

Oracle® Coherence

Release Notes for Oracle Coherence

Release 3.6

E17969-02

August 2010

Oracle Coherence Release Notes for Oracle Coherence, Release 3.6

E17969-02

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

Primary Author: Thomas Pfaeffle

Contributing Author: Noah Arliss, Mark Falco, Alex Gleyzer, Gene Gleyzer, Jason Howes, Adam Leftik, Rob Misek, Patrick Peralta

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	v
Audience	v
Documentation Accessibility	v
Related Documents	vi
Conventions	vi
1 Technical Changes and Enhancements	
Download and Install the Latest Software Patch	1-1
New and Improved Coherence Data Grid Functionality	1-1
Oracle Coherence for Java 3.6	1-2
Coherence*Extend Enhancements and Fixes	1-2
New Features	1-2
Enhancements	1-2
Fixes	1-3
Coherence*Web Enhancements and Fixes	1-3
New Features	1-3
Enhancements	1-3
SPI-Specific Enhancements and Fixes	1-4
Fixes	1-4
Management Framework Enhancements and Fixes	1-5
Security Framework Enhancements and Fixes	1-5
Serialization Framework Enhancements and Fixes	1-5
Other Enhancements and Fixes	1-6
Oracle Coherence for .NET 3.6	1-6
New Features	1-7
Other Enhancements and Fixes	1-7
Oracle Coherence for C++ 3.6	1-7
New Features	1-7
Other Enhancements and Fixes	1-7
2 Documentation Errata	
Packaging EAR Files When Using the Coherence Resource Adapter	2-1
Using Oracle Coherence for C++ with MSVC 2005 SP1	2-1
Avoid Using the size() Operation on Transactional Caches	2-2
The CohQL BACKUP CACHE Command Does Not Support POF	2-2

Deprecated Elements Referenced in Developer’s Guide	2-2
Maximum Value for Time-Based Configuration Elements.....	2-2
Typo in the manifest.mf File in the ActiveCache Documentation	2-2
Error in .NET SSL Example	2-2
Updated Instructions for Configuring Cluster Nodes for Coherence*Web	2-3
Configure Cluster Nodes (WebLogic Server 10.3.3 and Later)	2-3
Configuring Application Server-Scoped Cluster Nodes.....	2-4
Configuring EAR-Scoped Cluster Nodes.....	2-4
Configuring WAR-Scoped Cluster Nodes	2-5
Configure Cluster Nodes (WebLogic Server 10.3.2 and Earlier).....	2-6
Configuring Application Server-Scoped Cluster Nodes.....	2-7
Configuring EAR-Scoped Cluster Nodes.....	2-7
Configuring WAR-Scoped Cluster Nodes	2-8

Index

Preface

This document describes changes and enhancements that have been made to Oracle Coherence for the 3.6 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.

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> or visit <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*

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.
<i>italic</i>	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.

Technical Changes and Enhancements

This chapter describes the changes and enhancements made to the Oracle Coherence product for the 3.6 release. This document is accurate at the time of publication. Oracle updates the release notes periodically after the software release.

- [Download and Install the Latest Software Patch](#)
- [New and Improved Coherence Data Grid Functionality](#)
- [Oracle Coherence for Java 3.6](#)
- [Oracle Coherence for .NET 3.6](#)
- [Oracle Coherence for C++ 3.6](#)

Download and Install the Latest Software Patch

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

<https://support.oracle.com/CSP/ui/flash.html>

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

The following new functionality has been added to the Coherence 3.6 release.

- Introduced a new MVCC-based Transaction Framework that provides ACID transaction guarantees across partitions and caches even in the event of failure, and supports the use of `NamedCache` operations, queries, aggregation, and entry processors within the context of a transaction. The three components that are included are a `<transactional-scheme>` cache configuration element, a Connection-Based API and a fully XA-Compliant Resource Adapter.
- Introduced the Coherence Query Language (CohQL) which provides a light-weight SQL-like syntax that can be used to perform cache operations on a

Coherence cluster. The language can be used both programmatically and from a new command-line tool.

- Introduced Quorum support which provides a mechanism that allows customers to configure how Coherence should behave at various service and/or cluster sizes.
- Introduced the Coherence `ASP.NET` Session Management module which allows you to manage `ASP.NET` session state in a Coherence cluster in the same way `Coherence*Web` provides session management in JEE containers.
- Added SSL support for `Coherence*Extend` and `TCMP`.
- 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.
- Introduced a new `BinaryEntryStore` interface, which is similar to the `CacheStore`, but operates on `BinaryEntry` objects. The interface allows implementations to use optimistic concurrency for database updates (using the `OriginalValue` accessor) and interoperate with the backing map by modifying the entry's value.
- Added fast machine-level death detection (`IPMonitor`).
- Added deterministic process-level death detection to `TcpRing`.
- Added the `BroadPartitioningStrategy` implementation, for use with key association, that "spreads" the partition assignment for a given group based on a "span" value.

