractEvictionPolicy
implements LocalCache.EvictionPolicy
```

Changes to init-params and class-name Elements

There has been a change in the descriptions of the init-params and class-name elements in Appendix A, Operational Configuration Elements, in the Developer’s Guide for Oracle Coherence. These elements can now be used as sub-elements of the default-serializer element.

Changes to Well Known Addresses

There is an error in Appendix A, Coherence Features by Edition in Getting Started with Oracle Coherence. The description states that Multicast-free operation (also known as "Well Known Addresses", or "WKA") is supported by the Standard Edition. It is not.

There is also an error in the description of the well-known-addresses element in the Developer’s Guide for Oracle Coherence. The description states: "Use of the Well Known Addresses (WKA) feature is not supported by Caching Edition..." Instead of Caching Edition, this should read Standard Edition.

Changes to packet-buffer Description

The following section has been added to the description of the packet-buffer element in Appendix A, Operational Configuration Elements, in the Developer’s Guide for Oracle Coherence:

**Latency Versus Throughput**

When setting this for transmit (that is, within packet-publisher), higher values may allow for increased throughput, while lower values may allow for decreased latency. If you make any changes to this value, it is recommended that you evaluate how it effects performance in all of these dimensions.

### Table 2–1 Change to Setting Single Server Mode

<table>
<thead>
<tr>
<th>Old Text</th>
<th>New Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>The unicast IP address is defined by the &lt;address&gt; sub-element of the</td>
<td>The unicast IP address is defined by the &lt;address&gt; sub-element of the</td>
</tr>
<tr>
<td>unicast-listener element. This sub-element specifies the IP address</td>
<td>unicast-listener element. This sub-element specifies the IP address</td>
</tr>
<tr>
<td>that a Socket will listen or publish on. Setting this sub-element to</td>
<td>that a Socket will listen or publish on and must be routed to loopback</td>
</tr>
<tr>
<td>an IP address that is never used will prevent Coherence from joining</td>
<td>as well. Setting this sub-element to an IP address that is never used</td>
</tr>
<tr>
<td>the network.</td>
<td>will prevent Coherence from joining the network.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The localhost setting may not work on systems that define</td>
</tr>
<tr>
<td></td>
<td>localhost as the loopback address; in that case, specify the machine</td>
</tr>
<tr>
<td></td>
<td>name or the specific IP address.</td>
</tr>
</tbody>
</table>
Changes to local-scheme Element

Changes have been made to the descriptions of the low-units and high-units attributes of the local-scheme element in Appendix B, Cache Configuration Elements in the Developer’s Guide for Oracle Coherence.

Table 2–2  Changes to high-units Description

<table>
<thead>
<tr>
<th>Old text</th>
<th>New Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>high-units—(Optional) Used to limit the size of the cache. Contains the maximum number of units that can be placed in the cache before pruning occurs. An entry is the unit of measurement, unless it is overridden by an alternate unit-calculator. Once this limit is exceeded, the cache will begin the pruning process, evicting entries according to the eviction policy until the low-units size is reached.</td>
<td>high-units—(Optional) Used to limit the size of the cache. Contains the maximum number of units that can be placed in the cache before pruning occurs. An entry is the unit of measurement, unless it is overridden by an alternate unit-calculator. Once this limit is exceeded, the cache will begin the pruning process, evicting entries according to the eviction policy.</td>
</tr>
</tbody>
</table>

Table 2–3  Changes to low-units Description

<table>
<thead>
<tr>
<th>Old Text</th>
<th>New Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>low-units—(Optional) Contains the number of units that the cache will be pruned down to when pruning takes place. An entry is the unit of measurement, unless it is overridden by an alternate unit-calculator. When pruning occurs entries will continue to be evicted according to the eviction policy until this size.</td>
<td>low-units—(Optional) Contains the lowest number of units that a cache will be pruned down to when pruning takes place. A pruning will not necessarily result in a cache containing this number of units, however a pruning will never result in a cache containing less than this number of units. An entry is the unit of measurement, unless it is overridden by an alternate unit-calculator. When pruning occurs entries will continue to be evicted according to the eviction policy until this size.</td>
</tr>
</tbody>
</table>

Changes to tcp-acceptor and tcp-initiator Elements

In the Developer’s Guide for Oracle Coherence, the reusable subelement of tcp-acceptor element has been deprecated. Use the new reuse-address element instead. The reuse-address element specifies whether a TCP/IP socket can be bound to an address if a previous connection is in a timeout state. When a TCP/IP connection is closed the connection may remain in a timeout state for a period after the connection is closed (typically known as the TIME_WAIT state or 2MSL wait state). For applications using a well-known socket address or port, it may not be possible to bind a socket to a required address if there is a connection in the timeout state involving the socket address or port. If no value for reuse-address is specified, the default value is true.