Oracle Coherence for Java 3.6

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

- [Coherence*Extend Enhancements and Fixes](#)
- [Coherence*Web Enhancements and Fixes](#)
- [Management Framework Enhancements and Fixes](#)
- [Security Framework Enhancements and Fixes](#)
- [Serialization Framework Enhancements and Fixes](#)
- [Other Enhancements and Fixes](#)

Coherence*Extend Enhancements and Fixes

The following is a list of new features, improvements, and bug fixes to `Coherence*Extend` for the 3.6 release.

New Features

- Added SSL support for `Coherence*Extend`.

Enhancements

- Optimized streaming of query results between cache servers and proxies.
- Changed the default value of the `reuse-address` configuration element to `true`.
- Fixed issue with `PofBufferedReader.getParentParser()` accessor, where the wrong parent could be returned.

Fixes

- Fixed an issue in which a misconfiguration of network filters by clients could terminate the proxy service.
- Hardened `NamedCacheProxy.unregisterChannel()` to protect against a concurrent release or destroy of the wrapped cache.
- Fixed an issue in which `WrapperSet.$KeySet.remove()` could potentially result in multiple `RemoveRequests`.
- Fixed an issue in which multiple network filters were being applied in inverse order for send and receive.
- Hardened the `TCPAcceptor` behavior when null is returned from the configured `AddressProvider`.
- `Initiator.openConnection()` now respects the connect timeout setting.

Coherence*Web Enhancements and Fixes

The following is a list of new features, improvements, and bug fixes to Coherence*Web for the 3.6 release.

New Features

- Introduced a Coherence*Web-specific report for the Reporter that consolidates the more relevant JMX statistics for managing and monitoring Coherence*Web.
- Implemented session affinity for Tomcat 5.5 and JBoss.
- Added support for restricting the return of session cookies over secure protocols only.
- Implemented a Tomcat 6.0 `DefaultFactory` to support the Servlet 2.5 API.

Enhancements

- Added various optimizations and fixed several issues with session reaping.
- Optimized session reaping by implementing parallel reaping routines.
- Hardened deserialization routines involving `SessionExpiryExtractor` by using `Binary` decorations.
- Added a new locking mode that allows concurrent updates to the session by way of a "last writer wins" policy.
- Hardened support for single session ID referencing of a unique `HttpSession` instance in each Web application using WebLogic Server SPI.
- Optimized WebLogic Server SPI session and session attribute event dispatching.
- Optimized session cookie encoding.
- Hardened `AbstractHttpSessionCollection.isExistent()` implementation for cases when session ID is null.
- Introduced enforcement of consistent Coherence*Web configuration across Web applications that share session data.
- Introduced ability for users to explicitly disable the automatic suspect attribute.
- Introduced the ability to customize attribute serialization.

- Introduced the ability to configure `Coherence*Web` to automatically print a thread dump on a node holding a session lock.
- Optimized the update of session attributes when using the split session model.
- Introduced a configuration parameter to allow session ownership to be lazily acquired.
- Optimized flush behavior on the `HttpSession` model by deferring the flush until the last concurrent request completes.
- Converted to using `StringBuilder` rather than `StringBuffer`.
- Optimized logging.
- Decoupled local session attributes from sticky session optimization.
- Added the `coherence-session-get-lock-timeout` context parameter to configure lock timeouts for `Coherence*Web`.
- Introduced the ability to override context parameters by using system properties.

SPI-Specific Enhancements and Fixes

- Changed the default locking model in the WebLogic Server SPI to be consistent with the default in non-SPI configurations.
- Enabled session metric reporting in the WebLogic Server console using the SPI.
- Enabled side-by-side deployment in WebLogic Server using the SPI.

Fixes

- Fixed the `IllegalStateException: session is not in a 'active' state` exception that could occur if thread locking is disabled and there are concurrent requests for the same session.
- Fixed the issue where `Coherence*Web` would not honor the `session-timeout` value configured in the `web.xml` file.
- Fixed a potential deadlock that could occur during remote include and forward when using `Coherence*Web` with the WebLogic Server SPI.
- Fixed session ownership issues after executing WebLogic Server proprietary logout methods.
- Introduced the caching of `HttpSessions` after application redeployment in WebLogic Server.
- Fixed `ClassNotFoundException` that could occur if multiple Web applications were deployed when using application or EAR-scoped cluster nodes.
- Hardened `ServletRequestListener` session exit behavior after it executes.
- Fixed instances where JSP XML files could potentially be improperly encoded by the installer.
- Hardened update rejections on sessions that have been concurrently invalidated when using the optimistic locking mode.
- Fixed an issue regarding the management of non-serializable attribute retention.
- Fixed `ClassNotFoundException` during session invalidation.
- Fixed issue with JBoss and TagLibs caused by `com.tangosol.coherence.servlet.api23.JspServlet` not implementing `_jspInit()`.

Management Framework Enhancements and Fixes

The following is a list of new features, improvements, and bug fixes to the management framework for the 3.6 release.

- Added support for MBean notifications
- Added ability to reduce the number of registered MBeans based on the name patterns
- Added a new `TransactionManagerMBean`.
- Added membership notifications to the `ClusterMBean`.
- Added the `MemberDeparted` attribute to the `ClusterMBean` showing a list of recently departed members.
- Added the `QuorumStatus` attribute to the `ClusterNodeMBean` and `ServiceMBean`.
- Fixed `ClassCastException` when using `DoubleSum.process` with the `Reporter`.
- Fixed the `rolename` attribute initialization issue that could occur when running with `JRockit`.
- Hardened `CacheMBean` to insulate itself from the cache being concurrently destroyed.
- Reduced the memory footprint for `ClusterNodeMBean`.
- Fixed an issue with reporting negative values when services are restarted.

Security Framework Enhancements and Fixes

The following is a list of new features, enhancements and bug fixes to the security framework for 3.6.

- Added pluggable identity handling between client and proxy to enable identity validation and custom security tokens.
- Added subject scoping to automatically associate Subjects with remote cache and invocation service references.
- Added ability to wrap remote cache service and remote invocation service to allow custom authorization code in the proxy.
- Fixed bug that would ignore security exceptions for expired certificates.
- Fixed concurrency bug in security framework when under heavy load.

Serialization Framework Enhancements and Fixes

The following is a list of new features, improvements, and bug fixes to the serialization framework for the 3.6 release.

- Optimized `PacketPublisher` by replacing the use of streams with buffers.
- Optimized aggregator and entry processor serialization by using `LiteMap` and `LiteSet`.
- Added `EnumPofSerializer`.
- Removed overly strict assertions from `AbstractEvolvable`.
- Optimized `Number` deserialization in POF.

- Fixed an issue with `PofExtractor` where it could fail to extract values from uniform collections.
- Fixed issue with `PofBufferedReader.getParentParser()` accessor, where the wrong parent could be returned.

Other Enhancements and Fixes

The following is a list of other improvements and bug fixes for the 3.6 release.

- Allowed for a disabled guardian to monitor and log SLA violations.
- Optimized configuration updates during distribution of partitions.
- Minimized operating system level UDP socket contention.
- Added a `RegexFilter` implementation.
- Reduced index overhead for indices with high cardinality.
- Added support for `KeyExtractor`-based filter to the `ContinuousQueryCache`.
- Added the ability to determine which node a partition is backed up on.
- Added support for custom filtering of the authorized hosts, determining if a host is allowed to join the cluster.
- Optimized forward index by reusing previously extracted `Collection` references
- Scaled-back the aggressiveness of write-delay after a failure.
- Hardened the `ReadWriteBackingMap` implementation to protect the write-behind queue from excessive growth.
- Changed the default setting for `rollback-cachestore-failures` to `true`.
- Allowed the `MachineId` element to be specified as a string.
- Improved cluster join protocol to enforce the cluster size limitation imposed by maximum packet size.
- Optimized memory footprint for `PacketPublisher`.
- Added ability to layer network protocol filters.
- Optimized the handling of write-behind thread timeouts.
- Added the ability to scroll through the history of commands in the command line tool.
- Fixed a race condition during simultaneous service start (with ownership-disabled senior), which could potentially lead to ownership desynchronization.
- Resolved race condition related to well-known and non-well-known members forming a cluster.
- Fixed an issue where `ServiceEvents` were potentially not dispatched during a hard stop.
- Fixed issue where a malformed or rogue `SeniorMemberHeartbeat` could result in cluster restart.

Oracle Coherence for .NET 3.6

The following is a list of new features, improvements, and bug fixes in Oracle Coherence for .NET 3.6:

New Features

- Added SSL support for Coherence*Extend.