The reuse-address element has also been added as a sub-element of tcp-initiator. The definition of reuse-address element is the same, but in this case the default value is false.

Changes to JVM Tuning

The following text resolves inconsistencies and consolidates information on JVM tuning that appears in the Best Practices appendix in Getting Started with Oracle Coherence and the Performance Tuning chapter in the Developer’s Guide for Oracle Coherence.

- Basic Sizing Recommendation
- Heap Size Considerations
Changes to JVM Tuning

Basic Sizing Recommendation

The recommendations in this section are sufficient for general use cases and require minimal setup effort. The primary issue to consider when sizing your JVMs is a balance of available RAM versus garbage collection (GC) pause times.

Cache Servers

The standard, safe recommendation for Coherence cache servers is to run a fixed size heap of up to 1GB. Additionally, it is recommended to utilize an incremental garbage collector to minimize GC pause durations. Lastly, run all Coherence JVMs in server mode, by specifying the `server` on the JVM command line. This allows for several performance optimizations for long running applications.

For example:

```shell
java -server -Xms1g -Xmx1g -Xincgc -Xloggc: -cp coherence.jar com.tangosol.net.DefaultCacheServer
```

This sizing allows for good performance without the need for more elaborate JVM tuning. For more information on garbage collection, see "GC Monitoring and Tuning" on page 2-7.

---

**Note:** It is possible to run cache servers with larger heap sizes; though it becomes more important to monitor and tune the JVMs to minimize the GC pauses. It may also be necessary to alter the storage ratios such that the amount of scratch space is increased to facilitate faster GC compactions. Additionally it is recommended that you make use of an up to date JVM version such as HotSpot 1.6 as it includes significant improvements for managing large heaps. See "Heap Size Considerations" on page 2-4.

---

TCMP Clients

Coherence TCMP clients should be configured similarly to cache servers as long GCs could cause them to be misidentified as being dead.

Extends Clients

Coherence Extend clients are not technically speaking cluster members and, as such, the effect of long GCs is less detrimental. For extend clients it is recommended that you follow the existing guidelines as set forth by the application in which you are embedding coherence.

Heap Size Considerations

This section will help you decide:

- How many CPUs you will need for your system
- How much memory you will need for each system
- How many JVMs to run per system
- How much heap to configure with each JVM
Since all applications are different, this section should be read as guidelines. You will need to answer the following questions to choose the configuration that is right for you:

- How much data will be stored in Coherence caches?
- What are the application requirements in terms of latency and throughput?
- How CPU or Network intensive is the application?

Sizing is an imprecise science. There is no substitute for frequent performance and stress testing.

The following topics are included in this section:

- General Guidelines
- Moving the Cache Out of the Application Heap

**General Guidelines**

Running with a fixed sized heap will save your JVM from having to grow the heap on demand and will result in improved performance. To specify a fixed size heap use the `-Xms` and `-Xmx` JVM options, setting them to the same value. For example:

```
java -server -Xms4G -Xmx4G ...
```

A JVM process consumes more system memory than the specified heap size. The heap size settings specify the amount of heap which the JVM makes available to the application, but the JVM itself will also consume additional memory. The amount consumed differs depending on the OS, and JVM settings. For instance, a HotSpot JVM running on Linux configured with a 1GB JVM will consume roughly 1.2GB of RAM. It is important to externally measure the JVMs memory utilization to ensure that RAM is not over committed. Tools such as `top`, `vmstat`, and Task Manager are useful in identifying how much RAM is actually being utilized.

**Storage Ratios** The basic recommendation for how much data can be stored within a cache server of a given size is to use up to 1/3rd of the heap for primary cache storage. This leaves another 1/3rd for backup storage, and the final 1/3rd for scratch space. Scratch space is then used for things such as holding classes, temporary objects, network transfer buffers, and GC compaction. You may instruct Coherence to limit primary storage on a per-cache basis by configuring the `<high-units>` element and specifying a BINARY value for the `<unit-calculator>` element. These settings are automatically applied to backup storage as well.

Ideally both the primary and backup storage will also fit within the JVMs tenured space (for HotSpot based JVMs). See HotSpot’s Tuning Garbage Collection guide for details on sizing the collectors generations:

```
http://java.sun.com/docs/hotspot/gc5.0/gc_tuning_5.html
```

**Cache Topologies and Heap Size** For large datasets, Partitioned or Near caches are recommended. As the scalability of the Partitioned cache is linear for both reading and writing, varying the number of Coherence JVMs will not significantly affect cache performance. Using a Replicated cache will put significant pressure on GC.