Other Enhancements and Fixes

- Added connection-specific information to the `ConnectionException` class.
- Improved the information reported when a `ConnectionException` occurs due to socket disconnect.
- Optimized `RemoteNamedCache.BinaryCache.Query()` by using `ImmutableMultiList`.
- Introduced the ability to associate identity with cache and service references.
- Added an `EnumPofSerializer`.
- Converted `CacheEntry` to a value type.
- Fixed an issue where `GetLocalHostAddress()` could potentially return the IP address of a DNS Server.
- Fixed the incorrect usage of `IDisposable`.
- Fixed an issue where network filters were being applied in inverse order, for send and receive.
- `Initiator.OpenConnection()` now respects the connect timeout setting.
- Fixed issue with `PofBufferReader.getParentParser()` accessor, where the wrong parent could be returned.

Oracle Coherence for C++ 3.6

The following is a list of new features, improvements, and bug fixes in Oracle Coherence for C++ 3.6:

New Features

- Exposed `TreeMap` and Universally Unique Identifier (UUID) implementations.
- Added support for Visual Studio 2010.

Other Enhancements and Fixes

- Improved the information reported when a `ConnectionException` occurs upon socket disconnect.
- Optimized the buffering of reads and writes to the underlying `Socket` in `TcpInitiator::readMessageLength()` and `writeMessageLength()`.
- Optimized `RemoteNamedCache::BinaryCache::query()` by using `ImmutableMultiList`.
- Added `static Array<T>::hashCode(const T*)` helper.
- Made `GenericPrincipal` class available in C++.
- Fixed issue where network filters were being applied in inverse order for send and receive.
- `Initiator::openConnection()` now respects the connect timeout setting.

- Fixed potential deadlock between `TcpInitiator` and `TcpReader` during shutdown.
- Fixed segmentation faults that could occur when unloading the Coherence library.
- Fixed issue with `PofBufferReader::getParentParser()` accessor, where the wrong parent could be returned.

Documentation Errata

This chapter describes changes, enhancements, and corrections made to the Oracle Coherence documentation library for the 3.6 release. The library can be found at the following URL:

http://download.oracle.com/docs/cd/E15357_01/index.htm

Documentation issues:

- [Packaging EAR Files When Using the Coherence Resource Adapter](#)
- [Using Oracle Coherence for C++ with MSVC 2005 SP1](#)
- [Avoid Using the size\(\) Operation on Transactional Caches](#)
- [The CohQL BACKUP CACHE Command Does Not Support POF](#)
- [Deprecated Elements Referenced in Developer's Guide](#)
- [Maximum Value for Time-Based Configuration Elements](#)
- [Typo in the manifest.mf File in the ActiveCache Documentation](#)
- [Error in .NET SSL Example](#)
- [Updated Instructions for Configuring Cluster Nodes for Coherence*Web](#)

Packaging EAR Files When Using the Coherence Resource Adapter

The "Include the Required Libraries" section in the *Oracle Coherence Developer's Guide* provides instructions for packaging an EAR file when using the Coherence resource adapter. The instructions recommend placing the `coherence.jar` file in the root of the EAR file. This is a generic recommendation. When deploying to WebLogic Server, the `coherence.jar` file should be placed in the `/APP-INF/lib` directory of the EAR file. For example:

```
/coherence-transaction.rar  
/APP-INF/lib/coherence.jar
```

Using Oracle Coherence for C++ with MSVC 2005 SP1

Using Oracle Coherence for C++ with MSVC 2005 SP1 (x86 and x64) requires both the Microsoft Visual C++ 2005 Service Pack 1 Redistributable and the Microsoft Visual C++ 2005 Service Pack 1 Redistributable Package ATL Security Update. Coherence will not run without the security update. For more information on this update, see:

<http://support.microsoft.com/?kbid=973544>

The security update is available from the Microsoft Update Web site or directly from this URL:

<http://www.microsoft.com/downloads/details.aspx?displaylang=en&familyID=766a6af7-ec73-40ff-b072-9112bab119c2>

Avoid Using the size() Operation on Transactional Caches

The `size()` operation, when performed transactional caches, is fairly complex. Avoid using this operation in a tight loop as it may result in blocking partition transfers.

The CohQL BACKUP CACHE Command Does Not Support POF

The "Writing a Serialized Representation of a Cache to a File" section of the *Oracle Coherence Developer's Guide* fails to mention that the CohQL BACKUP CACHE command does not back up cached POF objects. Objects must implement `Serializable` or `ExternalizableLite` to be backed up by this command.

Deprecated Elements Referenced in Developer's Guide

The *Oracle Coherence Developer's Guide* refers to the following elements which are no longer supported in the `coherence.dtd`:

`<burst-mode>`—This was a subelement of `<packet-publisher>`.

`<serializer-name>`—This was a subelement of `<serializer>`.

Maximum Value for Time-Based Configuration Elements

Time-based configuration elements, such as `<expiry-delay>` that are entered as an integer will max out to `Integer.MAX_VALUE` (2147483647) regardless of how they are entered (as milliseconds, seconds, minutes, hours, or days).

Typo in the manifest.mf File in the ActiveCache Documentation

There is a typo in the `manifest.mf` file in Example 6-3 in the *Oracle® Fusion Middleware Using ActiveCache 11g Release 1 (10.3.3)* documentation. "Implementation" was mis-spelled. Following is the corrected configuration:

```
Extension-List: active-cache
active-cache-Extension-Name: active-cache
active-cache-Specification-Version: 1.0
active-cache-Implementation-Version: 1.0
```

Error in .NET SSL Example

Example 5-8 in the *Oracle Coherence Client Guide* contains an element sequence error which prevents the example from successfully running. To run the example, the `<stream-provider>` node must precede the `<remote-addresses>` node within the `<tcp-initiator>` element as follows:

```
<tcp-initiator>
  <stream-provider>
    <ssl>
      <protocol>Tls</protocol>
      <local-certificates>
        <certificate>
```

```

        <url>C:\</url>
        <password>password</password>
        <flags>DefaultKeySet</flags>
    </certificate>
</local-certificates>
</ssl>
</stream-provider>
<remote-addresses>
  <socket-address>
    <address>198.168.1.5</address>
    <port>9099</port>
  </socket-address>
</remote-addresses>
<connect-timeout>10s</connect-timeout>
</tcp-initiator>

```

Updated Instructions for Configuring Cluster Nodes for Coherence*Web

There were errors in the instructions for configuring Coherence*Web for cluster nodes in the *User's Guide for Oracle Coherence*Web*. The following sections provide the corrected instructions.

Configure Cluster Nodes (WebLogic Server 10.3.3 and Later)

Coherence*Web can have *application server-scope*, *EAR-scope*, or *WAR-scope*. Like Coherence clusters, scoping of Coherence*Web depends on the placement of the `coherence.jar` in the classloader's hierarchy.

WebLogic Server 10.3.3 provides several features that allow your applications to more easily interact with the Coherence cache. For a complete discussion of these features see *Using ActiveCache*. To employ ActiveCache functionality in your applications, you must also deploy the `active-cache.jar` file, which you can find in the `WL_HOME/common/deployable-libraries` directory.

- [Configuring Application Server-Scoped Cluster Nodes](#)
- [Configuring EAR-Scoped Cluster Nodes](#)
- [Configuring WAR-Scoped Cluster Nodes](#)

Note: The application server-scoped cluster configuration should be considered very carefully and never used in environments where application interaction is unknown or unpredictable.

An example of such an environment may be a deployment where multiple application teams are deploying applications written independently, without carefully coordinating and enforcing their conventions and naming standards. With this configuration, all applications are part of the same cluster and the likelihood of collisions between namespaces for caches, services, and other configuration settings is quite high and may lead to unexpected results.

For these reasons, Oracle Coherence strongly recommends that you use EAR-scoped and WAR-scoped cluster node configurations. If you are in doubt regarding which deployment topology to choose, or if this warning applies to your deployment, then do not choose the application server-scoped cluster node configuration.

Configuring Application Server-Scoped Cluster Nodes

If you are adding Coherence*Web for session management to a Coherence cluster, follow these steps:

1. Edit your WebLogic Server system classpath to include `coherence.jar` and `WL_HOME/common/deployable-libraries/active-cache.jar` in the system classpath. The `active-cache.jar` should be referenced only from the `deployable-libraries` folder in the system classpath and should not be copied to any other location.
2. Use the WebLogic Server Administration Console or the command line to deploy `coherence-web-spi.war` as a shared library.
3. Enable Coherence*Web in your Web application.

Add the library reference stanza illustrated in [Example 2-1](#) to the `weblogic.xml` in each WAR file deployed in the WebLogic server that intends to use Coherence*Web.

Example 2-1 Library Reference for a WAR File

```
<weblogic-web-app>
...
  <library-ref>
    <library-name>coherence-web-spi</library-name>
  </library-ref>
...
</weblogic-web-app>
```

4. (Optional) If you must configure Coherence cluster properties, create a `CoherenceClusterSystemResourceMBean` and reference it in the `ServerMBean`.