**Deciding How Many JVMs to Run Per System** The number of JVMs (nodes) to run per system depends on the system’s number of processors/cores and amount of memory. As a starting point, we recommend starting with a plan to run one JVM for every two cores. This recommendation balances the following factors:
Changes to JVM Tuning

- Multiple JVMs per server allow Coherence to make more efficient use of network resources. Coherence’s packet-publisher and packet-receiver have a fixed number of threads per JVM; as you add cores, you’ll want to add JVMs to scale across these cores.
- Too many JVMs will increase contention and context switching on processors.
- Too few JVMs may not be able to handle available memory and may not fully utilize the NIC.
- Especially for larger heap sizes, JVMs must have available processing capacity to avoid long GC pauses.

Depending on your application, you can add JVMs up toward one per core. The recommended number of JVMs and amount of configured heap may also vary based on the number of processors/cores per socket and on the machine architecture.

**Sizing Your Heap**  When considering heap size, it is important to find the right balance. The lower bound is determined by per-JVM overhead (and also, manageability of a potentially large number of JVMs). For example, if there is a fixed overhead of 100MB for infrastructure software (for example, JMX agents, connection pools, internal JVM structures), then the use of JVMs with 256MB heap sizes will result in close to 40% overhead for non-cache data. The upper bound on JVM heap size is governed by memory management overhead, specifically the maximum duration of GC pauses and the percentage of CPU allocated to GC (and other memory management tasks).

GC can affect the following:

- The latency of operations against Coherence. Larger heaps will cause longer and less predictable latency than smaller heaps.
- The stability of the cluster. With very large heaps, lengthy long garbage collection pauses can trick TCMP into believing a cluster member is dead since the JVM is unresponsive during GC pauses. Although TCMP takes GC pauses into account when deciding on member health, at some point it may decide the member is dead.

We offer the following guidelines:

- With older JVMs such as Sun 1.4 and 1.5, we recommend not allocating more than 1GB and 2GB heap respectively.
- For newer JVMs (Sun 1.6 or JRockit, for example), we recommend allocating up to a 4GB heap. We also recommend using Sun’s Concurrent Mark and Sweep GC or JRockit’s Deterministic GC.

The length of a GC pause scales worse than linearly to the size of the heap. That is, if you double the size of the heap, pause times due to GC will, in general, more than double. GC pauses are also impacted by application usage:

- Pause times increase as the amount of live data in the heap increases. We recommend not exceeding 70% live data in your heap. This includes primary data, backup data, indexes, and application data.
- High object allocation rates will increase pause times. Even "simple" Coherence applications can cause high object allocation rates since every network packet generates many objects.
- CPU-intensive computation will increase contention and may also contribute to higher pause times.

Depending on your latency requirements, you can increase allocated heap space beyond the above recommendations, but be sure to stress test your system.
Moving the Cache Out of the Application Heap

Using dedicated Coherence cache server instances for Partitioned cache storage will minimize the heap size of application JVMs as the data is no longer stored locally. As most Partitioned cache access is remote (with only 1/N of data being held locally), using dedicated cache servers does not generally impose much additional overhead. Near cache technology may still be used, and it will generally have a minimal impact on heap size (as it is caching an even smaller subset of the Partitioned cache). Many applications are able to dramatically reduce heap sizes, resulting in better responsiveness.

Local partition storage may be enabled (for cache servers) or disabled (for application server clients) with the tangosol.coherence.distributed.localstorage Java property (for example, -Dtangosol.coherence.distributed.localstorage=false).

It may also be disabled by modifying the <local-storage> setting in the tangosol-coherence.xml (or tangosol-coherence-override.xml) file as follows:

Example 2–1 Disabling Partition Storage

<!--
Example using tangosol-coherence-override.xml
-->
<coherence>
  <cluster-config>
    <services>
      <!--
id value must match what's in tangosol-coherence.xml for DistributedCache service
-->
      <service id='3'>
        <init-params>
          <init-param id='4'>
            <param-name>local-storage</param-name>
            <param-value system-property="tangosol.coherence.distributed.localstorage">false</param-value>
          </init-param>
        </init-params>
      </service>
    </services>
  </cluster-config>
</coherence>

At least one storage-enabled JVM must be started before any storage-disabled clients access the cache.

GC Monitoring and Tuning

Lengthy GC pause times can negatively impact the Coherence cluster and are, for the most part, indistinguishable from node death. During these pauses, a Java application is unable to send or receive packets and in the case of receiving the operating system buffered packets, the packets may be discarded and need to be retransmitted. For these reasons, it is very important that cluster nodes are sized and/or tuned to ensure that their GC times remain minimal. As a good rule of thumb, a node should spend less than 10% of its time paused in GC, normal GC times should be under 100ms, and maximum GC times should be around 1 second.