You can use WLST to reference the MBean. See `createServerScopedCoherenceSystemResource` in *Using ActiveCache*.

Configuring EAR-Scoped Cluster Nodes

Follow these steps to use Coherence*Web for session management:

1. Use the WebLogic Server Administration Console or the command line to deploy the `coherence.jar`, `active-cache.jar` and `coherence-web-spi.war` files as shared libraries to all of the target servers where the application will be deployed.

See "Install a Java EE Library" in the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Help*.

2. Reference the `coherence.jar`, `active-cache.jar`, and `coherence-web-spi.war` files in the `weblogic-application.xml` file. Store the file in the EAR's `META-INF/` directory.

[Example 2-2](#) illustrates a sample `weblogic-application.xml` file.

Example 2-2 Coherence, Coherence Web SPI, and ActiveCache Referenced in weblogic-application.xml

```
<weblogic-application ... >
...
  <library-ref>
    <library-name>coherence</library-name>
  </library-ref>
```

```

    <library-ref>
      <library-name>coherence-web-spi</library-name>
    </library-ref>
  <library-ref>
    <library-name>active-cache</library-name>
  </library-ref>
  ...
</weblogic-application>

```

3. (Optional) If you must configure Coherence cluster properties, create a `CoherenceClusterSystemResourceMBean` and reference it as a `coherence-cluster-ref` element in `weblogic.xml` file. This element allows the applications to enroll in the Coherence cluster as specified by the `CoherenceClusterSystemResourceMBean` attributes. For more information, see *Using ActiveCache*.

[Example 2-3](#) illustrates a sample configuration. The `myCoherenceCluster` MBean in the example is of type `CoherenceClusterSystemResourceMBean`.

Example 2-3 Identifying a Coherence Cluster for EAR-Scoped Cluster Nodes

```

<weblogic-web-app>
...
  <coherence-cluster-ref>
    <coherence-cluster-name>
      myCoherenceCluster
    </coherence-cluster-name>
  </coherence-cluster-ref>
...
</weblogic-web-app>

```

Configuring WAR-Scoped Cluster Nodes

If you are using Coherence*Web for session management, follow these steps:

1. Use the WebLogic Server Administration Console or the command line to deploy `active-cache.jar` and `coherence-web-spi.war` as shared libraries to all of the target servers where the application will be deployed. See "Install a Java EE Library" in the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Help*.
2. Copy `coherence.jar` to the WAR file's `WEB-INF/lib` directory.
3. If you deploy the `coherence-web-spi.war` file as a shared library, you must also create a shared library reference by adding the stanza illustrated in [Example 2-4](#) to the `weblogic.xml` file in the WAR file's `WEB-INF` directory.

Example 2-4 Library Reference for a Web Application

```

<weblogic-web-app>
...
  <library-ref>
    <library-name>coherence-web-spi</library-name>
  </library-ref>
...
</weblogic-web-app>

```

4. Create a `manifest.mf` file to reference the `active-cache.jar` file. Copy the file to each WAR file's `META-INF` directory.

Example 2-5 manifest.mf File to Reference active-cache.jar

```
Extension-List: active-cache
active-cache-Extension-Name: active-cache
active-cache-Specification-Version: 1.0
active-cache-Implementation-Version: 1.0
```

5. (Optional) If you must configure Coherence cluster properties, create a `CoherenceClusterSystemResourceMBean` and reference it as a `coherence-cluster-ref` element in `weblogic.xml` or `weblogic-ejb-jar.xml` file. [Example 2-5](#) illustrates a sample configuration for WAR-scoped cluster nodes in the `weblogic.xml` file. The `myCoherenceCluster` MBean is of type `CoherenceClusterSystemResourceMBean`. For more information, see *Using ActiveCache*.

[Example 2-6](#) illustrates a sample configuration for WAR-scoped cluster nodes in the `weblogic.xml` file. The `myCoherenceCluster` MBean is of type `CoherenceClusterSystemResourceMBean`.

Example 2-6 Identifying a Coherence Cluster for EAR-Scoped Cluster Nodes

```
<weblogic-web-app>
...
  <coherence-cluster-ref>
    <coherence-cluster-name>
      myCoherenceCluster
    </coherence-cluster-name>
  </coherence-cluster-ref>
...
</weblogic-web-app>
```

Configure Cluster Nodes (WebLogic Server 10.3.2 and Earlier)

WebLogic Server 10.3.2 and earlier does not support `ActiveCache`.

Coherence cluster nodes are class loader scoped. Therefore, you must configure the number of unique Coherence cluster nodes in a Coherence*Web deployment before packaging the application(s). The packing and configuration options are described in the following sections:

- [Configuring Application Server-Scoped Cluster Nodes](#)
- [Configuring EAR-Scoped Cluster Nodes](#)
- [Configuring WAR-Scoped Cluster Nodes](#)

Note: The application server-scoped cluster configuration should be considered very carefully and never used in environments where application interaction is unknown or unpredictable.

An example of such an environment may be a deployment where multiple application teams are deploying applications written independently, without carefully coordinating and enforcing their conventions and naming standards. With this configuration, all applications are part of the same cluster and the likelihood of collisions between namespaces for caches, services, and other configuration settings is quite high and may lead to unexpected results.

For these reasons, Oracle Coherence strongly recommends that you use EAR-scoped and WAR-scoped cluster node configurations. If you are in doubt regarding which deployment topology to choose, or if this warning applies to your deployment, then do not choose the application server-scoped cluster node configuration.

Configuring Application Server-Scoped Cluster Nodes

1. Deploy `coherence-web-spi.war` as a shared library on each WebLogic Server.
2. Edit your WebLogic Server system classpath to include `coherence.jar` or copy the JAR to your `$DOMAIN_HOME/lib` directory.
3. Enable Coherence*Web in your Web application.

Add the library reference stanza illustrated in [Example 2-7](#) to the `weblogic.xml` in each WAR file deployed in the WebLogic server that intends to use Coherence*Web.

Example 2-7 Library Reference for a WAR File

```
<weblogic-web-app>
  ...
  <library-ref>
    <library-name>coherence-web-spi</library-name>
  </library-ref>
  ...
</weblogic-web-app>
```

Configuring EAR-Scoped Cluster Nodes

1. Use the WebLogic Server Administration Console to deploy `coherence.jar` and `coherence-web-spi.war` as shared libraries to all of the target servers where the application will be deployed. See "Install a Java EE Library" in the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Help* for more information.
2. Reference the `coherence.jar` and `coherence-web-spi.war` files in the `weblogic-application.xml` file. Store the file in the EAR's `META-INF` directory.

[Example 2-8](#) illustrates a `weblogic-application.xml` file.

Example 2-8 Coherence and Coherence Web SPI Referenced in `weblogic-application.xml`

```
<weblogic-application ...>
```

```
...
    <library-ref>
        <library-name>coherence</library-name>
    </library-ref>
    <library-ref>
        <library-name>coherence-web-spi</library-name>
    </library-ref>
...
</weblogic-application>
```

Configuring WAR-Scoped Cluster Nodes

1. Use the WebLogic Server Administration Console or the command line to deploy `coherence-web-spi.war` as shared libraries to all of the target servers where the application will be deployed. See "Install a Java EE Library" in the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Help*.
2. Copy `coherence.jar` to the WAR file's `WEB-INF/lib` directory.
3. If you deploy `coherence-web-spi.war` as shared libraries, create a shared library reference by adding the stanza illustrated in [Example 2-9](#) to the `weblogic.xml` file in the WAR file's `WEB-INF` directory.

Example 2-9 Library Reference for a Web Application

```
<weblogic-web-app>
...
    <library-ref>
        <library-name>coherence-web-spi</library-name>
    </library-ref>
...
</weblogic-web-app>
```

Index

parameter, 2-2
parameter, 2-2
parameter, 2-2
parameter, 2-2

A

AbstractEvolvable, 1-5
AbstractHttpSessionCollection.isExistent, 1-3
ActiveCache, 2-2
active-cache.jar file, 2-3, 2-4, 2-5, 2-8
AddressProvider, 1-3
aggregator serialization, 1-5
application server-scoped cluster nodes, 2-4, 2-7
ASP.NET, 1-2
attribute serialization, 1-3
automatic suspect attributes, 1-3

B

BACKUP CACHE command, 2-2
Binary decorations, 1-3
BinaryEntry objects, 1-2
BinaryEntryStore interface, 1-2
BroadPartitioningStrategy, 1-2

C

CacheEntry, 1-7
CacheMBean, 1-5
ClassCastException, 1-5
ClassNotFoundException, 1-4
cluster join protocol, 1-6
cluster nodes, 2-3
ClusterNodeMBean, 1-5
Coherence ASP.NET Session Management, 1-2
Coherence caches
 using with EAR-scoped cluster nodes, 2-4
 using with WAR-scoped cluster nodes, 2-5, 2-8
Coherence Query Language (CohQL), 1-1
Coherence Resource Adaptor, 2-1
Coherence*Web configuration, 1-3
Coherence*Web, configuring for cluster nodes, 2-3
coherence-cluster-ref element, 2-6
CoherenceClusterSystemResourceMBean, 2-5, 2-6
coherence.jar file, 2-4, 2-5, 2-8

coherence-session-get-lock-timeout parameter, 1-4
coherence-web-spi.war file, 2-4
CohQL, 2-2
Collection references, 1-6
concurrency, 1-5
ConnectionException class, 1-7
context parameters, overriding, 1-4
ContinuousQueryCache, 1-6
custom filtering, 1-6