Changes to Features By Edition

Changes to Features By Edition

A sub-row for C++ has been added to the API Language row of the Coherence Client Editions table in Appendix A, Coherence Features by Edition in Getting Started with Oracle Coherence. The sub-row indicates that C++ API are available for the Data Client and the Real Time Client configured as an Extend/TCP Client. There is no available C++ API for the Real Time Client configured as a Compute Client.

Changes to Local Cache Example

A correction was made to the example in the Configuring the Local Cache section of Getting Started with Oracle Coherence. In the example, the configuration of the <flush-delay> element has been removed.

Changes to Supported Environments for Coherence for C++

There has been a change to the table in the Supported Environments section of the Requirements, Installation, and Deployment chapter in the Client Guide for Oracle Coherence.

Note 6 below the table should now read as follows:

6. Specifically, GCC 3.4.6-8 and above, as well as GCC 4.x versions are supported.

Changes to the Logger Configuration Example for .NET Clients

The example in the Logger section of the Configuration and Usage for .NET Clients chapter of Client Guide for Oracle Coherence has been corrected. The example incorrectly displayed log4net as the value for the destination element. This value has been corrected to common-logger:

```
<coherence>
  <logging-config>
    <destination>common-logger</destination>
    <severity-level>5</severity-level>
    <message-format>(thread={thread}): {text}</message-format>
  </logging-config>
</coherence>
```
Changes to .NET Installation

The following changes have been made to the Requirements, Installation and Deployment for Coherence for .NET chapter in the Client Guide for Oracle Coherence.

- An additional package requirement has been added for running .NET:
  MSHelp 2.x Runtime, which is included in Visual Studio 200x, as well as the applications at this URL:
  http://www.helpware.net/mshelp2/h20.htm#MS_H2_Runtime
- Note that after Step 2, "Run the installer by double clicking on the installer file":
  If the installer indicates that it is rolling back the installation, then you must run the installer in elevated execution mode. For example, executing the MSI file from a command line shell that was started as Administrator should enable the installation process to complete.

Changes to Tomcat Server Type Aliases for Coherence*Web

In previous releases, the User’s Guide for Oracle Coherence*Web listed the Server Type Aliases for the Tomcat 5.5.x and Tomcat 6.0.x application servers as Generic. This has changed in the current release. Use Tomcat/5.5.x as the Server Type Alias for Tomcat 5.5.x and Tomcat/6.0.x as the alias for Tomcat 6.0.x.

Deprecation of the Coherence*Web APIs

As of this release, the Coherence*Web API should be considered to be deprecated, and their use should be discouraged. They will be removed in a future release.

Disable Page Tag Pooling when Deploying on IBM WebSphere

When deploying on IBM WebSphere thread and page tag pooling must be disabled. For more information on disabling page tag pooling on IBM Websphere, see:

Index

A
AbstractProcessor, 1-2
ArrayIndexOutOfBoundsException, 1-1

C
C++ API support, 2-8
class-name element, 2-2
Coherence for C++, 2-8
Coherence*Web
  APIs, 2-9
  server type aliases, 2-9
ContinuousQueryCache, 1-2
custom eviction policy, 2-2

D
DaemonPool.checkHungTasks, 1-2
destination element, 2-8

E
ensureCache(), 1-2

F
flush-delay element, 2-8

G
GC monitoring, 2-7
Guardian timeouts, 1-2

H
heap size, 2-4
high-units attribute, 2-3

I
IllegalArgumentException, 1-2
IllegalStateException, 1-1, 1-2
init-params element, 2-2
installation changes
  .NET, 2-9
  invokeAll(), 1-2

J
JMX security properties, 1-1
JVM tuning, 2-3

L
local scheme element, 2-3
logger configuration, 2-8
low-units attribute, 2-3

M
MapEventFilter, 1-2

N
.NET
  clients, 2-8
  installation changes, 2-9
  package requirements, 2-9
node restarts, 1-2
NullPointerException, 1-2
NullPointersException, 1-2

O
OldCache, 1-2

P
package requirements, .NET, 2-9
packet-buffer element, 2-2
packet-publisher element, 2-2
Partition does not exist at
  PartitionSplittingBackingMap, 1-1
Peer.assertIdentityToken(), 1-2
PofExtractor, 1-2

R
ReplicatedCache, 1-2
Reporter, 1-1
reusable element, 2-3
reuse-address element, 2-3
S
server type aliases, Coherence*Web, 2-9
Servlet 2.5 API, 1-1
ServletRequestListener, 1-1
session reaping, 1-1
session-timeout element, 1-1
single server mode, 2-1
supported environments, C++, 2-8

T
tcp-acceptor element, 2-3
tcp-initiator element, 2-3
Tomcat application server
  5.5.x, 1-1
  6.0, 1-1
    server type aliases, 2-9

U
unicast-listener element, 2-2

W
well known addresses, 2-2
worker threads, 1-2