D

DefaultFactory, 1-3
DoubleSum.process, 1-5

E

EAR-scoped cluster nodes, 2-4, 2-7
entry processor serialization, 1-5
EnumPofSerializer, 1-5, 1-7
ExternalizableLite interface, 2-2

G

GenericPrincipal class, 1-7
GetLocalHostAddress, 1-7

H

HttpSession model, 1-4

I

IDisposable, 1-7
IllegalStateException, 1-4
ImmutableMultiList, 1-7
index overhead, 1-6
Initiator.OpenConnection, 1-7
Initiator.openConnection, 1-3, 1-7
Integer.MAX_VALUE, 2-2
IPMonitor, 1-2

J

JBoss, 1-3, 1-4
JMX statistics, 1-3
JRockit, 1-5

JSP XML files, 1-4
jspInit method, 1-4
JspServlet class, 1-4

K

KeyExtractor-based filters, 1-6

L

lazy acquisition, 1-4
local session attributes, 1-4
locking model, 1-4
logging, 1-4
logout methods, 1-4

M

MachineId element, 1-6
manifest.mf file, 2-2
maximum values, 2-2
MBean notifications, 1-5
MemberDeparted attribute, 1-5
membership notifications, 1-5
MSVC 2005 SP1, 2-1

N

NamedCache, 1-1
NamedCacheProxy.unregisterChannel, 1-3
network filters, 1-7
network protocol filters, 1-6
non-serializable attribute retention, 1-4
Number deserialization, 1-5

O

OriginalValue accessor, 1-2

P

PacketPublisher, 1-5, 1-6
pluggable identity handling, 1-5
PofBufferReader.getParentParser, 1-7
PofBufferReader.getParentParser, 1-2, 1-6, 1-8
PofExtractor, 1-6
proxy service, 1-3

Q

query results, streaming, 1-2
Quorum, 1-2
QuorumStatus attribute, 1-5

R

race conditions, 1-6
readMessageLength, 1-7
ReadWriteBackingMap implementation, 1-6
RegexFilter implementation, 1-6
remote cache service, 1-5

remote include and forward, 1-4
remote invocation service, 1-5
RemoteNamedCache.BinaryCache.Query, 1-7
RemoteNamedCache.BinaryCache.query, 1-7
RemoveRequests, 1-3
Reporter, 1-3, 1-5
reuse-address element, 1-2
rolename attribute, 1-5
rollback-cachestore-failures parameter, 1-6

S

security exceptions, 1-5
segmentation faults, 1-8
SeniorMemberHeartbeat, 1-6
Serializable interface, 2-2
ServiceMBean, 1-5
ServletRequestListener session, 1-4
session attributes, updating, 1-4
session cookie encoding, 1-3
session invalidation, 1-4
session metric reporting, 1-4
session ownership, 1-4
session reaping, 1-3
SessionExpiryExtractor, 1-3
session-timeout parameter, 1-4
side-by-side deployment, 1-4
single session ID referencing, 1-3
size() operation, 2-2
SLA violations, 1-6
split session model, 1-4
SSL, 1-2, 1-7
sticky session optimization, 1-4
streaming query results, 1-2
Subjects, 1-5

T

TagLibs, 1-4
TCPAcceptor, 1-3
TcpInitiator, 1-8
TcpReader, 1-8
TcpRing, 1-2
thread dumps, 1-4
thread locking, 1-4
Tomcat 5.5, 1-3
Tomcat 6.0, 1-3
Transaction Framework, 1-1
transactional caches, 2-2
transactional-scheme element, 1-1
TransactionManagerMBean, 1-5
TreeMap, 1-7

U

UDP socket contention, 1-6
UUID, 1-7

V

Visual Studio 2010, 1-7

W

WAR-scoped cluster nodes, 2-5, 2-8
WebLogic Server SPI, 1-3
weblogic-application.xml file, 2-5
weblogic-ejb-jar.xml file, 2-6
weblogic.xml file, 2-5, 2-6
web.xml, 1-4
WrapperSet.\$KeySet.remove, 1-3
write-behind thread timeouts, 1-6
writeMessageLength, 1-7

